

FEEC



Unicamp



André Luis Ogando Paraense

Uma Metodologia de Desenvolvimento de Sistemas de Amplificação de Inteligência Orientada a Semiótica

Tese de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Ricardo Ribeiro Gudwin
Co-orientador: Rodrigo Almeida Gonçalves

Departamento de Engenharia de Computação e Automação Industrial - DCA
Faculdade de Engenharia Elétrica e de Computação - FEEC
Universidade Estadual de Campinas - UNICAMP - Caixa Postal 6101
13083-970, Campinas, SP, Brasil
Junho de 2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

P212m Paraense, André Luis Ogando
Uma metodologia de desenvolvimento de sistemas de
amplificação de inteligência orientada a semiótica / André
Luis Ogando Paraense. –Campinas, SP: [s.n.], 2008.

Orientadores: Ricardo Ribeiro Gudwin, Rodrigo Almeida
Gonçalves
Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Engenharia de software – Metodologia. 2. Processo
decisório. 3. Semiótica. I. Gudwin, Ricardo Ribeiro. II.
Gonçalves, Rodrigo Almeida. III. Universidade Estadual de
Campinas. Faculdade de Engenharia Elétrica e de
Computação. IV. Título.

Título em Inglês: A semiotic-oriented development process for intelligence
augmentation systems
Palavras-chave em Inglês: Software engineering, Decision support systems,
Intelligence augmentation systems, Wicked problems,
Computational semiotics
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca Examinadora: Ivan Luiz Marques Ricarte, Maria Cecília Calani Baranauskas
Data da defesa: 19/06/2008
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: André Luis Ogando Paraense

Data da Defesa: 19 de junho de 2008

Título da Tese: "Uma Metodologia de Desenvolvimento de Sistemas de Amplificação de Inteligência Orientada a Semiótica"

Prof. Dr. Ricardo Ribeiro Gudwin (Presidente): Ricardo Ribeiro Gudwin

Profa. Dra. Maria Cecília Calani Baranauskas: Maria Cecília Calani Baranauskas

Prof. Dr. Ivan Luiz Marques Ricarte: Ivan Luiz Marques Ricarte

André Luis Ogando Paraense

Uma Metodologia de Desenvolvimento de Sistemas de Amplificação de Inteligência Orientada a Semiótica

Tese de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Aprovação em 19/06/2008

Banca Examinadora:

Prof. Dra. Clarisse Sieckenius de Souza - PUC-Rio

Prof. Dr. Ivan Luiz Marques Ricarte - DCA-FEEC-UNICAMP

Prof. Dr. Fernando Antônio Campos Gomide - DCA-FEEC-UNICAMP

Prof. Dra. Maria Cecília Calani Baranauskas - IC-UNICAMP

Prof. Dr. Ricardo Ribeiro Gudwin - DCA-FEEC-UNICAMP

Departamento de Engenharia de Computação e Automação Industrial - DCA

Faculdade de Engenharia Elétrica e de Computação - FEEC

Universidade Estadual de Campinas - UNICAMP - Caixa Postal 6101

13083-970, Campinas, SP, Brasil

Junho de 2008

Resumo

Problemas não estruturados são problemas que dependem de variáveis não quantificáveis, não mensuráveis, imprecisas ou incertas, e que podem depender de fatores culturais, políticos, sociais, econômicos e ambientais. Sistemas “Otimizantes”, onde há uma forte dependência na lógica e fraca interação homem-máquina, geralmente não são capazes de resolver este tipo de problema, pois não há nestes sistemas um processamento semântico adequado para manipular convenientemente as imprecisões e intratabilidades algorítmicas.

Sistemas de Amplificação de Inteligência (SAI), que têm sua gênese nos Sistemas de Suporte à Tomada de Decisão (SSTD) e consideram a participação humana no circuito de resolução dos problemas, são capazes de suportar a tomada de decisão humana na resolução de problemas não estruturados.

Para construir SAI, as técnicas e ferramentas das metodologias existentes de desenvolvimento de sistemas computacionais são necessárias, porém não suficientes. Este trabalho propõe uma metodologia para o desenvolvimento sistemático de SAI que estende o Processo Unificado de Desenvolvimento, adicionando a ele um fluxo principal de trabalho denominado Modelagem Cognitiva, que concentra-se em aspectos cognitivos dos agentes humanos que resolvem os problemas não estruturados existentes. Para ser capaz de modelar estes aspectos, a metodologia se baseia na teoria da semiótica de Charles Sanders Peirce.

Como caso de estudo, é apresentada a aplicação do fluxo de Modelagem Cognitiva para desenvolver um Sistema de Amplificação Inteligência para suportar a resolução de um problema não estruturado do mundo real: o problema de estocagem e embarque de pelotas de minério-de-ferro de uma mineradora brasileira.

Palavras-chave: Engenharia de Software – Metodologia, Processo Decisório, Semiótica.

Abstract

Wicked problems are problems which depend on unquantifiable, unmeasurable, imprecise or uncertain variables, which can depend on cultural, political, social, environmental and economic factors. “Optimizing” Systems, which depend heavily on computer logic and have unsatisfactory human-computer interaction, often are incapable of solving wicked problems, because they can not manipulate imprecision and algorithmic intractabilities based on a semantic processing.

Intelligence Augmentation Systems (IAS), which have their genesis in Decision Support Systems (DSS) and consider humans in the loop of the problem resolution, are capable of supporting human decision making in solving wicked problems.

Existing tools and techniques for developing computational systems are necessary but not sufficient to build IAS. This work proposes a software development process to systematically develop IAS which extends the Unified Software Development Process, adding to it one core workflow called Cognitive Modeling, which concentrates in the cognitive aspects of the human agents who solve the existing wicked problems. In order to be able to model these aspects, the methodology is based on the semiotic theory of Charles Sanders Peirce.

We present as the case study the application of the core workflow Cognitive Modeling to build an IAS to support the decision making of a real world wicked problem: the stocking and shipping of iron ore pellets of a Brazilian mining company.

Keywords: Software Engineering, Decision Support Systems, Intelligence Augmentation Systems, Wicked Problems, Computational Semiotics.

Agradecimentos

A minha lista de agradecimentos, como a de qualquer um que finaliza uma tese, é inumerável. Com efeito, todas as pessoas com quem eu convivi ao longo da minha vida e desenvolvi um sentimento de confiança influenciaram no resultado desse trabalho, a começar pela minha família, passando pelos meus amigos de infância, faculdade, todos os professores que me ensinaram algo valioso, chegando aos meus orientadores e professores que avaliaram este trabalho na banca.

Se eu fosse um semeador, minha família teria me ensinado o valor do meu trabalho, me educado para realizá-lo com disciplina e a ajudar os outros semeadores sempre que possível.

Meus amigos seriam outros semeadores, que estiveram do meu lado ao longo dos intermináveis dias sofridos de trabalho no campo, trocando palavras, idéias, técnicas, piadas e aventuras nos momentos de folga.

Meus professores de colégio e faculdade teriam me treinado nas técnicas básicas do plantio das sementes.

Eu teria sido um semeador que buscava plantar uma árvore diferente, cuidar dela e vê-la com certo destaque entre todas as outras que eu e meus amigos já tinham plantado.

Meu co-orientador, Rodrigo, teria me dado a semente que poderia ser dessa árvore especial, e aconselhado: “vai, e planta”.

Meu orientador, Ricardo, teria me guiado neste plantio, fornecendo informações sobre os melhores solos para plantar, como deveria ser realizado o plantio para que a árvore crescesse sobre bases sólidas, e me aconselhado ao longo do crescimento da árvore.

Quando a árvore estivesse quase adulta, os professores da banca chegariam para a poda, elevando a copa da árvore para que ela parecesse ainda mais respeitável.

E eu, teria trabalhado dia após dia, debaixo de sol quente, chuva, frio, rindo e às vezes me lamentando ao lado dos meus amigos, os outros semeadores, e de minha família, que apesar da distância eu carregava na memória e no coração, sem os quais eu não teria chegado a ver a árvore como eu havia inicialmente figurado.

Agora imagine que todo esse trabalho se deu na minha mente. . .

À minha família

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xvii
Trabalhos Publicados Pelo Autor	xix
1 Introdução	1
2 Problemas não estruturados	5
2.1 Problemas não estruturados do mundo real	5
2.1.1 Intratabilidade de problemas não estruturados	6
2.1.2 Humanos no circuito	9
2.1.3 Definição de problemas não estruturados	10
2.1.4 Problemas não estruturados no nível operacional: problemas de planejamento e execução dinâmicos (P&ED)	12
2.2 Sistemas de suporte à tomada de decisão: a roupa nova do rei?	15
2.2.1 Estatísticas de fracassos e causas de falhas em projetos de TI	16
2.3 Resumo	19
3 Sistemas de suporte à tomada de decisão	21
3.1 Introdução	22
3.1.1 História dos sistemas de suporte à tomada de decisão	23
3.1.2 Definições e componentização	24
3.1.3 Tipos de SSTD	27
3.1.4 Atributos de um sistema de suporte à tomada de decisão	30
3.1.5 Sistemas descritivos ou prescritivos	31
3.2 Abordagens de SSTD	32
3.2.1 Abordagens centralizada no computador, os “sistemas otimizantes”	32
3.2.2 Abordagens híbridas humano-computador de sistemas de suporte à tomada de decisão	39
3.2.3 Sistemas de Amplificação de Inteligência	42
3.3 Resumo	44

4	Semiótica: representação e significação em operações mentais	47
4.1	Introdução	48
4.2	A semiótica de Charles Sanders Peirce	51
4.2.1	Conceitos relevantes da semiótica Peirceana	53
4.3	Semiótica e semiótica computacional	64
4.3.1	Conceitos importantes da semiótica computacional	66
4.4	Resumo	67
5	Amplificação de inteligência: técnicas para potencializar o intelecto humano	69
5.1	Introdução	72
5.2	Desenvolvimentos da área de amplificação de inteligência	75
5.2.1	Uma ferramenta mental avançada para amplificar a inteligência	79
5.3	Amplificação de inteligência versus inteligência artificial	80
5.3.1	Histórico	80
5.3.2	IA x AI	81
5.3.3	IA	82
5.3.4	AI	87
5.4	Agentes computacionais inteligentes: amplificando a inteligência continuamente	90
5.4.1	Definição de agentes	91
5.4.2	Agentes amplificando o intelecto humano	92
5.5	Engenharia de SAI	94
5.5.1	Atributos de sistemas de amplificação de inteligência	94
5.5.2	Análise dos atributos de SAI	102
5.5.3	Componentização padrão de um SAI	105
5.6	Diagrama de fluxo cognitivo	106
5.7	Redes Semiônicas	112
5.8	Resumo	120
6	Uma metodologia para o desenvolvimento de Sistemas de Amplificação de Inteligência	123
6.1	Engenharia de software	123
6.1.1	Requisitos de software	124
6.1.2	Processos de desenvolvimento de software	129
6.2	O Processo Unificado de Desenvolvimento	133
6.3	Processo Unificado Cognitivo: uma metodologia de desenvolvimento de SAI	140
6.3.1	Modelagem Cognitiva	143
6.3.2	Fluxo de iteração genérico do PUC	150
6.3.3	Benefícios do fluxo principal Modelagem Cognitiva	152
6.4	Resumo	154
7	Caso de estudo: desenvolvendo um sistema de amplificação de inteligência	155
7.1	Introdução	156
7.1.1	Contextualização	156
7.1.2	Objetivos	157
7.1.3	O problema em poucas palavras	158

7.2	MES Pátio - Samarco	162
7.2.1	Mapear workflow operacional	162
7.2.2	Identificar estruturas estáticas e dinâmicas do domínio	164
7.2.3	Identificar sistemas externos	190
7.2.4	Prototipar motores de solução	192
7.2.5	Desenvolver diagrama de fluxo cognitivo	192
7.2.6	Identificar tipos de tomada de decisão	197
7.2.7	Identificar pontos de amplificação de inteligência	197
7.2.8	Modelar amplificação de inteligência em Redes Semiônicas	198
7.2.9	Identificar lugares nas Redes Semiônicas	203
7.2.10	Identificar casos de amplificação de inteligência	205
7.2.11	Prototipar a interface de diálogo	205
7.2.12	Propor componentes do SAI	217
7.2.13	Estimar resultados e ganhos	223
7.2.14	Planejar ciclos de desenvolvimento	224
7.2.15	Identificar requisitos de amplificação de inteligência	227
7.3	Um sistema otimizante para a estocagem e embarque da Samarco	227
7.4	Resumo	229
8	Conclusão e trabalhos futuros	231
8.1	Contribuições	231
8.2	Problemas, limitações e trabalhos futuros	233
8.3	Implicações de médio prazo	234
8.4	Considerações finais	234
A	Apêndices	235
A.1	Código da rede semiônica das figuras 5.11 e 5.12 para SNToolkit	235
A.2	Atividades do fluxo principal de trabalho Modelagem Cognitiva detalhadas	235
A.2.1	Mapear workflow operacional	235
A.2.2	Identificar estruturas estáticas e dinâmicas do domínio	236
A.2.3	Identificar sistemas externos	237
A.2.4	Prototipar motores de solução	238
A.2.5	Desenvolver diagrama de fluxo cognitivo	238
A.2.6	Identificar tipos de tomada de decisão	239
A.2.7	Identificar pontos de amplificação de inteligência	240
A.2.8	Modelar amplificação de inteligência em Redes Semiônicas	241
A.2.9	Identificar lugares nas Redes Semiônicas	242
A.2.10	Identificar casos de amplificação de inteligência	244
A.2.11	Prototipar a interface de diálogo	246
A.2.12	Propor componentes do SAI	248
A.2.13	Estimar resultados e ganhos	249
A.2.14	Planejar ciclos de desenvolvimento	253
A.2.15	Identificar requisitos de amplificação de inteligência	254
A.3	Código da rede semiônica das figuras das figuras 7.31, 7.32 e 7.33 para SNToolkit	258

A.4	Descrição dos casos de amplificação de inteligência do MES Pátio Samarco	258
A.5	Estimativas de resultados e ganhos do MES Pátio Samarco	271
A.6	Requisitos de amplificação de inteligência do MES Pátio Samarco	277
Referências bibliográficas		306

Lista de Figuras

1.1	Metodologia Processo Unificado Cognitivo, uma extensão do Processo Unificado com a adição de um fluxo principal de trabalho denominado Modelagem Cognitiva . . .	2
2.1	Processo de tomada de decisão em um SSTD	9
2.2	Novo paradigma para processo de tomada de decisão em SSTD. Adaptado de Courtney (2001)	10
2.3	Categorias de atividades de tomada de decisão segundo Anthony (1965)	12
2.4	Processo de P&ED em tempo ótimo. (a) - Atividades de planejamento e execução realizadas sequencialmente. (b) - Atividades de planejamento e execução realizadas em paralelo. (Adaptado de Aitken et al. (2007))	13
2.5	Ao ocorrer uma mudança no ambiente, a execução do plano deixa de ser efetiva, e as atividades não são realizadas em tempo ótimo, perdendo o efeito desejado. (Adaptado de Aitken et al. (2007))	14
2.6	Tipos de perdas de eficiência na utilização dos ativos. <i>Fonte: A.T. Kearney</i>	16
3.1	Componentes de um SSTD segundo Bonczek (1980) e Shim et al. (2002)	26
3.2	Componentes de um SSTD segundo Turban e Aronson (1997).	29
4.1	Charles S. Peirce - o pai da semiótica	50
4.2	Modelo triádico de signo (adaptado de (Queiroz, 2004))	53
4.3	Semiose como uma rede (Queiroz, 2004)	55
4.4	Série aleatória de números	57
4.5	Série determinística de números	58
4.6	Série de números que procura atingir uma referência, uma meta.	59
4.7	Relação de iconicidade depende das propriedades de R, ou seja, da similaridade entre R e O.	60
4.8	A figura de uma chave, um signo icônico.	61
4.9	Relação indexical depende de relações espaço-temporais de R-O	61
4.10	A figura de uma chave faz relação com a fechadura, evidenciando uma relação espaço-temporal.	62
4.11	Relação simbólica depende da mediação do interpretante (I) para que a relação R-O ocorra.	62
4.12	A palavra chave faz relação com o objeto chave devido a uma lei, no caso, a convenção linguística.	62
4.13	Tipos de semiose R (icônica), R-O (indexical) e R-O-I (simbólica).	63

5.1	Sistema de controle	74
5.2	Áreas que compõe a IA (Russel & Norvig, 1995)	81
5.3	Quarto chinês (Searle, 1980)	84
5.4	Composição do H-LAM/T system (Engelbart, 1962)	89
5.5	Inserção de uma operação mandatória representando o diálogo entre agentes humanos e computacionais	98
5.6	Componentes genéricos de um SAI	105
5.7	Componentes do diagrama de fluxo cognitivo (DFC)	108
5.8	Exemplo de construções possíveis com o DFC	110
5.9	Estrutura de um agente semiônico (Gudwin, 2003)	113
5.10	Estrutura de uma rede semiônica com agentes semiônicos confinados a lugares (Gudwin, 2003)	114
5.11	Tradução do DFC da figura 5.8 em uma rede semiônica	116
5.12	Subrede implementada pelo superlugar ACoC da rede da figura 5.11	116
5.13	Agentes computacionais, a partir do conhecimento de UCo1, UCo2 e UCo3, propõem uma solução UCoX	117
5.14	O agente humano, a partir do conhecimento de UCo1, UCo2, UCo3 e de UCoX, critica a solução UCoX propondo UCoY	118
5.15	Os agentes computacionais, levando em consideração UCoY, propõem a solução UCoZ	118
5.16	Em (a), a rede inicia sua operação com a atividade cognitiva A, criando em (b) as unidades de conhecimento UCo1 e UCo2. Em (c), a unidade de conhecimento UCo3 é criada através da atividade cognitiva B, que se utiliza de UCo1 e UCo2 para tal. Em (d), temos a UCo4 criada depois do diálogo que acontece entre agente humano e computacional dentro do superlugar sp1, que representa a atividade cognitiva C, uma atividade inteligente. Essa rede é uma tradução do DFC da figura 5.8	119
6.1	Processo de desenvolvimento de software	129
6.2	Tipos de modelo de processo de desenvolvimento de software	132
6.3	Riscos mais sérios são reduzidos nas primeiras iterações do modelo iterativo e incremental (Jacobson et al., 1999b)	135
6.4	Retrabalho nos modelos em cascata e iterativo (Jacobson et al., 1999b)	135
6.5	Ciclos do PU, do início ao fim da vida do processo (Jacobson et al., 1999b)	136
6.6	Um ciclo do PU com suas fases e iterações (Jacobson et al., 1999b)	136
6.7	Os cinco fluxos principais de trabalho do PU ocorrem em iterações ao longo das quatro fases (Jacobson et al., 1999b)	137
6.8	Atividades do fluxo de iteração genérico do PU (Jacobson et al., 1999b)	138
6.9	Alocação média de recursos e tempo das fases do PU (Jacobson et al., 1999b)	139
6.10	Fluxos principais de trabalho do RUP	141
6.11	Um ciclo do PUC, com os seis fluxos principais de trabalho - Modelagem Cognitiva, Requisitos, Análise, Projeto, Implementação e Teste - acontecendo ao longo das quatro fases: concepção, elaboração, construção e transição.	143
6.12	O conjunto de requisitos fica acrescido dos requisitos de amplificação de inteligência no PUC, que procuram capturar os atributos necessários de um SAI.	144
6.13	Iterações do PUC se sobrepondo	144

6.14	Modelagem Cognitiva é realizada principalmente na concepção	145
6.15	Artefatos e papéis envolvidos no fluxo principal de trabalho Modelagem Cognitiva	146
6.16	O Modelo Cognitivo contém todos os artefatos produzidos no fluxo Modelagem Cognitiva	148
6.17	Diagrama com as principais atividades da Modelagem Cognitiva	149
6.18	Fluxo de iteração genérico do PUC	151
6.19	Distribuição média de tempo e recurso entre as fases do PUC para projetos de SAI	152
6.20	O Modelo Cognitivo serve de artefato de entrada para a captura de requisitos.	153
7.1	Localização geográfica da Samarco. (Fonte: http://www.samarco.com)	156
7.2	Planta da Samarco	157
7.3	Itens de controle. (Fonte: http://www.samarco.com)	158
7.4	Empilhamento de pelotas de qualidades bi-horárias diferentes. A forma como as pelotas se organizam na pilha não está reproduzido fielmente.	160
7.5	Gráfico ilustrativo de qualidade x tempo(horas) de um possível item de controle.	161
7.6	Workflow de operação da Samarco	163
7.7	Diagrama do cenário de julho de 2007 do pátio da Samarco	165
7.8	Processos na usina	166
7.9	Diagrama do cenário de 2008 do pátio da Samarco	172
7.10	Linha 1 - Empilhamento de pellet feed	175
7.11	Linha 2 - Empilhamento de pellet feed	176
7.12	Linha 1 - Empilhamento de pelotas	177
7.13	Linha 3 - Empilhamento de pelotas	177
7.14	Linha 2 - Empilhamento de pelotas (rota alternativa)	178
7.15	Linha 2 - Empilhamento de pelotas (emergência das usinas I e II)	178
7.16	Empilhamento de pellet screening (sinter feed)	179
7.17	Linha de recuperação A	180
7.18	Linha de recuperação B	180
7.19	Linha de recuperação A e B	181
7.20	Linha de recuperação A + usinas	182
7.21	Retorno de pellet feed	182
7.22	Mapa de situação do pátio e das pilhas utilizado pelos operadores da Samarco na sala de controle	183
7.23	Forma de empilhamento em cones alternados	185
7.24	Forma de empilhamento em cones Shell	186
7.25	Quotas para embarque	187
7.26	Médias das usinas I e II	188
7.27	Produção das usinas	189
7.28	Qualidade da mistura	189
7.29	Entradas geradas pelos sistemas externos durante o processo de produção	191
7.30	Diagrama de fluxo cognitivo para o problema de estocagem e embarque da Samarco	194
7.31	Rede Semiônica para DFC da figura 7.30	199
7.32	Rede Semiônica referente a atividade de decidir empilhar	200
7.33	Rede Semiônica referente a atividade de decidir recuperar	201

7.34	Sequência de atividades cognitivas e geração de unidades de conhecimento da Rede Semiônica referente a atividade de decidir empilhar.	202
7.35	Protótipo da interface de diálogo diagrama de situação do pátio (DSP) sendo utilizado para empilhamento	206
7.36	Protótipo da interface de diálogo diagrama de situação do pátio (DSP) funcionando no modo recuperação	211
7.37	Outra proposta de interface de diálogo representando as mesmas unidades de conhecimento de outra maneira	214
7.38	Proposta de interface de diálogo do MES Pátio capaz de apresentar estados do pátio em um período de tempo	216
7.39	Componentes do MES Pátio	218
7.40	Etapas do plano de ciclos de desenvolvimento do software	224
7.41	Evolução da utilização do sistema otimizante para a estocagem e embarque da Samarco	229

Lista de Tabelas

2.1	Principais causas para falhas em projetos de TI segundo o levantamento “Chaos” do Standish Group (1995)	18
3.1	Definições de SSTD - parte I (Turban & Aronson, 1997)	25
3.2	Atributos de um DSS segundo Alter (1980)	26
3.3	Definições de SSTD - parte II (Turban & Aronson, 1997)	27
3.4	Definições de SSTD - parte III (Turban & Aronson, 1997)	28
5.1	Implicações nos SAI em função das características dos ambientes dinâmicos	104
7.1	Números da Samarco. (<i>Fonte: http://www.samarco.com</i>)	156
7.2	Produtos	168
7.3	Equipamentos do pátio no cenário futuro	173
A.3	Tabela modelo para descrição de casos de amplificação de inteligência	246
A.4	Fatores de conversão para os valores lingüísticos de estimativa de ganhos e riscos	250
A.5	Tabela modelo para análise de estimativas e ganhos dos subsistemas da componentização	252
A.6	Tabela para avaliação numérica de ganhos e riscos. Seu preenchimento modelo é equivalente ao exemplo da tabela A.5	253
A.7	Tabela modelo para descrição dos requisitos de amplificação de inteligência	257
A.30	Consolidação dos ganhos, riscos e custos	275
A.31	Avaliação de ganhos e riscos	277

Trabalhos Publicados Pelo Autor

1. Paraense, A.L.O.; Gudwin, R.R.; Gonçalves, R.A. Brainmerge: a Semiotic-Oriented Software Development Process for Intelligence Augmentation Systems. Integration of Knowledge Intensive Multi-Agent Systems KIMAS'07: Modeling, EVOLUTION and Engineering. April 29 - May 3, 2007 Waltham, Massachusetts, USA.
2. Paraense, A.L.O.; Gudwin, R.R.; Gonçalves, R.A. Designing Intelligence Augmentation System with a Semiotic-Oriented Software Development Process. Intelligent Systems Design and Applications ISDA'07. October 22 - 24, 2007 Rio de Janeiro, Brazil.

Capítulo 1

Introdução

A fábula “A Roupas Nova do Rei”, escrita pelo dinamarquês Hans Christian Andersen em 1837, famoso pelas suas fábulas, conta a história de um rei vaidoso e de dois vigaristas que se fizeram passar por tecelões de um tecido maravilhoso com a propriedade de parecer invisível aos estúpidos. Como ninguém queria passar por tolo, todos, inclusive o rei, fingiam ver o tecido que os dois teciam a peso de ouro.

Em 2002, Roger Penrose, professor na universidade de Oxford, ganhador do prêmio Wolf Prize em 1998, dividido com Stephen Hawking pela sua contribuição no entendimento do universo, usou esta história para fazer uma analogia com a inteligência artificial. Em seu livro “A mente nova do rei” (Penrose, 1989), ele compara a “mente computacional” à “roupa nova do rei”.

Neste trabalho, também usamos uma analogia com a fábula da “roupa nova do rei”, porém para fazer questionamentos diferentes. Durante as últimas décadas do século XX, foi gerada muita frustração em torno de projetos de “sistemas de suporte à tomada de decisão” (SSTD)¹ ou “otimização” ou ainda “inteligência artificial” aplicados à classe de **problemas não estruturados** (Gorry & Morton, 1971). De forma geral, projetos desta natureza têm um grande valor percebido e pouco valor medido ou realizado na prática, quando os problemas abordados estão em um nível menos tático e mais operacional. Muitas empresas e gestores já colocam abertamente que este tipo de sistema “otimizante” é o “novo sistema do rei”. Os problemas não estruturados abordados em nível operacional são também denominados **problemas de planejamento e execução dinâmicos**, e essa classe de problemas será melhor detalhada no capítulo 2.

O principal objetivo deste trabalho é propor uma metodologia capaz de desenvolver sistemas de suporte à tomada de decisão adequados para resolver problemas não estruturados em nível operacional, ou problemas de planejamento e execução dinâmicos (Aitken et al., 2007).

A metodologia de desenvolvimento de SSTD proposta neste trabalho torna claro o entendimento dos SSTD na solução de problemas não estruturados em nível operacional, desanuviando o entendimento do problema, as relações homem-máquina, o papel e a contribuição de cada agente na construção das soluções. Esta metodologia, batizada de Processo Unificado Cognitivo, é uma extensão do Processo Unificado de Desenvolvimento² (Jacobson et al., 1999b). Construídos com esta filosofia, os SSTD são também denominados de “Sistemas de Amplificação de Inteligência” (SAI)³.

¹SSTD é ao mesmo tempo singular e plural (sistema e sistemas).

²*The Unified Software Development Process* em inglês.

³SAI é ao mesmo tempo singular e plural (sistema e sistemas).

Para entender o posicionamento da metodologia proposta neste trabalho com relação ao Processo Unificado de Desenvolvimento, pode-se utilizar a figura 1.1 (este posicionamento e diferenciação serão refinados ao longo do texto, principalmente no capítulo 6). O processo Unificado Cognitivo é uma extensão do Processo Unificado, adicionando a ele um fluxo principal de trabalho denominado Modelagem Cognitiva, que será executado principalmente na fase de concepção do SAI. O objetivo do fluxo principal Modelagem Cognitiva é o de encontrar as premissas necessárias para que o sistema a ser desenvolvido possua os atributos necessários a um SAI. Seria possível desenvolver um SAI utilizando apenas o Processo Unificado, no entanto, encontrar essas premissas nesse caso dependeria da competência dos profissionais que assumissem os papéis do fluxo principal Requisitos. No caso do Processo Unificado Cognitivo, encontra-se essas premissas de maneira sistemática, garantida pelo fluxo principal Modelagem Cognitiva.

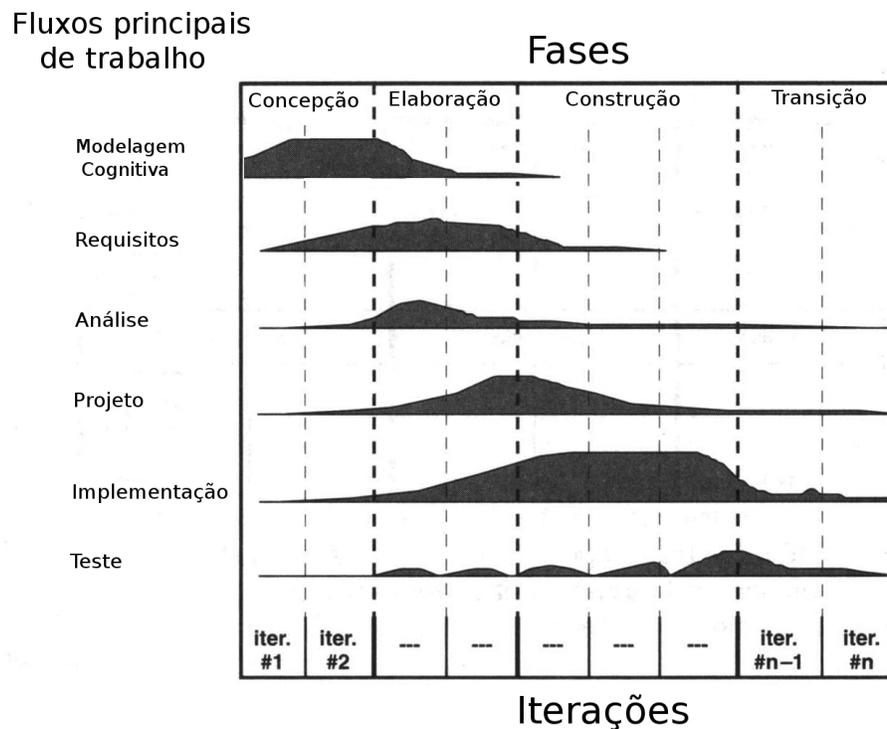


Fig. 1.1: Metodologia Processo Unificado Cognitivo, uma extensão do Processo Unificado com a adição de um fluxo principal de trabalho denominado Modelagem Cognitiva

O capítulo 2 mostra que o fracasso dos projetos de SSTD aplicados à classe de problemas não estruturados está relacionado com o desentendimento do problema, das ferramentas disponíveis e da metodologia utilizada. Não seriam as ferramentas otimizantes genéricas, encontradas nas “prateleiras” da Internet, prontas a serem aplicadas a qualquer tipo de problema não-estruturado, o “novo toolbox do rei”? No mesmo capítulo, são estudadas as características dos problemas em que normalmente ocorrem os fracassos na criação de sistemas de suporte à tomada de decisão e apontadas algumas linhas mestras para o desenvolvimento de uma metodologia que minimizaria os problemas na construção deste tipo de sistema.

O capítulo 3 apresenta os sistemas de suporte à tomada de decisão. O que é exatamente um

sistema de suporte à tomada de decisão? Uma pesquisa bibliográfica mostra que esta pergunta não tem uma resposta clara. Assim, este capítulo compila diversas visões de vários autores e coloca uma definição para este tipo de sistema. Resumindo em uma frase, concluímos que um SSTD é um sistema que “amplifica a inteligência”.

O capítulo 4 detalha alguns conceitos da semiótica, uma ferramenta teórica na qual está fundamentado o entendimento do que é um sistema que “amplifica a inteligência” e também fundamentada a metodologia apresentada neste trabalho.

O capítulo 5 explora o conceito de amplificação de inteligência e explora seus atributos. O objetivo é que a metodologia aqui apresentada seja orientada a estes atributos, ou seja, que os sistemas criados por meio desta metodologia de fato tenham os atributos necessários para “amplificar a inteligência”, ou seja, suportar à tomada de decisão. Neste capítulo, a “amplificação de inteligência” também é comparada à “inteligência artificial”. O capítulo sugere porque agentes computacionais inteligentes são adequados para ser utilizados como ferramentas de amplificação de inteligência na construção de SSTD e na interação com o usuário na construção da solução para problemas do mundo real. Além disso, o capítulo apresenta o diagrama de fluxo cognitivo e as Redes Semiônicas, ferramentas que serão utilizadas na metodologia proposta para identificar os atributos necessários aos SAI.

O capítulo 6 revisa conceitos importantes de engenharia de software e apresenta a metodologia de criação de sistemas de amplificação de inteligência Processo Unificado Cognitivo, que é uma extensão da metodologia Processo Unificado de Desenvolvimento.

O capítulo 7 mostra em detalhes um estudo de caso da utilização da metodologia aqui apresentada.

Por fim, o capítulo 8 faz uma conclusão e apresenta uma visão da evolução deste trabalho.

A contribuição deste trabalho se consubstancia nas técnicas de construção de sistemas de amplificação de inteligência, e não no desenvolvimento de algoritmos de otimização, de técnicas de inteligência artificial, de sistemas multi-agentes ou de protocolos de comunicação e negociação entre agentes. Apesar de endereçar esses assuntos, a contribuição do trabalho está na criação de uma metodologia de desenvolvimento adequada para sistemas de amplificação de inteligência.

Capítulo 2

Problemas não estruturados

Essencialmente, o objetivo deste capítulo é apresentar a motivação e a relevância deste trabalho, que vêm da investigação da importância e das características dos problemas não estruturados¹ (Rittel & Webber, 1973).

O capítulo está organizado da seguinte maneira:

- Seção 2.1: apresenta as motivações intelectuais para construir sistemas de suporte à tomada de decisão, fechando o escopo do significado de “problema não estruturado do mundo real”, apresentando o conceito de problemas não estruturados no nível operacional e demonstrando um pouco da intratabilidade de problemas não estruturados com um exemplo prático. Sugere-se também a necessidade de humanos no circuito na resolução desse tipo de problema;
- Seção 2.2: apresenta as motivações econômicas para construir sistemas de suporte à tomada de decisão, as oportunidades de ganhos desses sistemas e como devem ser tratados para não se tornarem simplesmente “a nova roupa do rei”. Além disso, reporta algumas estatísticas de falhas de projetos de TI com abordagem existentes e inadequadas nas últimas décadas;
- Seção 2.3: apresenta um resumo com as principais contribuições do capítulo.

2.1 Problemas não estruturados do mundo real

A internet e as tecnologias de telecomunicações influenciam cada vez mais as organizações, tornando-as mais complexas, globais e coesas em seus vários níveis de decisão. Cada vez mais a cadeia de suprimentos se interliga. O fluxo de processos e materiais, desde a matéria-prima até os consumidores finais, é cada vez mais integrado e controlado, em vários níveis, por sistemas computacionais. Em alguns casos, os processos controlados ou suportados por ferramentas computacionais são bem estruturados e conhecidos, como, por exemplo, controle de estoques, automação de pontos de vendas, etc. Por outro lado, há processos que são não estruturados (Gorry & Morton, 1971) ou tão complexos que não são definidos com clareza. Estes processos são definidos por Rittel e Webber (1973) como sendo problemas não estruturados. Estes problemas normalmente dependem de variáveis que são não quantificáveis, não mensuráveis, imprecisas ou incertas e que podem depender

¹No trabalho original dos autores, em inglês, problemas não estruturados são denominados *wicked problems*.

de fatores culturais, políticos, sociais, econômicos, ambientais, etc. São estes problemas que são considerados como sendo “problemas não estruturados” dentro do contexto deste trabalho.

Em um primeiro instante, entendemos que sistemas de suporte à tomada de decisão são sistemas computacionais projetados para suportar a tomada de decisão humana na solução de problemas não estruturados (Shim et al., 2002). Esta seção tem por objetivo aprofundar o entendimento de “problemas não estruturados” e, a partir deste entendimento, extrair alguns atributos que devem estar presentes nos SSTD para que eles possam ter os resultados esperados.

2.1.1 Intratabilidade de problemas não estruturados

Estamos vivendo um momento de transição no entendimento do papel das máquinas na resolução de problemas não estruturados. Conforme antecipado por Simon em 1960:

“Muitos, talvez a maioria, dos problemas que precisam ser resolvidos pela média e alta gerência não têm se tornado mais propícios ao tratamento matemático, e provavelmente nunca se tornarão” (Simon, 1960)

Penrose (1989) argumenta que todos os computadores digitais operam por algoritmos, ou seja, regras que o computador segue passo a passo. Segundo ele, fazendo uma interpretação da prova do Teorema da Incompletude de Gödel², há coisas que não podem ser calculadas algorítmicamente, e que para estes casos os seres inteligentes usam mecanismos de “intuições”³ que ainda estão longe de ser compreendidos plenamente. Em outras palavras, há proposições lógicas que são verdadeiras por causa de seu significado e não por causa de sua relação sintática com os outros axiomas do sistema lógico. Apesar disto, ainda há uma crença, presente no inconsciente coletivo, de que é possível modelar matematicamente ou algorítmicamente toda uma organização e otimizá-la em tempo real. Como argumentado por Beynon et al. (2002), a visão contemporânea acerca de computadores e o paradigma atual de programação ainda estão dominados por uma dependência forte na lógica e no formalismo para a construção das aplicações. Essa abordagem desencoraja a interação homem-máquina. Nas últimas décadas, os avanços nas técnicas de otimização contribuíram para a perpetuação desse paradigma, apesar do fato de simples sistemas otimizantes (seção 3.2.1) não serem capazes de suportar a tomada de decisão de uma maneira efetiva.

Nos processos naturais, no nível atômico e quântico, a física newtoniana é imprecisa. Nestes casos, faz-se necessário utilizar as físicas quântica, atômica e estatística. Além disto, há fenômenos que são contra-intuitivos, como por exemplo a dualidade da natureza da luz e o tunelamento de barreira. Adicionalmente, em se tratando de elementos subatômicos, temos que nos contentar em medir sua velocidade ou sua posição uma vez que é impossível medir as duas coisas, com precisão, simultaneamente (princípio da incerteza). Esta impossibilidade nada tem a ver com o avanço tecnológico dos

²O teorema da incompletude de Gödel, às vezes também designado por teorema da indecidibilidade, é o nome atribuído a dois teoremas demonstrados por Kurt Gödel:

- Teorema 1: “Se o conjunto axiomático de uma teoria é consistente, então nela existem teoremas que não podem ser demonstrados (ou negados)”;
- Teorema 2: “Não existe procedimento construtivo que demonstre que uma tal teoria seja consistente”.

³A palavra “intuição” será usada como sinônimo de *insight*.

instrumentos utilizados para a medição. Sabe-se que simplesmente é impossível medir as duas coisas, com precisão, simultaneamente. Trata-se de uma lei natural e nada podemos fazer além de entender e conviver com esta limitação.

Assim como nos processos naturais a física newtoniana descreve com precisão somente sistemas macroscópicos em baixa velocidade, o paradigma de sistemas computacionais “convencionais” é insuficiente para suportar a decisão de problemas não estruturados. Pode-se dizer inclusive que há um princípio análogo ao princípio da incerteza que foi definido por Zadeh (1973) como “princípio da intratabilidade”:

“... a essência desse princípio é que à medida que a complexidade do sistema aumenta, nossa habilidade para realizar declarações precisas e também relevantes a respeito do seu comportamento diminui até um limite ser atingido, abaixo do qual precisão e significado (ou relevância) se tornam praticamente características mutuamente exclusivas.”

Segundo Einstein (1922):

“Quanto mais as proposições matemáticas se referem a realidade, mais elas não estão corretas; e quanto mais elas estão corretas, mais elas não se referem a realidade.”

Assim, há uma relação entre relevância e precisão em problemas não estruturados que independe da tecnologia utilizada para abordar o problema. Trata-se de uma lei natural e nada podemos fazer além de entendê-la e contorná-la, tratando a imprecisão.

Beynon et al. (2002) dão um exemplo que caracteriza o cenário em questão. Considere a tarefa de um humano especialista que precisa construir manualmente uma tabela de horários com o escalonamento de apresentações orais de uma classe de alunos para um grupo específico de juízes. Cada apresentação requer um espaço de meia-hora entre 9 da manhã e 5 da tarde de segunda a sexta. A apresentação será realizada com a presença do supervisor do projeto, um avaliador e um moderador para presidir a apresentação. Ao alocar moderadores e avaliadores dos trabalhos, faz-se necessário garantir que eles estejam qualificados tecnicamente para essas tarefas, além de ser importante a necessidade de dividir igualmente o trabalho entre as pessoas disponíveis, o que é muitas vezes conflitante. Há um número da ordem de 100 estudantes para apresentar trabalhos, mas esse número cresce a cada ano.

Apesar de ser perfeitamente possível escrever uma representação abstrata para uma tabela de horários factível, e também conceber um sistema computacional que considere todas as restrições com relação aos alunos e aos juízes e que gere tal tabela automaticamente, sabemos que não é somente dessa maneira (abstrata e lógica) que o especialista humano entende e modela o problema ao resolvê-lo. A alocação da tabela de horários é mais do que simplesmente decidir um determinado recurso alocado para um determinado tópico balizado por restrições abstratas. O especialista humano possui muitas expectativas tácitas⁴ sobre certas qualidades que a tabela de horários deve possuir, e a exploração de diferentes soluções para construir a tabela será guiada por essas expectativas. Por exemplo, o especialista humano irá se preocupar com:

- necessidade de tornar as seções tematicamente coerentes;

⁴Expectativas que existem na mente de uma pessoa, difíceis ou impossíveis de serem representadas, escritas ou codificadas (Courtney, 2001).

- competência dos membros da banca na avaliação de determinados trabalhos ou assuntos;
- minimizar retrabalho futuro - pode ser que um membro chave da banca não saiba até o domingo se ele poderá avaliar trabalhos na segunda ou na terça, o que irá atrapalhar todas as outras alocações;
- aspectos humanos - pode ser que ele saiba que um determinado membro da banca tenha problemas pessoais com outro membro da banca e que preferencialmente os dois não deveriam trabalhar juntos.

Apesar de ser possível admitir que estas situações podem ser modeladas com novos atributos, regras e restrições no modelo do sistema, deve-se considerar que:

- em alguns casos é muito menos custoso desconsiderar alguma questão que tem baixa probabilidade de ocorrência;
- é impossível antecipar todos os potenciais problemas. Por exemplo, pode-se descobrir que um dos membros da banca tem uma orientação religiosa fundamentalista, ou que seja racista e que sua religião ou princípios éticos vão prejudicar o julgamento de alguns trabalhos;
- uma solução boa é alcançada a partir das relações que ela possui com o cenário que se apresenta no mundo real ao qual ela se refere e a sua vizinhança no espaço de busca que está fora do escopo do problema (Beynon et al., 2002).

Mesmo se fosse possível antecipar e implementar atributos, regras e restrições para tratar todos os casos de exceção, provavelmente o sistema se tornaria caro demais e extremamente difícil ou impossível de ser utilizado por excesso de parâmetros. O especialista humano simplifica o problema desconsiderando algumas variáveis e aproximando outras. Assim, para o modelo ser relevante, é necessário tratá-lo com uma certa imprecisão. Caso se invista em uma solução extremamente precisa, considerando todas as possíveis condições, atributos, regras e restrições, perde-se a relevância do sistema porque o excesso de parâmetros fará com que ele não seja utilizável. Em outras palavras, neste caso a relevância e a precisão são inversamente correlacionadas. Neste caso, qual das duas tabelas traria mais ganhos? A tabela encontrada automaticamente por um algoritmo de otimização, ou uma tabela que seria resultado da intervenção humana na decisão otimizada, considerando os aspectos não estruturados? Certamente, seria a tabela alterada pelas decisões inteligentes do humano.

O surgimento de lógicas multivaloradas e da lógica fuzzy trouxe um enorme avanço no tratamento de informações incertas e imprecisas. Todavia, a necessidade de arcabouços⁵ fuzzy para a solução de problemas não estruturados parece ser apenas um indício de um problema maior: *a necessidade de haver processamento semântico dos dados para poder manipular convenientemente a imprecisão*. Este problema semântico é tratado pela IA e descrito na seção 5.3.3. Segundo Beynon et al. (2002):

“Há uma confusão fundamental entre dados abstratos que são interpretados por seres humanos através de uma associação direta com seu referente no mundo real em determinada situação, e dados abstratos que são manipulados de acordo com regras computacionais que só podem levar em consideração situações pré-programadas dessa associação.”

⁵Tradução destes autores para o termo em inglês “framework”, que será utilizada em todo este trabalho.

O problema da construção da tabela de horários é o tipo de problema do mundo real enfrentado pelos SSTD em que a interação entre humanos e sistema é indispensável. Neste exemplo, faz-se necessário se adaptar a mudanças inesperadas nos requisitos da resolução do problema, tanto na geração da tabela quanto na hora de aplicá-la, mas constitui-se em um problema pouco dinâmico. Alguns problemas do mundo real apresentam-se em um ambiente muito mais dinâmico, onde a necessidade da intervenção humana em decisões otimizadas é ainda mais evidente e as restrições de tempo real são mais fortes. Este é o caso do problema apresentado no capítulo 7.

2.1.2 Humanos no circuito

Para que haja um equilíbrio adequado entre a participação humana e computacional na construção de SSTD para problemas não estruturados, faz-se necessário considerar os aspectos cognitivos do processo de busca pela solução. Enquanto os problemas de tratamento da semântica dos dados não são resolvidos pela IA (seção 5.3.3), uma estratégia interessante é a sugerida pelo problema apresentado na seção anterior: *incluir os humanos no circuito* para que eles tratem, pelo menos em parte, a incerteza, imprecisão, contextualização, semântica da informação, ponderação entre múltiplos objetivos, etc.

A figura 2.1 (Courtney, 2001) mostra um típico processo de tomada de decisão. A ênfase se dá no desenvolvimento do modelo e na análise do problema. São criadas várias alternativas a partir dos modelos e a escolha é feita, de acordo com Shim et al. (2002). À medida que a complexidade do problema aumenta, as fases deste ciclo se fundem, repetem, atropelam ou se expandem fractalmente, uma vez que cada uma delas podem se tornar um novo processo de tomada de decisão.

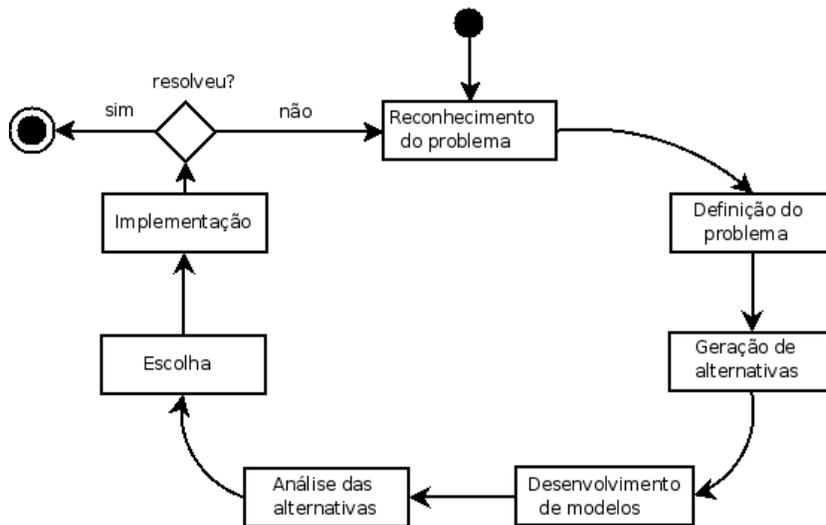


Fig. 2.1: Processo de tomada de decisão em um SSTD

Courtney (2001) sugere um novo processo de tomada de decisão, que foi ligeiramente adaptado na figura 2.2. Ao invés de realizar uma análise puramente técnica, focada nos aspectos de engenharia da tomada de decisão, o processo proposto por Courtney consiste no desenvolvimento de perspectivas de diferentes tipos, para dar suporte aos aspectos “flexíveis” da tomada de decisão. Neste processo, os

modelos mentais dos stakeholders⁶, com variadas perspectivas, são o núcleo do processo de decisão, não mais havendo a hegemonia da perspectiva puramente técnica dos processos tradicionais. Perceba que muitos desses aspectos são humanísticos, e portanto não quantificáveis, como ética e individualidade, o que dificultará o tratamento dos problemas. Isso abre um vasto campo para a aplicação de técnicas de inteligência computacional.

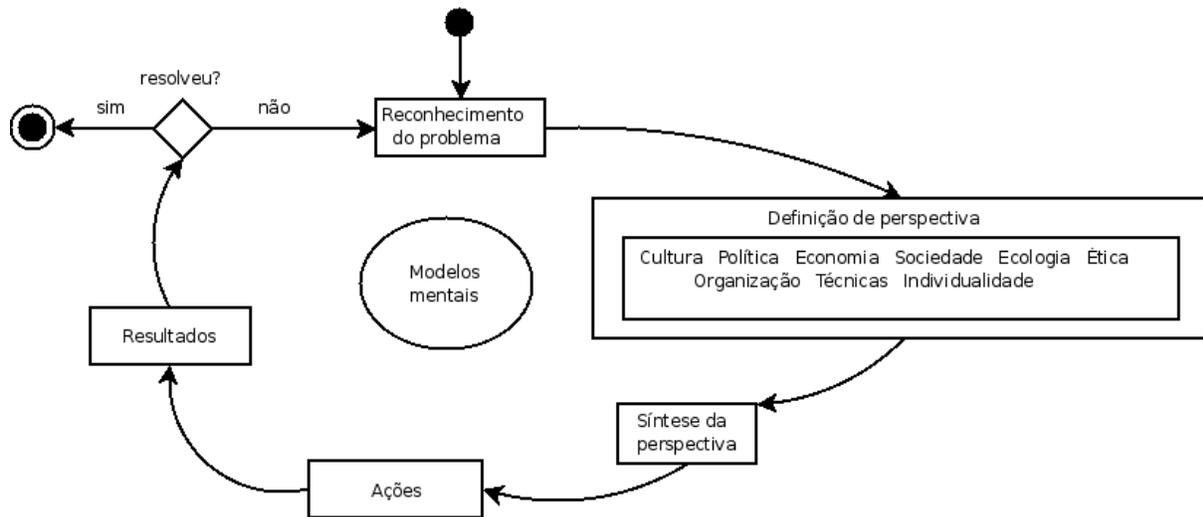


Fig. 2.2: Novo paradigma para processo de tomada de decisão em SSTD. Adaptado de Courtney (2001)

Conforme já mencionado anteriormente, apesar do enorme avanço promovido por ferramentas teóricas, como por exemplo, a lógica *fuzzy*, que permite o tratamento e a representação de informações imprecisas pelo computador, no momento, a capacidade de tratamento de imprecisões e problemas não estruturados ainda é bastante limitada. Portanto, a melhor estratégia de implementar o processo proposto por Courtney e tratar problemas não estruturados é dividí-los em problemas menores e utilizar o ser humano para tratar as imprecisões, postergando, pelo menos por enquanto, o cérebro eletrônico que substituiu o homem e o super-sistema que otimiza, sozinho, toda a cadeia produtiva de uma empresa em tempo real.

2.1.3 Definição de problemas não estruturados

A proposta deste trabalho é apresentar uma metodologia para o desenvolvimento de SSTD para problemas não estruturados. Para definir melhor a classe dos problemas-alvo, partimos da definição de Rittel e Webber (1973). Os autores definem os problemas não estruturados através de 10 propriedades:

1. Não há uma formulação definitiva para um problema não estruturado. Segundo Courtney (2001), formular o problema é o problema;

⁶A palavra *stakeholder* é utilizada para se referir a qualquer pessoa que influencie direta ou indiretamente os requisitos do sistema. Usuários finais e engenheiros que desenvolvem o sistema são exemplos de stakeholders.

2. Problemas não estruturados não possuem regra de parada. Os planejadores param porque acabou o tempo, o dinheiro, a paciência, ou porque chegaram a uma solução “boa o suficiente” (Courtney, 2001), não porque alcançaram o ótimo;
3. Soluções para problemas não estruturados não são certas ou erradas, mas sim boas ou ruins. Isso se dá porque os valores utilizados para avaliar as soluções variam entre os stakeholders (Courtney, 2001);
4. Não há teste imediato ou definitivo de uma solução para um problema não estruturado;
5. Toda solução de um problema não estruturado é uma operação única; como não há oportunidade para aprender por tentativa e erro, toda tentativa custa significativamente, e soluções não podem ser desfeitas;
6. Problemas não estruturados não possuem um conjunto enumerável de possíveis soluções, nem há um conjunto bem descrito de operações permitidas que podem ser incorporadas em um plano. Pode mesmo ser que não haja uma solução;
7. Todo problema não estruturado é essencialmente único. Apesar de muitas similaridades, cada problema não estruturado possui características diferentes que o torna único (Courtney, 2001);
8. Todo problema não estruturado pode ser considerado um sintoma de outro problema. Devido às conexões entre o problema e o ambiente, resolver um problema não estruturado pode resultar em criar outros problemas (Courtney, 2001);
9. A existência de uma discrepância na representação de um problema não estruturado pode ser explicada de maneiras diferentes. A escolha por uma determinada explicação determinará também a natureza da solução para o problema. Essa discrepância na representação diz respeito a estados atuais e estados desejados, projetados de acordo com os objetivos do planejador;
10. O planejador não tem o direito de errar. Diferentemente de cientistas, que podem formular hipóteses que no futuro serão refutadas, planejadores procuram melhorar algum aspecto do mundo (Courtney, 2001).

Apesar da definição de Rittel e Webber ser abrangente devido ao grande escopo da aplicação dos SSTD (negócios, engenharia, medicina, entre outros), e ao fato de que SSTD desenvolvidos para cada uma dessas áreas possuírem atributos diferentes, temos que nos restringir a um certo conjunto de problemas para analisar os requisitos que um SSTD deve possuir e desenvolver uma abordagem correta para sua construção. Com efeito, no restante deste capítulo, e também neste trabalho como um todo, considerar-se-á como alvo principal dos SSTD os problemas não estruturados no nível operacional, denominados de “problemas de planejamento e execução dinâmicos” (P&ED), ou seja, os problemas que apresentam as propriedades propostas por Rittel e Webber em um ambiente dinâmico e com alto nível de detalhamento (alta granularidade).

2.1.4 Problemas não estruturados no nível operacional: problemas de planejamento e execução dinâmicos (P&ED)

Segundo a visão de Anthony (1965), análise e suporte à tomada de decisão podem ser aplicados em três diferentes categorias de atividades de gerenciamento:

- Nível estratégico: análises estáticas do uso dos ativos e estudos dinâmicos de simulação apontam pontos de perda do sistema, gargalos, avaliam políticas alternativas e estimam o retorno de investimento de capital em ativos.
- Nível tático: ferramentas de simulação e otimização apoiam o planejamento de médio e longo prazo, aumentando a efetividade da utilização dos ativos.
- Nível operacional: ferramentas de suporte à tomada de decisão em tempo real implementam de modo sustentável boas práticas operacionais, diminuem o tempo de resposta frente a mudanças de cenários operacionais e otimizam a utilização dos ativos.

O nível operacional é a base da pirâmide da figura 2.3. É dinâmico, incerto, impreciso e está permeado de problemas não estruturados e combinatoriais. Normalmente, a complexidade dos problemas e suas interfaces com outros sistemas obscurecem os potenciais ganhos e dificultam a criação de ferramentas computacionais. Sendo assim, o nível operacional está sempre repleto de oportunidades de ganhos.

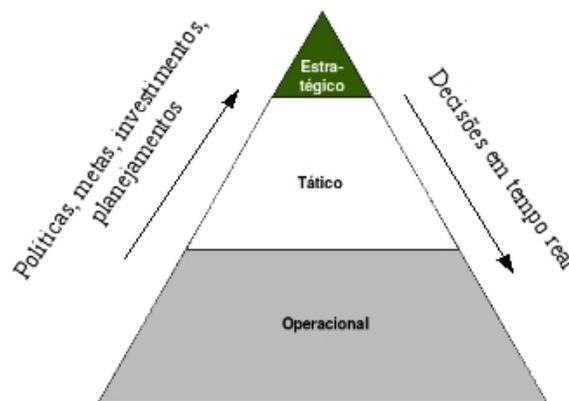


Fig. 2.3: Categorias de atividades de tomada de decisão segundo Anthony (1965)

Para finalizar a classificação dos problemas que são tratados neste trabalho, completamos a definição dos problemas não estruturados com mais uma propriedade necessária para abarcar a dinâmica normalmente encontrada no nível operacional. Esta propriedade é definida por Aitken et al. (2007) como:

“A capacidade de continuamente desenvolver, analisar, selecionar e executar de maneira flexível um robusto curso de ações baseadas nas intenções, consciência situacional e capacidades de um comandante.”

O trabalho de Aitken, Humiston e Patel, um dos poucos da literatura sobre o assunto, estuda problemas não estruturados no nível operacional utilizando o contexto militar. Os autores chamam os problemas não estruturados no nível operacional de problemas de planejamento e execução dinâmica (P&ED).

Os autores afirmam que o ambiente de operação militar modificou-se depois do fim da Guerra Fria, passando a configurar um ambiente onde há presença de desordem, complexidade e incertezas, ou seja, um ambiente dinâmico. Segundo os autores, tornou-se difícil operar no tempo ótimo necessário para que as campanhas possuam o efeito desejado em ambientes de batalha aérea, devido a essa mudança constante e rápida do ambiente de operação, que deixa pouco tempo para o planejamento, e que a maneira de planejar precisa de adaptação para continuar dando vantagens operacionais no futuro, ou seja, encontrar uma solução em resposta a situações que se desenvolvem dinamicamente.

Segundo Aitken et al. (2007), para que os efeitos desejados para suas ações sejam alcançados, faz-se necessários que as atividades corretas sejam realizadas no tempo ótimo, representado na figura 2.4. Uma série de fatores devem estar combinados para que os efeitos sejam alcançados, sendo um dos mais relevantes a capacidade de planejar e agir dentro de janelas de tempo adequadas. Em outras palavras, deve-se ter tempo para planejar e tempo para colocar em execução o plano “ótimo” em todas as campanhas. O planejamento pode ser feito sequencialmente ou concorrentemente à execução. Para esta estratégia funcionar, faz-se necessário possuir conhecimento correto, atualizado e completo de todas as variáveis relevantes do sistema controlado. No entanto, quando o ambiente mudar, como na figura 2.5, ocorrem situações inesperadas. A execução irá continuar neste novo cenário, mas dificilmente será efetiva como era antes da mudança no ambiente, já que o plano foi construído baseado num cenário que já não existe. A execução subsequente somente será efetiva *após a inserção de informações relevantes e atualizadas*. Isso significa que, mesmo se um curso de ação ótimo for encontrado em cada planejamento, sua execução não chegará a termo. Assim, apesar da solução final ser a soma de vários planejamentos ótimos, ela não será ótima.

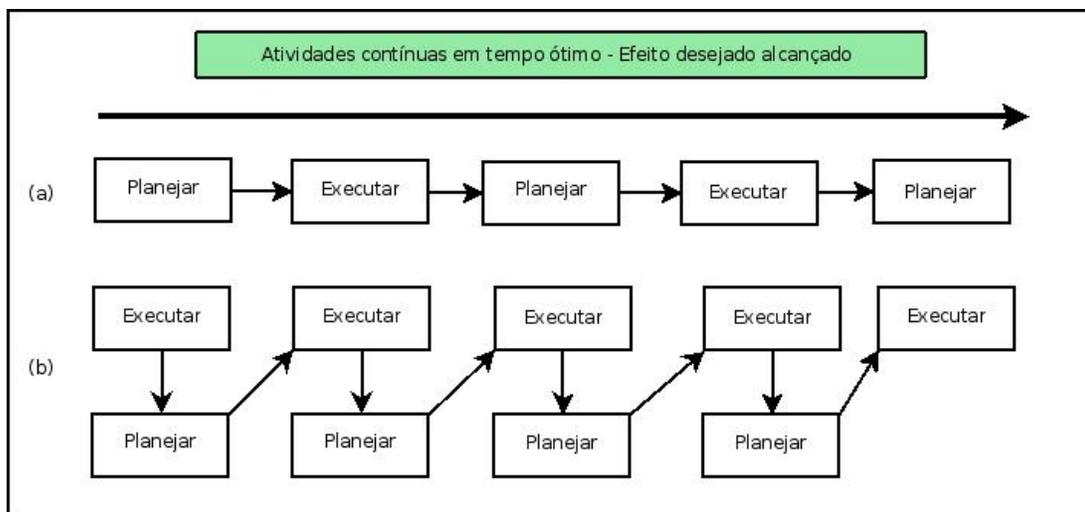


Fig. 2.4: Processo de P&ED em tempo ótimo. (a) - Atividades de planejamento e execução realizadas sequencialmente. (b) - Atividades de planejamento e execução realizadas em paralelo. (Adaptado de Aitken et al. (2007))

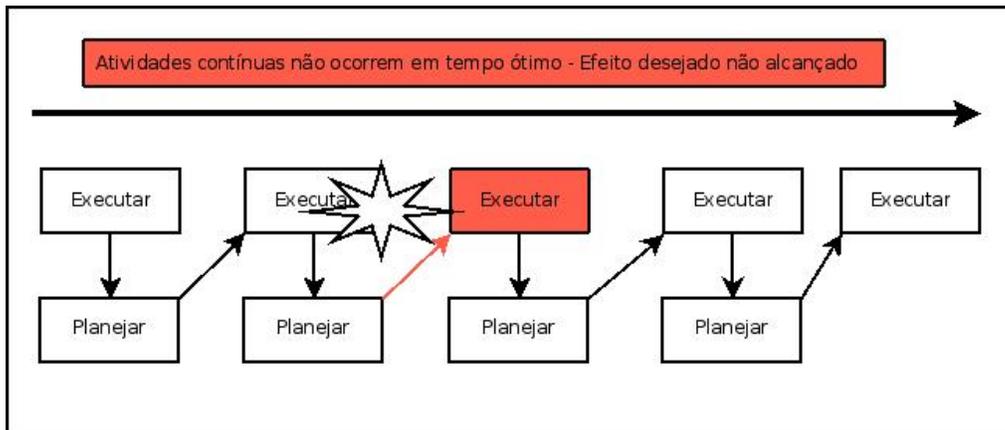


Fig. 2.5: Ao ocorrer uma mudança no ambiente, a execução do plano deixa de ser efetiva, e as atividades não são realizadas em tempo ótimo, perdendo o efeito desejado. (Adaptado de Aitken et al. (2007))

No desenvolvimento do seu trabalho, Aitken et al. (2007) chegaram a uma série de conclusões a respeito do que é necessário para resolver o problema de P&ED, ou seja, o que é necessário implementar num processo de P&ED para lidar com a dinamicidade de um problema. Na seção 6.3, esses atributos serão considerados na definição da engenharia de SAI. Os atributos levantados por Aitken et al. (2007) são:

1. **Execução e planejamento desacoplados:** ao invés de realizar o planejamento e a execução sequencialmente, deve-se realizá-los de maneira desacoplada, ou seja, como dois processos paralelos. Dessa maneira, o processo de planejamento apenas se preocupa em receber informações atualizadas do mundo real e ajustar os planos daquele instante em diante. Assim, mudanças e eventos inesperados possuem impactos mínimos na execução, que apenas recebe sempre os planos mais atualizados e procura segui-los.
2. **Subjetividade no planejamento:** os planos devem ser gerados e interpretados por humanos, que são determinados por suas características e personalidade, como conhecimento, treinamento, experiência, confiança, moral, agressividade e atitude diante de riscos, características que, atualmente, não podem ser implementadas em uma máquina.
3. **Linha temporal:** segundo os autores, representações estáticas de planos não são fáceis de atualizar e costumam gastar mais tempo para tal do que está disponível.
4. **Replanejamento:** segundo os autores, a cada nível hierárquico ao qual o plano é submetido, modificações são realizadas devido a mudanças no ambiente antes da execução. Representações estáticas dos planos não contém nada da lógica, interdependência e razão que existem nas considerações de alteração de um plano.
5. **Coordenação:** mudanças em planos realizadas por subordinados devem ser feitas em concordância com outros subordinados.

6. **Representações semânticas no plano:** segundo os autores, para que um plano possa ser mudado dinamicamente de maneira efetiva e eficiente, o planejador deve entender a lógica e o significado semântico do plano anterior, além de ser capaz de rapidamente gerenciar interdependências afetadas pelas mudanças. Para tal, faz-se necessário uma representação dinâmica e alguma forma de suporte automatizado do planejamento. A representação, segundo os autores, deve ser capaz de facilitar as trocas de informação humano-humano, humano-computador e computador-computador, e que crítica para tal é a habilidade de capturar e representar o contexto e considerações acerca dele.

2.2 Sistemas de suporte à tomada de decisão: a roupa nova do rei?

Em empresas de capital intensivo, o retorno e o valor da empresa estão fortemente ligados com a eficiência do capital e dos ativos. Em outras palavras, a criação de valor econômico é diretamente impactada pelas escolhas feitas no gerenciamento dos ativos.

Essencialmente, há três formas de aumentar a eficiência da utilização dos ativos que são mostradas na figura 2.6:

- Reduzir perdas de utilização: perdas de oportunidade de produção. Sempre que há uma oportunidade de produzir algo e não se produz, há uma perda de utilização.
- Reduzir perdas de *throughput*: perdas de taxa de produção. Sempre que se opera em uma taxa mais baixa do que a taxa ideal, há perdas de taxa de produção.
- Reduzir perdas de aceitação: perdas devido à qualidade da produção, casos em que o produto produzido não atende completamente as especificações exigidas pelos clientes ou pela própria empresa produtora, causando problemas como retrabalho, diminuição do preço de venda e insatisfação do cliente.

A redução das perdas pode ser realizada através de, entre outras coisas, investimento em ativos, através de ferramentas de gestão e através de ferramentas de suporte à tomada de decisão. Normalmente, os profissionais das empresas tem sua competência técnica focada no principal negócio das mesmas e sustentam o aumento da produtividade em peças de engenharia e investimentos em ativos. Contudo, as empresas que possuem um melhor desempenho buscam sustentar seu desempenho na efetividade do uso de seus ativos, usando o capital como um **alavancador** ao invés de um substituto à excelência operacional.

Os sistemas de suporte à tomada de decisão são, sem dúvida, uma forma eficaz e barata de aumentar, de forma sustentada, a eficiência da utilização de ativos em ambiente corporativos e operacionais complexos.

Outros fatores que corroboram na importância dos sistemas de suporte à tomada de decisão são:

- Aumento da tecnologia nos negócios: atualmente, a internet e as tecnologias *wireless* dão origem a organizações que necessitam conectar diretamente sua cadeia de suprimentos, desde a

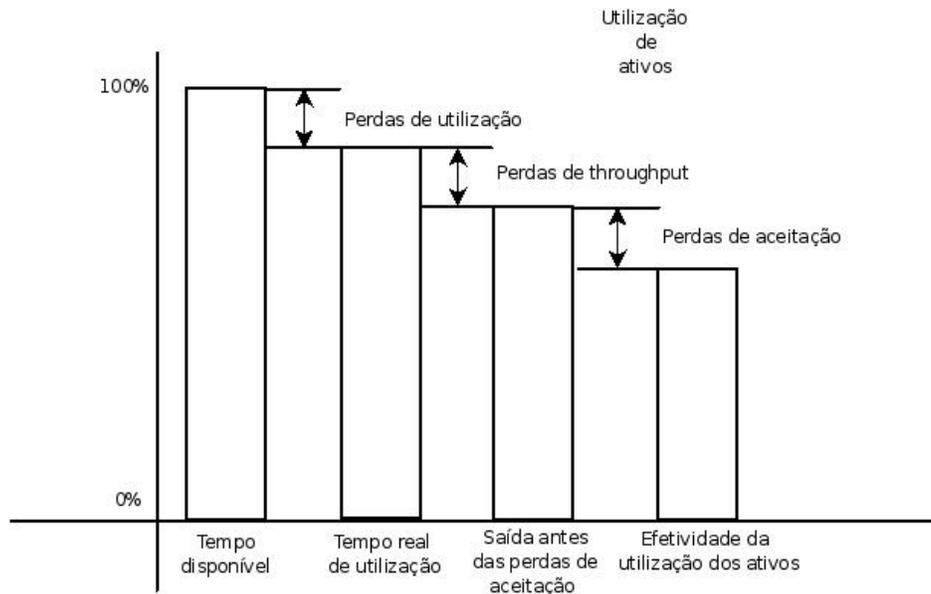


Fig. 2.6: Tipos de perdas de eficiência na utilização dos ativos. *Fonte: A.T. Kearney*

matéria prima até os consumidores finais, além de lidar com um volume de dados extremamente grande, tornando cada vez mais importantes e necessárias as atividades dos sistemas de suporte à tomada de decisão;

- Globalização: as organizações terão que lidar, em vários níveis, incluindo o nível operacional, com problemas tais como diversidade cultural, política, social, econômica e até ecológica (Shim et al., 2002).

Infelizmente, apesar de aparentemente não haver uma estatística oficial a respeito, a taxa de sucesso em projetos de SSTD é baixa e os gestores de Tecnologia da Informação (TI) das empresas têm a percepção que este tipo de projeto traz poucos resultados na prática. Para citar um exemplo, Lang et al. (2000), executivos da indústria ferroviária, em um artigo criticando sistemas de suporte à tomada de decisão para ferrovias, afirmam:

“... a perfeição buscada é simplesmente inalcançável...”

A falta de estatísticas oficiais a respeito de falhas em projetos de TI se explica pelo receio dos profissionais em reportar falhas. Neste tipo de projeto, dentro de uma organização, há muitos profissionais envolvidos, e falhas na implementação de um projeto estratégico da empresa acabam sendo atreladas à imagem do profissional e sendo utilizadas para desmoralizá-lo ou impedi-lo de participar de outros projetos, ameaçando sua carreira.

2.2.1 Estatísticas de fracassos e causas de falhas em projetos de TI

O último estudo do nosso conhecimento a respeito de falhas em projetos de TI data de 2001. Alguns outros estudos foram realizados antes dessa data⁷.

⁷Fonte: <http://www.it-cortex.com/Failure.htm>

Levantamento de Robbins-Gioia (2001)

Robbins-Gioia é um provedor de consultoria em gerência localizado em Alexandria, Virgínia, EUA, que realizou um estudo sobre a percepção de empresas na implantação de um pacote E.R.P.⁸.

O escopo do levantamento foram 232 empresas de vários setores, incluindo governo, TI, comunicações, finanças e saúde.

Um total de 36% dessas empresas já haviam ou estavam no processo de implantação de um sistema ERP. Os principais resultados foram:

- 51% reportaram a implementação do ERP como sendo um fracasso;
- 46% reportaram que a organização não havia entendido como utilizar o sistema para melhorar a forma como seu negócio era conduzido;

Segundo especialistas, um problema desse estudo é a falta de critérios objetivos para medir o fracasso, pois essa medição foi realizada baseada em uma sensação subjetiva dos entrevistados. No entanto, a vantagem dele é que essa “sensação” acaba resumindo uma série de aspectos. De fato, ele é muito parcial, e a resposta pode ser diferente caso o entrevistado tenha participado ou não da implantação do projeto.

Levantamento da KPMG Canada (1997)

Em abril de 1997, a KPMG Canada enviou um questionário sobre aspectos de gerenciamento de projetos de TI para as 1450 maiores organizações do setor público e privado do Canadá. O principal objetivo era levantar as causas de falhas em projetos de TI.

O escopo do levantamento foi a análise de 176 dos 1450 questionários enviados.

Os principais resultados foram:

- 61% dos projetos analisados foram considerados um fracasso pelos entrevistados. Mais de 75% atrasaram em mais de 30% da estimativa de cronograma. Mais de 50% excederam o orçamento inicial por uma margem considerável. Considerando a estimativa de um gasto de US\$25 bilhões em projetos de TI no Canadá, estima-se que as perdas atingiram alguns bilhões de dólares.

Levantamento “Chaos”, do Standish Group (1995)

O levantamento “Chaos” (The Standish Group, 1995) é considerado um marco no estudo de falhas em projetos de TI, sendo citado por todos os autores que falam sobre o assunto.

O escopo deste levantamento foram gerentes executivos. A amostra incluiu empresas de todos os tamanhos e dos principais segmentos da indústria dos EUA: bancos, seguradoras, manufaturas, empresas de saúde, serviços e empresas do setor público, federais, estaduais e locais. Foram 365 amostras representando 8385 aplicações. O Standish Group conduziu entrevistas pessoais.

Principais resultados:

- 31,1% dos projetos seriam cancelados antes de completos. 52,7% dos projetos iriam custar mais de 189% do estimado. Só nos EUA, as perdas poderiam ser estimadas em trilhões de dólares no ano de 1995.

⁸Software para planejamento de recursos nas empresas, do inglês Enterprise Resource Planning.

- Estima-se que 80000 projetos foram cancelados em 1995.

As principais causas identificadas por este levantamento para falhas em projetos de TI estão na tabela 2.1.

Fatores de falhas em projetos de TI		% das respostas
1	Requisitos incompletos	13,1%
2	Pouco envolvimento dos usuários	12,4%
3	Poucos recursos	10,6%
4	Expectativas irreais	9,9%
5	Falta de suporte da gerência	9,3%
6	Requisitos e especificações mutantes	8,7%
7	Falta de planejamento	8,1%
8	Não era mais necessário	7,5%
9	Falta de gerenciamento de TI	6,2%
10	Desconhecimento da tecnologia utilizada	4,3%
11	Outros	9,9%

Tab. 2.1: Principais causas para falhas em projetos de TI segundo o levantamento “Chaos” do Standish Group (1995)

Conclusões

Apesar dos levantamentos apresentados anteriormente se referirem a projetos de TI em geral, e não a projetos específicos de SSTD, percebe-se na tabela 2.1 que algumas das principais causas de falhas se referem aos pontos críticos de um SSTD:

- 1. Requisitos incompletos (13,1%). Problemas não estruturados não possuem uma formulação de requisitos única ótima e as formulações possíveis não são triviais, conforme apresentado nas seções 2.1.1 e 2.1.3.
- 2. Pouco envolvimento dos usuários (12,4%). Os usuários precisam estar envolvidos no projeto de SSTD para que possa ser corretamente especificada a interação homem máquina e a participação destes no circuito de solução (seção 2.1.2).
- 4. Expectativas irreais (9,9%). Sistemas “otimizantes” geram expectativas irreais de resolução de todos os problemas na concepção de um SSTD.
- 6. Requisitos e especificações mutantes (8,7%). Idem de 1.
- 8. Não era mais necessário (7,5%). Devido às mudanças constantes no ambiente de problemas não estruturados, uma especificação completa, precisa e centralizada no computador acaba se tornando obsoleta quando o mundo muda.

A soma desses pontos críticos aos SSTD é responsável por 51,6% das falhas de projetos de TI, sendo que entre estes pontos críticos estão as duas maiores razões da tabela 2.1.

A maioria dos fracassos de projetos de SSTD está relacionada ao fato de que eles são tratados como sistemas otimizantes. Conforme se propõe neste trabalho, a maior parte dos SSTD deveria ser tratada, ao invés, como sistemas de amplificação de inteligência. Esta diferença de abordagem, que ficará clara no capítulo 3, tem reflexos em todo o processo produtivo do sistema, desde a fase de propostas e contratação, passando pela engenharia e implantação do sistema, até a garantia ou manutenção do mesmo.

A experiência na utilização da metodologia proposta neste trabalho, não só no estudo de caso apresentados, mas também em outros projetos, demonstra que, com uma abordagem adequada, os SSTD deixam de ser “a roupa nova do rei” para ocupar efetivamente seu papel de alavancador de valor nas empresas.

2.3 Resumo

Sugeriu-se neste capítulo que sistemas que resolvem problemas não estruturados demandam decisões inteligentes, mas nem sempre sistemas inteligentes computacionais podem suprir sozinho essa demanda, devido à necessidade de haver processamento semântico dos dados para poder manipular convenientemente a imprecisão. A solução para resolver tais problemas não estruturados é o desenvolvimento de sistemas de amplificação de inteligência, ou seja, sistemas híbridos, onde há interação entre inteligência humana e artificial.

Além disso, sugeriu-se também que em grande parte dos setores da economia há uma grande diferença entre a relação custo-benefício de um projeto de SSTD e de um projeto de investimento em ativos. Em outras palavras, investir em SSTD é normalmente muito mais vantajoso para as empresas do que o investimento em ativos. Por outro lado, até o momento, o risco do investimento em SSTD é muito maior do que o risco no investimento em ativos. Este capítulo mostrou porque este risco normalmente é alto e porque este trabalho corrobora na redução e mitigação deste risco.

Capítulo 3

Sistemas de suporte à tomada de decisão

Este capítulo irá definir o conceito de sistemas computacionais de suporte à tomada de decisão (SSTD), baseando-se nos desenvolvimentos e no estado da arte da área e procurando tornar explícito os atributos mais importantes dos SSTD. Os atributos apresentados são, com efeito, o que um sistema computacional necessita para ser considerado um SSTD, e a correta implementação destes resultará num SSTD de maior qualidade e que obterá mais sucesso no seu papel. Do ponto de vista de uma metodologia de desenvolvimento, isso significa que estes atributos devem estar de alguma maneira mapeados nos requisitos de um futuro SSTD. O capítulo procura identificar também a motivação para a existência dos SSTD além de fazer um breve histórico dos mesmos.

Os objetivos deste capítulo são:

- Fazer uma revisão bibliográfica de SSTD, apresentando suas definições e os atributos necessários para sua correta implementação;
- Contrastar as abordagens centralizada no computador ¹ e híbrida humano-computador ² dos SSTD;
- Definir Sistemas de Amplificação de Inteligência e seu diferencial frente aos SSTD conforme descrito na literatura.

O capítulo está organizado da seguinte maneira:

- A seção 3.1 traz definições e conceitos acerca de SSTD;
- A seção 3.1.1 conta um pouco da história dos SSTD;
- A seção 3.1.2 apresenta definições e componentização de SSTD;
- A seção 3.1.3 descreve tipos de SSTD;
- A seção 3.1.4 apresenta atributos de SSTD;
- A seção 3.1.5 diferencia SSTD descritivos e prescritivos;

¹Tradução destes autores para o termo “computer-centered”

²Tradução destes autores para o termo “hybrid human-computer”

- A seção 3.2 fala das abordagens centralizada no computador, híbrida humano-computador e da abordagem de sistemas de amplificação de inteligência;
- A seção 3.3 traz um resumo das contribuições deste capítulo ao trabalho.

3.1 Introdução

Sistemas de suporte à tomada de decisão são sistemas computacionais projetados para suportar a tomada de decisão humana na solução de problemas não estruturados (Shim et al., 2002). Esses sistemas são aplicados para suportar a tomada de decisão em diversas áreas, incluindo negócios, engenharia e medicina.

Nos últimos 30 anos de história dos SSTD, foram aproveitadas e utilizadas de maneiras oportunistas todo tipo de tecnologia que pôde ser trazida para tal contexto: planilhas, bancos de dados, redes, agentes inteligentes, programação visual, técnicas avançadas de otimização, entre outros. Apesar do avanço em muitos aspectos, como velocidade de resolução de problemas, capacidade de armazenamento, classificação e recuperação de dados, projeto de interfaces, a melhora efetiva da utilização de SSTD e a extensão destes para diferentes contextos foi modesta (Beynon et al., 2002). Se em algumas aplicações, como processadores de texto, e-mail e web browsers, onde a interação entre humanos e computadores é mais simples, avançaram rapidamente em escala de uso e sofisticação, o mesmo não pode ser dito das aplicações onde essa interação torna-se mais complexa, onde o contexto físico e a imprevisibilidade do ambiente tornam necessário uma interação com qualidade superior (Beynon et al., 2002), como aplicações onde há ambientes dinâmicos com problemas não estruturados. Isso mostra que a qualidade da interação entre humanos e computadores não é muito beneficiada pelo simples avanço tecnológico do computador, sendo necessário concentrar esforços nesse foco.

Há um desequilíbrio entre os avanços tecnológicos computacionais e a evolução na interação homem-computador nos SSTD. Conforme explicado na seção 2.1.2, faz-se necessário “incluir o ser humano no circuito”. Ou seja, ao invés de buscar novas funcionalidades técnicas, deve-se cada vez mais dar ênfase ao agente humano no processo de tomada de decisão, acabando com a abordagem centralizada na resposta fornecida pelo computador e criando uma abordagem híbrida entre humano e computador na busca pela solução, dando uma importância equivalente às contribuições que cada um faz neste processo, atribuindo suas ações para as áreas nas quais suas intuições terão maior sucesso (humanos lidam melhor com aspectos não estruturados, enquanto computadores conseguem melhores resultados nas partes estruturadas do problema). Idealmente, deseja-se sistemas que estabeleçam uma espécie de simbiose entre a mente humana e o computador, fornecendo um alto grau de interação entre ambos (Chen & Lee, 2003). Os SSTD existentes carregam uma série de modelos quantitativos e fornecem um acesso simples a uma grande quantidade de dados, mas os aspectos cognitivos da tomada de decisão recebem pouca atenção, apesar de terem sido reconhecidos há muito como um aspecto importante (Chen & Lee, 2003) (Hoch & Schkade, 1996).

Hoch & Schkade (1996) chamam a atenção para o fato de que há uma preocupação em incorporar novas tecnologias aos SSTD sem a preocupação com o fato de que essas tecnologias podem não ser compatíveis com os processos cognitivos dos usuários dos sistemas. Nem sempre sistemas mais avançados tecnologicamente irão trazer melhores resultados na tomada de decisão. Segundo os autores, faz-se necessário concentrar-se mais em como a decisão é realizada em termos cognitivos

e em como melhorá-la. Apesar disso, Hoch e Schkade (1996) não apresentam um arcabouço para tratar os processos de tomada de decisão, apenas consideram o reconhecimento de padrões como um importante aspecto cognitivo humano e o exploram.

Pode-se dizer então que há duas abordagens de conceitualização de SSTD:

- Abordagem centralizada no computador;
- Abordagem híbrida humano-computador.

A seção 3.2 apresenta uma breve discussão acerca de algumas abordagens centralizadas no computador de desenvolvimento de sistemas de suporte à tomada de decisão, de algumas abordagens cognitivas (híbridas humano-computador) e da abordagem dos chamados Sistemas de Amplificação de Inteligência, uma extensão de sistemas de suporte à tomada de decisão cognitivos que se utiliza de técnicas da semiótica computacional do capítulo 4 e da área de amplificação de inteligência, apresentada no capítulo 5. Antes, porém, faz-se uma revisão da história dos SSTD e um resumo de alguns aspectos importantes da literatura.

3.1.1 História dos sistemas de suporte à tomada de decisão

De acordo com Shim et al. (2002), sistemas de suporte à tomada de decisão evoluíram de duas áreas de pesquisa:

- Estudos teóricos de tomada de decisão organizacional, conduzidos no Carnegie Institute of Technology no fim da década de 1950 e início da década de 1960;
- Trabalho técnico conduzido no MIT durante a década de 1960.

O conceito original foi cunhado por Gorry & Morton (1971), que argumentaram que a natureza imprecisa da informação tratada em um sistema de tomada de decisão torna necessário bancos de dados relacionais e linguagens de *queries* flexíveis, e que a imprecisão estrutural do processo de decisão tornava necessário ambientes de modelagem dos problemas. Eles integraram a visão de Anthony (1965) das categorias de atividades de gerenciamento (estratégico, tático e operacional) e a descrição dos tipos de tomada de decisão de Simon (1960). Segundo Simon, os problemas existem em um plano contínuo que começa nos problemas programados (repetitivos, rotineiros, bem estruturados e de fácil resolução) até os não programados (originais, mal estruturados, difíceis de resolver). Ao invés de utilizar-se dos termos programado e não programado, Gorry e Scott Morton (1971) se utilizaram dos termos estruturado, semi estruturado e não estruturado, e combinaram estes termos com as categorias de Anthony (1965), estratégico, tático e operacional. Eles também se utilizaram das definições do arcabouço de Simon (1960) dividido em inteligência (busca por problema), projeto (desenvolvimento de alternativas) e escolha (análise das alternativas e escolha de uma para implementação). Assim, um SSTD era definido como um sistema computacional que lidasse com um problema pelo menos semi estruturado, ou não estruturado, no qual um sistema era construído para lidar com a parte estruturada da tomada de decisão, enquanto o julgamento humano era usado na parte não estruturada, constituindo um sistema de resolução homem-máquina.

SSTD se desenvolveram bastante desde seu início no começo da década de 1970. A pesquisa nesta área tem procurado melhorar a eficiência com a qual o usuário toma a decisão e a efetividade da decisão (Shim et al., 2002). Esse desenvolvimento acompanhou as três eras de crescimento da indústria

de computadores, a era de processamento de dados, a era dos microcomputadores e a era das redes. A internet possibilitou o desenvolvimento de diferentes tecnologias para SSTD, criando a oportunidade de desenvolver sistemas distribuídos para tomada de decisão de equipes geograficamente distribuídas e a utilização da tecnologia móvel, para que a tomada de decisão possa ser suportada em todos os lugares e o acesso às informações seja permanente. A evolução neste período das interfaces homem-computador, utilizadas como a linguagem entre agentes humanos e o sistema, colaborou muito para que o suporte à tomada de decisão pudesse ser mais eficiente. As interfaces gráfica com o usuário, refinadas na Xerox, popularizadas no Macintosh e depois incorporadas ao Windows proporcionaram esse avanço. As interfaces em três dimensões, touch screen e realidade virtual estão complementando atualmente esse cenário de melhoria.

A partir da década de 1980, criaram-se os sistemas de suporte à tomada de decisão de grupos³ (SSTDG), que possibilitaram a comunicação para suporte à tomada de decisão em equipe e o brainstorm. Os sistemas colaborativos estão incluídos nessa categoria. Nessa mesma década, iniciou-se a utilização de técnicas de inteligência artificial e sistemas especialistas para melhorar o suporte à tomada de decisão desses sistemas. Os desenvolvimentos na área de pesquisa operacional também foram rapidamente absorvidos pela área e diretamente aplicados nos SSTD, o que proporcionou um rápido aproveitamentos desses desenvolvimentos por parte dos usuários, criando uma interessante simbiose entre SSTD e técnicas de otimização, assim como com as técnicas de inteligência artificial. Os SSTD se aproveitam dos desenvolvimentos desses campos para suportar melhor as decisões, enquanto esses campos se aproveitam dos problemas do mundo real, dinâmicos, cheios de restrições e combinatoriais enfrentados pelos SSTD para avançar seus algoritmos. Uma tendência recente é a habilidade de desenvolver, armazenar e analisar diversos cenários de soluções, o que ajuda o usuário a contextualizar a situação e escolher a solução mais conveniente para a conjuntura que o cenário apresenta em determinado instante.

A partir da década de 1990, quatro poderosas ferramentas emergiram para ajudar no processo de construção dos SSTD (Shim et al., 2002). As três primeiras ajudam-no a lidar com dados e construir interfaces com sistemas externos: “data warehouse”(integra dados de diversos bancos de dados operacionais), processamento analítico on-line (rápida compreensão dos dados utilizando diferentes pontos de vista) e data mining (análise inteligente de dados). A quarta ferramenta é a World Wide Web, que deu origem aos SSTD baseados na Web e tornou possível suportar decisões através de browsers, ajudando equipes distribuídas geograficamente.

3.1.2 Definições e componentização

Em grande parte da literatura, os sistemas de suporte à tomada de decisão têm um significado específico, designando uma classe de sistemas de suporte à tomada de decisão para negócios. A revisão das definições de SSTD feita por Turban & Aronson (1997) (tabelas 3.1, 3.3 e 3.4) mostra que não há um consenso na definição do que é SSTD. Além disso, grande parte das definições são subjetivas, dando abertura a diversas interpretações.

As definições apresentadas por Turban e Aronson (1997) não podem ser de fato comparadas, pois possuem focos diferentes. Exceto pela definição dos próprios autores, todas ignoram um tema central dos SSTD: suportar e melhorar a tomada de decisão. A definição de Turban e Aronson (1997)

³Do inglês “group decision support systems” (GDSS).

<i>Autor</i>	<i>Em que está baseada a definição</i>	<i>Detalhe da definição</i>
Gorry & Morton (1971)	No tipo do problema	definiram SSTD como sistemas computacionais que lidam com um problema pelo menos em um nível semi-estruturado ou não estruturado
Little (1970)	Nas funcionalidades do sistema, características da interface	SSTD são compostos por um conjunto de procedimentos para processar dados e julgamentos baseado em um modelo, para ajudar um gerente na sua tomada de decisão
Alter (1980)	No tipo de uso e objetivos do sistema	Alter define DSS através de uma comparação com sistemas convencionais de processamento de dados (tabela 3.2)
Moore & Chang (1980)	No tipo de uso e capacidade do sistema	SSTD são sistemas expansíveis capazes de suportar análise de dados <i>ad hoc</i> e modelagem de decisões, orientados a um planejamento futuro, utilizado em intervalos irregulares e não planejados

Tab. 3.1: Definições de SSTD - parte I (Turban & Aronson, 1997)

é a mais completa das apresentadas, apesar de também ser genérica (apresenta um escopo grande, contemplando desde SSTD básicos até SSTD ideais). Por outro lado, todas as definições apresentadas concordam que os SSTD lidam com ambientes não estruturados permeados de imprecisões, incertezas, variações no tempo, restrições de tempo de resposta, restrições de operação e necessitam considerar um agente humano e suas intuições.

Segundo Turban e Aronson (1997), Bonczek (1980) procura definir SSTD pela sua componentização, formada por três componentes principais:

1. algum tipo de linguagem para a comunicação entre os agentes humanos (usuários) e o sistema computacional;
2. um modelo ou representação do mundo e do problema;
3. uma unidade de resolução deste problema.

Shim et al. (2002) também definem os SSTD através de sua componentização:

1. Gerenciamento de banco de dados sofisticado para tratamento de dados internos e externos;

<i>Dimensões</i>	<i>SSTD</i>	<i>Sistemas convencionais de processamento de dados</i>
Uso	Ativo	Passivo
Usuário	Nível executivo (superior)	Nível técnico
Meta	Eficácia	Eficiência
Horizonte de tempo	Presente e futuro (principalmente futuro)	Passado
Objetivo	Flexibilidade	Consistência

Tab. 3.2: Atributos de um DSS segundo Alter (1980)

2. Funções de modelagem e “motores”⁴ de solução (algumas vezes com algoritmos de otimização) formando o núcleo do sistema;
3. Interface poderosa e amigável com funções e gráficos interativos.

Percebe-se que esses três componentes são um mapeamento dos elementos colocados por Bonczek (1980), representados na figura 3.1.

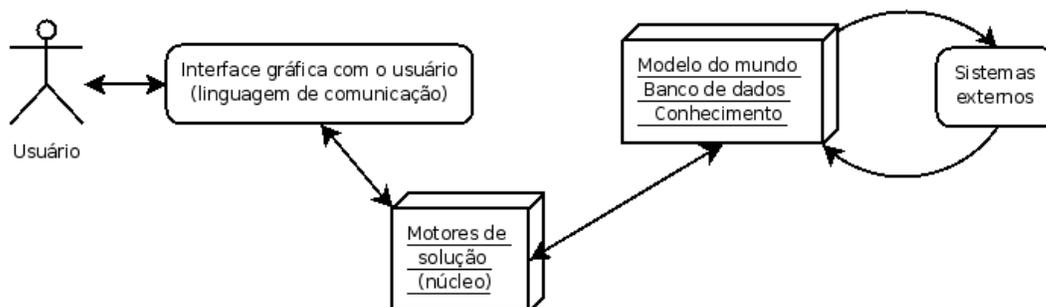


Fig. 3.1: Componentes de um SSTD segundo Bonczek (1980) e Shim et al. (2002)

Turban e Aronson (1997) apresentam uma componentização dos SSTD mais completa, conforme figura 3.2. Para os autores, os componentes dos SSTD são descritos da seguinte maneira:

- Gerenciamento de base de dados: inclui a base de dados gerenciada por um software denominado sistema de gerenciamento de base de dados (SGBD);
- Modelo: modelos quantitativos que proporcionam capacidade analítica ao sistema. Esses modelos podem ser estratégicos, táticos e/ou operacionais.
- Conhecimento: subsistema que suporta algum outro subsistema ou atua de maneira independente. Proporciona inteligência para suportar a decisão.
- Interface com o usuário: subsistema através do qual o usuário se comunica e controla o SSTD.

⁴O termo “motores” foi utilizado como tradução destes autores para o termo do inglês “engines”, e será utilizado deste ponto em diante no trabalho.

<i>Autor</i>	<i>Em que está baseada a definição</i>	<i>Detalhe da definição</i>
Bonczek (1980)	Nos componentes do sistema	SSTD são sistemas computacionais que consistem de três componentes: um sistema de linguagem (um mecanismo para possibilitar a comunicação entre usuário e outros componentes do sistema), um sistema de inteligência (o repositório de conhecimento do domínio do problema embutido no sistema, em formato de dados ou funções), e um sistema de processamento do problema(o elo entre os outros dois componentes, contendo uma ou mais capacidades de manipulação do problema necessárias para a tomada de decisão)
Keen (1980)	Nos processos de desenvolvimento	SSTD são aplicados “a situações onde um sistema final só pode ser construído através de um processo adaptativo de aprendizagem e evolução”, ou seja, como produto de um processo de desenvolvimento no qual o usuário do SSTD, o implementador e o próprio SSTD se influenciam mutuamente, resultando na evolução do sistema e de padrões de utilização.

Tab. 3.3: Definições de SSTD - parte II (Turban & Aronson, 1997)

- Usuário: o usuário tem contribuições indispensáveis proporcionadas através da interação entre ele e o sistema computacional.

3.1.3 Tipos de SSTD

Assim como a definição de SSTD, não há um consenso na literatura sobre a classificação dos SSTD. Baseado nas categorias de atividades gerenciais descritas por Anthony (1965), podemos dizer que os sistemas de suporte à tomada de decisão se classificam da seguinte forma:

1. SSTD estratégicos: suporte ao planejamento estratégico, ou seja, decisões que levam em consideração missões e metas globais. Essas decisões dizem respeito a períodos de tempo mais longos, como anos ou décadas. Por exemplo, projeções de atendimento anual de demanda.
2. SSTD táticos: suporte ao planejamento tático, ou seja, decisões de gerência média inspiradas nas metas da organização. O período de tempo referenciado é de dias ou semanas. Por exemplo, planejamento da produção semanal.

<i>Autor</i>	<i>Em que está baseada a definição</i>	<i>Detalhe da definição</i>
Turban e Aronson (1997)	No que o sistema deve fazer	SSTD são sistemas computacionais interativos, flexíveis e adaptativos feitos especialmente para suportar a solução de um problema não estruturado de gerenciamento para melhorar a tomada de decisão. Ele usa dados, fornece uma interface amigável ao usuário, e pode incorporar as intuições do usuário. Além disso, os SSTD usam modelos, são construídos através de um processo interativo (geralmente por usuário finais), suportam todas as fases de tomada de decisão e podem incluir um componente inteligente.

Tab. 3.4: Definições de SSTD - parte III (Turban & Aronson, 1997)

3. SSTD operacional: suporte ao controle operacional, ou seja, decisões de tarefas operacionais específicas em tempo real. Por exemplo, decisões de replanejamento operacional.

Segundo Turban e Aronson (1997), Holsapple & Whinston (1996) classificam os SSTD em 6 famílias:

- SSTD orientados a textos: informações guardadas em formato de texto e acessadas pelo usuário. Faz-se necessário processar e representar a informação de maneira eficiente, já que a mesma cresce exponencialmente. Esse SSTD mantém uma cópia eletrônica de informações relevantes para a tomada de decisão, tornando possível a criação de textos e documentos quando necessário. Tecnologias como hiper-texto e agentes inteligentes podem ser aplicadas, e tecnologias baseadas na internet têm revolucionado este tipo de sistema.
- SSTD orientados a base de dados: bases de dados possuem o papel central, organizando os dados em uma estrutura de alto nível (relacional ou orientada a objetos).
- SSTD orientados a planilhas: planilhas são uma linguagem de modelagem que permite que o usuário escreva modelos para serem executados no SSTD. Elas criam, modificam, apresentam e instruem o sistema a executar suas operações, e são utilizadas em SSTD voltados a usuários finais, como Microsoft Excel e Lotus 1-2-3.
- SSTD orientados a resolvedores: possuem algoritmos ou procedimentos escritos como programas computacionais para realizar cálculos que resolvem determinados problemas. Resolvedores podem ser incorporados aos SSTD a partir de ferramentas computacionais ou podem ser escritos como programas computacionais em alguma linguagem de programação.
- SSTD orientados a regras: possuem um subsistema de conhecimentos e regras para realizar inferências como um sistema especialista. As regras podem ser qualitativas ou quantitativas.

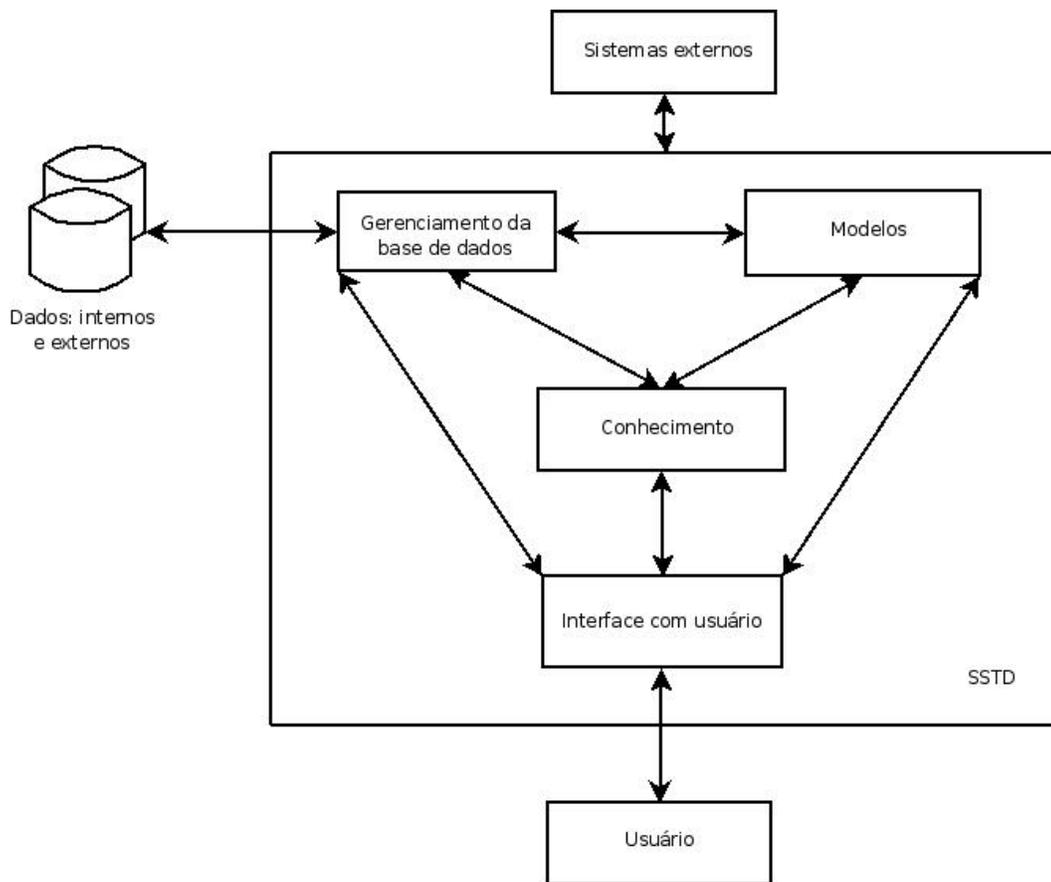


Fig. 3.2: Componentes de um SSTD segundo Turban e Aronson (1997).

- SSTD composto: são feitos a partir de dois ou mais dos cinco tipos explicados anteriormente.

Uma classificação alternativa mais simples, aqui proposta, é a seguinte:

1. Sistemas de suporte transacionais: esses SSTD ajudam usuários e grupos de trabalho a compartilhar informações importantes para a tomada de decisão. O foco deste tipo de suporte é interface entre os usuários, armazenamento e recuperação otimizada de informação. Esses sistemas geralmente não carregam um motor de solução com algoritmos de otimização, concentrando seu foco em coletar informações de diferentes fontes e agrupá-las, ou seja, processar e organizar a informação para os usuários, levando em consideração as interconexões existentes nas informações. No caso de lidar com grupos de trabalho, esses sistemas concentram seu esforço em times de negócio e trabalho cooperativo, coordenando atividades de decisão, já que diferentes pessoas em um mesmo time compartilham metas que se sobrepõem parcialmente e que precisam ser transformadas em metas homogêneas. Nesse contexto, pode-se destacar sistemas de suporte à grupos, sistemas colaborativos e times virtuais.
2. Sistemas de suporte baseados em otimização (Shim et al., 2002): também chamados de “baseados em modelos”, esses sistemas podem ser aplicados às três categorias de atividades gerenciáveis. De acordo com Shim et al. (2002), esses sistemas trabalham em três estágios: formulação

(a especificação de um problema de tomada de decisão é transformada em uma forma algébrica que pode ser entendida por um algoritmo), solução (motores de otimização encontram a melhor solução) e análise (soluções aplicadas a múltiplos cenários e geração de relatórios). A diferença entre este tipo de suporte baseado em otimização e o suporte transacional é que, neste caso, o sistema também sugere uma solução para o problema, ao invés de somente fornecer os meios para que os usuários a encontrem por si próprios.

3.1.4 Atributos de um sistema de suporte à tomada de decisão

Segundo Little (1970), um sistema de suporte à tomada de decisão deve ser simples, robusto, fácil de controlar, adaptativo, completo nas questões importantes, e de fácil interação com o usuário. Já Alter (1980) define os principais atributos de um SSTD através de uma comparação com sistemas convencionais de processamento de dados em cinco dimensões (vide tabela 3.2), evidenciando as diferenças entre esses tipos.

Turban e Aronson (1997) definiram 13 atributos que um SSTD deve possuir para que sua tarefa possa ser exercida com excelência:

1. Um SSTD deve fornecer suporte para situações semi estruturadas ou não estruturadas. Segundo Turban e Aronson, isto pode ser feito unindo informações processadas pelo ser humano e por computador;
2. Um SSTD deve suportar diferentes níveis de tomada de decisão gerencial;
3. Um SSTD deve suportar o tratamento de indivíduos ou grupos de pessoas;
4. Um SSTD deve tratar decisões sequenciais ou interdependentes;
5. Um SSTD suporta várias fases do processo de tomada de decisão;
6. Um SSTD suporta processos de suporte à tomada de decisão com diferentes estilos;
7. Um SSTD é adaptativo ao longo do tempo permitindo que o usuário possa confrontar cenários passados com os atuais e tratar novas situações adicionando, removendo, modificando ou rearranjando elementos básicos;
8. Um SSTD deve ter uma interface amigável com o usuário;
9. Um SSTD deve focar na eficácia do processo de tomada de decisão (precisão, tempo de resposta, qualidade) ao invés da eficiência (custo) da tomada de decisão;
10. O usuário deve ter completo controle em todos os passos do processo de tomada de decisão. O SSTD deve ser criado para dar apoio ao usuário e não substituí-lo;
11. Usuários finais devem ser capazes, por eles mesmos, de realizar modificações simples;
12. Um SSTD utiliza um modelo para realizar análises;
13. Um SSTD deve prover acesso a uma grande variedade de fontes de dados, formatos, e tipos de dados.

No entanto, muitos dos atributos colocados por Turban e Aronson (1997) são vagos ou mal definidos. Por exemplo, o que significa exatamente “uma interface amigável com o usuário”? Por que ela é necessária? O que significa controlar os passos do processo de tomada de decisão? Como isso pode ser realizado? O que deve estar presente para que o sistema possa ser adaptativo? Como deve funcionar a interação entre as informações processadas pelo ser humano e pelo computador? A engenharia de sistemas de amplificação de inteligência responde a essas perguntas na definição dos atributos desse tipo de sistema (seção 5.5.1).

3.1.5 Sistemas descritivos ou prescritivos

Quando se utiliza otimização em um SSTD, identifica-se dois tipos de sistemas resultantes dependendo de duas maneiras diferentes de analisar o modelo (Makowski, 1994):

1. SSTD descritivos ou preditivos: utilizado para fazer previsões do comportamento do modelo. Geralmente, este tipo serve para responder perguntas do tipo “o que aconteceria se”⁵. A simulação é a ferramenta mais utilizada neste caso, sendo possível variar o valor de certas variáveis, para criar cenários “o que aconteceria se”. Dessa maneira, é possível aproveitar a intuição do usuário para setar o valor de algumas variáveis e considerá-los na simulação, incorporando assim a intuição do humano na construção da resposta;
2. SSTD prescritivos ou normativos: utilizado para obter o comportamento do modelo guiado por um conjunto de objetivos. Um SSTD prescritivo ajuda a responder perguntas do tipo “quais decisões levam a melhor solução”. As técnicas de otimização são largamente utilizadas neste caso para escolher uma solução que é considerada “a melhor”. Perceba que o termo “melhor” neste caso refere-se à solução que possui o melhor valor para uma dada função objetivo. Normalmente, o que a comunidade de pesquisa operacional considera é que os modelos matemáticos correspondem exatamente à realidade ou são próximos da mesma o suficiente para que seja possível gerar uma função objetivo que represente as intenções de um usuário humano em dado instante, frente a uma dada situação, levando a uma solução que será a melhor no mundo real. No entanto, conforme tem sido objetivo deste capítulo sugerir, essa conclusão é equivocada e a otimização por si só não leva à melhor solução no mundo real.

Faz-se necessário utilizar os dois modos para obter sucesso em um SSTD. Por exemplo, antes de usar o modo prescritivo do sistema, o usuário poderia usar o modo descritivo, para que pudesse descobrir valores interessantes de algumas variáveis e verificar inconsistências entre suas intuições e as respostas analíticas do modelo. Inconsistências neste ponto demonstram que ou o usuário teve uma percepção errada ou as premissas e especificações do modelo estão erradas. Podem acontecer ambos os casos em situações diferentes, pois modelos matemáticos não modelam o mundo real completamente (nem é o objetivo deles), e os usuários (por mais experientes e especialistas que sejam em determinado domínio) nem sempre terão julgamentos corretos, como é típico de agentes humanos.

⁵A expressão “cenários o que aconteceria se” é uma tradução destes autores para a expressão do inglês “what-if scenarios”, e será utilizado de agora em diante neste trabalho.

3.2 Abordagens de SSTD

3.2.1 Abordagens centralizada no computador, os “sistemas otimizantes”

Abordagens centralizadas no computador são aquelas que não enfatizam a participação do agente humano no processo de busca pela solução, ou seja, que não consideram o processo cognitivo desse agente humano nessa busca para lidar com problemas não estruturados em nível operacional, discutidos por Shim et al. (2002), Courtney (2001) e Beynon et al. (2002). Essas abordagens concentram-se em construir um sistema com um otimizador central que entrega uma resposta “ótima” para um usuário e criar mecanismos para conseguir extrair o máximo desse otimizador, sem considerar uma possível interação com o agente humano da maneira correta. Cada vez mais percebe-se que essa abordagem centralizada no computador de construção de SSTD não é adequada para lidar com problemas não estruturados. Apesar disto, mesmo dada a agenda para o futuro dos SSTD e os problemas que deverão ser enfrentados (Shim et al., 2002), ainda é possível encontrar trabalhos recentes que se baseiam neste paradigma.

Esta abordagem utiliza basicamente só o modo prescritivo (veja seção 3.1.5), o que pode trazer uma série de problemas. Alguns destes problemas foram identificados por Makowski (1994), e ajudam a entender um pouco mais porque é equivocada a idéia de que utilizar somente otimização pode levar à melhor solução para um problema no mundo real:

- uma especificação única de um modelo matemático e de uma função multi-objetivo é muito difícil, quando não é impossível, para a maioria das situações no mundo real. Assim, um ciclo de análises de respostas dadas pelo modelo e modificações dessas respostas realizadas pelo usuário humano (modificações baseadas em suas intuições) antes de se chegar a uma resposta final é uma propriedade desejada em um SSTD;
- modelos são simplificações da realidade, e a otimização está sempre restrita ao modelo que inclui objetivos, por sua vez simplificações de estados desejados do usuário. Dessa maneira, uma solução ótima de um modelo não é necessariamente ótima na realidade, conforme percebida pelo usuário. Este deve ter a capacidade de modificar algumas soluções “ótimas”, para que alguns fatores ainda não levados em consideração pelo modelo em questão sejam incorporados. Pode acontecer, inclusive, de alguns fatores conhecidos serem deixados fora do modelo, porque poderiam torná-lo computacionalmente intratável, o que não significa que eles não irão influenciar a aplicação da solução no mundo real. O processo de decisão de um problema é composto de sub-processos de decisão, resolvidos independentemente; no entanto, a solução ótima do processo como um todo não é a soma das soluções ótimas de cada um dos sub-problemas. Não há como fazer com que o computador seja capaz de perceber no mundo real esses detalhes que faltam na solução como alternativa a essa necessidade do usuário humano compor a solução, já que o computador não está “plugado” no mundo real através de sensores, e tampouco é capaz de atribuir significado aos referenciais do modelo matemático no mundo real, conforme discutido no problema de fundamentação do símbolo⁶ do capítulo 5.

Geoffrion & Maturana (1995) e Maturana et al. (2004), que apresentam trabalhos complementares, fazem uma abordagem interessante no sentido de reduzir tanto o tempo quanto o custo da

⁶A expressão “problema de fundamentação do símbolo” é uma tradução do inglês *symbol grounding problem* adotada por estes autores neste trabalho.

construção de SSTD, a qual eles denominam de sistemas de suporte à tomada de decisão baseados em otimização⁷ (SSTDBO): o projeto e a implementação de um gerador automático de sistemas de suporte à tomada de decisão. O objetivo dos autores é construir ferramentas que possibilitem a construção de SSTD com pouco esforço de programação, por analistas e modeladores com pouca experiência. Segundo os autores, SSTDBO demandam um tempo geralmente muito grande para seu desenvolvimento, e esse desenvolvimento demanda uma equipe com diferentes habilidades, como especialistas em otimização, especialistas em interface humano-computador, desenvolvedores experientes e até mesmo projetistas e artistas. Apesar de ser uma abordagem interessante, e a preocupação em reduzir tempo e custo de construção de SSTDBO ser uma preocupação pertinente, os sistemas de suporte à tomada de decisão considerados como produto da geração automática desta abordagem falham em uma série de aspectos.

A definição de SSTD encontrada por Geoffrion e Maturana (1995) já é um indício que essa abordagem apresentará falhas no que diz respeito à incorporação do agente humano no processo de busca pela solução:

“Um sistema de suporte à tomada de decisão baseado em otimização também pode ser visto como uma aplicação específica, um sistema de suporte, que serve um motor de otimização, que é um software necessário para colocar um motor de otimização para trabalhar de forma útil”.

Mais além, os autores dizem que:

“Bons sistemas de suporte são essenciais se benefícios significativos precisam ser obtidos de um otimizador”.

A abordagem desse trabalho se concentra demais em tentar produzir um motor de otimização que seja capaz de resolver os problemas do mundo real, e considera-se que o sistema de suporte à tomada de decisão deve funcionar de maneira a conseguir aproveitar ao máximo a utilização do otimizador em seu núcleo. É uma definição interessante, mas não funcionará bem ao enfrentar problemas de planejamento e execução dinâmicos, não resolvendo os problemas apontados por Shim et al. (2002) para o futuro dos SSTD, constituindo a primeira falha dessa abordagem. Não podemos deixar de recordar que, apesar de não estarem de acordo com o que é apontado para o futuro dos SSTD, essa abordagem atinge os objetivos dos autores, que é a de criar ferramentas capazes de gerar SSTD de maneira rápida e sem custo ou utilização de especialista.

Segundo os autores, há alguns requisitos necessários para construir bons SSTD, encontrados por Geoffrion & Maturana (1995):

1. Estrutura interna versus estrutura de superfície: segundo os autores, um bom SSTD deve fazer uma distinção correta entre uma “estrutura de superfície” e uma “estrutura interna”. A estrutura interna seriam aspectos de estrutura matemática que impactam a habilidade do algoritmo de otimização de resolver o problema enfrentado. Todo o resto é estrutura de superfície. A idéia dos autores é que a estrutura de superfície pode ser flexível em um SSTDBO, mas a estrutura interna deve ser mantida fixa;

⁷Os autores chamam os SSTD no seu trabalho de “optimization-based decision support systems” - OBDSS

2. Princípio da flexibilidade graduada: segundo os autores, um bom SSTD deve possuir um modelo de representação e um projeto que obedeçam ao princípio denominado “flexibilidade graduada”, que afirma que:

“a estrutura matemática de um modelo e seus dados detalhados devem ser implementados de tal maneira que, ao longo do ciclo de vida da aplicação, as mudanças mais comumente requeridas sejam as mais fáceis de serem feitas, enquanto as menos requeridas possam ser mais difíceis”.

A estrutura interna segundo este princípio, teria uma estrutura mais difícil de modificar do que a estrutura de superfície.

3. O terceiro requisito para um SSTD de qualidade seria o fato de uma determinada linguagem de modelagem ter o papel principal na arquitetura do sistema. Esta linguagem de modelagem é capaz de gerar modelos que são traduzidos para uma representação adequada a um resolvedor.

Os autores se baseiam neste requisitos para fazer o projeto de SSTD, utilizando-se de um gerador de interfaces gráficas e resolvedores encontrados no mercado, como o CPLEX. O sistema é dividido em seis partes: interface com usuário, gerenciador de cenário, sistema de gerenciamento de base de dados, gerador de problema, interpretador da solução e resolvedor. Os três pontos colocados pelos autores como chave para as decisões de projeto da arquitetura são os seguintes:

1. Customização do usuário: decidir entre deixar o usuário customizar as estruturas de superfície ou não.
2. Gerador de problema: utilizar linguagens de modelagem e um tradutor para ler os dados da base de dados, ler o modelo gerado pela linguagem de modelagem e gerar o problema na forma de entrada do resolvedor.
3. Integração do resolvedor: o resolvedor pode ser fracamente ou fortemente integrado com o resto do sistema. Se a integração for fraca, ele interage com o resto do sistema escrevendo e lendo arquivos. Se for forte, ele interage compartilhando certas estruturas de dados. A integração fraca é mais fácil de implementar, porém menos eficiente.

Os sistemas de suporte à tomada de decisão resultado desta abordagem certamente não terão um desempenho adequado frente aos problemas não estruturados, pois faltam alguns atributos para que eles obtenham sucesso. Para citar alguns:

- Os autores não contemplam a interação entre o agente humano usuário e o sistema, de maneira a fazer com que o agente humano fique responsável por administrar os problemas na parte não estruturada, como estabelecer a ligação entre os símbolos e os conceitos representados no sistema e seus referentes no mundo real. Isso seria realizado através de uma interface gráfica adequada, e da interação entre agentes humanos e computacionais. Os autores citam a necessidade de desenvolver interfaces gráficas avançadas (Geoffrion & Maturana, 1995), mas em nenhum momento apresentam exatamente o que deve estar presente nessas interfaces gráficas ou como elas devem ser projetadas. Naturalmente, como essa abordagem procura construir um gerador de SSTD, tudo é bastante genérico. No entanto, os problemas atuais e futuros dos

SSTD necessitam de um projeto mais específico, com interfaces gráficas que representem conceitos que precisam ser manipulados pelo usuário de maneira mais simples e algoritmos que resolvam o problema de maneira rápida o suficiente para que haja uma interação entre o usuário e o sistema computacional na busca pela solução. Como gerar uma interfaces gráfica e um algoritmo desse nível automaticamente?

- Os autores se preocupam muito com a necessidade de encontrar uma solução ótima para o problema em questão. No entanto, no caso dos problemas não estruturados, quase sempre lidamos com problemas de planejamento e execução dinâmicos. Assim, o tempo necessário para se encontrar a solução ótima dificilmente é compatível com a constante de tempo do sistema, ou seja, com o tempo médio entre mudanças de cenário, resultando no problema descrito na seção 2.1.4 e ilustrado pela figura 2.5.

Para ilustrar a dificuldade deste tipo de abordagem em um problema não estruturado, considere o problema da tabela de horários descrito na seção 2.1.1. Neste caso da tabela de horários, seria mais interessante que o sistema de suporte à tomada de decisão projetado para escalonar equipes de avaliadores encontrasse uma solução boa em alguns segundos, permitindo que o usuário pudesse fazer algumas modificações, fixando-as, do que gastar um tempo considerável para achar uma solução ótima, que pode se tornar inválida após um simples telefonema de um dos membros do grupo de avaliação informando que está doente e não poderá atuar em um dos dias para o qual ele estava escalonado. No primeiro caso, bastaria ao usuário mudar essa alocação para outro dia e pedir que o sistema encontrasse outra solução considerando aquela alocação realizada manualmente como fixa. Essa conversa entre usuário e sistema levará a uma solução final que é evolutiva, e certamente melhor do que a que o humano ou o computador achariam sozinhos. No caso da tabela de tempos, provavelmente o usuário terá tempo suficiente para encontrar uma solução. No entanto, isso não é verdade para uma aplicação em tempo real. Essa diferença torna-se ainda mais relevante, já que não poderíamos permitir que o algoritmo passasse um tempo consideravelmente grande para encontrar a solução ótima.

Um outro trabalho que considera a utilização de geradores de SSTD é o de Bhargava et al. (1999). Neste trabalho, Bhargava et al. avaliam uma série de ferramentas geradoras de SSTD disponíveis no mercado. Os autores defendem a idéia de que SSTD não foram adotados de maneira efetiva em muitas aplicações devido ao seu longo tempo e alto custo de desenvolvimento. Na visão dos autores, ferramentas sofisticadas capazes de gerar modelos, dados e interfaces gráficas com o usuário seriam a saída para diminuir o tempo e o custo do desenvolvimento de SSTD, sendo portanto cruciais para o sucesso da aplicação desses sistemas. O escopo dos autores neste trabalho é basicamente o de SSTD em negócios, o que difere bastante do escopo deste trabalho, que é o de SSTD baseados em otimização e utilizados para resolver problemas não estruturados com processos de tomada de decisão dinâmicos. No entanto, os conceitos apresentados pelos autores podem ser generalizados e utilizados para este escopo.

Dutta (1996) apresenta um survey a respeito da integração de técnicas de otimização e inteligência artificial para suportar a decisão. Segundo Dutta (1996), o fato de técnicas de inteligência artificial terem sido criadas com o intuito de emular o comportamento intelectual humano faz com que elas tenham potencial para realizar atividades de tomada de decisão tipicamente humanas, ou seja, não estruturadas. Uma integração dessas ferramentas com técnicas de otimização, que tem um modelo

bem estruturado, traria ganhos em potencial, segundo o autor. Ele ainda ressalta que as técnicas de inteligência artificial encontram lugar em outros aspectos das ferramentas de tomada de decisão, como na análise pós-solução e atividades de meta-modelagem. Um aspecto interessante da abordagem de Dutta (1996) é a sua sugestão de que as soluções encontradas pelas ferramentas devem ser analisadas, testadas e modificadas antes de aplicadas no mundo real, e que essa atividade poderia ser realizada por técnicas de inteligência artificial, já que as mesmas foram criadas com o intuito de emular atividades cognitivas humanas. Outro aspecto interessante citado pelo autor são as desvantagens de utilizar otimização e um modelo matemático para obter uma resposta final aplicada ao mundo real, enumerados assim:

- métodos de solução podem ficar presos em ótimos locais;
- mudanças incrementais em modelos podem ser difíceis de ser realizadas;
- formulação de modelos necessitam de conhecimento específico avançado, nem sempre presente;
- valores numéricos de entrada estão associados a incertezas típicas do mundo real;
- um modelo raramente captura todos os aspectos de um problema real e geralmente fará simplificações e considerações.

Segundo Dutta (1996), as soluções provenientes da otimização necessitam ser testadas e modificadas antes de serem aplicadas ao mundo real, pois há alguns fatores que o modelo não consegue representar. Neste ponto, o autor está correto. No entanto, sua proposta de utilizar técnicas de inteligência artificial para realizar estes testes, análises pós-solução e modificações dificilmente resolverá este problema. Apesar das técnicas de inteligência artificial conseguirem lidar com as incertezas provenientes do mundo real e terem sido construídas para emular a cognição humana, essas técnicas até hoje ainda não conseguem trazer o entendimento do mundo real para dentro do computador, ou seja, o problema de fundamentação do símbolo. Não há como relacionar os conceitos representados nos modelos e nos algoritmos com seus referenciais no mundo real. Dessa maneira, sempre haverá uma informação ou aspecto do mundo real que não será absorvido pelo sistema, e que tem potencial para tornar a solução de baixa qualidade ou até mesmo ineficaz. Para resolver tal problema, faz-se necessária uma abordagem em que o agente humano realize este ajuste, não só na fase pós-solução, mas também participando da construção da solução em si. Dessa maneira, o agente humano, que está “plugado” no mundo real através dos seus sensores e atuadores, poderá incorporar no sistema informações e aspectos relacionados aos referentes dos conceitos representados presentes no mundo real. Assim, técnicas de inteligência artificial ajudam a adaptar a solução encontrada para o mundo real, mas não são suficientes. Ou seja, podem e devem ser utilizadas, mas não são capazes de suprir a necessidade do diálogo com o agente humano.

Dutta (1996) ainda estabelece que a grande dificuldade, segundo sua visão, para o que ele chama de “integração sem coesão”⁸ entre otimização e técnicas de inteligência, onde o usuário poderia ter a opção de variar entre uma abordagem estruturada (computações quantitativas) ou não estruturadas (análises heurísticas qualitativas) conforme necessário durante as fases de tomada de decisão, é o fato

⁸A expressão “integração sem coesão” é uma tradução destes autores para o termo “seamless integration” do original em inglês.

de que a integração necessita trocar informações não só de estruturas de dados óbvias como matrizes ou regras heurísticas, mas também informações de controle. Como exemplo, ele cita um usuário que gostaria de intervir em um procedimento de branch and bound conforme ele ocorre para mudar a direção da busca a medida que ela surge. Segundo o autor, seria necessário guardar informações tais quais número de nós, ramos entre outras formas de controle do algoritmo. Ainda segundo ele, a grande variedade de modelos e métodos de solução tornaria difícil conseguir prever as informações de controle que necessitariam ser trocadas.

Felizmente, há uma maneira mais simples de resolver essa questão levantada por Dutta, para que assim o agente humano possa construir parte da solução juntamente com as técnicas computacionais. Para realizar tal façanha, deve-se desenvolver uma linguagem de representação a qual possa ser utilizada tanto pelos agentes computacionais como pelos agentes humanos para representar a solução para o problema. Dessa maneira, à medida que o branch and bound de Dutta constrói a solução, o usuário poderia reconhecê-la nessa linguagem e fazer modificações na mesma, obrigando o algoritmo a continuar sua busca.

Uma outra boa abordagem que será aqui classificada como centralizada no computador, mas que na verdade está na fronteira entre centralizada no computador e híbrida humano-computador é a abordagem de Wierzbicki et al. (2000). Nesta abordagem, os autores se concentram em sistemas de suporte à tomada de decisão baseados em modelos⁹(SSTDBM), apresentando uma metodologia para construir esse tipo de SSTD. Essa abordagem está na fronteira, pois não utiliza somente o modo prescritivo, como fazem geralmente as abordagens centralizada no computador, mas também não chega a considerar mecanismos capazes de prover meios ao usuário de construir parte da solução e fazer com que a otimização considere essa parte construída. O que se considera nessa abordagem é o uso de ambos os modos prescritivo e descritivo, mas um de cada vez, utilizados para que um valide o outro, e não fundidos um no outro, como na abordagem cognitiva. Nesta abordagem, os autores consideram que há três aspectos importantes na construção de um SSTDBM:

- informação sobre a situação atual e sobre o passado: desenvolver maneiras adequadas de guardar os dados e apresentá-los de maneira organizada e coerente;
- relação entre processos básicos e as ações ou decisões: estabelecer um modelo matemático cujos parâmetros estão relacionados a decisões como entrada e as saídas estão relacionadas aos resultados do processo;
- o processo de decisão;

Segundo os autores, muitos consideram alguns desses aspectos dominantes sobre os outros, e são capazes de citar problemas em que este exemplo se confirma. No entanto, no caso de problemas não estruturados, os três aspectos são de igual importância e dominantes, na visão dos autores, e uma metodologia deve ser capaz de atacar todos estes aspectos e integrá-los.

A metodologia apresentada pelos autores, denominada “reference point methodology” (RPM), se baseia em otimização multi-objetivo como uma ferramenta para suportar a seleção entre variadas possibilidades de decisão, garantindo a possibilidade de entender a saída para uma determinada decisão escolhida, que é predita por modelos matemáticos. Assim, é possível gerar tanto um resultado

⁹No trabalho original em inglês, os autores denominam os sistemas de suporte à tomada de decisão estudados de “model-based decision support systems” (MBDSS).

final a partir de uma decisão para uma situação inicial, como também é possível criar um cenário final, e obter que processo de decisão levaria até este cenário final iniciando no cenário que se possui. Essa possibilidade é especial principalmente para o tipo de aplicação dos autores, que são aplicações ambientais. Eles podem gerar cenários ambientais caóticos e descobrir que tipo de decisões partindo do cenário de hoje em dia levariam até aquela situação.

A RPM utiliza o conceito de otimalidade de Pareto e fronteira de Pareto, que é essencial em SSTD que lidam com problemas não estruturados, pois estes problemas apresentam objetivos múltiplos e conflitantes. Uma solução é Pareto-ótima quando não é possível melhorar um de seus critérios sem piorar outro. Assim, a otimalidade de Pareto somente pode ser utilizada quando todos os critérios são minimizados ou todos são maximizados. Quando se faz coisas diferentes com diferentes critérios, ou seja, maximizar um e minimizar outro, fala-se no conceito de eficiência (Wierzbicki et al., 2000). Assim, eficiência neste contexto generaliza o conceito de otimalidade de Pareto, um subtipo da mesma. As soluções Pareto-ótimas formam a fronteira de Pareto, o limite de onde todas as outras soluções são Pareto-dominadas. Uma explicação mais formal de Pareto-otimalidade é dada no capítulo 6.

A utilização do conceito de Pareto-otimalidade dá ao SSTD a possibilidade de encontrar as soluções na fronteira de Pareto e apresentá-las ao usuário, para que o mesmo escolha a mais adequada. Como todas são Pareto-dominadas, do ponto de vista da otimização multi-objetivo, nenhuma é melhor que a outra: no entanto, ao referenciá-las para o mundo real (o que só pode ser realizado pelo usuário, já que o sistema e o computador não possuem tal capacidade cognitiva), uma acaba se mostrando mais interessante que a outra. No exemplo da geração da tabela de tempos e escalonamentos de juízes para trabalhos escolares, imagine que dois dos objetivos sejam utilizar juízes com experiência no assunto do trabalho e distribuir ao máximo a carga de trabalho de todos os juízes. No caso de muitos trabalhos aparecerem acerca do mesmo tópico, os objetivos serão conflitantes. Haverá soluções nas quais se privilegia a utilização de juízes com experiência no assunto e soluções nas quais se privilegia a distribuição de carga horária na fronteira de Pareto. Não há como o sistema saber qual é a mais adequada no momento, mas um usuário, por estar ligado ao mundo real e experienciando os fatos, sabe que naquela semana é mais importante que os juízes tenham cargas horárias bem divididas. Portanto, entre as soluções apresentadas pelo SSTD, ele apontará uma que atenda este requisito, e esta será a solução final.

Assim, no caso de uma fronteira de Pareto, pode-se apresentar as soluções na fronteira para que o usuário decida entre uma delas, já que do ponto de vista da otimização multi-objetivo nenhuma é melhor que a outra. O sistema não dará uma solução final, mas ele irá separar as decisões que podem ser escolhidas das que não devem ser escolhidas. Em sistemas otimizantes, a utilização de tal técnica é considerada um problema, já que o sistema não é capaz de encontrar uma solução única e final para apresentar. No entanto, no caso de SSTD, nos quais o usuário supostamente troca informações com o sistema, essa abordagem é considerada uma vantagem. Para usar esta vantagem, conforme dito, basta deixar que o usuário controle a escolha de uma dessas soluções através de algum parâmetro de entrada. A abordagem RPM apresenta duas maneiras do usuário realizar esta escolha:

1. coeficientes de ponderação, especificando um peso para cada um dos objetivos. A decisão referente ao objetivo de maior peso é escolhida;
2. utilizando objetivos específicos no espaço de busca. A decisão que alcançar ou se aproximar mais do objetivo é escolhida.

Com relação a interfaces gráficas, os autores acreditam que o papel desse componente de um SSTD é:

- organizar o SSTD e ajudar a organizar o processo de decisão;
- prover os comandos que o usuário possui e prover determinadas formas de interação;
- apresentar ao usuário os resultados e análises realizados pelo SSTD.

A abordagem dos autores com relação a interfaces gráficas é interessante e sofisticada, principalmente no que diz respeito a consciência de que a mesma deve estabelecer diversas maneiras de interação com o usuário, ajudando-o a organizar os processos de decisão. Apesar de aprofundar mais este conceito além do simples “amigável ao usuário”, falta um tratamento da interface como facilitadora da participação do ser humano no processo de decisão em si, ou seja, na incorporação de suas intuições e de suas informações relevantes com relação ao mundo real, as quais não podem estar presentes em nenhum tipo de SSTD e que são importantes para que a solução encontrada seja a melhor possível com relação ao contexto real existente naquele instante.

A abordagem dos autores é muito forte no aspecto de otimização, utilizando multi-objetividade e Pareto-otimalidade e eficiência da maneira mais adequada para SSTD. No entanto, apesar do usuário ser capaz de escolher entre as soluções disponíveis na fronteira de Pareto, realizando alguma forma de referência ao mundo real, ele não participou do processo de construção daquelas soluções. Falta na metodologia alguma maneira do usuário incorporar seus intuições na construção dessas soluções que irão popular a fronteira de Pareto, para depois escolher uma delas. A busca por uma solução ótima na abordagem também pode ser prejudicial na construção de SSTD a serem aplicados como sistemas críticos de tempo real.

3.2.2 Abordagens híbridas humano-computador de sistemas de suporte à tomada de decisão

A abordagem híbrida humano-computador é uma fusão de sistemas descritivos e prescritivos (vide seção 3.1.5), o que significa que, ao invés de usar os dois tipos cada um de uma vez, ela utiliza os dois de uma só vez. Dessa maneira, é possível para o agente humano analisar as soluções sugeridas pelo sistema (prescritivo), mudando valores de variáveis e analisando cenários “o que aconteceria se” (descritivo), e adaptá-las para serem utilizadas no mundo real. No entanto, o usuário não é capaz de participar do processo de construção da solução em si, incorporando suas intuições nesta construção. Caso os julgamentos do sistema tenham sido equivocados, ele será capaz apenas de perceber este fato ao se deparar com a solução final, que não terá o efeito esperado. Ele terá a oportunidade de aprender, apagar a solução e começar novamente. Por isso, é considerada uma interação fraca, uma abordagem distribuída híbrida humano-computador, não adequada.

Beynon et al. (2002) sugerem um novo paradigma de SSTD cognitivos utilizando um arcabouço que eles denominam de “centralizado no ser humano” e baseados nos conceitos de “observável”¹⁰ (entidades do mundo que serão representadas para construir um modelo), “ dependência”¹¹ (inter-

¹⁰Tradução deste autores para o termo original do inglês “observable”.

¹¹Tradução deste autores para o termo original do inglês “dependency”.

conexões e relações entre as partes do modelo) e “agência”¹² (ação que modifica o modelo e a representação mental, em direção a uma solução). Os autores consideram o uso de planilhas como uma das melhores maneiras de melhorar a qualidade da interação entre humanos e computadores. Segundo eles, a idéia principal por trás das planilhas, que não foi explorada devidamente até hoje no paradigma de computadores e programação, é a de mudança de estado através dos conceitos de “dependência” e “agência”. Ou seja, os conceitos de estado e de mudança de estado são fundamentais, e realizados através da interação e modificação da forma como o problema é representado durante a execução do programa, sendo o usuário capaz de interpretar o estado atual de uma maneira que não tenha sido pré-concebida. A planilha funciona como uma representação da situação na mente do humano na tomada de decisão, e as possíveis redefinições das células representam suas expectativas acerca das consequências da mudança de estado devido a suas ações em determinada situação. Assim, o conteúdo das células representa o que está atualmente gravado como a situação representacional da mente, sendo possível adicionar novas células e novas definições na planilha. Isso significa que o conteúdo reflete o que o humano considera serem os “observáveis” mais importantes para caracterizar a situação. O responsável pela tomada de decisão deve possuir um bom entendimento da situação localmente, que é adquirida através da experiência, para alocar novos valores para as células da planilha de maneira a fazer boas previsões. Em situações pouco familiares e complexas, faz-se necessário explorar maneiras mais poderosas de interação, como adicionar novas células e modificar as “dependências” existentes. Dessa maneira, a abordagem se baseia em hipóteses empíricas sobre os mecanismos de mudança de estado, levando em consideração a “agência” do computador, a participação humana e o ambiente. O computador é considerado como um dispositivo confiável de mudança de estado, e a programação é considerada não somente algoritmos e tratamento de dados, mas também a configuração de uma série de componentes que formam um sistema, inclusive os agentes humanos. Devido à ênfase na experiência e observação, os autores classificam sua iniciativa no escopo do “Modelamento Empírico”¹³ (ME). ME possui um paradigma baseado na experiência como ingrediente principal. A idéia chave é usar experiências geradas em computador como uma alternativa de representação do conhecimento. A natureza experimental e situada dos problemas de negócio não permitem uma formulação abstrata sem realizar simplificações que acabam removendo algumas dificuldades que deveriam ser enfrentadas e resolvidas, dando origem a representações de problemas diferentes dos originais. O suporte cognitivo no caso do ME vem a partir das experiências para representar essas dificuldades e conceitos de ciências sociais e negócios. Há uma série de scripts escritos para caracterizar as “dependências”, e a facilidade de revisão desses scripts, ou seja, de interação com o modelo, fornece uma maior liberdade com relação ao planejamento e desenvolvimento desses scripts, que representam o conhecimento do humano acerca do domínio, de uma maneira similar a que uma planilha incorpora às operações mentais do usuário, e ao mesmo tempo servindo como uma maneira de representar um modelo computacional evolutivo desse mesmo domínio. Há uma qualidade situacional nesse modelo, pois ele advém da experiência pessoal do usuário, estando conectado a uma referência do mundo real através de comparações e interações do usuário com o mundo. Devido a essa qualidade e esse estilo evolutivo, os autores denominam esses modelos de “modelos de situações interativas”¹⁴ (MSI). Esses modelos podem ser atualizados por múltiplos usuários com diferentes pontos de vista e conhecimentos interagindo num modelo comum e tentando chegar a um

¹²Tradução deste autores para o termo original do inglês “agency”.

¹³Tradução destes autores para o termo original do inglês “Empirical Modelling” (EM).

¹⁴Tradução destes autores para o termo original em inglês “interactive situation models” (ISM).

consenso a partir de um objetivo, podendo ser utilizado para suportar times colaborativos. O objetivo dessa abordagem é criar artefatos no formato de MSI, nos quais a observação humana alinhada com estados computacionais e suas derivações, dependências e atualizações, tragam uma maneira mais adequada de interação homem-computador, com um suporte cognitivo baseado na relação que o humano é capaz de estabelecer entre as representações abstratas e os seus relativos no mundo real. Beynon et al. (2002) apresentam dois exemplos de aplicação desse paradigma, um na construção de uma tabela de horários de apresentações de alunos, similar ao dado como exemplo para ilustrar o cenário em questão neste capítulo, e um MSI para o projeto de um elevador.

A conclusão de Beynon et al. (2002) acerca da necessidade de melhorar a interação entre humanos e computadores nos novos paradigmas de construção de SSTD, somados ao entendimento de que esses novos paradigmas devem atacar de alguma maneira o aspecto cognitivo do processo de tomada de decisão é também adequada. No entanto, o paradigma apresentado por esses autores, apesar de apontar no sentido desses argumentos e ter obtido sucesso nas aplicações apresentadas, ainda não apresenta a maneira mais adequada de interação homem-computador, e tampouco um tratamento cognitivo adequado aos processos de tomada de decisão. No paradigma apresentado, onde planilhas são como metáfora de estados a serem atualizados pelo usuário humano, que relaciona os conceitos abstratos aos referentes no mundo real apresentado, os autores não especificaram como deve ser desenvolvida a linguagem de comunicação entre computador e agentes humanos, apesar de estarem conscientes da necessidade de uma linguagem adequada. Isso não foi descrito no trabalho, sendo isso uma falha de muitas abordagens, a falta dos mecanismos que devem ser desenvolvidos para tornar a interação homem-máquina mais adequada. O suporte à cognição ocorre apenas através de observações e experiências do usuário humano. No entanto, há uma necessidade de entender como ocorre o fluxo do pensamento humano no processo de tomada de decisão, quais entidades são representadas e como se dá as operações mentais deste usuário, para que sejamos capazes de construir sistemas que possam cooperar com esse fluxo de pensamento. Isso tampouco é explorado no paradigma apresentado por eles. Não se apresenta como deve ser a interação entre humanos e computador para atualização das células da planilha (estados), mas apenas que estas devem existir. Um screenshot da aplicação de tabela de horários é mostrado, evidenciando uma forma de representação dos conceitos pobre e conseqüentemente uma linguagem entre humanos e computador (interface gráfica) pouco desenvolvida. Os autores chegam a afirmar que melhores interfaces gráficas e linguagens de comunicação devem existir e ser alvo de esforços, mas não dizem o que uma linguagem dessas deve possuir para ser adequada, ou que mecanismos devem existir para suportar uma melhor interação entre humanos e computadores. Dessa maneira, a abordagem é muito bem intencionada e aponta na direção correta, mas faltam conceitos que materializem e formalizem alguns argumentos apresentados, que acabam tornando-se vaporosos.

Chen & Lee (2003) apresentam algumas outras “ferramentas cognitivas” relacionadas com o processo descrito por Beynon et al. (2002). O intuito dessas ferramentas é sempre o de complementar o modelo simbólico utilizado no SSTD, oferecendo a possibilidade de agregar informação advinda das experiências do usuário com o mundo real, determinando uma espécie de interação entre as entidades do modelo e seus referentes no mundo real. A este novo paradigma de modelagem computacional, o autor denomina “modelos de situações interativas” (MSI), conforme citado por Beynon et al. (2002) anteriormente. Os autores dão exemplos de quatro MSI e explicam as contribuições ao suporte efetivo de tomada de decisão, entre eles uma aplicação de operação de ferrovias e uma aplicação de tabela de horários.

Uma bem sucedida abordagem híbrida humano computador é a de Hsu et al. (1993), onde Hsu et al. descrevem um SSTD para escalonamento chamado MacMerl. O sistema é composto por dois módulos, sendo um deles responsável por gerar escalas automaticamente enquanto o outro permite que o usuário humano construa uma solução, podendo utilizar esta solução construída com rotinas no núcleo do escalonamento. Os autores chamam a abordagem de “escalonamento de iniciativa híbrida”¹⁵, na qual humano e computadores atuam de maneira cooperativa para construir soluções para problemas não estruturados de escalonamento. O que falta para a abordagem é uma maneira eficiente e adequada de entender melhor os processos cognitivos humanos na busca pela solução para que processos computacionais sejam inseridos no sistema de maneira a suportar a decisão humana da maneira mais eficiente possível, para poder aplicar poder computacional exatamente nos pontos em que o agente humano possui dificuldade de lidar.

3.2.3 Sistemas de Amplificação de Inteligência

Courtney (2001) coloca que a globalização trará cada vez mais problemas não estruturados para as organizações, e que novos métodos serão necessários para construir SSTD que possam ser efetivos nestas situações. Segundo ele:

“Planilhas e outros modelos, SSTD baseados em otimização e SSTD baseados em grupos podem ajudar a resolver esses problemas. No entanto, parece que ferramentas mais poderosas são necessárias. Especialmente necessária é uma perspectiva mais abrangente em termos de pesquisa em SSTD.”

Para Chen & Lee (2003):

“Um sistema de suporte à tomada de decisão deve ser parte de um ambiente de colaboração e aprendizado humano-computador onde o computador realiza um papel mais ativo em facilitar o pensamento criativo do tomador de decisão, fornecendo ferramentas para trazer a tona crenças e premissas tácitas, além de ajudar a sua sequência de raciocínios.”

Sistemas de Amplificação de Inteligência (SAI) são essencialmente sistemas computacionais que, por construção, amplificam a inteligência da mente que está a cargo da solução de problemas de planejamento e execução dinâmicos. Em outras palavras, são sistemas que apoiam tomadas de decisão inteligentes baseadas na cooperação resultante de um diálogo entre agentes humanos operando o sistema (usuários) e o próprio sistema. O principal resultado é que a decisão final não é realizada nem pelo ser humano nem pelo sistema isoladamente, e sim pelo time sistema-usuário, que é um sistema cognitivo distribuído (Giere & Moffatt, 2003). Os SAI consideram, por construção, a participação humana na criação da solução para planejamentos e tomadas de decisão em ambientes dinâmicos, o que significa que o agente humano irá construir parte da solução final. Por esta abordagem considerar uma interação de fato entre a inteligência artificial do sistema e a inteligência humana, é considerada mais adequada do que a centralizada no computador, na qual não há interação, e do que a híbrida humano-computador, na qual há uma interação fraca. No caso dos SAI, delega-se partes específicas da resolução do problema para o sistema computacional:

¹⁵Tradução destes autores para o termo original do inglês “mixed-initiative scheduling”.

- Tarefas estruturadas, determinísticas, mecânicas ou aleatórias: facilmente realizadas pelo computador;
- Tarefas não-estruturadas, mal-definidas: realizadas por um diálogo entre humano e sistema computacional.

Assim, há um processo de cognição distribuída (Giere & Moffatt, 2003) entre os artefatos e o humano, que é liberado para se concentrar nas tarefas que demandam inteligência. Considera-se que a parte da solução construída pelo ser humano trará o entendimento ao sistema acerca dos referentes do mundo real relacionados aos símbolos nele representados, além de informações e atualizações (visto que o ambiente e o planejamento são dinâmicos) que fogem aos modelos e ao computador em si. Além disso, a parte da solução construída pelo ser humano também irá tratar os problemas não estruturados presentes no ambiente, os quais requerem um julgamento humano qualitativo que as máquinas não podem realizar para diferentes contextos, pela dificuldade semântica inerente.

Os SAI estão fundamentados em duas áreas do conhecimento:

- Semiótica computacional: área da pesquisa de sistemas inteligentes onde a noção de “signo” é usada de maneira explícita, herdando parte da teoria da semiótica, com sistemas artificiais que realizam semiose e manipulam signos, conforme discutido no capítulo 4.
- Amplificação de inteligência: tarefas cognitivas que seriam realizadas por um ser humano são realizadas por um sistema artificial, com o intuito de melhorar a eficiência de um processo específico, deixando o ser humano livre para se concentrar em outras partes do processo e amplificando seu intelecto, conforme discutido no capítulo 5.

O objetivo deste trabalho é apresentar uma engenharia de SAI, em outras palavras, uma metodologia para o desenvolvimento sistemático e fundamentado de SAI. Esta metodologia está centrada na própria definição de SAI, ou seja, primeiro define-se quais são os atributos que um SAI deve ter para garantir sua efetividade para depois definir uma metodologia que garanta a presença destes atributos no sistema em desenvolvimento.

Os atributos que um SAI deve ter para que ele alcance os resultados esperados são apresentados na seção 5.5.1. Se fosse necessário resumir estes requisitos em apenas dois, seriam os seguintes:

1. pensamentos exosomáticos: o processo de pensar deve extrapolar o corpo humano. O sistema tem que trabalhar em simbiose com o ser humano (usuário). Com a utilização do SAI, a mente que resolve o problema deve ser a mente formada pela união das mentes dos agentes humanos e computacionais envolvidos.
2. diálogo: os agentes computacionais devem “dialogar” com os agentes humanos. O SAI deve fornecer meios pelos quais estes diálogos devem ocorrer (linguagem de comunicação, meio de comunicação, etc).

Um SAI é uma dualidade, pois ao mesmo tempo que sugere uma solução para a tomada de decisão, fornece uma linguagem, uma interface de diálogo para as soluções dos agentes humanos e computacionais, de maneira que possa adequar-se a como o pensamento humano ocorre e absorver suas intuições, advindos da importante relação que eles realizam entre a solução abstrata e o mundo real.

A metodologia que será apresentada no capítulo 6 é, em sua essência, uma “engenharia de SAI”. Ela se fundamenta em quatro ferramentas:

- semiótica computacional (capítulo 4): ferramental utilizado para tratar os aspectos cognitivos e interface homem-máquina;
- amplificação de inteligência (capítulo 5): ferramental utilizado para orientar os requisitos do SAI (diálogo com intencionalidade, formulação, escolha e execução de planos, etc);
- agentes computacionais com motores de solução (seção 5.4): ferramental utilizado para implementar autonomia inteligente e contínua nos processos.
- engenharia de software (seção 6.1): uma vez que o sistema de amplificação de inteligência se consubstanciará em um sistema computacional, ou seja, em um software, é natural que a engenharia de SAI seja um complemento da engenharia de software;

A semiótica computacional irá abordar a sequência das atividades cognitivas realizadas pelo especialista humano e representá-las, para que haja um maior entendimento de como ocorrem esses processos e para que os agentes computacionais carregando motores de solução, presentes no sistema e trabalhando de maneira autônoma para suportar a decisão e amplificar a inteligência continuamente, possam se adequar a essas atividades, enquanto a amplificação de inteligência irá focar no desenvolvimento de atributos necessários ao sistema computacional para que haja a interação entre humano e computador. Deseja-se que a mente humana seja utilizada no processo de tomada de decisão nas tarefas que ela realiza melhor (julgamentos qualitativos, problema de fundamentação do símbolo e informações dinâmicas do mundo real incorporadas aos planos), enquanto os processos computacionais sejam utilizados para o que eles fazem de melhor (poder de processamento, exploração do espaço de busca combinatório), de tal maneira que a solução final seja o resultado de um processo evolutivo entre intuições inseridas pelo humano e atividades estruturadas realizadas pelo computador, chegando a uma solução melhor do que a que cada um deles chegaria separadamente. Com efeito, problemas não estruturados apresentam situações nas quais há um número de atividades cognitivas a serem realizadas que, para serem realizadas em tempo hábil, necessitam de um sistema cognitivo complexo (Giere & Moffatt, 2003), do qual humanos são apenas componentes. Nenhum humano seria capaz de fisicamente realizar todas as tarefas necessárias sem que as tarefas cognitivas estejam distribuídas entre eles e os artefatos de amplificação nesses casos (Giere & Moffatt, 2003).

Faz-se necessário entender os processos cognitivos humanos na busca pela solução, para poder entender como processos computacionais podem amplificá-los. Para tal, precisa-se de uma fundamentação teórica adequada para entender a representação dessas operações mentais. Neste trabalho, propomos a semiótica computacional como esta fundamentação. Os atributos necessários aos SAI e o arcabouço de representação mental, assim como o processo de desenvolvimento, serão apresentados no capítulo 5.

3.3 Resumo

Este capítulo apresentou o primeiro bloco teórico necessário para introduzir o conceito de Sistemas de Amplificação de Inteligência: os sistemas de suporte à tomada de decisão. Foram explicadas

suas definições e sua composição, bem como motivação para sua existência e problemas enfrentados no presente e no futuro por esta categoria de sistema.

Como contribuições deste capítulo, pode-se citar a apresentação de explicações para o fato de sistemas otimizantes falharem em aplicações típicas de sistemas de suporte à tomada de decisão, onde um agente humano (usuário) e suas intuições realizam uma importante parte do trabalho de encontrar uma solução boa que possa ser aplicada no mundo real.

Finalmente, o capítulo apresentou a definição de Sistemas de Amplificação de Inteligência, sendo extensões de SSTD onde há interação homem-máquina na construção da solução, ou seja, uma cognição distribuída (Giere & Moffatt, 2003) entre o humano usuário e o artefato de amplificação máquina, estabelecendo que esses sistemas cognitivos distribuídos são capazes de resolver problemas não estruturados em ambientes dinâmicos, processos de planejamento e execução dinâmicos.

Sistemas otimizantes resolvem modelos, sistemas de amplificação de inteligência, com a correta abordagem desenvolvida neste trabalho, resolvem problemas do mundo real, pois há humanos participando da construção da solução, comprometidos em entender os processos e estabelecendo as conexões semânticas necessárias com o mundo real.

Capítulo 4

Semiótica: representação e significação em operações mentais

Neste capítulo, são apresentados importantes conceitos da semiótica e de sua ramificação conhecida como semiótica computacional. Esses conceitos são necessários para o entendimento da metodologia apresentada neste trabalho. A utilização de conceitos da semiótica é a base do processo de desenvolvimento de software apresentado e também um novo paradigma nessa área de engenharia de sistemas inteligentes. Estes conceitos serão posteriormente utilizados para modelar e representar operações mentais dos seres humanos para ajudar a entender como sua inteligência pode ser amplificada através de um sistema computacional. Esta amplificação se dá através da substituição de determinadas atividades cognitivas por processos computacionais que dialoguem com o agente humano.

A busca pela solução de problemas não estruturados fomenta a interação de diversas áreas e conseqüentemente a criação de trabalhos interteóricos. A interdisciplinaridade deste trabalho, utilizando-se de conceitos advindos da semiótica para criar uma metodologia de construção de sistemas de amplificação de inteligência, não é uma simples busca por um novo paradigma: a utilização da semiótica de Charles Sanders Peirce (1839-1914) tem fundamentos fortes neste contexto. A metodologia aqui proposta propõe-se a criar sistemas que sejam capazes de amplificar a mente do agente humano na resolução de problemas, integrando os agentes computacionais com os agentes humanos e focando a máquina nos processos cognitivos que os humanos realizam de forma ineficiente. Para isto, faz-se necessário entender como a informação se transmite, processa e como as representações mentais e os processos de semiose (cognição) do agente humano ocorrem. A semiótica trata exatamente disso.

Na seção 4.1, apresenta-se os conceitos da semiótica e sua relação com representações mentais. Faz-se ainda nesta seção uma rápida discussão acerca da definição de inteligência. Além disso, apresenta-se um pouco da história da semiótica e as maiores contribuições no campo. Na seção 4.2, fala-se da semiótica de Charles Sanders Peirce, e apresenta-se seus conceitos pertinentes ao desenvolvimento deste trabalho. Na seção 4.3, fala-se de semiótica computacional e seus conceitos. A seção 4.4 apresenta o resumo das contribuições do capítulo.

Para tornar algumas das citações de Peirce feitas neste capítulo mais destacadas, há uma representação especial para elas que é adotada pelos estudiosos da semiótica. Quando precedidas de MS e feitas entre parênteses, referem-se aos manuscritos inéditos, conforme a numeração feita por R. S. Robin em *Annotated Catalogue of the Papers of Charles S. Peirce*, Amherst, 1967; aquelas precedidas por CP e feitas entre parênteses, são dos *Collected Papers of Charles Sanders Peirce*, Vols. 1 a 6, editadas

por C. Hartshorne e P. Weiss; volumes 7 e 8, editadas por A.W. Burks, Cambridge, 1931-1958. O primeiro algarismo é o número do volume e a numeração subsequente, a do parágrafo. Por exemplo, (CP3, 144) se refere ao parágrafo 144 do volume 3 dos *Collected Papers*, conforme anteriormente. As precedidas por EP e feitas entre parênteses, referem-se à obra *The Essential Peirce. Selected Philosophical Writings.*, volume 1 (1867-1893) (editado por Nathan Houser e Christian Kloesel) e volume 2 (1893-1913) (editado por “the Peirce Edition Project”), em Bloomington e Indianapolis, da Indiana University Press, 1992 e 1998.

4.1 Introdução

Semiótica é a disciplina ou doutrina que estuda os signos, as representações e os fenômenos de significação e comunicação. A etimologia da palavra está relacionada ao grego *semeiotikos*, estudo dos signos. Com efeito, não há um consenso a respeito de uma definição única e precisa para semiótica, havendo entre os muitos autores da área tantas definições para a “ciência dos signos” que alguns deles, como Greimas, são capazes de refutar que a semiótica seja definida como uma ciência dos signos (Nöth, 1995). Dessa maneira, há uma variedade de questionamentos que são realizados a respeito da semiótica, se é uma ciência ou um movimento, um projeto ou uma doutrina, uma moda ou uma revolução. Fato é que a semiótica e os signos nunca foram tão explorados como atualmente, por diferentes pontos de vista. A interdisciplinaridade é uma característica dessa área, sendo a pesquisa em semiótica levada adiante por diferentes pesquisadores, como lingüistas, lógicos, filósofos, sociólogos, psicólogos, antropólogos, biólogos e cientistas da computação. No entanto, o status de ciência ainda é discutido quando se fala em semiótica (Nöth, 1995), e somente aos poucos a semiótica vai se estabelecendo como uma disciplina respeitada.

Representação mental tem uma estreita ligação com cognição e inteligência. A sobrevivência das espécies depende da sua adaptação às modificações do ambiente em que se encontra. Levando em consideração o fato de que a cognição é o meio pelo qual os ajustes comportamentais podem acontecer, a representação é o meio que torna essa evolução possível, visto que as representações mentais são necessárias para a cognição (Queiroz, 2004). Um cachorro que sai de casa para dar umas voltas na cidade e depois consegue retornar, hipoteticamente precisa se utilizar de representações para se deslocar de maneira orientada no espaço, em vista do fato de não possuir mapa, bússola ou qualquer tipo de instrumento de navegação. Portanto, ele precisa representar a região do espaço onde se desloca, de alguma maneira. Assim, placas, mapas, partituras musicais, fórmulas matemáticas e todas as tarefas que fazemos dependem de representações; ao representar algo, o substituímos por algum tipo de processo (Queiroz, 2004). Utilizar o ferramental semiótico é uma maneira de entender essas representações das diversas tarefas realizadas, como elas são produzidas, sua lógica de funcionamento, as estruturas a elas associadas e como simulá-las em sistemas computacionais. Assim, pode-se ganhar em rigor, precisão e generalidade ao utilizar-se desse ferramental para desenvolver uma metodologia de criação de sistemas computacionais mais inteligentes, apesar de se perder em inteligibilidade.

A busca pelo desenvolvimento de sistemas computacionais mais inteligentes e capazes de realizar cognição tem sido realizada nos estudos de inteligência artificial. Essa busca tem sido duramente criticada, principalmente após o fracasso da agenda de pesquisa dessa área montada no seu início, na década de 1950. Essa polêmica é causada em parte pelo fato do próprio conceito de inteligência não possuir uma definição geral e completa. Somos capazes de dizer que inteligência envolve conheci-

mento, raciocínio, pensamentos, idéias, capacidade de resolver problemas, capacidade de compreender situações em diferentes contextos, capacidade de planejar o futuro e realizar as ações necessárias para que o plano se concretize, capacidade de aprendizado, entre outros atributos que podemos identificar em um sistema que consideramos inteligente. Sabemos também diferenciar os variados tipos de inteligência: inteligência lógico-matemática, inteligência musical, inteligência físico-motora (típica de atletas), inteligência emocional, inteligência inter-pessoal (sociabilidade), enfim, podemos dizer os diferentes domínios da inteligência e sua intensidade em cada um deles, sendo uma propriedade composta e derivada. Nos achamos capazes, assim, de classificar os diferentes sistemas em mais ou em menos inteligentes, e de dizer que a inteligência é tipicamente desenvolvida nos seres humanos. No entanto, nos falta uma definição completa e geral.

Rolf Pfeiffer, em seu livro “Understanding Intelligence”, afirma que, em 1921, o *Journal of Education Psychology* (vol. 12, pp. 123-147, 195-216) perguntou a 14 especialistas no ramo qual era a sua definição de inteligência. Como resultado, o JEP recebeu 14 respostas diferentes. Algumas delas foram as seguinte:

- A habilidade de levar adiante pensamentos abstratos (L.M. Terman);
- Ter aprendido ou ter a habilidade de aprender a se adaptar ao ambiente (S.S. Colvin);
- A habilidade de se adaptar adequadamente a novas situações na vida (R. Pintner);
- Mecanismo biológico pelo qual os efeitos de estímulos complexos são agrupados e associados a um efeito unificado de comportamento (J. Peterson);
- A capacidade de adquirir capacidade (H. Woodrow);
- A capacidade de aprender ou se beneficiar das experiências (W. F. Dearborn);

Nota-se que são diferentes concepções, mas todas falam de pontos considerados importantes, como aprendizado e adaptação. Apenas alguns relacionam inteligência ao ambiente, que foi largamente negligenciado em muitas pesquisas.

Importante, e também colocado por Pfeiffer, são os conceitos populares de inteligência. Entre eles, podemos citar a capacidade de pensar e resolver problemas; a competência de falar, ler e escrever; intuição e criatividade; aprendizado e memória; emoções; sobrevivência em um mundo complexo; e consciência. Para todos esses pontos, pode-se falar em graus de inteligência.

No campo da inteligência artificial tradicional, o conceito de inteligência foi desenvolvido associado aos computadores. A idéia de inteligência foi entendida como um programa de computador, recebendo uma entrada que era processada e gerava uma saída. O cérebro humano nada mais era do que um computador super potente, recebendo entradas do mundo através de seus sensores (olhos, ouvidos, pele, etc), processando-os e tendo como saída uma ação ou comportamento. No entanto, quando pesquisadores de IA começaram a aplicar essas idéias na construção de sistemas que interagiam com o mundo real, foi uma decepção. Os robôs tinham grande dificuldade em realizar as tarefas mais simples de maneira correta, como se locomover em um trajeto dentro de um ambiente, reconhecer objetos, trazer objetos para um alvo. Os problemas se demonstraram sérios o suficiente para que os pesquisadores comesçassem a buscar alternativas diferentes.

Sendo a semiótica o estudo dos signos e dos processos de significação, vertentes recentes de estudo de sistemas inteligentes identificam a capacidade de processar signos como a fonte de inteligência, em todas as classes de sistemas. Isso significa que um sistema é inteligente porque processa signos, e a intensidade de sua inteligência irá depender da quantidade e da variedade dos signos que ele pode processar. Tomando esses conceitos como base, estudar semiótica passa a ser a chave para o entendimento dos sistemas inteligentes e para a criação de sistemas mais inteligentes.

A semiótica tornou-se relevante porque estuda como o significado é construído e entendido, a base para o entendimento dos fenômenos da cognição e comunicação. O estudo da cognição visa entender como ocorre a captura e o processamento dos fenômenos do ambiente enquanto o estudo da comunicação visa compreender como o conhecimento produzido no processo cognitivo é transmitido entre os seres inteligentes. A semiótica se aproxima muito da semiologia em seus objetivos, porém se distingue no enfoque utilizado, uma vez que a semiologia é direcionada à lingüística. Como resultado de tantos estudos e relevância, a semiótica está atualmente cada vez mais aplicada pragmaticamente. Entre suas aplicações, estão a análise de textos e mensagens em diferentes contextos, apreciação de meios áudio visuais, estratégias de comunicação para meios de massa (TV, internet), análise crítica de discursos e projeto de situações onde a interação entre humanos e o ambiente precisa ser mais eficiente, como na arquitetura, no desenvolvimento de interfaces homem-máquina e configuração de instrumentos para auxiliar o ser humano em suas atividades.

É difícil ter um conhecimento claro da história da semiótica, devido ao fato de muitas contribuições a ela feitas terem surgido em campos diferentes do conhecimento, fazendo com que o desenvolvimento da teoria dos signos não coincida com a pesquisa chamada semiótica (Nöth, 1995). Um historiador preocupado em considerar a história de todas as disciplinas relacionadas à semiótica pode acabar confundindo a história da semiótica com a da ciência em si. No entanto, não se pode considerar apenas a pesquisa explicitamente realizada sob o termo semiótica. Por isso, Nöth (1995) coloca que levar em consideração teorias da antiguidade a respeito da natureza dos signos não é criar uma falsa genealogia. Apesar de idéias semióticas da natureza dos signos e da significação terem sido desenvolvidas na antiguidade e na idade média, uma teoria geral dos signos não surgiu antes do período da semiótica moderna de Charles Sanders Peirce.



Fig. 4.1: Charles S. Peirce - o pai da semiótica

O uso mais antigo do termo semiótica está relacionado com um ramo da medicina, relacionado aos estudos de sintomas e diagnóstico de doenças. Galeno de Pérgamo (139-199) referenciava diagnóstico como um processo de semiose, ou seja, cognição (Nöth, 1995). No século XVIII, “semiotica” ou “semiotique” tornou-se um termo médico oficial para a doutrina dos sintomas na Europa. A primeira

referência no campo da filosofia é feita por Locke, no seu *Essay Concerning Human Understanding*, onde a doutrina dos signos é definida porém não desenvolvida, apenas colocada como uma proposta para uma futura disciplina. Ela só foi sistematizada por Peirce.

A história da semiótica pode ser subdividida da seguinte maneira (Nöth, 1995):

- História antiga (greco-romana, até Santo Agostinho (354-430)), medieval (até Dante) e renascentista (até Campanella): neste período surgiram a lógica, a retórica, a poesia, o desenvolvimento da semântica e da natureza do signo. Entre os personagens dessa época estão Platão (427 a.c.-347 a.c.), Aristóteles (384 a.c.-322 a.c.), os estóicos (300 a.c.-200 d.c.), os epicuristas e Santo Agostinho (354-430), Roger Bacon (1215-1294), John Duns Scotus (1270-1308), William de Ockham (1290-1349). Para saber mais sobre suas contribuições, refira-se a Nöth (1995).
- Iluminismo e racionalismo: durante o século XVII, as tradições da antiguidade e medievais da semiótica foram deixadas de lado, e por isso Locke em 1690 pareceu inovar ao postular a semiótica como uma área da ciência (Nöth, 1995). Esse postulado foi levado em consideração nesta época, e uma teoria formal dos signos começou a ser produzida. Nessa era do racionalismo, que dava ênfase ao poder intelectual do homem, foram desenvolvidos dois tópicos importantes para a semiótica. Um deles era a noção mental de signo, enquanto o outro era a procura por uma gramática universal de todas as línguas, para que fosse construída uma língua universal. Entre os personagens dessa época que contribuíram neste sentido, pode-se citar Descartes (1596-1650), Francis Bacon (1561-1626), Leibniz (1646-1716), Hobbes (1588-1679), Locke (1632-1704), Berkeley (1685-1753), Condillac (1715-1780), Diderot (1713-1784), Lambert (1728-1777) e Kant (1724-1803). Todos estes filósofos possuem contribuições importantes que podem ser vistas em Nöth (1995).
- Semiótica moderna: surgimento das teorias modernas de significado, semântica lingüística e início da pesquisa em linguagem. Podemos citar entre os filósofos que contribuíram Hegel (1770-1831), Humboldt (1767-1835), Bolzano (1781-1848), Peirce (1839-1914), Lady Welby (1837-1912), Husserl (1859-1938), Cassirer (1874-1945), Bühler (1879-1963), Uexküll (1864-1944), Whitehead (1861-1947), Russel (1872-1970), Wittgenstein (1889-1951), Carnap (1891-1970) e Sebeok (1920-2001). Hoje em dia há um grande número de pesquisadores na área em todo o mundo.

Todos os autores citados acima, ao longo da história do desenvolvimento da semiótica, possuem sua teoria de representação e significação de operações mentais, cada um com sua definição de signo e significação. Neste trabalho, utiliza-se os conceitos da semiótica peirceana, desenvolvida por Charles Sanders Peirce. Esses conceitos serão apresentados neste capítulo, pois serão utilizados como ferramental para o desenvolvimento da metodologia proposta neste trabalho.

4.2 A semiótica de Charles Sanders Peirce

Charles Sanders Peirce é considerado o fundador da teoria moderna dos signos (Burks & Weiss, 1945). Ele definiu a semiótica como “a doutrina da natureza essencial e fundamental de todas as

variedades possíveis de semiose” (Peirce, 1960), ou seja, de qualquer tipo de cognição e seus desdobramentos. Apesar de ser aclamado hoje em dia com um dos maiores filósofos americanos, ele foi completamente ignorado pelos seus contemporâneos, em parte por ter escrito milhares de artigos que ficaram muito tempo sem serem publicados. Na parte mais avançada de sua vida, Peirce isolou-se e desenvolveu sua teoria de maneira introspectiva, escrevendo uma obra extensa sem apresentar as idéias a quase ninguém, exceto por algumas correspondências que mantinha com algumas figuras da época, entre elas Victoria Lady Welby. Apenas em meados do século XX é que seus manuscritos foram explorados e deram origem a projetos de edição, como o Peirce Edition Project, Indiana University Press. A obra publicada mais conhecida são os *Collected Papers of Charles Sanders Peirce*, em 8 volumes, sendo volumes I a VI editados por Paul Hartshorne e Paul Weiss no período de 1931-1935 e os volumes VII e VIII editados por Arthur W. Burks em 1958. Há outras obras importantes, como os *Writings of Charles Sanders Peirce*. No entanto, parte de seus manuscritos ainda continuam não publicados e não organizados cronologicamente, o que representa um problema, já que ao longo do tempo Peirce realizou algumas modificações na sua teoria e na sua nomenclatura. Peirce nunca escreveu um esquema coerente e completo de sua teoria dos signos, o que dificulta o acesso aos conceitos postulados por ele (Nöth, 1995). Além disso, sua escrita rebuscada e prolixa faz com que seu texto não possua uma inteligência imediata, tornando alguns conceitos de sua semiótica formal difíceis de serem captados. O conjunto desses fatores parece tornar sua obra um tanto quanto hermética, mas, uma vez compreendidos seus conceitos, nota-se que há ali uma poderosa ferramenta de representação e significação.

O trabalho de Peirce se concentra no ramo da semiótica epistemológica (estudo do conhecimento) e metafísica, tendo sido desenvolvida independentemente do ramo lingüístico de Saussure e Hjelmslev. Por isso, a princípio, sua semiótica foi reconhecida no campo filosófico, somente mais tarde sendo reconhecida na lingüística. Em verdade, a teoria de Peirce tem o objetivo de ser universal.

A principal característica da teoria de Peirce é a sua visão “pansemiótica” do universo. Segundo essa visão, que será melhor compreendida no próximo tópico, ao serem apresentados os principais conceitos da semiótica Peirceana, objetos, sensações, pensamentos e até mesmo pessoas são semióticas em sua essência. Como acontece com um signo, um pensamento sempre se refere a um outro pensamento e a objetos do mundo, e todas essas reflexões fazem referência a algo no passado. Segundo o próprio Peirce:

“O universo inteiro está permeado de signos, se não é exclusivamente composto de signos” (CP5, 448),

ou ainda,

“o fato de que todo pensamento é um signo, considerado em conjunto com o fato de que a vida é um trem de pensamentos, prova que o homem é um signo” (EP1, 54)

Dessa maneira, a semiótica de Peirce estuda os processos semióticos sem se preocupar com a base material no qual eles acontecem (Queiroz, 2004), seja no interior de células (citossemiose), entre os animais (zoossemiose), plantas (fitossemiose), em humanos (antropossemiose) ou até mesmo em robôs e computadores (cyberssemiose).

Para entender melhor o que Peirce quis dizer com sua visão pansemiótica, além de outros conceitos de sua teoria que irão compor um ferramental importante para a metodologia proposta neste trabalho, explica-se a seguir seus conceitos.

4.2.1 Conceitos relevantes da semiótica Peirceana

Todo o ferramental de conceitos da semiótica Peirceana que é relevante para o desenvolvimento da metodologia proposta neste trabalho será apresentado a seguir.

Signo, semiose e semiose ilimitada

Segundo o conceito de Peirce, o signo é uma relação triádica, diferente de alguns outros modelos diádicos propostos por outros semiotistas. Esse modelo triádico pressupõe três entidades: o objeto, o representamen e o interpretante. Para Peirce, o signo é qualquer coisa que possa representar um objeto causando na mente um interpretante idêntico ao que o objeto causaria se estivesse presente de fato. Formalmente, há um primeiro, chamado representamen, que está numa relação triádica para um segundo, o objeto, de forma a ser capaz de determinar um terceiro, seu interpretante (Nöth, 1995).

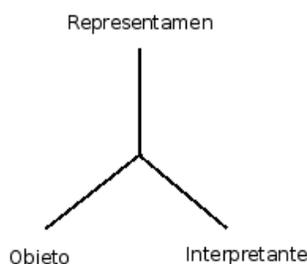


Fig. 4.2: Modelo triádico de signo (adaptado de (Queiroz, 2004))

Em algumas partes de sua teoria, em determinados momentos, Peirce chama o primeiro correlato da relação de “signo”, ao invés de “representamen”. Ele distingue ambos afirmando que o primeiro correlato pode ser chamado de signo se o interpretante ocorre na mente, e pode ser chamado de representamen caso o interpretante ocorra na mente ou fora da mente. Nesse trabalho, será utilizado sempre o termo representamen para representar o primeiro correlato, e signo para representar a tríade completa.

Assim, tem-se uma tripla com representamen, objeto (coisa significada) e interpretante (mediador da relação objeto-representamen, como uma cognição produzida na mente, por exemplo). Para ilustrar com um exemplo: a foto de um lugar querido é um signo deste lugar onde: o objeto é o lugar representado; o representamen é a foto e o interpretante pode ser uma sensação igual a que a visão do próprio lugar causaria. Neste exemplo o objeto é material (a foto), mas este pode ser qualquer coisa, até mesmo uma sensação ou evento capaz de gerar um interpretante na mente. Assim, pode ser uma idéia, sem necessidade de existir no mundo real. O interpretante pode ser uma idéia na mente de um intérprete (humano ou não), atuando como mediador na relação objeto-representamen, mas pode também ser uma ação no mundo real, como ocorre na semiose externa (seção 4.3.1).

Peirce procura definir formalmente as três estruturas do modelo triádico de signo no seu trabalho. As principais definições são:

- Um signo é qualquer coisa que determina qualquer outra coisa (seu interpretante) a se referir a um objeto ao qual ele mesmo se refere (seu objeto) do mesmo modo, o interpretante se tornando por sua vez um signo, e assim por diante, *ad infinitum*. (CP2, 303)

- Um representamen é o sujeito de uma relação triádica de um segundo, chamado de seu objeto, para um terceiro, chamado de seu interpretante, esta relação triádica sendo de tal ordem que o representamen determina que seu interpretante fique na mesma relação triádica com o mesmo objeto para algum interpretante. (CP1, 541)
- Um signo, ou representamen, é um primeiro que está em uma tal relação genuína com um segundo, chamado seu objeto, de modo que seja capaz de determinar um terceiro, chamado seu interpretante, para assumir a mesma relação triádica com seu objeto na qual ele próprio está com o mesmo objeto. (CP2, 274)
- Um representamen é o primeiro correlato de uma relação triádica, o segundo correlato, sendo chamado de seu objeto, e o possível terceiro correlato sendo denominado seu interpretante, por cuja relação triádica o possível interpretante é determinado como o primeiro correlato da mesma relação triádica para o mesmo objeto, e para algum possível interpretante. (CP2, 242)
- O signo é determinado pelo objeto relativamente ao interpretante, e determina o interpretante em referência ao objeto, de tal modo que produza o interpretante a ser determinado pelo objeto através da mediação do signo. (MS318, 81)

Peirce definiu essa ação de gerar um interpretante na mente de um intérprete como semiose. Segundo Peirce, a semiose é um processo cognitivo no qual os signos são interpretados. Uma das características mais relevantes do modelo sógnico de Peirce é o fato de o interpretante gerado em processo de semiose ter a capacidade de atuar como representamen em outro processo de semiose. Dessa maneira, um interpretante gerado pode dar origem a outro interpretante, e assim sucessivamente, *ad infinitum*. Naturalmente, as semioses concatenadas podem durar toda a eternidade, pois elas são potencialmente ilimitadas, dando origem ao conceito que Eco (1976) chamou de “semiose ilimitada”, referindo-se a Peirce. Segundo Eco (1976):

“... para estabelecer o significado de um significante (Peirce fala, não obstante, em “signo”) é necessário nomear o primeiro significante por meio de um outro significante, que a seu turno conta com outro significante que pode ser interpretado por outro significante, e assim sucessivamente. Temos, destarte, um processo de SEMIOSE ILIMITADA. Embora a solução possa parecer paradoxal, a semiose ilimitada é a única garantia de um sistema semiótico capaz de explicar-se a si próprio, em seus próprios termos. A soma das várias linguagens seria um sistema auto-explicativo, ou um sistema que se explica por sucessivos sistemas de convenções a se esclarecerem entre si.”

Não há um primeiro nem um último signo nesse processo que também não é cíclico (Nöth, 1995). Assim, o signo não é somente uma estrutura que relaciona um objeto, um representamen e um interpretante. Ele é, com efeito, uma função recursiva que pode ser aplicada a si própria quantas vezes for necessário, gerando interpretações que se tornam argumentos da próxima aplicação da função (de Souza, 2005). Essa idéia sofisticada do pensamento se refere ao fato de que todo pensamento se dá através de um diálogo, um diálogo entre diferentes fases do ego, sendo composto de signos (CP 4.6). Segundo o próprio Peirce:

“[...] o pensamento sempre acontece na forma de um diálogo [...], sendo dialógico, ele é essencialmente composto de signos” (CP 4.6)

“dizer, portanto, que o pensamento não pode acontecer em um instante, mas requer um tempo, é uma outra forma de dizer que todo pensamento deve ser interpretado em outro, ou que todos os pensamentos se dão em signos” (CP5.253).

Assim, todo pensamento se dá em signos e na forma de um diálogo. Como todo pensamento aponta para outro, ele nunca pode ser finalizado, apenas interrompido (CP 5.284). Teoricamente, um interpretante sempre pode dar origem a outro.

Merrel (1997) afirma que um diagrama para representar a semiose ilimitada deve possuir um padrão de uma rede.

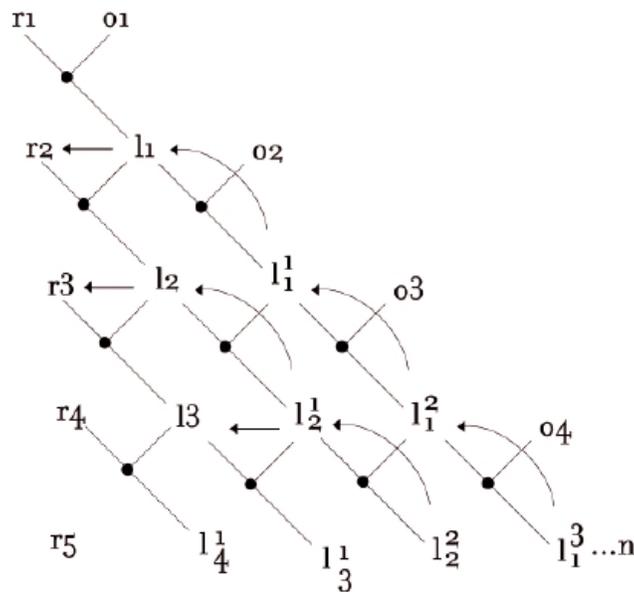


Fig. 4.3: Semiose como uma rede (Queiroz, 2004)

Perceba neste esquema da figura 4.3, como os interpretantes tornam-se eles próprios objetos ou representamen de outros signos, dando origem a outro interpretante, que pode inclusive ser objeto de outra relação triádica, e assim por diante.

Categorias universais de Peirce

Na filosofia, há um ramo denominado *teoria do conhecimento* ou *epistemologia* que estuda a significação objetiva do pensamento humano (Hessen, 1980). A teoria do conhecimento pode ser dividida em dois grupos:

- Teoria geral do conhecimento: estuda a relação do pensamento humano com os objetos em geral, ou seja, a referência de todo pensamento aos objetos é o objeto formal dessa teoria (Hessen, 1980).
- Teoria especial do conhecimento: estuda os conteúdos do pensamento nos quais a relação entre pensamento e objeto encontra a sua expressão mais elementar (Hessen, 1980). Essa teoria

investiga como brotam os conceitos das leis essenciais do pensamento em concorrência com o caráter do que é dado empiricamente.

Assim, a *teoria especial do conhecimento* estuda conceitos básicos e gerais, por meio dos quais os objetos são definidos. Estes conceitos supremos são denominados de *categorias* (Hessen, 1980).

As *categorias* foram agrupadas, ao longo do decurso da história da filosofia, em diferentes sistemas de categorias:

- Aristóteles: foi o primeiro a agrupar categorias para formar um sistema de categorias. Ensinava que o conhecimento humano é uma reprodução dos objetos e, portanto, as categorias, que são os conceitos fundamentais do conhecimento, devem representar propriedades gerais dos objetos, qualidades objetivas do ser (Hessen, 1980). Portanto, deriva suas 10 categorias das *classes de palavras*:

1. Substância ou essência. Exemplo: homem, cavalo.
2. Quantidade: Exemplo: dois, três.
3. Qualidade. Exemplo: sábio, culto.
4. Relação. Exemplo: maior que.
5. Lugar. Exemplo: em casa.
6. Tempo. Exemplo: hoje, amanhã.
7. Posição. Exemplo: em pé.
8. Estado. Exemplo: está feliz.
9. Ação. Exemplo: ir, ferir.
10. Paixão ou passividade. Exemplo: ser ferido.

Segundo Hessen (1980), contra o sistema aristotélico das categorias, argumenta-se que sua base, o paralelismo entre as classes das palavras e as categorias, não é exato.

- Kant: significou um progresso sobre Aristóteles quando derivou as categorias das *classes de juízos*, e não das classes de palavras (Hessen, 1980). Ensinava que o pensamento produz os objetos, e as categorias são puras determinações do pensamento, formas e funções *a priori* da consciência. Para Kant, o entendimento é a faculdade de julgar. Deriva, assim, 12 categorias, divididas em 4 classes de juízos (Hessen, 1980):

1. Categorias da quantidade: unidade (este S é P), pluralidade (alguns S são P), totalidade (todos os S são P).
2. Categorias da qualidade: realidade (S é P), negação (S não é P), limitação (S é um não P).
3. Categorias da relação: substância-acidente (S é P), causa-efeito (se S é P, não é Q), ação recíproca (S é P ou Q).
4. Categorias da modalidade: existência (S é P), possibilidade (S é talvez P), necessidade (S é necessariamente P).

- Peirce: diferentemente de Aristóteles e Kant, Peirce criou um sistema de categorias baseado em meta-categorias, ou seja, categorias que podem ser utilizadas para gerar outras categorias. Assim sendo, segundo Peirce, faz-se necessário descrever apenas três categorias:
 - Primeiridade: “é o modo de ser daquilo que é tal como é, positivamente e sem referência a qualquer outra coisa” (CP 8.328). Essa categoria se refere a idéias de novidade, criação, liberdade, originalidade e potencialidade (Nöth, 1995). Um exemplo de primeiridade seria a sensação de uma cor ou uma criação de conhecimento.
 - Secundidade: “é o modo de ser daquilo que é tal como é, com respeito a um segundo, mas independente de um terceiro” (CP 8.328). Predominante nas idéias de causação e reação, comparação, oposição, polaridade, diferenciação e existência (Nöth, 1995).
 - Terceiridade: “é o modo de ser daquilo que é tal como é, colocando em relação recíproca um segundo e um terceiro” (CP 8.328). Predominante nas idéias de mediação, meio, intermediário, continuidade, representação, generalidade, infinitude, difusão, crescimento, inteligência e intencionalidade (Nöth, 1995).

Pode-se realizar uma pequena análise e uma simples analogia para tornar mais claro o entendimento dessas meta-categorias. Por exemplo, por que os conceitos de novidade, criação, liberdade, originalidade e potencialidade são “o modo de ser daquilo que é tal como é, positivamente e sem referência a qualquer outra coisa”? Porque esses conceitos se referem à geração de outros conceitos que não tem referência com conceitos anteriores, ou seja, “sem referência a qualquer outra coisa”. Uma analogia útil seria imaginar uma sequência de números qualquer. Se o próximo número da sequência é escolhido de maneira aleatória, ou seja, sem nenhuma referência aos números anteriores, como na figura 4.4, ele é uma primeiridade.

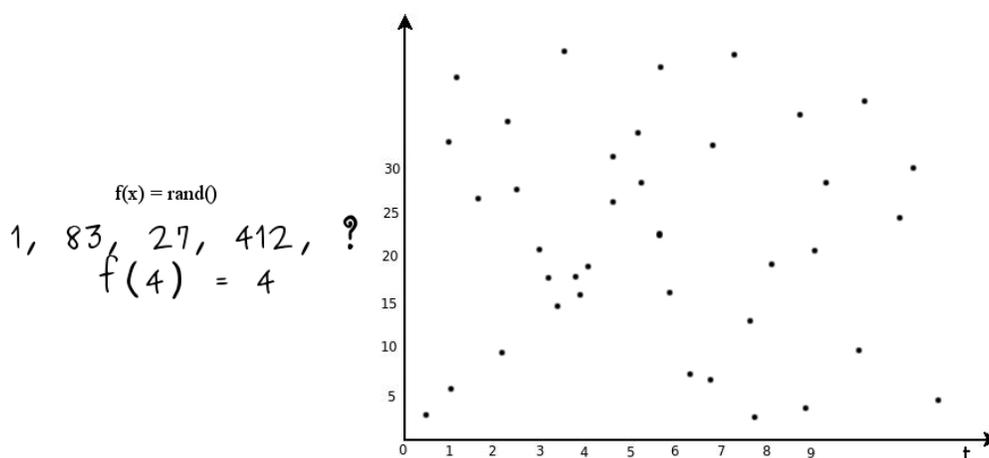


Fig. 4.4: Série aleatória de números

Gudwin & Queiroz (2007) equacionaram um sistema aleatório como sendo

$$S(t + 1) = \text{random}(), \quad (4.1)$$

o que significa que se considerarmos $t+1$ como sendo o presente, temos um sistema cujo presente é aleatório.

Com relação à secundidade, os conceitos de causação, reação, comparação, oposição, polaridade, diferenciação e existência são considerados “o modo de ser daquilo que é tal como é, com respeito a um segundo, mas independente de um terceiro”, pois eles se referem à geração de um conceito a partir de um outro existente, ou seja, “com respeito a um segundo”. Utilizando-se da mesma analogia da sequência de números, se o próximo número da sequência é escolhido a partir de uma função dos números anteriores, como na figura 4.5, ele é considerado uma secundidade.

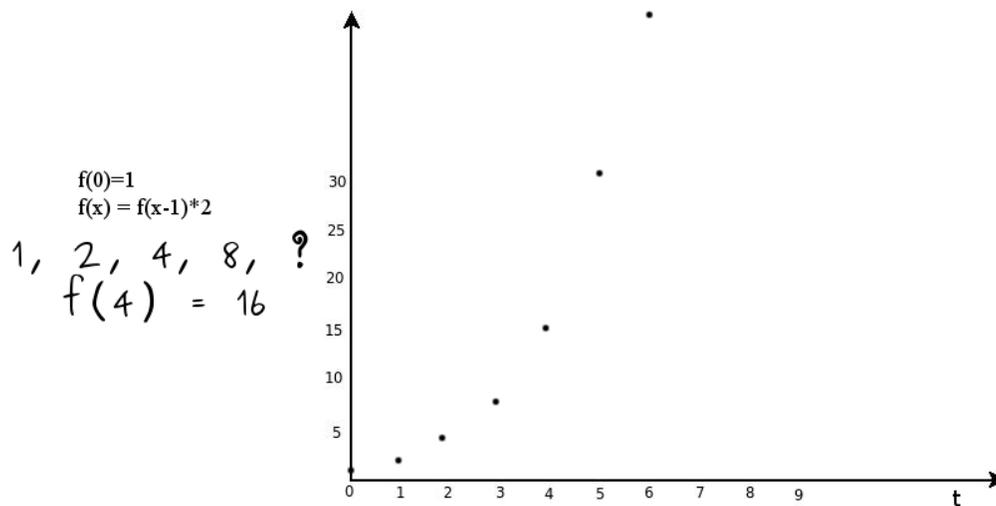


Fig. 4.5: Série determinística de números

Gudwin & Queiroz (2007) equacionaram um sistema determinístico como sendo

$$S(t+1) = f(S(t)), \quad (4.2)$$

o que significa que se considerarmos $t+1$ como sendo o presente, temos um sistema cujo presente é completamente determinado pelo passado.

A idéia de terceiridade é mais complexa: os conceitos de mediação, meio, intermediário, continuidade, representação, generalidade, infinitude, difusão, crescimento, inteligência e intencionalidade são considerados “o modo de ser daquilo que é tal como é, colocando em relação recíproca um segundo e um terceiro” pois esses conceitos se referem à geração de novos conceitos baseada em uma lei, uma convenção que media essa geração. Na analogia da sequência numérica anterior, se o próximo número da sequência fosse escolhido a partir da predição de uma série de números possíveis baseando-se em uma lei para selecionar um dentre os possíveis números futuros, como na figura 4.6, ele seria considerado uma terceiridade.

Gudwin & Queiroz (2007) equacionaram um sistema inteligente como sendo

$$S(t+1) = f(S(t+\tau)), \quad (4.3)$$

1, 5, 10, 13, 17, 20, 18, 16, 14, 15, 15, 15, 15...

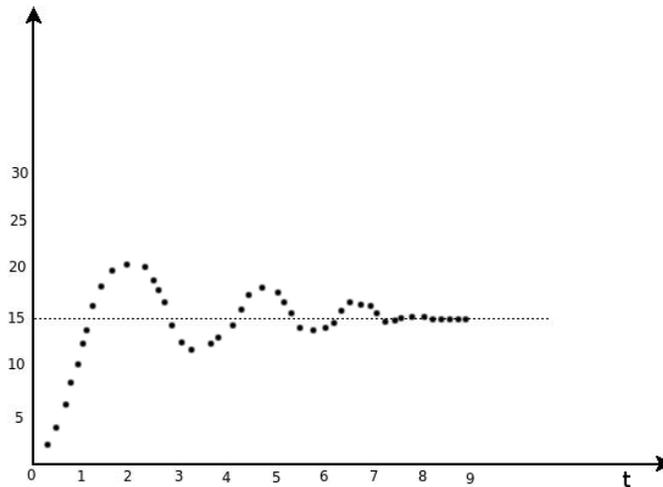
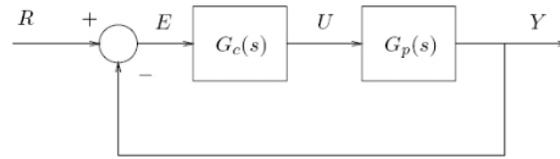


Fig. 4.6: Série de números que procura atingir uma referência, uma meta.

o que significa que se considerarmos $t+1$ como sendo o presente, temos um sistema cujo presente é determinado pelo futuro.

Mas como isto é possível, se o tempo evolui do passado para o presente? Para entender melhor o que isto significa, deve-se modificar a equação para

$$S(t+1) = f(E(S(t+\tau))(t)). \quad (4.4)$$

Neste caso, $E(.)$ é uma estimativa no tempo t de um estado futuro $S(t+\tau)$.

As categorias universais de Peirce são a base da sua semiótica (Nöth, 1995). Enquanto Aristóteles postulou dez e Kant postulou doze classes ontológicas, Peirce postulou apenas três. No entanto, essas três são categorias que podem ser usadas para criar outras categorias, de maneira tal que pode-se desenvolver 66 ou até 3^{10} (59.049) classes para classificar os signos. A princípio, e partindo das definições do próprio Peirce, parece difícil de captar os conceitos dessas categorias, mas são idéias sofisticadas que podem ser aproveitadas ao serem interpretadas em um determinado contexto.

Neste trabalho, o conceito das categorias é utilizado para classificar e entender as atividades cognitivas que representam tomadas de decisão de seres humanos que resolvem problemas não estruturados. Para tal, a seguinte interpretação lhes é atribuída, no contexto deste trabalho:

- **Primeiridade:** devido a sua relação com idéias de novidade, criação e originalidade, a primeiridade está relacionada com atividades cognitivas aleatórias. Uma escolha randômica entre várias alternativas disponíveis é classificada como uma atividade de decisão de primeiridade.

No contexto de engenharia de computação, a geração da primeira população em um algoritmo genético é classificada como uma atividade de primeiridade, por ser aleatória, por exemplo.

- **Secundidade:** por referenciar idéias de causação, reação e existência, a secundidade está relacionada com atividades mecânicas, que consideram apenas informações presentes, sem fazer estimativas do futuro baseadas em objetivos. Elas podem até mesmo ser um processo de decisão eficiente, mas se baseiam em regras simples a serem seguidas. A aplicação de uma função matemática sobre uma entrada, obtendo uma saída, é uma atividade mecanicista, uma secundidade. Dessa maneira, no contexto de engenharia de computação, as principais atividades de uma rede neural artificial são classificadas como secundidades.
- **Terceiridade:** por se referir a idéias de inteligência e intencionalidade, a terceiridade está relacionada com tarefas onde há estimação de estados futuros, baseadas na formulação de objetivos, planos para alcançá-los, escolha de um melhor plano e execução desse melhor plano, ou seja, tarefas inteligentes, segundo Peirce. No contexto de engenharia de computação, a tarefa realizada por um algoritmo de otimização ao encontrar uma solução ótima para um problema definido é classificada como uma terceiridade.

Signo como terceiridade

Segundo a teoria de Peirce, o objeto por si só poderia causar o aparecimento de uma idéia na mente do intérprete. Entretanto, na ausência do objeto, o signo é capaz de produzir na mente deste essa mesma idéia. Logo, o signo é um mediador, o meio pelo qual um objeto ausente produz uma idéia na mente do intérprete. Dessa maneira, o signo é um exemplo de terceiridade. Todo fenômeno de terceiridade é um signo.

Ícone, índice e símbolo

O ícone, representante da primeiridade na relação representamen-objeto, acontece nas relações de similaridade e analogia com o objeto. Essas relações de similaridade podem prescindir de qualquer relação física que o representamen(R) apresenta com o objeto (O). Dessa maneira, a semiose R-dependente (icônica) depende das propriedades do representam R, como pode ser visto na figura 4.7.

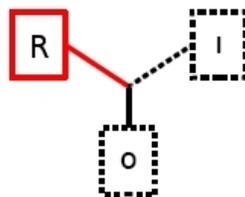


Fig. 4.7: Relação de iconicidade depende das propriedades de R, ou seja, da similaridade entre R e O.

Um exemplo de ícone pode ser uma foto de um carro. Na foto, a figura do carro apresenta diferentes tipos de similaridade com o objeto carro, que de fato existe no mundo, com o mesmo formato, mesmas proporções, cores, etc. Dessa maneira, esse signo, que é icônico, é capaz de causar

na nossa mente uma série de interpretante (idéias, sensações, etc) que o próprio carro causaria. Como a complexidade do ícone é menor que a do índice e do símbolo, a cognição (semiose, interpretação do signo) é mais simples, rápida e fácil de se fazer diante de um ícone do que diante dos outros componentes da segunda tricotomia. A mente humana trabalha de maneira fácil com ícones, sendo capaz de recuperar muitas informações a partir dele de maneira mais rápida e direta. Mesmo se mudarmos um pouco as relações de similaridade, como por exemplo, distorcemos a figura do carro, a mente ainda consegue recuperar as informações, pois o ícone é compreendido como um todo, e não somente por partes. Essa distorção parece ser suportada até determinados limites. A figura 4.8 apresenta um exemplo.

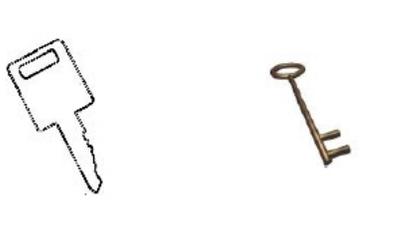


Fig. 4.8: A figura de uma chave, um signo icônico.

O ícone é um signo que está para alguma coisa somente pelo fato de se assemelhar a esta coisa. Eles são completamente substituídos pelos seus objetos de tal forma que dificilmente podem ser diferenciados deles.

O índice representa a secundidade na relação representamen-objeto. O índice ocorre em virtude de uma relação de causa e efeito de R com O. Assim sendo, a semiose R-O dependente (indexical) é dependente de relações espaço-temporais de R-O, carregando em si uma idéia de reação, como pode ser visto na figura 4.9

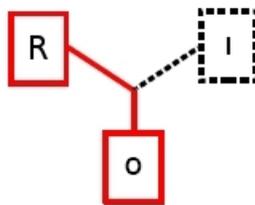


Fig. 4.9: Relação indexical depende de relações espaço-temporais de R-O

Um exemplo de relação indexical é a capacidade de certas pessoas que moram há bastante tempo em uma região de relacionar a cor do céu, as nuvens, o vento e até cheiros com uma precipitação (chuva). Essas pessoas estabeleceram uma relação de indexicalidade entre uma determinada situação (causa) e seu efeito, uma chuva (reação). A relação de indexicalidade é mais sofisticada do que a relação de iconicidade, de tal maneira que sua semiose apresenta um pouco mais de complexidade. A figura 4.10 é um exemplo de uma relação de indexicalidade.

O símbolo representa a terciaridade da relação representamen-objeto. O símbolo ocorre quando a relação R-O é mediada pelo signo, de tal maneira que o símbolo é uma lei dessa relação. A semiose R-O-I dependente (simbólica) é dependente da mediação do interpretante (I) para que a relação R-O ocorra, como pode ser visto na figura 4.11.



Fig. 4.10: A figura de uma chave faz relação com a fechadura, evidenciando uma relação espaço-temporal.

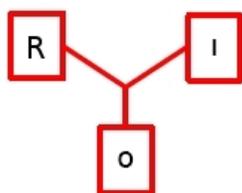


Fig. 4.11: Relação simbólica depende da mediação do interpretante (I) para que a relação R-O ocorra.

Um exemplo de símbolo são as palavras de uma língua qualquer. Existe uma convenção social entre os falantes da língua portuguesa, por exemplo, que determina que a palavra “cadeira” se refira ao objeto utilizado pelas pessoas para sentar-se, o qual todos conhecem. Assim, ao ler em um texto a palavra “cadeira”, será criado um interpretante na mente do leitor que relacionará esse símbolo ao objeto citado (a não ser que a palavra seja aplicada em um contexto onde conote outra coisa). A relação simbólica é extremamente complexa e sofisticada, e pode mudar com o tempo, com diferentes contextos, interpretações ou acordos. Por exemplo, as palavras de uma língua podem mudar de significado ao longo do tempo. Provavelmente, em meados do século XX, ninguém imaginava que a palavra “sinistro” poderia ser usada para conotar alguma coisa boa e interessante, como aconteceu com jovens brasileiros no início do século XXI. A figura 4.12 mostra um exemplo.

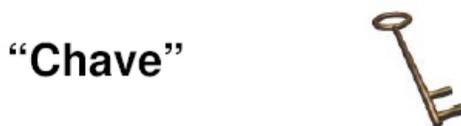


Fig. 4.12: A palavra chave faz relação com o objeto chave devido a uma lei, no caso, a convenção linguística.

Assim, um símbolo está relacionado a um objeto devido a uma lei. Ele depende de uma convenção, de um hábito ou de uma disposição natural do interpretante.

A figura 4.13 ilustra um processo de interpretação levando em consideração os tipos de semiose R (icônica), R-O (indexical) e R-O-I (simbólica).

Um exemplo de como a classificação de um signo em ícone, índice ou símbolo pode ser relativo é o interessante fato de que, depois da primeira interpretação, um mesmo signo que é interpretado

como ícone pode eventualmente ser interpretado como um símbolo. Um exemplo clássico é o ícone da lixeira da área de trabalho de um sistema operacional: no primeiro contato com esse signo, ele é interpretado como um ícone, pois seu formato igual ao de uma lixeira faz entender ao intérprete sua função; no entanto, da segunda vez em diante, ele passa a ser interpretado como um símbolo para aquele mesmo intérprete, que já sabe sua função devido a uma convenção que ele adotou após aquela primeira interpretação anterior, e não porque analisou o seu formato icônico, semelhante ao de uma lixeira.

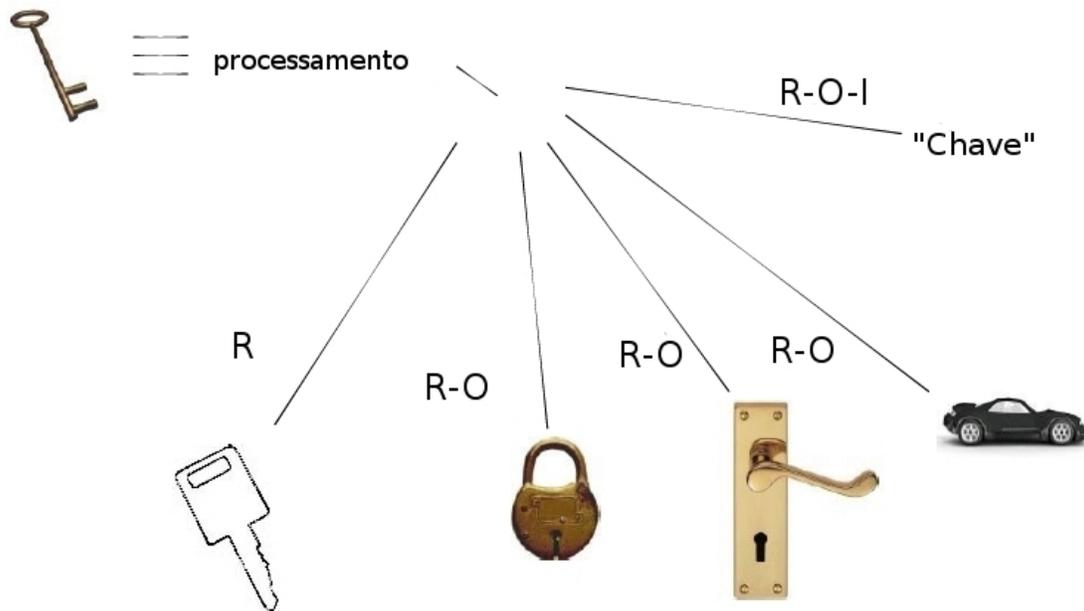


Fig. 4.13: Tipos de semiose R (icônica), R-O (indexical) e R-O-I (simbólica).

Conceito de mente em Peirce

O conceito de exosomatização do pensamento, ou seja, representação dos pensamentos fora do corpo humano, deriva do conceito de mente em Peirce e é essencial na metodologia de engenharia de sistemas inteligentes proposta neste trabalho. Para Peirce, **mente é tudo aquilo que exerce um poder mediador**. Essa definição aumenta o escopo da definição tradicional de mente, a qual estamos habituados, pois não se limita à mente humana ou de seres pensantes: com efeito, **qualquer elemento que exerce o papel de terceiraidade é considerado uma mente**, pois através desse elemento pode-se perceber o fenômeno da semiose. Isso também significa afirmar que a semiose não ocorre somente na mente humana somática, mas pode ocorrer em outras mentes, seja essa mente uma célula, um ambiente onde ocorre um processo físico-químico ou uma máquina.

Essa noção de mente é muito mais ampla do que noções biológicas, e nada tem a ver com a idéia de alma ou de algo oposto ao corpo (tradicional dicotomia corpo x mente). Essa mente é, com efeito, um fator regulador que determina a finalidade de um processo causal. Esta causalidade aqui referida pode ser de duas naturezas, segundo Peirce (1960):

- Causa eficiente: uma causa puramente mecanicista, que não busca atingir um estado futuro que representa uma predição, plano ou objetivo. É uma secundidade.
- Causa final: uma causa com finalidade, ou seja, baseado em um estado futuro de predição, plano ou objetivo criado e que é perseguido. É uma terceiridade.

Baseando-se nesse conceito de mente peirceano, surgiram grupos de pesquisadores para estudar a semiose nos diversos meios onde ela ocorre, além da mente humana. Identificar uma mente em um ambiente fora da mente humana é reconhecer que a semiose pode ocorrer neste meio externo, ou seja, que o pensamento pode ser exosomatizado (se localizar e ser representado fora da mente humana). Entre as ramificações de estudo da semiótica que estudam a semiose fora da mente humana pensante, pode-se incluir mas não limitar-se a estas:

- Biosemiótica: estuda os processos semióticos em todos os níveis biológicos, ou seja, um estudo semiótico dos seres vivos. Considera principalmente os aspectos da comunicação e significação entre os seres vivos, postulando que a comunicação é a característica essencial da vida dos seres vivos, e que os organismos são como mensagens para as futuras gerações de como sobreviver e reproduzir. Os organismos vivos funcionam como mente.
- Semiótica organizacional: estuda a integração de sistemas computacionais dentro de sistemas sociais. A ênfase desta linha de pesquisa é obter uma melhor eficiência na integração da tecnologia da informação em práticas de trabalho e na cultura dos usuários de computadores de organizações. Adapta os conceitos da semiótica para o meio empresarial, das corporações. Uma empresa funciona como uma mente, e os processos internos de organização e tomadas de decisão são semiose.
- Semiótica urbana: uma cidade inteira é comparada a uma mente, e seus processos e desenvolvimentos são estudados como semioses interdependentes.
- Físio-semiótica: considera a semiose um processo tão vasto quanto o próprio universo físico. Procura entender o mundo físico e seus fenômenos a partir de conceitos semióticos.
- Semiótica computacional: adapta os conceitos da semiótica para o arcabouço computacional na tentativa de criar máquinas e sistemas capazes de manipular signos, ou seja, realizar semiose (cognição) através de inteligência artificial e representação do conhecimento, além de tentar realizar a engenharia do processo da semiose no estudo e projeto de interações humano-computador.

4.3 Semiótica e semiótica computacional

O estudo do fenômeno da inteligência vem sendo realizado há muito tempo e através de várias abordagens diferentes. Nas ciências exatas, mais particularmente na ciência da computação, a preocupação de se embutir inteligência aos computadores é tão antiga quanto os próprios computadores. Nas ciências humanas, o estudo do fenômeno da inteligência teve início nos trabalhos dos primeiros grandes filósofos gregos. Platão já se preocupava com o fenômeno da percepção e da representação dos fenômenos do mundo. Desde que Von Neumann publicou suas primeiras idéias sobre arquitetura

e organização dos computadores (Neumann, 1958) (Neumann, 1969), surgiram os questionamentos a respeito da possibilidade daquela nova máquina reproduzir a inteligência humana. Naquele instante, pode-se dizer que houve a gênese da área de inteligência artificial. Muitos ramos surgiram na inteligência artificial, entre eles os ramos da inteligência computacional e da computação flexível, os quais procuraram inspirações da biologia para tentar construir sistemas mais inteligentes. Mais recentemente, engenheiros e cientistas de computação têm procurado na semiótica uma fonte de inspiração para realizar esta tarefa. Para que este trabalho fosse facilitado, foi criado um ramo na área de sistemas inteligentes denominado “semiótica computacional”, onde a principal preocupação é estudar como a semiótica pode ser usada para construir sistemas com habilidade de realizar semiose e evoluir comportamentos inteligentes (Gudwin, 1999). Neste ramo, o conceito de signo é utilizado de maneira explícita, herdando grande parte do corpo da teoria da semiótica. A idéia é desenvolver sistemas inteligentes capazes de realizar semiose e manipular signos.

Como dito anteriormente, a semiótica é uma ciência que estuda significação e representação, envolvendo os fenômenos da cognição e comunicação entre sistemas vivos. O estudo de sistemas inteligentes artificiais possui, entre muitos dos seus objetivos, desenvolver sistemas capazes de realizar comunicação e cognição. A semiótica computacional é uma disciplina que tenta aproveitar os conceitos desenvolvidos nas ciências humanas dentro do contexto da inteligência artificial para a implementação de sistemas computacionais com características inteligentes. Há um desafio em sintonizar os conceitos originalmente aplicados apenas a seres humanos e biológicos e fazer com que eles se adequem ao arcabouço de sistemas artificiais (Gudwin, 1999). A semiótica computacional tem sido aplicada em diferentes contextos:

- Semiótica computacional no escopo da análise de textos: também chamada de linguística computacional. Estuda sistemas de interpretação de textos baseados nos conceitos da semiótica.
- Semiótica computacional na engenharia de interfaces: estuda aspectos relacionados à comunicação entre usuários e sistemas computacionais orientados a síntese de interfaces homem-máquina. São estudados a emulação de linguagem humana, realidade virtual, aspectos teóricos de interface no contexto de desenvolvimento de novos sistemas operacionais, etc.
 - Engenharia Semiótica: variação da semiótica computacional na engenharia de interfaces introduzida por de Souza (1993), destaca-se na área de interfaces humano-computador. Nesta abordagem, considera-se que a interface do sistema é uma mensagem enviada do projetista para o usuário, levada através da mídia computacional. Como essa mensagem é também capaz de trocar outras mensagens com o usuário, a interface pode ser entendida como um artefato de metacomunicação¹ (de Souza, 1993). Segundo essa teoria, a mensagem carregada pela interface do sistema, um artefato de metacomunicação, deve ser capaz de responder a duas perguntas fundamentais:
 1. Quais problemas o sistema é capaz de resolver?
 2. Como os usuários podem interagir com o sistema para implementar estas soluções?
- Semiótica aplicada ou controle situacional semiótico: encabeçada por Pospelov e outros pesquisadores russos. Seu trabalho consiste em uma representação formal de conhecimento baseada

¹Um “artefato de metacomunicação” é um artefato que comunica mensagens sobre (aspectos da própria) comunicação.

na semiótica. Ele também propõe uma álgebra para trabalhar com esta representação formal de conhecimento. Outros trabalhos foram desenvolvidos utilizando como ponto de partida o trabalho de Pospelov, especulando sobre aprendizagem e evolução de sistemas semióticos.

- Semiótica computacional, modelagem e inteligência artificial (linha americana): busca uma interpretação das arquiteturas propostas por Meystel e Albus dentro do escopo da semiótica, utilizando para isto algumas das idéias da semiótica aplicada.
- Semiótica computacional, modelagem e inteligência artificial (linha brasileira): introduzida por Gudwin Gudwin (1999). A semiótica computacional faz uma projeção da taxonomia dos conhecimentos proposta na semiótica no espaço dos sistemas computacionais, propõe uma representação formal para os conhecimentos e uma ferramenta formal para a utilização desta representação.

4.3.1 Conceitos importantes da semiótica computacional

Conforme dito anteriormente, a semiótica computacional procura adaptar conceitos da semiótica para estudar a possibilidade de criar sistemas e máquinas capazes de manipular signos. A terminologia utilizada é praticamente a mesma da semiótica tradicional. Fala-se de signos, objetos, interpretantes, intérpretes, primeiridade, secundidade, terceiridade, entre outros, mas do ponto de vista de um entendimento computacional. A seguir, apresenta-se alguns conceitos semióticos com este viés computacional, desenvolvidos por Gudwin (1999).

Intérprete

O intérprete é o ser semiótico, o sistema que hospeda o processamento sígnico (Gudwin, 1999). No entendimento puro da semiótica tradicional, o conceito de intérprete não é necessário. No caso da semiótica computacional, este conceito é central, pois o interesse é justamente sintetizar este ser semiótico capaz de manipular signos e realizar semioteses. O intérprete é o agente computacional, o principal objeto de estudo.

Unidade de conhecimento

Com o propósito de causar menos impacto na comunidade de inteligência artificial devido à utilização de alguns termos não conhecidos e estranhos a ela provenientes da teoria semiótica, o que poderia provocar uma possível aversão a utilização desses conceitos nesta comunidade, Gudwin (1999) preferiu introduzir este conceito de unidade de conhecimento.

A região sob o foco de atenção de algum espaço, ou seja, um signo ou conjunto de signos, estejam eles causando ou não comportamento no intérprete, é chamada de unidade de conhecimento. O comportamento básico de um intérprete é selecionar unidades de conhecimento de ambos os espaços internos e externos e gerar novas unidades de conhecimento em ambos espaços.

Esta estrutura pode ser abstraída como sendo uma estrutura matemática qualquer, como por exemplo números, listas, árvores e grafos.

Vale a pena enfatizar que, segundo a definição acima, uma unidade de conhecimento é um signo, segundo a visão pansemiótica de Peirce. Dessa maneira, sempre que for utilizado o conceito de unidade de conhecimento neste trabalho, estaremos tratando de signos, na visão de Peirce.

Semiose externa

A semiose externa ocorre quando o interpretante gerado pelo signo ocorre no espaço externo, ou seja, no espaço fora do corpo do intérprete, exosomático. Esse tipo de semiose causa uma mudança no ambiente, sendo compartilhável com outros intérpretes, diferentemente de uma semiose que ocorre no espaço interno, que é acessível apenas ao intérprete que a gerou. Sendo assim, a semiose externa pode agir como um novo signo para o mesmo intérprete ou para outros intérpretes.

Podem ocorrer intérpretes que não possuem estados internos. Um exemplo disso são processos semióticos acontecendo em moléculas ou reações químicas, assim como processos semióticos em organismos simples.

Semiose interna

A semiose interna se dá quando o interpretante ocorre em qualquer um dos espaços internos. Uma cadeia semiótica típica ocorre inicialmente com um signo no espaço externo, que gera um conjunto de interpretantes no espaço interno, que tornam-se por sua vez signos que poderão gerar novos interpretantes internos numa cadeia potencialmente infinita. Algum deles pode se tornar também um interpretante externo.

4.4 Resumo

Este capítulo apresentou importantes conceitos da semiótica de Peirce e da semiótica computacional, que representa uma adaptação dos conceitos da semiótica com o intuito de aplicá-los na construção de softwares e sistemas inteligentes. O capítulo não apresentou nenhuma contribuição original, apenas teve o intuito de introduzir de forma didática e sintética alguns conceitos e ferramentas cujo conhecimento prévio se faz necessário para o correto entendimento do processo de desenvolvimento de sistemas chamado Engenharia de Sistemas de Amplificação de Inteligência, que é objeto de estudo deste trabalho. Vistos dessa maneira, avulsos, esses conceitos podem parecer desconexos e sem relação com uma possível metodologia de criação de sistemas. No entanto, com a correta compreensão destes conceitos, o leitor será capaz de entender os mecanismos criados para representar as operações mentais no processo de tomada de decisão inteligente, e o significado extraído desses processos com o intuito de desenvolver sistemas computacionais capazes de amplificar a inteligência do agente humano na resolução do problema, de modo que um leitor que não possua conhecimento prévio da semiótica de Peirce e da semiótica computacional dificilmente conseguirá seguir em frente neste texto e compreendê-lo de fato, sem antes ter passado pelos conceitos apresentados aqui.

Capítulo 5

Amplificação de inteligência: técnicas para potencializar o intelecto humano

A análise do surgimento, desenvolvimento e rápido domínio da espécie humana, *Homo sapiens sapiens*, no globo terrestre, comumente causa admiração e espanto, devido ao sucesso evolutivo que essa espécie rapidamente obteve no ambiente terrestre, e devido a sua incapacidade de encontrar um equilíbrio populacional, comum a todas as outras espécies, o que pode vir a causar o esgotamento dos recursos naturais do planeta. O sucesso da espécie humana, entre outros fatores, está relacionado ao desenvolvimento do seu conhecimento. Não somente do conhecimento específico de cada ser humano, mas principalmente do conhecimento compartilhado pela humanidade. As criaturas humanas possuem uma capacidade única, que provavelmente as leva a essa hegemonia: elas são capazes de utilizar ferramentas mentais (culturais), como a linguagem, para transmitir conhecimentos para os seus semelhantes. Dessa maneira, cada nova geração de criaturas não precisa redescobrir o que já foi alcançado pela geração anterior, podendo concentrar-se em aprender rapidamente o que já existe de conhecimento e aplicar suas energias para testar novos talentos e possibilidades, construindo novas tecnologias a partir do que já havia sido construído, estabelecendo uma evolução contínua. São as chamadas “criaturas gregorianas”, segundo as definições de tipos de mente de Dennett (1996). Para potencializar ainda mais essa capacidade de evoluir e de criar novas tecnologias, além da capacidade de evoluir seus modelos mentais internos (conhecimento) através da utilização de ferramentas mentais, essas criaturas gregorianas possuem também a capacidade de refinar ou criar novas ferramentas mentais que proporcionam sua evolução.

Desde que os humanos tomaram conhecimento da existência de civilizações humanas primitivas e exóticas ainda habitando o globo terrestre, surgiu uma discussão no ocidente a respeito da relação desses grupos com os grupos mais conhecidos da humanidade. Essa discussão se estendia desde a moral dessas civilizações aos seus costumes, mas o mais importante era a curiosidade com relação a qualidade de seus processos mentais, como por exemplo, se eles possuíam a mesma lógica que grupos civilizados (Gardner, 2003). As primeiras idéias que surgiram postulavam que esses indivíduos exóticos, como índios americanos ou nômades árabes, eram verdadeiras degenerações comparadas ao ser humano “puro”. As viagens extensas do período Renascentista contribuíram com essa visão da degeneração, pois os analistas enfatizavam as semelhanças entre o selvagem e os antecessores do homem civilizado, colocando os selvagens em um estágio inferior ao dos europeus contemporâneos. Com o Iluminismo, surgiu a crença fervorosa na igualdade, e o debate se acirrou. No entanto, a raci-

onalidade era o padrão do iluminista, e a mente ocidental parecia ser qualitativamente mais avançada que a do selvagem, que parecia confuso e contraditório. À medida que práticas como escravidão e crenças de superioridade de um povo sobre o outro foram sendo consideradas retrógradas, criou-se uma atmosfera de pontos de vista igualitários. Esses pontos de vista inclusive criaram dificuldades para aqueles que possuíam convicções puramente religiosas, devido ao fato de necessitarem explicar porque alguns grupos cultuavam um Deus único, enquanto outros eram politeístas. Membros das igrejas constituídas da época afirmavam que os selvagens não podiam ser ajudados, e que de fato pertenciam a uma outra espécie. Com as teorias de Darwin, esses membros precisaram enfrentar suas demonstrações baseadas em fatos e argumentos científicos de que todos os humanos eram descendentes de uma linhagem de antecessores que remontava a milhões de anos, e mais ainda: estes não poderiam ser concebidos como separados de qualquer parte da ordem natural (Gardner, 2003).

Estudos de antropólogos como Lucien Lévy-Bruhl sobre processos de pensamento de povos primitivos há quase um século, apontavam para uma direção que era geral da comunidade antropológica: o modelo evolucionista (Gardner, 2003). Segundo esse modelo, membros da civilização ocidental eram “avançados”, e representavam o auge do raciocínio humano, enquanto os indivíduos do resto do mundo eram simplesmente cópias inferiores, com uma capacidade mental menor. Ao longo de sua vida, Lévy-Bruhl viu esse pensamento da comunidade mudar, e ele passou a ser considerado conservador nas suas idéias. Diferentemente da maioria dos estudiosos, que costumam defender suas idéias ao serem atacados por gerações seguintes, Lévy-Bruhl acabou deixando de lado suas idéias, e no final da sua vida adotou uma visão completamente contrária a tudo o que tinha publicado até então:

“O passo que acabo de dar, e espero que seja definitivo, consiste, resumidamente, em abandonar um problema mal colocado... mesmo levando em consideração os numerosos casos característicos de *participação* dos quais meus seis volumes estão cheios, ainda existem dúvidas sobre a explicação... Comecei postulando uma mentalidade primitiva diferente da nossa... uma posição que nunca consegui defender bem, e que a longo prazo é insustentável... A tese de tal forma atenuada e enfraquecida não é mais defensável... Desistamos totalmente de explicar a participação através de algo peculiar à mente humana... Não existe uma mentalidade primitiva distinguível da outra” (citado por Gardner (2003)).

O consenso que existe hoje na antropologia é que não há diferença entre as habilidades cognitivas dos seres humanos do ponto de vista racial, que as capacidades cognitivas dos indivíduos primitivos eram muito parecidas com as nossas, e que eles classificam em grande parte nas mesmas linhas e das mesmas maneiras que as pessoas civilizadas (Gardner, 2003). Isso explica o fato de até hoje existirem tribos isoladas que exibem comportamento típico de seres humanos primitivos: as diferenças de comportamento são diferenças proporcionadas pela cultura, pelos estímulos, pelo ambiente ao qual as pessoas são expostas e pelas ferramentas mentais utilizadas. Ao invés de procurar fatores subjacentes às estruturas ou processos do mundo, a mente primitiva tenta classificar os objetos e a experiência do dia-a-dia em termos das propriedades perceptivas e sensoriais aparentes.

A grande evolução do intelecto humano, dos tempos primitivos aos de hoje em dia, não foi uma evolução de estruturas físicas do sistema nervoso humano, como redes neurais naturais ou outras estruturas encefálicas, mas sim das ferramentas mentais utilizadas pelo humano e a revolução que as mesmas causaram nos seus modelos internos, ou seja, na sua maneira de realizar cognição. A linguagem e a escrita foram, com efeito, o marco inicial das ferramentas mentais potentes o suficiente para

realizar uma amplificação do intelecto humano capaz de refinar seus processos cognitivos. Mas, sem dúvida, nunca existiu uma ferramenta mental capaz de realizar tal papel tão bem quanto o computador. Ironicamente, os primeiros modelos de arquitetura e organização de computadores, propostos por Neumann (1958) na década de 1950, eram modelos baseados na idéia da mente humana (daí os conceitos de memória, unidade central de processamento e periféricos), sendo idéias dessa época descrever de forma tão precisa a inteligência e o aprendizado de maneira tal que se pudesse fazer com que uma máquina o simulasse. Atualmente, os computadores são as principais ferramentas mentais do ser humano, e eles têm revolucionado seus modelos mentais.

A respeito dessa evolução do intelecto humano por meio da criação de sistemas cognitivos distribuídos (Giere & Moffatt, 2003), Henderson (1998) afirma:

“Claramente, a implementação de novas ferramentas representacionais (como sistemas gráficos computacionais) mudam essa cultura, enquanto as convenções existentes influenciam como as novas ferramentas podem ser utilizadas”.

O objetivo deste capítulo é essencialmente explorar a área de pesquisa denominada “amplificação de inteligência” (do inglês *intelligence augmentation*), que trata das técnicas e ferramentas mentais que amplificam o intelecto humano. Naturalmente, este estudo estará focado no contexto de ferramentas computacionais de amplificação de inteligência. É objetivo deste trabalho explorar a utilização dessas ferramentas para construir sistemas de computação que possam auxiliar o ser humano na resolução de problemas não estruturados, amplificando seu intelecto nessas tarefas.

Este capítulo está organizado da seguinte maneira:

- Seção 5.1: explica o conceito de amplificação de inteligência;
- Seção 5.2: apresenta-se um pouco da história da área, com alguns dados cronológicos e um exemplo de uma ferramenta avançada de amplificação de inteligência em 5.2.1;
- Seção 5.3: faz uma comparação entre as áreas de amplificação de inteligência e de inteligência artificial, traçando paralelos, apresentando as intersecções e ortogonalidades dos conceitos;
- Seção 5.4: sugere que os agentes computacionais inteligentes são uma ferramenta adequada para amplificar a mente humana na resolução de problemas não estruturados;
- Seção 5.5: é apresentada a engenharia de SAI, com os atributos necessários aos SAI e sua componentização padrão;
- Seção 5.6: a ferramenta “diagrama de fluxo cognitivo”, desenvolvida para modelar o fluxo cognitivo dos agentes humanos que resolvem os problemas a ser amplificados, é descrita em detalhes;
- Seção 5.7: a teoria de Redes Semiônicas é apresentada como uma maneira de simular dinamicamente os diagramas de fluxo cognitivo e modelar a inserção do futuro sistema de amplificação de inteligência nesse fluxo;
- Seção 5.8: apresenta um resumo do capítulo e de suas contribuições.

5.1 Introdução

Amplificar a inteligência é aumentar a capacidade humana de resolução de problemas a partir da utilização de ferramentas mentais. A princípio, qualquer ferramenta utilizada durante a resolução de um problema, como um papel e um lápis, é um instrumento de amplificação de inteligência, mas o foco deste capítulo será na utilização de computadores e sistemas computacionais como ferramentas de amplificação. A humanidade se utilizou de muitos instrumentos de amplificação de inteligência e ferramentas mentais ao longo de sua evolução, como linguagem, artefatos e metodologias criadas em diferentes culturas. A invenção da escrita, por exemplo, possibilitou ao ser humano gravar informações fora do cérebro. Conforme dito no prólogo deste capítulo, os computadores são a ferramenta de amplificação de inteligência mais poderosa que a humanidade já produziu, sendo responsáveis atualmente por uma revolução na maneira humana de pensar e resolver problemas.

Essa maneira humana de pensar é, com efeito, um fluxo que ocorre em vários meios além do cerebral (Ransdell, 2003). Suponha um arquiteto que realiza um projeto de uma casa utilizando uma prancheta, papel, lápis e borracha. A prancheta de trabalho é o meio no qual seus pensamentos ocorrem e se materializam. Cada traço desenhado, além de ser a materialização de um pensamento anterior, é um signo que dá origem a um novo interpretante, ou seja, é parte do pensamento do arquiteto. Assim, durante o processo de criação, a prancheta fez parte do processo cognitivo, trabalhando em conjunto com o cérebro. Sendo a prancheta um objeto inanimado, sua contribuição durante o processo cognitivo foi infinitamente menor do que a contribuição da mente humana. Por outro lado, caso a ferramenta fosse um CAD, a contribuição seria maior, visto que ele ajuda o arquiteto a expressar seus pensamentos de forma mais efetiva. Se fosse um CAD inteligente, que fosse capaz de criticar o trabalho do arquiteto, dando feedbacks e fazendo parte do trabalho sozinho, seu papel no processo cognitivo seria maior ainda.

Assim, as representações fora da mente são parte do pensamento. O problema não foi resolvido somente pelo arquiteto, mas sim pela interação entre duas entidades: arquiteto e prancheta (ou CAD). Neste caso, dizemos que a inteligência que gerou o projeto não estava localizada somente no arquiteto e sim no sistema arquiteto-CAD, ou seja, ela está de certa forma também representada além do corpo humano (exosomática). Nos dois casos, a ferramenta utilizada possibilitou o fluxo do pensamento, mas no último caso, além de viabilizar tal fluxo, a ferramenta introduz informações de forma a direcionar e ajudar a coordenar o pensamento, aumentando a capacidade cognitiva do arquiteto. Neste caso, a ferramenta é designada como sendo uma ferramenta de amplificação de inteligência (Ransdell, 2003). Fica claro, então, que um dos requisitos para que ocorra amplificação de inteligência é a capacidade de representar parte do pensamento ou fluxo cognitivo exosomaticamente, para que a mente humana fique mais livre para realizar outros processos cognitivos relevantes para resolver determinado problema.

Mas esta não é a única maneira de amplificar a inteligência, havendo inúmeros exemplos de ferramentas. O simples fato de fornecer novos meios de expressão (como, por exemplo, o hipertexto), resulta em novas formas de pensamento e conseqüentemente na ampliação da capacidade cognitiva. O hiper-texto é uma forma de representar um texto que se utiliza do artifício de links, entre outras coisas. A possibilidade de utilizar links em determinadas porções do texto, levando a outros textos relacionados, é uma possibilidade que amplifica a mente humana. Escrever um texto com este artifício ajuda a organizar os pensamentos do autor de maneira diferente e mais interessante.

A amplificação de inteligência está baseada em dois princípios (Ransdell, 2003):

1. Todo pensamento pode ser materialmente corporificado ou incorporado (exosomático)
2. Todo pensamento ocorre em forma de diálogo

Ambos os princípios estão baseados na semiótica de Peirce, no princípio que diz que “todo pensamento se dá em signos”¹. O primeiro princípio suporta a idéia de que todo pensamento ou conhecimento pode ser representado por um signo ou representamen, segundo Peirce. Sendo possível representar um pensamento como um signo, e sendo o mundo composto por signos, na visão peirceana, então pensamentos e conhecimentos podem ser convenientemente representados fora da mente, ou seja, exosomaticamente. Segundo Ransdell (2003):

“O ponto de partida para entender a amplificação de inteligência é considerar a localização exosomática da mente no ambiente material”.

Entendê-la é também o ponto de partida para entender como a semiótica de Peirce é capaz de fornecer uma base teórica para a amplificação de inteligência. A afirmação de Ransdell resume em poucas palavras uma das formas de amplificar a mente humana, com efeito a mais importante delas, a essência da amplificação de inteligência na resolução dos problemas que serão confrontados: exosomatizar o pensamento.

O segundo princípio procura demonstrar a estrutura das atividades cognitivas humanas: uma estrutura tipicamente construída na forma de um diálogo, sendo este diálogo constituído de feedbacks negativos, como num sistema convencional de controle, representado na figura 5.1. É como se o processo cognitivo humano acontecesse a partir do teste de vários planos de ação no modelo interno da mente. Esses planos teriam um erro realimentando a entrada, erro este comparado a um estado futuro estimado. Assim, os planos são refinados e o plano final é o escolhido e executado, por ter a melhor avaliação, sendo obtido a partir desses esquemas de feedback. Na verdade, essa estrutura pode ser usada para modelar atividades cognitivas inteligentes, sendo classificadas por Peirce como terceiridades. Fazendo uma leitura de Peirce aplicada à engenharia (seção 4.2.1), existem também atividades cognitivas aleatórias (primeiridades) ou mecanicistas (secundidades), sendo as primeiras uma geração de comportamento aleatório e a segunda uma simples aplicação de uma regra, sem estimação de estados futuros. Assim, promover ferramentas que auxiliem o fluxo cognitivo através de feedbacks negativos é uma maneira de amplificar a mente. Um exemplo disso seria o CAD utilizado pelo arquiteto criticar a colocação de uma coluna em algum ponto do projeto devido a excesso de carga ou restrições de orçamento, fazendo com que o arquiteto replaneje e realize novamente a colocação, considerando o erro apontado anteriormente no input de sua atividade. O fluxo deste discurso baseado em diálogos é composto de uma interpretação após a outra, e o desenvolvimento de processos de controle crítico desta cadeia de interpretações torna o diálogo mais eficiente e efetivo relacionado ao seu fim, promovendo a possibilidade do desenvolvimento de raciocínios mais interessante, amplificando assim a inteligência.

¹A expressão original em inglês é “all thought is in signs”, e sua tradução é complicada. A tradução utilizada neste trabalho é escolha pessoal destes autores. Outras traduções equivalentes poderiam ser:

- “todo pensamento se dá por meio de signos”, ou
- “todo pensamento ocorre por meio de signos”, ou ainda
- “todo pensamento é em signos”.

Entendemos que nenhuma destas traduções é tão sintética e expressiva como o original em inglês.

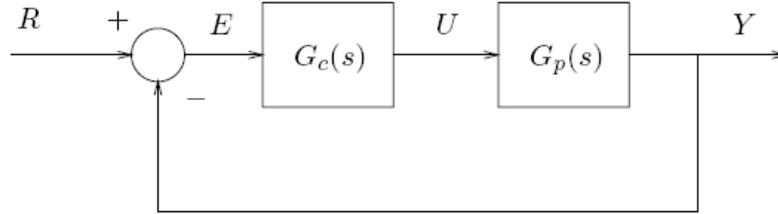


Fig. 5.1: Sistema de controle

Uma forma eficiente de amplificar a inteligência é delegar algumas tarefas cognitivas, necessárias para a resolução de um problema não estruturado, a uma ferramenta mental. Este processo se torna mais eficiente quando a ferramenta mental realiza algumas tarefas que são realizadas de maneira ineficiente pela mente humana, deixando que ela fique livre para se concentrar nas tarefas que ela realiza melhor. No caso do CAD e do arquiteto, o CAD pode realizar cálculos matemáticos estruturais de maneira muito mais eficiente do que a mente do arquiteto, deixando que ele se concentre nas tarefas de criar projetos interessantes, uma atividade que necessita de criatividade, tipicamente melhor exercida por uma mente humana do que por um sistema computacional como o CAD. Perceba que é muito difícil, para não dizer impossível, conseguir realizar esta forma de amplificar a mente sem que o conhecimento seja representado exosomaticamente de alguma maneira e sem que haja formas de expressá-lo que sejam adequadas, pois o CAD necessita expressar as soluções que sugere através de uma linguagem que é uma ontologia² comum entre os dois agentes, o humano e o computacional.

Encontra-se na teoria de Peirce o arcabouço teórico e filosófico necessário para entender e avançar o projeto de amplificar a mente humana.

Resumindo, as principais formas de amplificar a mente são:

- Representar o conhecimento exosomaticamente. Isso equivale a **fornecer novos meios de representação adequados**. O CAD que o arquiteto utiliza representa a solução construída aos poucos exosomaticamente, caracterizada através de uma interface gráfica, amplificando a mente do arquiteto ;
- Utilizar novos artifícios para expressar o pensamento. Isso equivale a **fornecer novos meios de**

²A palavra “ontologia” é utilizada aqui e em todo este trabalho no contexto de sistemas multi-agentes, e possui um significado diferente do contexto filosófico. No contexto deste trabalho, “ontologia” é uma estrutura de base que um agente precisa ter para se referir a um determinado domínio. A estrutura de base de uma ontologia é a seguinte:

- Vocabulário de termos utilizados no domínio;
- Modelagem semântica deste vocabulário;
- Regras para uso do vocabulário;
- Conjunto de tautologias válidas para o domínio em questão.

Exemplo: Sistema multi-agentes que monitora o funcionamento de um automóvel. Vocabulário: motor, pneu, suspensão, temperatura, alta, porta, vidro, alarme, freio, etc. Pode-se ter um agente que monitora a temperatura do motor e se comunica com um agente central. Exemplo de mensagem: “Temperatura do motor é alta”. Todos os agentes devem conhecer o vocabulário e a semântica para poder interpretar essa mensagem. Para se comunicar, os agentes precisam compartilhar a mesma ontologia.

expressão. A linguagem e o hiper-texto são exemplos desse tipo de amplificação, que dá mais capacidade de organização das semioses;

- **Controlar o fluxo cognitivo.** Isso é feito através de feedbacks negativos que realimentam o processo cognitivo, organizando e estruturando os passos em busca da solução;
- **Realizar tarefas cognitivas** que a mente humana não realiza de maneira eficiente, deixando que a mente humana se concentre nas tarefas que ela faz melhor, configurando um sistema cognitivo distribuído (Giere & Moffatt, 2003).

5.2 Desenvolvimentos da área de amplificação de inteligência

O termo “intelligence augmentation”, que significa aumento ou amplificação de inteligência, foi cunhado por Douglas C. Engelbart como um slogan em seu manifesto lançado em 1962 (Engelbart, 1962), o “Augmenting Human Intellect”, onde lançou as idéias e metas da amplificação de inteligência de desenvolver ferramentas eletrônicas para amplificar o intelecto do ser humano, inspirado pelo artigo de Bush (1945). Nos quinze anos seguintes, Engelbart dirigiu o “Augmentation Research Center” (ARC) no Stanford Research Institute (SRI). Juntamente com outros nomes como Licklider do MIT e Ivan Sutherland, inventor do bitmap, deu origem à computação interativa, editores de texto, redes de computadores, mouse e interfaces gráficas (Skagestad, 1993). No entanto, foi Licklider em 1960 (Licklider, 1960) que inaugurou todo o programa de amplificação de inteligência, exceto pelo nome e formalização dados depois por Engelbart. Diferentemente do paradigma da inteligência artificial, que procurava simular a mente humana em máquinas, Licklider sugeriu que houvesse uma simbiose entre humanos e máquinas. Ele previu, então, que por volta de 1980 os computadores seriam mais capazes que a mente humana em todos os tipos de processamentos, mas que os avanços intelectuais ainda teriam que ser realizados pelos humanos, que poderiam se beneficiar da ajuda dos computadores para tal. Para ilustrar tal situação, ele começou a classificar como ele gastava suas horas de trabalho, chegando a uma conclusão perturbadora de que quase 85% de seu tempo era gasto preparando-se para pensar, ou seja, realizando atividades mecânicas como reunir informações, plotar gráficos, converter unidades, enfim, preparando-se para uma decisão ou uma intuição. Licklider via a possibilidade de os computadores realizarem essa tarefa de preparação através de mecanismos de interação com o humano, amplificando sua mente.

Neste trabalho, utiliza-se o termo amplificação de inteligência, porém a tradução do termo “intelligence augmentation” para “amplificação de inteligência” pode trazer alguma confusão. O próprio Engelbart colocou em um de seus trabalhos que o uso do termo “intelligence amplification” não era adequado, pois o que se buscava não era aumentar a inteligência nativa humana, mas apenas criar ferramentas que possibilitassem aos humanos utilizar melhor a sua inteligência nativa. No entanto, Engelbart voltou atrás nessa concepção, aceitando o termo “intelligence amplification” como válido, deixando isto explícito no seu manifesto (Engelbart, 1962):

“Aceitar o termo “intelligence amplification” não implica nenhuma tentativa de aumentar a inteligência nativa do ser humano. O termo “intelligence amplification” parece aplicável ao nosso intuito de amplificar o intelecto humano no qual a entidade a ser produzida

vai exibir mais daquilo que pode ser chamado de inteligência do que um ser humano sozinho poderia; nós teremos amplificado a inteligência de um humano através da organização de suas capacidades intelectuais em níveis superiores de estruturação sinérgica”.

Para ilustrar como as ferramentas mentais são importantes para auxiliar e amplificar o pensamento humano, Peirce chega a comparar a experiência de arrancar a parte do seu cérebro que comanda a fala com roubar sua tinta de escrever, no sentido de que em ambos os casos ele não seria capaz de continuar sua discussão, ou seja, seus pensamentos não viriam até ele. Segundo ele, isso se dá porque sua faculdade de discutir está igualmente localizada na sua tinta de escrever e na sua fala. Quando assim colocou, ele não estava afirmando o óbvio de que não seria capaz de escrever seus pensamentos sem sua tinta de escrever, mas sim que os pensamentos dele chegam através do ato de escrever, principalmente aqueles que compõem agrupamentos e cadeias muito longas para serem desenvolvidos simplesmente na consciência mental (Skagestad, 1996). Assim, ele destaca que o conhecimento está menos em estados mentais do que na capacidade de meios externos induzirem estados mentais. Einstein uma vez disse, a esse respeito, que seu lápis era mais inteligente do que ele próprio. Sobre isso, parafraseando Benjamin Lee Whorf (Whorf, 1964), e sua hipótese de que a visão do mundo de uma determinada cultura é limitada pelo repertório que sua linguagem consegue representar, Engelbart coloca: “tanto a língua utilizada por uma cultura, como sua habilidade para atividades intelectuais efetivas, são diretamente afetadas durante sua evolução pelos meios pelos quais os indivíduos controlam a manipulação externa de símbolos”. Whorf estabelece que a falta de palavras para um determinado conceito em uma língua torna mais difícil expressar tal conceito, o que por sua vez tornará mais difícil o aprendizado acerca do conceito, o que tornará mais difícil a existência daquele conceito. Assim, a medida que uma língua se espalha e começa a ser utilizada, ela afetará o desenvolvimento de novos conceitos. A partir desse fato de que a evolução está ligada a manipulação externa de conceitos através de ferramentas mentais, Engelbart sugere integrar as capacidades das máquinas nas atividades intelectuais humanas. Segundo ele, a humanidade passou por quatro estágios de evolução das capacidades intelectuais até hoje, sendo estes:

1. Manipulação de conceitos, simples formação de abstrações mentais. Esses conceitos são conceitos ainda primários, numa forma não-verbal, manipulações não-simbólicas de conceitos, como perceber que a queda de uma pedra poderá machucar alguém e tomar uma atitude para impedir isso. Os conceitos foram formados através de abstrações. Perceba como esses pensamentos simples são suficientes para atividades como construir abrigo ou caçar.
2. Manipulação de símbolos, não significando habilidade de comunicação por símbolos, mas simples habilidade individual de representar objetos simbolicamente. Essa foi uma grande evolução do homem, ao ser capaz de representar conceitos na sua mente através de símbolos específicos. Ao invés de ter que gravar como cada uma das cinquenta ovelhas se pareciam, o pastor poderia apenas gravar um número cardinal em sua mente, para ter idéia da quantidade de ovelhas.
3. Manipulação externa de símbolos, linguagem e escrita, como escrever com papel e lápis. Essa evolução particularmente ajudou muito na resolução de problemas, devido a capacidade de externalizar símbolos, particularmente de maneira gráfica. Isso potencializou a memória e a capacidade de visualização do humano, e está associado a evolução da presente forma de manipulação de conceitos.

4. Manipulação externa e automatizada de símbolos: integração das capacidades das máquinas nas atividades intelectuais humanas.

A respeito do quarto estágio, que é o atual, de cooperação cognitiva entre humanos e máquinas, Engelbart diz o seguinte:

“Neste estágio, símbolos com os quais os humanos representam os conceitos que eles manipulam podem ser colocados diante de seus olhos, movidos, guardados, recuperados, operados de acordo com regras complexas tudo com respostas muito rápidas para um mínimo de informação provida pelo humano, por meio de dispositivos tecnológicos especiais de cooperação” (Engelbart, 1962).

Dessa maneira, o fato de a mente humana estar evoluindo não significa somente que as estruturas internas do sistema nervoso estão mudando, mas também que as formas de representação, os instrumentos materiais e a “mídia de cognição” evolui, como com o uso de interfaces gráficas com o usuário, mouse, hipertextos, processador de texto, planilhas de cálculo, entre outros (Ransdell, 2003). O desenvolvimento de processadores de texto como ferramentas computacionais possibilitou, por exemplo, que escritores pudessem inserir com facilidade no meio do texto pensamentos que chegassem em outro tempo. Na leitura de hipertextos, não há sequer uma ordem predeterminada para os pensamentos chegarem na mente. Com relação a esta evolução proporcionada pelo quarto estágio, a introdução de interfaces gráficas pela Apple em 1984 foi tão importante quanto a própria introdução dos computadores na década de 1950 (Skagestad, 1996).

Assim, Engelbart lança a hipótese que ele mesmo define como Neo-Whorfiana: não somente a linguagem utilizada por uma cultura influencia as capacidades de realizar atividades intelectuais, mas também a evolução dos meios e ferramentas mentais de manipulação externa de símbolos. Segundo ele, essa evolução intelectual através de ferramentas mentais se dá principalmente devido à uma estrutura e organização particular e mais adequada proporcionada pela ferramenta. Segundo Engelbart (1962), um humano selvagem jamais conseguiria dirigir um carro no trânsito, devido ao fato dele não ser capaz de preencher o espaço entre seu background cultural e o mundo onde existem carros e tráfego. No entanto, seria possível realizar um programa especial de treinamento que levaria o selvagem a ser capaz de dirigir corretamente. Isso se dá porque, segundo Engelbart, a mente humana não aprende ou atua através de longos passos, mas pequenos passos organizados, estruturados e dependentes entre si. O ser humano não é capaz de dar longos passos em inovação, compreensão e execução de processos, comparados aos passos necessários para resolver um problema não estruturado (como dirigir para o selvagem), mas de fato humanos resolvem problemas não estruturados. Isso porque as ferramentas mentais de amplificação utilizadas por eles servem para quebrar o problema em pequenos passos de maneira que o raciocínio humano possa caminhar ao longo dele com seus pequenos passos, e a estrutura e organização desses pequenos passos farão a diferença entre um processo mais ou menos eficiente. A amplificação de inteligência procura introduzir ferramentas mentais novas e sofisticadas para essa organização, estruturação e manipulação.

Hoje em dia, conforme dito por Peter Skagestad (1998), acredita-se que a interface de diálogo mais avançada para amplificar a mente na resolução de um problema seria um ambiente totalmente imerso em realidade virtual, onde o usuário estaria com todos os seus sentidos imersos em um ambiente altamente interativo e direcionado para ajudá-lo no controle da manipulação externa de símbolos. No entanto, segundo Biocca (1997), a realidade virtual ainda está muito imatura. Biocca fala sobre

um interessante padrão no desenvolvimento das interfaces de realidade virtual o qual ele chama de “incorporação progressiva”³. Segundo este padrão, os desenvolvimentos nas tecnologias de sensores e displays levam cada vez mais a um acoplamento entre corpo e interface na realidade virtual, estando o corpo presente tanto no espaço físico quanto no espaço virtual, ou cyberspace. Com isso, as interfaces se adaptam ao corpo, e o corpo se adapta às interfaces. Biocca apresenta, assim, uma série de questionamentos psicológicos e sociológicos a este respeito, e termina por apresentar uma questão a qual denominou “o dilema do cyborg”⁴: o dilema das interfaces de realidade virtual se tornarem cada vez mais naturais e humanas, faz com que elas se adaptem mais ao corpo e à mente humanas. Quanto mais ocorre essa adaptação, mais o corpo e a mente humanas se adaptam à interface não-humana. Assim, quanto mais natural esta interface não-humana, mais o humano se tornaria não-natural, ou seja, um cyborg. Este interessante questionamento está de fato ligado à amplificação de inteligência proporcionada pela exosomatização dos pensamentos e manipulação externa de signos proporcionadas pela realidade virtual. Uma ferramenta de amplificação de inteligência deste tipo seria como uma reconstrução da realidade, mas uma realidade onde a inteligência humana estaria amplificada, ou seja, onde parte dos processos cognitivos seriam feitos para nós pelo sistema, que apenas nos apresentaria unidades de conhecimentos e informações prontas para serem usadas em tarefas deixadas para nossa inteligência. Um exemplo disso seria um CAD com realidade virtual, que poderia ser usado por um arquiteto para passear por dentro de sua construção. Nessa realidade virtual, os números e signos apareceriam na frente do arquiteto enquanto ele caminha pela construção para lhe indicar problemas na estrutura. É a criação de uma realidade, onde nosso intelecto estaria amplificado devido a este suporte da ferramenta mental. A respeito da possibilidade dos humanos evitarem este dilema, rejeitando novas tecnologias, Biocca coloca que seria equivalente a negar que a relação humana com a tecnologia é uma relação de evolução, ou seja, que os humanos evoluem através de ferramentas criadas, mesma visão apresentada anteriormente neste capítulo. Além disso, Biocca assume que as tecnologias criadas pelos humanos não devem ser vistas como algo extraterrestre, mas sim como um reflexo do próprio humano, sendo natural dele, portanto, abraçar essas tecnologias criadas. Ele diz:

“Pode mesmo ser que o corpo e a mente do ser humano tenham evoluído para incorporar totalmente essas externalizações de processos mentais e amplificações do corpo que são nossas tecnologias.” (Biocca, 1997)

Portanto, a “incorporação progressiva” não é mais do que parte de um padrão, que é a evolução cultural dos humanos através dos artefatos mentais.

Para ilustrar um exemplo negativo de amplificação de inteligência, poderia-se pedir que duas pessoas com as mesmas capacidades em determinado assunto escrevessem uma dissertação sobre esse tema, dando a uma lápis e papel e à outra uma pedra e uma parede, tornando o trabalho de escrita da segunda extremamente laborioso. Obviamente, a primeira pessoa desenvolveria raciocínios mais avançados que a segunda, o que ilustra o fato de a mente evoluir desenvolvendo artefatos que exosomatizem o conhecimento, que parecem adicionar novas dimensões à capacidade humana, como quando se usa lápis e papel para fazer uma soma que não se consegue realizar somente na mente. Um exemplo bastante simples e intuitivo é a comparação entre um CAD rodando em uma tela de 14” com resolução de 640x480 pixels e o mesmo programa rodando em um computador com 2 monitores de

³Tradução destes autores para o termo em inglês “progressive embodiment”

⁴tradução destes autores para o original em inglês “the cyborg’s dilemma”

19” com resolução de 1280x1024. Percebe-se claramente que é muito mais difícil fazer um projeto no primeiro caso do que no segundo, apesar do sistema utilizado ser o mesmo.

5.2.1 Uma ferramenta mental avançada para amplificar a inteligência

Para ilustrar um tipo de sistema que poderia ser criado com um meio de expressão altamente competente, descreve-se a seguir um sistema que poderia ser criado neste sentido, baseando-se em um exemplo contido na introdução de Engelbart (1962). É um exemplo ilustrativo e fictício, apenas para tornar mais clara a idéia que vem sendo desenvolvida.

Imagine um arquiteto em seu trabalho de fazer o projeto de um novo edifício, utilizando-se de uma ferramenta de amplificação de inteligência avançada. Ele se senta na sua estação de trabalho em frente a uma tela de 2m X 3m (ela forma uma parede em sua frente), que é sua superfície de trabalho controlada por um computador. O arquiteto se comunica com a ferramenta através de reconhecimento de voz, toques na tela, teclado, pointer laser, além da possibilidade de utilizar um ambiente de realidade virtual que inclui óculos, luva e um cockpit para simular a realidade perante seus sentidos. Outros dispositivos também estão a serviço da comunicação. O arquiteto está fazendo o projeto de um edifício. Ele imagina várias possibilidades e inicia testando-as na tela, entrando com todos os dados para seu “case” no sistema através de uma conversa com a ferramenta, que reconhece sua fala e também sintetiza voz. Alguns dados são criticados pela ferramenta de amplificação, que mostra a impossibilidade de levantar certos arcos ou a tensão excessiva em certas treliças desenhadas. Após um debate entre os dois, chega-se a um acordo sobre os dados iniciais, tendo a ferramenta procurado otimizar todos os detalhes pedidos, como fator de segurança, uso de material mínimo, entre outros. No caso desses objetivos serem conflitantes e haver mais de uma solução não-dominada na fronteira de Pareto (uma discussão mais aprofundada sobre soluções na fronteira de Pareto pode ser encontrada no capítulo 6), a ferramenta apresenta as soluções na fronteira de Pareto interpretadas graficamente para que o arquiteto escolha a mais conveniente. Assim, o arquiteto pede à ferramenta que lhe dê uma visão perspectiva do edifício a partir da estrada que fica a alguns metros dali, além de uma visão das árvores que irão permanecer no loteamento. Com uma pistola de laser, o arquiteto aponta dois pontos de interesse na tela, e a ferramenta lhe fornece a distância e elevação entre eles no canto inferior esquerdo. Agora, com um pointer e o teclado, o arquiteto traça uma linha de referência na tela. Gradualmente, a tela começa a exibir o trabalho que ele está realizando, mostrando uma escavação interessante em uma subida de encosta ao lado do edifício. Depois de alguns minutos, o arquiteto coloca uma visão de cima da escavação. Alguns minutos de estudo e o arquiteto pede detalhes sobre uma lista de itens estéticos que poderiam ser construídas naquela posição para fins decorativos, grande parte das perguntas diz respeito a viabilidades técnicas. Ele marca cada um dos itens para um estudo posterior, solicitando que a ferramenta lembre-o de fazê-lo. Como próximo passo, o arquiteto começa a entrar com uma série de itens a serem colocados na região da escavação: paredes de concreto de 4 cm de espessura e 4 m de altura, pisos de 3 cm de espessura, entre outros. Quando ele termina, vê-se uma estrutura tomando forma na tela. Ele a examina, faz ajustes, uma pausa para buscar em seus handbooks informações relevantes, além de pedi-las à ferramenta. Dessa maneira, a ferramenta de amplificação vai traduzindo todo o realizado pelo arquiteto em uma lista de especificações, que vão sendo modificadas e adicionadas. Essa lista evolui, tornando-se cada vez mais detalhada, estruturalmente interligada, representando o pensamento maduro por trás do projeto atual.

Montando diferentes planos em diferentes posições, superfícies curvas, transladando toda a estru-

tura em alguns metros para a esquerda, ele finalmente chega ao equilíbrio que queria com relação aos materiais a serem utilizados e à função daquele edifício. Para inserir informação detalhada sobre o interior, o arquiteto prefere ficar imerso em realidade virtual. Sentado em um cockpit onde praticamente todos os seus sentidos são utilizados para interagir com o sistema, com óculos, luva, macacão, entre outros, ele adentra a construção, caminhando pela obra. Seta algumas texturas para paredes e entra em um dos quartos. Ele verifica se a reflexão da luz no vidro espelhado do prédio não irá atingir os motoristas na estrada que passa logo a diante, e a ferramenta lhe informa que uma das janelas irá refletir fortemente a luz entre as 7 e 8 da manhã nas manhãs de verão. Logo após, ele inicia uma análise funcional, ainda em realidade virtual. Com uma estimativa do número de pessoas que irão ocupar o edifício e suas posições, ele segue cada turno, verificando como as portas abrem, onde se faz necessário mais iluminação e onde há mais tráfego no edifício, para determinar regiões de maior espaço e saídas de emergência. Todas essas informações são armazenadas pelo sistema na forma de uma lista de especificações. Outro arquiteto, um engenheiro, mestre de obras e cliente podem manusear o trabalho com a ferramenta e propor sugestões e idéias que os interessem. Eles também podem colocar anotações especiais que são integradas ao manual de projeto da ferramenta para benefício próprio ou de outros, evoluindo o sistema. Neste arcabouço, a capacidade matemática do computador, especialmente na explosão combinatória de possibilidades, é utilizada sempre que necessário. Além disso, outras capacidades não-matemáticas do computador como organização, planejamento e análises também são utilizadas.

Obviamente, um sistema como este está mais apto a existir em filmes de ficção científica do que em produções tecnológicas atuais na resolução de problemas não estruturados. No entanto, o importante neste exemplo é a analogia e a utilização dos conceitos apresentados para a exosomatização do pensamento.

5.3 Amplificação de inteligência versus inteligência artificial

5.3.1 Histórico

Não se sabe ao certo quando a primeira distinção entre inteligência artificial (IA) e amplificação de inteligência (AI) foi estabelecida. Ela pode ter surgido em 1962, quando Engelbart (1962) descreveu a capacidade de utilizar a computação para amplificar a inteligência do ser humano como “intelligence augmentation”. Certamente, Peter Skagestad (1996) distingue explicitamente os termos inteligência artificial e amplificação de inteligência, e os associa a visões de construção de sistemas computacionais inteligentes, duas partes da “inteligência computacional” (IC) que não são necessariamente opostas e nem necessariamente complementares nas suas abordagens. De fato, é difícil distinguir se há uma intersecção nessas abordagens ou se são partes desconexas de um mesmo conjunto chamado inteligência computacional, mas pode-se notar que uma consegue tirar proveito dos avanços da outra. Portanto, podemos dizer que estão de certa maneira acopladas.

Para ilustrar esse acoplamento, alguns tipos de tecnologia, como reconhecimento e sintetização de voz, possuem raízes na IA, onde foram criadas com o intuito de tornar as máquinas capazes de simularem humanos. No entanto, hoje em dia, são aplicadas como meio de melhorar a interface entre usuário e computador como uma forma de AI. Apesar de estarem acopladas, em algumas situações elas são controversas devido aos diferentes conceitos de mente que existem em cada uma dessas

visões. O que Skagestad argumenta é que as duas visões são conceitualmente independentes uma da outra, sendo o computador de fato dois modelos de mente diferentes ao mesmo tempo: de um lado, um mecanismo com algoritmos incorporados capazes de simular funções cognitivas humanas; do outro lado, um instrumento para coordenar processos cognitivos humanos de maneira a amplificar a inteligência.

Skagestad distingue entre IA e AI como sendo dois diferentes paradigmas de programação (Skagestad, 1996) fundados em dois tipos de máquinas diferentes. No caso da IA, a máquina de origem é a Universal Machine, descrita por Turing (1936), enquanto no caso da AI tem-se o Memex, de Bush (1945). Ambas as máquinas tentam mecanizar funções da mente humana. Turing tentou mecanizar os processos de raciocínio e inferência, enquanto Bush os processos do funcionamento da memória. No entanto, a abordagem dessas máquinas era diferente no sentido que o Memex não tentava simular a mente humana, como acontecia na máquina de Turing e acontece na IA, mas sim estender o seu funcionamento, replicando o funcionamento da memória de maneira a fornecer informações guardadas do passado mais rapidamente e fazendo com que as gravações mais relevantes estivessem sempre disponíveis quando necessárias. Essa idéia inspirou o programa de pesquisa conhecido como “intelligence augmentation” (amplificação de inteligência), formulado em 1962 por Douglas Engelbart.

5.3.2 IA x AI

Pode-se diferenciar inteligência artificial de amplificação de inteligência afirmando que inteligência artificial é uma técnica de programação de computadores que busca criar programas e máquinas que pensem como o ser humano, possuindo inclusive vontades e emoções, enquanto amplificação de inteligência seria uma técnica de programação em que se busca criar programas que atuem aumentando a capacidade do ser humano de manipular signos ou unidades de conhecimento, ou seja, de realizar cognição ou semiose, para resolver problemas não estruturados, e não substituí-la por um processo em uma máquina (Ransdell, 2003).

No entanto, é importante lembrar que a inteligência artificial não se resume somente a tentar imitar a mente humana. Russel & Norvig (1995) apresentam a IA composta por quatro áreas, conforme figura 5.2.

	Como seres humanos	Racionalmente
Pensar	Ciências Cognitivas	Lógica
Agir	Teste de Turing	Agentes Inteligentes

Fig. 5.2: Áreas que compõe a IA (Russel & Norvig, 1995)

Ao falarmos de IA, a tomaremos segundo o intuito de realizar as capacidades tipicamente humanas, como raciocínio e emoção, em máquina.

Para construir sistemas cuja função é ajudar a mente humana, faz-se necessário possuir um entendimento profundo de como ela funciona. Assim como se tem as ciências cognitivas e Turing como

alicerces da inteligência artificial na criação de mentes artificiais, no caso da amplificação de inteligência, Skagestad coloca a semiótica Peirceana como provedora da base teórica para a amplificação de inteligência: o modelo semiótico da mente. Dessa maneira, a mente humana é um sistema semiótico, ou seja, capaz de manipular signos, que são definidos por Peirce como “algo que está relacionado a outra coisa de alguma maneira para alguém”. Há uma relação triádica para o signo, que representa um objeto causando o mesmo interpretante que aquele objeto causaria se estivesse em seu lugar.

É importante citar que essa relação de “estar relacionado a outra coisa” não existe na máquina de Turing, que manipula tokens mas não consegue relacioná-los a referentes do mundo real. Skagestad afirma que Engelbart seguiu os mesmos caminhos da teoria de Peirce que, desconhecida por ele, havia sido formulada sessenta anos antes no contexto de uma doutrina filosófica dos signos. Peirce antecipou o modelo de Engelbart, este inspirado por Bush, aquele merecendo o título de filósofo da amplificação de inteligência. Assim, Ransdell estabelece o primeiro princípio da AI a partir de um dictum de Peirce (Ransdell, 2003):

“Todo pensamento se dá em signos”, baseando-se na teoria semiótica de mente de Peirce para explicar a exosomatização do pensamento.

A distinção entre as duas categorias IA e AI está implícita na literatura nas últimas décadas. Só recentemente essas categorias têm sido reconhecidas como dois ramos distintos com igual importância em seus desenvolvimentos. Skagestad sugeriu que essa distinção explícita pode ser atribuída ao cientista da computação Frederick Brooks (citado por Ransdell (2003)), que afirmou:

“Eu acredito que o uso dos sistemas computacionais para amplificação de inteligência é muito mais poderoso hoje em dia, e será em qualquer momento no futuro, do que o uso dos computadores para inteligência artificial. Na comunidade de IA o objetivo é substituir a mente humana pela máquina, seus programas e sua base de dados. Na comunidade de AI, o objetivo é construir sistemas que amplifiquem a mente humana providenciando auxiliares baseados em computador que realizem coisas que a mente humana tem dificuldade em realizar”.

Essa crença também pode ser atribuída ao fracasso da IA com relação às expectativas criadas nas décadas de 50 e 60, devido a uma série de problemas e restrições identificados na tentativa de construir sistemas que realizem o papel de uma mente humana.

Há vários exemplos e discussões famosas que ilustram esses muitos problemas que a IA encontrou na busca de simular a mente. Esses problemas, que atingiram fortemente e sempre tentaram decepar as iniciativas da IA, não atingiram a AI dessa mesma maneira, pelo menos não à princípio, devido às diferentes abordagens de ambas, o que explica essa crença de Brooks mais favorável com relação à AI.

5.3.3 IA

Dentro da própria IA, há uma separação entre dois conceitos: IA forte X IA fraca (Gardner, 2003). Na visão da IA forte, um computador corretamente programado para simular a mente humana é, com efeito, uma mente, possuindo estados cognitivos. Na visão da IA fraca, por outro lado, o computador funciona apenas como um modelo do funcionamento da mente, mas não pode ser de fato considerado uma mente em si. Diferentemente é a situação da AI, já que uma ferramenta mental

de amplificação de inteligência não tem o intuito de ser uma mente, nem mesmo de ser um modelo de uma mente humana em funcionamento: o intuito é de amplificar a capacidade dos processos da mente humana. Considerando que o pensamento advém da ação do signo, gerando um interpretante, a ferramenta mental coordena o processo de manipulação dos símbolos da melhor maneira possível, ajudando nos processos de iconização, dando feedbacks negativos em certas situações, comprimindo informações, fornecendo informações relevantes sempre que necessário, entre outros. Assim, não importa que o que esteja acontecendo no programa seja apenas manipulação de estruturas de dados e informações, ou uma simples sequência algorítmica, como estipulam os ferrenhos críticos da IA, pois o conjunto humano+ferramenta mental manipula símbolos (ainda que a tarefa da referência ao existente no mundo real fique para o humano) e é capaz de resolver problemas não estruturados de uma maneira mais eficaz e chegar numa melhor resposta do que ambos chegariam separadamente. Portanto, pode-se dizer que o conjunto homem + ferramenta mental é uma mente. Mais ainda, é uma mente amplificada, pronta para resolver problemas não estruturados de maneira mais adequada. Entre outras críticas e discussões referentes à IA, vale citar alguns clássicos para entender qual é o seu relacionamento também com a AI:

- Quarto chinês (Searle, 1980) e problema de fundamentação do símbolo
- Problema do enquadramento ⁵
- Intencionalidade
- Cognição situada e incorporada

Quarto chinês (Searle, 1980) e problema de fundamentação do símbolo

O argumento do quarto chinês foi escrito por John Searle para combater a visão da IA forte com relação ao fato de os computadores possuírem estados cognitivos e intencionalidade. Imagine um falante de inglês que é totalmente ignorante em chinês. Imagine também que ele recebe um livro que é uma espécie de dicionário, o qual lhe ensina a escrever respostas para perguntas em chinês como numa espécie de dicionário algorítmico. Nesse livro, estariam teoricamente todas as perguntas que poderiam ser feitas em chinês, com os ideogramas desenhados, e as respostas corretas relativas a estas perguntas, também desenhadas na forma dos ideogramas característicos da língua. Esse falante de inglês é colocado juntamente com o livro dentro de uma sala fechada, onde recebe papéis com perguntas em chinês. Sua tarefa é respondê-las em outra folha de maneira correta, com a ajuda do livro. Assim sendo, ao receber a pergunta, nosso herói buscaria essa pergunta no livro e apenas copiaria a resposta na folha para devolvê-la para fora da sala por uma outra abertura do outro lado, conforme indicado no dicionário algorítmico, como ilustrado na figura 5.3.

No entanto, nosso personagem não tem a menor idéia do que está sendo perguntado, e tampouco do que ele está respondendo. Ainda assim, para um falante de chinês que está fora da sala, é como se lá dentro estivesse alguém que falasse chinês e que está respondendo às suas perguntas corretamente. Agora imagine que, juntamente com a pergunta em chinês, fosse também entregue uma pergunta em inglês, língua do nosso personagem do quarto. Ele responderia às duas perguntas e as devolveria para fora do quarto. Do ponto de vista dos personagens que estão fora do quarto, essa pessoa fala tanto

⁵Tradução destes autores para o termo original do inglês *Frame problem*

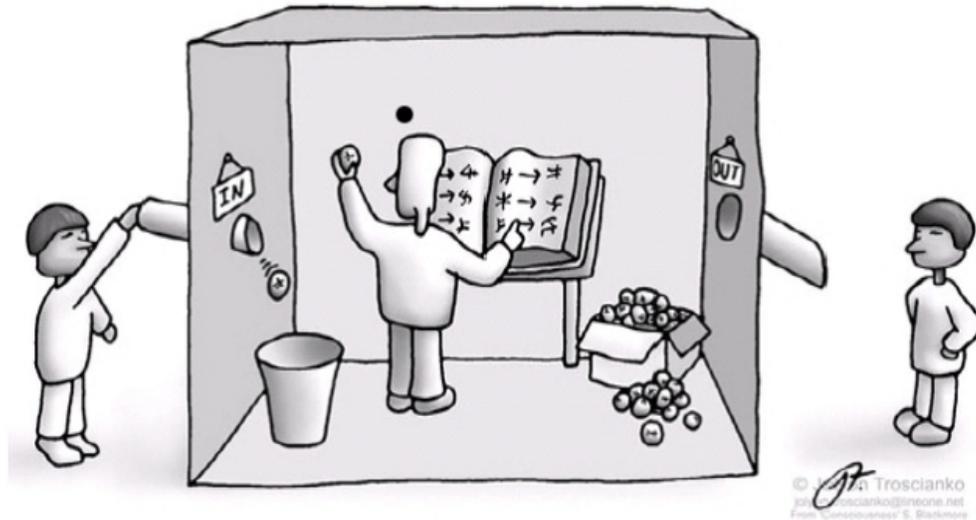


Fig. 5.3: Quarto chinês (Searle, 1980)

inglês quanto chinês. Porém, sabemos que o processo para responder as perguntas nos dois casos é completamente diferente. No caso da pergunta em chinês, nosso herói apenas busca no seu livro os ideogramas e copia a saída supostamente correta para o papel destinado a voltar para fora da sala, sem entender o que aqueles símbolos estranhos significam. No caso da pergunta em inglês, nosso herói a lê, entende os signos explicitamente e responde depois de raciocinar a respeito da questão. Fica claro que no primeiro caso não há estados cognitivos relacionados à semântica da pergunta, enquanto no segundo caso, há. Para Searle, o primeiro caso é o que acontece no computador: ele não possui estados mentais, apenas realiza operações segundo um algoritmo programado. O computador não é capaz de manipular os símbolos existentes no processo realizando a referência com o objetos do mundo real que aqueles signos representam, o que é conhecido como problema de fundamentação do símbolo.

Stevan Harnad (1990) explica o problema de fundamentação do símbolo de uma maneira própria e interessante, com o exemplo que ele chamou de “dictionary-go-round”. Segundo Harnad, há dois exemplos para o problema de fundamentação do símbolo: um que é muito difícil, e outro que é impossível. No primeiro deles, relativo à tarefa difícil, imagine que você tenha que aprender chinês como uma segunda língua, e que tudo o que esteja disponível para tal seja um dicionário. Seria muito difícil conseguir fazer mais do que simplesmente passear de símbolo para símbolo, sem conseguir extrair nenhum significado, entrando em uma espécie de loop que muitas vezes as definições dos dicionários nos levam. No entanto, arqueologistas conseguem realizar tarefas mais ou menos parecidas com estas ao desvendar códigos, línguas, desenhos e escrituras antigas, pois se baseam numa primeira língua e em conhecimentos prévios sobre determinadas situações. Agora, imagine que você tivesse que aprender chinês como uma primeira língua, somente com o mesmo dicionário: essa é a tarefa impossível. Não haveria como sair do loop de passar de símbolo para símbolo, sem conseguir nenhum significado. Como dar sentido a coisas que não passariam de símbolos sem sentido? Essa é a tarefa que os modelos de mente puramente simbólicos da IA enfrentam hoje em dia. Esse é o problema de fundamentação do símbolo. Muitos sistemas construídos segundo modelos simbólicos na IA, e

que a princípio pareciam demonstrar comportamento inteligente foram duramente criticados segundo esta ótica. As críticas pregam que não há nenhum comportamento inteligente, nenhuma intencionalidade nas decisões tomadas por esses sistemas: o que há não passa de uma simples execução de um algoritmo, com rotinas mecanicistas e aleatórias, porém, nada inteligentes. Não há como atribuir este mesmo tipo de problema a ferramentas de amplificação de inteligência. Independentemente do que esteja acontecendo no lado da máquina (considere que seja nada mais do que alguns algoritmos processando algumas estruturas de dados), o que acontece do lado do ser humano que está ali em um processo de busca por uma solução não pode ser entendido como simplesmente execução de algoritmos, havendo estados cognitivos em sua mente associados à interação homem-máquina. De fato, é o usuário humano que lida com o problema de fundamentação do símbolo, fazendo a referência necessária com os objetos do mundo real.

Taddeo & Floridi (2005) apresentam uma revisão de quinze anos de pesquisa para solução do problema de fundamentação do símbolo (PFS). Os autores apresentam oito propostas de estratégias para resolver o PFS e as avaliam segundo um requisito que chamam de “zero semantical commitment condition”, concluindo que todas as abordagens são comprometidas semanticamente, e logo não capazes de resolver o embate. As abordagens atacadas pelos autores se baseiam em estruturas representacionalistas.

Com tantos problemas para a IA, resta a dúvida se máquinas que realizem cognição são realmente possíveis. Muitas das iniciativas da IA em décadas iniciais falharam por basear-se no mesmo mecanismo de Turing, no qual as decisões e ações realizadas pela máquina eram puramente baseadas em lógica matemática simulada, deixando de lado importantes fatores que existem na mente humana, como julgamentos, intuição, espontaneidade e emoções. Esse tópicos têm sido considerados em pesquisas recentes.

Problema do enquadramento

O problema do enquadramento surgiu primeiro na lógica, nos laboratórios de IA, mas foi herdado como problema epistemológico pelos filósofos (Shanahan, 2004). Os sistemas construídos com os modelos atuais da IA sofrem com o problema de ter que modelar um mundo infinito e dinâmico. Há sempre o problema de ter que atualizar crenças com relação aos estados do mundo e de representar seus detalhes com o nível de abstração adequado, o que a mente humana parece fazer muito bem, na maioria das vezes, e quando se encontra em estado normal. Considere o exemplo de Shanahan (2004), onde há um robô cuja tarefa é preparar uma xícara de chá. Esse robô precisa tirar uma xícara do armário. A posição da xícara é representada internamente no robô por uma sequência de números em uma string, assim como todas as outras informações do ambiente, como temperatura, cor da parede, posição dos seus braços, entre outros. Ao pegar a xícara, o robô precisa atualizar essas informações. Obviamente, a posição da xícara precisa ser atualizada. Mas a questão é: quais outras informações também precisam ser atualizadas como consequência dessa ação? A temperatura não será afetada, mas uma colher que estava no pires e veio junto com a xícara também mudou de posição. O problema epistemológico relacionado é como o robô limitaria o escopo das proposições a serem atualizadas como consequência de uma ação, pois se o robô tiver uma inteligência próxima a do ser humano, será computacionalmente intratável percorrer todas as proposições e verificar se devem ser atualizadas. Apesar de muitas vezes os seres humanos errarem ao prever todas as consequências de suas ações, a mente humana parece desempenhar bem essa função, com um mecanismo de atenção

que limita o acesso ao espaço global.

Intencionalidade

O problema da intencionalidade se refere ao fato de um sistema inteligente apresentar vontades, crenças e objetividade em suas decisões. Daniel Dennett, ao criar o conceito de sistemas intencionais, procurou explicar o comportamento dos computadores atribuindo-lhes crenças e desejos. No entanto, este problema está relacionado também ao problema de fundamentação do símbolo e ao quarto chinês. O próprio Searle levantou a hipótese dos sistemas não apresentarem real intenção ao tomar decisões, da maneira análoga ao fato de não conseguirem manipular unidades de conhecimento, ou seja, signos. A questão que surgiu logo após essas críticas foi a seguinte: o que caracteriza intencionalidade em sistemas inteligentes? O próprio Searle procurou responder a questão em um livro escrito especialmente para esse fim (Searle, 1983). No entanto, não conseguiu grande apelo na comunidade. A intencionalidade é importante porque é considerada a marca do mental. Assim, demonstrar que um sistema possui intencionalidade é equivalente a dizer que o mesmo é uma mente. A questão da intencionalidade foi melhor resolvida um pouco depois por Bratman (1987). Segundo Bratman, para que haja intencionalidade, faz-se necessário haver formulação de planos, avaliação e escolha do melhor plano, e execução desse plano. Isso acabou sendo relacionado com a arquitetura Beliefs (crenças), Desires (desejos), Intentions (intenções), a arquitetura BDI, utilizada inclusive em agentes computacionais inteligentes. Dessa maneira, faz-se necessário possuir crenças, que é um conjunto de asserções construídas a partir de informações colhidas do mundo, desejos, que são basicamente metas, e intenções, que são os planos de ação. Perceba como a utilização de feedbacks negativos está relacionada a este conceito de intencionalidade: planos são formulados e testados introspectivamente, e feedbacks negativos ajudam a ajustá-los até a escolha do melhor plano de ação. Essa noção já existia desde a década de 1940, quando Rosenblueth et al. (1943), em um trabalho sobre propósito, intencionalidade, comportamento e teleologia, afirmaram:

“Todo comportamento com propósito pode ser considerado possuindo algum feedback negativo”

Nas ferramentas de amplificação de inteligência, o processo de busca pela solução é iterativo e interativo, ou seja, há uma crítica mútua entre o sistema e o usuário, que tentam adaptar as propostas de solução, que evoluem, até chegar em uma solução boa. Nesse processo de diálogo, o papel da ferramenta é o de amplificar a mente humana, controlando os processos e fluxos cognitivos da melhor maneira possível, além de atuar em áreas nas quais a mente humana geralmente não possui muito sucesso, como na explosão combinatória de certos problemas não estruturados, sendo possível identificar intencionalidade.

Cognição situada e incorporada

Um consenso que existe atualmente nas teorias cognitivas é que, para haver entendimento, a cognição necessita ser situada e incorporada. Isso equivale a dizer que, para existir uma mente, faz-se necessário existir um corpo, que sirva de interface entre a mente e o ambiente, “plugando” a mente no mundo, que será capaz de senti-lo através de sensores e atuar nele através de atuadores. Assim, os sistemas artificiais inteligentes necessitam de um avatar (corpo) e um ambiente, e a cognição é de fato

uma cadeia causal que envolve o cérebro, o corpo e o ambiente, resultado de uma interação complexa e dinâmica entre eles. Antonio Damasio (1994) chama a atenção para a tendência equivocada de pensar no corpo e na mente como dois sistemas separados:

“... o corpo como está representado na mente, constitui o enquadramento de referência indispensável para os processos neurais que nós experienciamos como mente; que nosso próprio organismo, antes de ser alguma experiência absoluta de realidade é utilizado como referência para as construções que nós fazemos do mundo que está ao nosso redor e para a construção do sempre presente senso de subjetividade que é parte e parcela de nossas experiências; que nossos mais refinados pensamentos e melhores ações, nossas maiores alegrias e piores dores, usam o corpo como uma referência de medida.”

O corpo funciona como uma espécie de simulador para a mente, sendo de fato uma interface de diálogo dela, um dispositivo de representação material das semioses internas, um veículo semiótico social para representar estados mentais internos, tais quais emoções, observações e planos (Biocca, 1997).

Os ambientes onde ocorrem cognição situada podem ser reais, como em robôs, ou simulados, com mundos artificiais, como acontece com criaturas artificiais e animats, que são uma espécie de animais simulados. Simular agentes autônomos inteligentes em ambientes reais é uma tarefa complicada, pois faz-se necessário lidar com todos os ruídos, imprecisões e problemas inesperados típicos da realidade. Muitas vezes o fato de lidar com este tipo de problema tira um pouco do foco da simulação nas tarefas mais relevantes do agente autônomo. Quando se simula em ambientes artificiais, é mais fácil focar nas atividades que se deseja simular, e pode-se fazer um modelo de mundo incremental, aumentando a dificuldade à medida que resultados são obtidos. No entanto, tem-se uma dificuldade de portar o experimento para o mundo real mais tarde. Ambientes simulados são muitas vezes considerados *toy-problems*, e alguns pesquisadores consideram que agentes autônomos necessitam lidar com problemas do mundo real para apresentar inteligência e cognição de fato.

5.3.4 AI

A AI não sofre essas mesmas críticas, pois o sistema analisado é o sistema formado pelo humano mais a ferramenta de amplificação. Segundo Engelbart, esse sistema híbrido entre humano e ferramenta de amplificação é chamado de H-LAM/T systems (Human using Language, Artifacts, Methodology, in which he is Trained). A respeito de computadores como H-LAM/T systems, Engelbart diz o seguinte (Engelbart, 1962):

“O sistema que queremos melhorar pode então ser visualizado como um humano treinado em conjunto com seus artefatos, linguagem e metodologia. O novo sistema explícito que contemplamos vai envolver como artefatos computadores, informações armazenadas por computadores, processamento de informação e dispositivos de apresentação dessa informação. Os aspectos desse arcabouço conceitual que são discutidos aqui são a princípio aqueles relacionados com a habilidade dos seres humanos de realizar uso significativo de tal equipamento em um sistema integrado.”

Quando um ser humano realiza uma tarefa cognitiva de alto nível, como escrever um texto, ele faz uso de vários sub-processos cognitivos. O processo de escrever um texto pode ser um sub-processo de uma tarefa de nível ainda maior, como organizar um evento. Cada indivíduo desenvolve um repertório de processos dos quais ele seleciona e evolui determinados processos para fazerem parte do processo que ele irá executar. A este repertório dá-se o nome de repertório hierárquico de capacidades (Engelbart, 1962). Ao executar um processo, o ser humano escolhe e adapta do seu repertório de capacidades, que é como um toolkit, os processos que irão compor a tarefa a ser executada. Todas as capacidades do repertório de um indivíduo estão relacionados a suas habilidades ou as habilidades de suas ferramentas mentais.

Segundo Engelbart (Engelbart, 1962):

“O que acontece, então, é que cada indivíduo desenvolve um certo repertório de capacidades de processo⁶ do qual ele seleciona e adapta aqueles que irão compor o processo que ele executa. Esse repertório é como um toolkit, e assim como um mecânico deve saber o que suas ferramentas podem fazer e como utilizá-las, também o trabalhador intelectual deve saber as capacidades de suas ferramentas e ter bons métodos, estratégias e regras para fazer uso destas. Todas as capacidades de processo no repertório do indivíduo descansam sobre as capacidades básicas dentro dele ou dentro de seus artefatos, e todo o repertório representa uma estrutura hierárquica (que nós geralmente denominamos de repertório hierárquico).”

Existem basicamente três categorias de processos no repertório hierárquico de capacidades de um indivíduo (Engelbart, 1962):

- processos executados completamente pelo humano, chamados de “capacidades de processo explícitas do ser humano”⁷;
- processos executados completamente por artefatos sem a intervenção do homem, chamados de “capacidades de processo explícitas do artefato”⁸;
- processos derivados de hierarquias contendo ambos os tipos anteriores, denominados “capacidades de processo compostas”⁹.

Ao escrever uma redação, utiliza-se uma série de capacidades do repertório como planejamento, desenvolvimento de raciocínios acerca do assunto, composição, entre outros. Se for um texto simples e curto, os dois primeiros processos (planejamento e raciocínios) serão explicitamente humanos, pois serão realizados internamente na mente, enquanto a composição será composta, já que utilizaremos ferramentas para escrever, como lápis e papel. No caso de ser uma redação complexa e grande, todos os processos necessitarão ser compostos, utilizando-se papel e lápis para planejar e produzir rascunhos. Sistemas que possuem a capacidade de realizar processos em qualquer um dos tipos de processos são sistemas H-LAM/T. Um indivíduo que vive em sociedade atualmente, é considerado

⁶Tradução desses autores para o termo original do inglês “repertoire of process capabilities” utilizado por Engelbart (Engelbart, 1962).

⁷Tradução destes autores para o termo original do inglês “explicit-human process capabilities”.

⁸Tradução destes autores para o termo original do inglês “explicit-artifact process capabilities”.

⁹Tradução destes autores para o termo original do inglês “composite process capabilities”.

um indivíduo amplificado intelectualmente, pois realiza uma série de processos cognitivos através de sistemas H-LAM/T, ou seja, utilizando a língua, artefatos e metodologias nas quais ele foi treinado para realizar suas tarefas. Como os processos presentes no repertório hierárquico de capacidades das ferramentas de AI, que configuram sistemas H-LAM/T, são dos três tipos mencionados, o poder cognitivo dessas ferramentas torna-se tamanho, aliando as habilidades humanas e das máquinas, que elas conseguem lidar com os problemas acima mencionados referentes à IA explorando as capacidades humanas e das máquinas, e a sinergia entre ambas traz uma resposta final ainda melhor. Os processos explícitos da ferramenta mental (computadores) não podem manipular símbolos e referenciar os existentes no mundo real, mas isso será realizado pelos processos explícitos do usuário humano. Tudo que ele precisa é uma maneira de inculcar essas informações no sistema, para que os processos explícitos da ferramenta mental possam se aproveitar dessa informação valiosa. Essa relação está mostrada na figura 5.4.

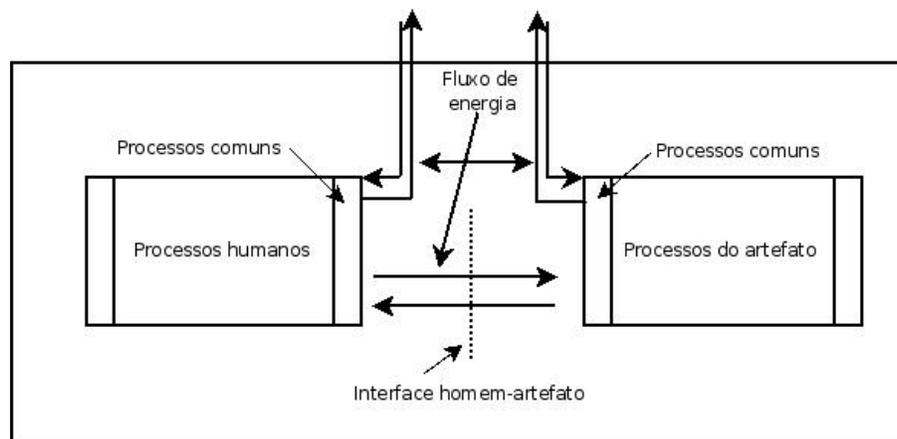


Fig. 5.4: Composição do H-LAM/T system (Engelbart, 1962)

Conforme dito anteriormente, todo processo do sistema pode ser decomposto em processos explícitos humanos ou explícitos da ferramenta. Em qualquer processo composto, haverá cooperação entre as partes, o que levará a uma troca de energia, grande parte dela para troca de informações. Assim, informações de cada um dos tipos de processo pode ser aproveitada pelo outro tipo, passando através da interface humano-artefato. Quando este artefato de amplificação é um computador, costumamos chamar a interface de interface humano-computador. Com efeito, interfaces desse tipo existem há séculos, desde que o homem começou a usar ferramentas mentais para amplificar seu intelecto. A troca de informações através dessa interface ocorrem quando processos humanos explícitos e da ferramenta estão acoplados. Alguns desses processos existem de fato somente para trocar informações. O mundo interage com o sistema H-LAM/T trocando energia com o humano ou a ferramenta, mas o foco da amplificação de inteligência está dentro do sistema, ou seja, nos processos que ditam a eficiência do sistema em garantir o entendimento humano e buscar seus objetivos. Essas trocas de energia que Engelbart fala a respeito, ocorrendo entre o mundo externo, humano e ferramenta, são de fato manipulações de signos, podendo ser semioses externas ou internas.

Quando fala-se que a evolução do intelecto humano tem ocorrido basicamente através da aplicação de ferramentas mentais que revolucionam a maneira de organizar o pensamento, o que se está modificando é exatamente este repertório hierárquico humano, enriquecido pela interação com os

repertórios hierárquicos das ferramentas mentais avançadas, criando novas capacidades cognitivas e novas formas de pensar para o seres humanos, ou seja, desenvolvendo novas capacidades no repertório hierárquico, potencializando o intelecto.

As abordagens da IA e AI não são opostas, no sentido de prover soluções conflitantes para o mesmo problema. De fato, elas atacam problemas diferentes e incorporam diferentes metas. Turing considera o ser humano essencialmente indistinguível de uma máquina, enquanto Bush considera o ser humano um usuário de máquinas, e sua intenção era desenvolver máquinas que manipulassem símbolos, máquinas para pensar “com o ser humano”, e não máquinas que “pensassem por si só” (Skagestad, 1993). A visão de Bush influenciou uma indústria que rapidamente muda a cultura e a sociedade, enquanto a visão de Turing se tornou um paradigma na pesquisa da IA e da ciência cognitiva. Segundo Skagestad, esse paradigma se tornou tão presente que muitas vezes coloca-se o modelo computacional como único modelo de mente disponível. No entanto, a AI se baseia em um outro modelo de mente, o modelo semiótico de Peirce, que apesar de não ter sido articulado por Bush, suporta completamente o programa de pesquisa iniciado por Engelbart e inspirado nas idéias dele.

Elas são rivais, no entanto, no sentido de criarem diferentes paradigmas para o uso dos computadores com relação à mente humana. A máquina de Turing aponta para a direção do uso dos computadores para simular a mente humana. Escolher esse modelo, como dito por Skagestad, significa comemorar cada avanço da tecnologia computacional como um passo a mais no sentido de replicar totalmente a mente humana em máquinas. Se for o Memex o escolhido como paradigma, a função passa a ser amplificar a mente. Com isso, sobram algumas questões interessantes. Conseguir replicar a mente humana em máquina seria também validar um modelo mecanicista da mente? E quanto à amplificação da mente através de máquinas, será que ela não traz elucidacões para a natureza da mente, visto que para ajudar no controle dos mecanismos da mente faz-se necessário entender profundamente seu funcionamento? O trunfo da AI é conseguir coordenar os vários fatores que existem no fluxo cognitivo, provendo meios de representação que ajudem a amplificar a inteligência.

5.4 Agentes computacionais inteligentes: amplificando a inteligência continuamente

No início da década de 90, a inteligência artificial passava por um momento de crise, arrastado por décadas. A abordagem tradicional da IA, baseada na manipulação abstrata de símbolos que não eram referenciados na realidade física, condenava todas as tentativas posteriores de desenvolvimento na área. Neste contexto, surgem metodologias que procuram se basear em uma interação com a realidade física, enfatizando uma interação contínua com o ambiente como fonte primária de restrições no desenvolvimento de sistemas inteligentes, apontando para novos rumos na IA. Rodney Brooks liderou uma dessas pesquisas no MIT (Brooks, 1990).

Essa nova metodologia considerava que a inteligência não ocorre a partir de um algoritmo ou procedimento centralizado, mas sim através de uma decomposição em módulos de comportamentos individuais, cuja coexistência e cooperação faziam com que um comportamento mais complexo (inteligente) emergisse. Os agentes e sistemas multi-agentes surgem neste contexto como uma demanda da cognição situada e incorporada, ou seja, da necessidade de possuir um corpo, que esteja “plugado” no ambiente, sentindo-o através dos seus sensores e atuando no mesmo através dos atuadores. A situabilidade, quando agentes estão situados em um ambiente, passa a ser a chave para a compreensão

do fenômeno cognitivo e da inteligência.

Engelbart não possuía a tecnologia de agentes computacionais quando definiu o sistema H-LAM/T (seção 5.3) como um sistema de dois domínios, sendo um deles representado pelo humano e outro pelo artefato de amplificação. No entanto, suas definições acerca da interação sugerem os agentes computacionais como uma maneira avançada de amplificar o intelecto humano. Segundo Engelbart (1962):

“Há então dois domínios separados de atividades dentro do sistema H-LAM/T: aquele representado pelo humano, no qual todos os processos explícitos humanos ocorrem; e aquele representado pelos artefatos, no qual todos os processos explícitos do artefato ocorrem. Em qualquer processo composto, há uma interação cooperativa entre esses dois domínios, que requer troca de energia (grande parte dela para o propósito único de troca de informações).”

Se o artefato de amplificação for capaz de realizar seus processos explícitos de maneira avançada, ou seja, inteligente, e tiver um poder maior de interação com o domínio dos processos explícitos humanos, então este artefato será mais poderoso na amplificação de inteligência. Assim são os agentes computacionais inteligentes.

5.4.1 Definição de agentes

O conceito de “agente” não é um conceito novo, pois sempre existiu para designar entidades com o poder de agir. De acordo com o dicionário Longman, agentes são “pessoas que agem por ou no lugar de outras, segundo autoridade por elas outorgada”. Neste contexto, agentes humanos realizam uma tarefa bem determinada, são especialistas naquilo que fazem, têm acesso a informações relevantes para sua tarefa, realizam o serviço em questão por um custo menor do que outros e frequentemente realizam um serviço que não poderia ser obtido em nenhum outro lugar. Agentes de seguro e de viagem são apenas dois exemplos dessa categoria.

Relativamente nova é a aplicação desse termo ao contexto de agentes artificiais. Neste novo contexto, eles podem ser:

- Agentes de hardware (robóticos): dispositivos mecânicos controlados por computador capazes de realizar tarefas de processamento, montagem e transporte, com finalidades industriais, de entretenimento e de pesquisa. Encontram-se no mundo real, o qual percebem através de seus sensores e modificam através de seus atuadores;
- Agentes de software: são programas de computador capazes de execução autônoma, tendo como ambiente o mundo dos computadores, o qual percebem e no qual atuam através de comandos de computador.

Não existe, porém, no contexto de agentes artificiais, um consenso na comunidade quanto a uma definição do que seja um agente. Stan Franklin e Art Graesser procuraram demonstrar em um trabalho (Franklin & Graesser, 1996) que as definições existentes eram muito abrangentes ou muito específicas para agentes, e propuseram uma definição formal de agentes autônomos que claramente distingue um agente de software de um outro programa qualquer.

Dessa maneira, Franklin e Graesser propõem uma definição em “estilo matemático”, isto é, uma definição que captura a essência do que é um agente, para definir a classe superior de agentes. A partir dessa definição, adicionando-se restrições, pode-se chegar a sub-classes de agentes. Segundo os autores:

“Um agente autônomo é um sistema que é parte de um ambiente, estando situado dentro dele, e sente e age sobre este ambiente, no tempo, de acordo com seus próprios propósitos, de modo a alterar o que sentirá no futuro”.

Dessa definição de agentes computacionais, pode-se começar a vislumbrar a sua adequação como ferramenta de amplificação de inteligência: se os agentes estão imersos em um ambiente, o qual podem sensoriar e atuar, isso significa que, segundo nossa visão de cognição situada e incorporada, essas ferramentas podem realizar semioses, ou seja, tarefas cognitivas. Neste sentido, faz-se possível aplicar agentes computacionais para realizar tarefas inteligentes em um processo juntamente com a mente humana, estabelecendo um diálogo que se aproveita das melhores qualidades de cada uma dessas mentes, enquanto outros subsistemas computacionais lidam com tarefas mecânicas ou aleatórias, liberando ainda mais a mente humana para se concentrar nas decisões e julgamentos não estruturados.

5.4.2 Agentes amplificando o intelecto humano

Na seção 5.3, o sistema H-LAM/T de Engelbart (1962) foi descrito com seus três processos típicos:

- processos explícitos humanos, nos quais as tarefas cognitivas são realizadas somente pelo agente humano;
- processos explícitos do artefato de amplificação, nos quais as tarefas cognitivas são realizadas pela ferramenta amplificação;
- processos compostos, nos quais há participação de ambos.

Um sistema computacional construído como um artefato de amplificação de inteligência obtém sucesso quando consegue **identificar os processos explícitos humanos em determinado fluxo cognitivo que são ineficientes e remodelá-los**. Remodelar estes processos explícitos humanos significa transformá-los em processos explícitos do artefato, ou seja, do sistema computacional, quando uma determinada tarefa cognitiva for estruturada, ou transformá-los em processos compostos, ou seja, tarefas cognitivas realizadas em conjunto por um agente humano e pelo sistema computacional, a fim de aproveitar o que cada um pode oferecer de melhor na resolução do problema, quando ela for não estruturada. Os processos compostos são necessários quando não se pode excluir o humano da resolução, geralmente porque a tarefa necessita de raciocínios inteligentes nos quais os humanos se sobressaem às máquinas.

Agentes computacionais são adequados para serem utilizados em **processos compostos**, pois seus atributos implementam as estratégias de amplificação de inteligência de controle do fluxo cognitivo e de realização de tarefas cognitivas. Esses atributos são:

- auto-gerenciabilidade;

- reatividade;
- autonomia;
- orientação à metas e pró-atividade;
- continuidade;
- comunicabilidade;
- aprendizagem;
- emocionabilidade.

Todos esses atributos tornam os agentes computacionais ferramentas de amplificação ativas e com vida própria, capazes de perceber quando devem iniciar a resolução de uma sub tarefa cognitiva juntamente com um agente humano, iniciar de maneira autônoma, resolvê-la com o agente humano através de diálogos com feedback negativo e chegar a uma solução final negociada entre ambos, provendo uma amplificação contínua nos processos compostos, participando de maneira autônoma e inteligente nos processos.

Os agentes computacionais implementam características essenciais para amplificação de inteligência:

- **cognição situada:** o agente deve ser capaz de estar presente no mesmo ambiente do agente humano, ambiente este que contém o problema em questão. Naturalmente, como são agentes computacionais, este ambiente deve ser um ambiente virtual, um meio pelo qual se represente o mundo real onde existe o problema. Fato é que estes agentes devem estar presentes neste meio, que por definição os “pluga” ao mundo real onde existe o problema, representando-o fielmente, e interagir com os agentes humanos que também irão atuar neste meio;
- **intencionalidade:** os agentes devem ser inteligentes, já que irão participar de um processo composto juntamente com agentes humanos. Isso significa uma necessidade de apresentar intencionalidade, que segundo Bratman (1987) se resume na capacidade de gerar planos de ação para resolver a atividade cognitiva, avaliar esses planos segundo objetivos, escolher o mais adequado e aplicá-lo;
- **comunicabilidade:** os agentes devem ser capazes de representar suas soluções, entender as soluções dos agentes humanos e dialogar com os mesmos para que juntos cheguem a uma solução mais adequada.

O agente será capaz de comunicar-se com o agente humano realizando semioses externas (seção 4.3.1) na interface de diálogo no qual estará inserido, manipulando seus signos para representar a sua proposta de solução.

A abordagem utilizando agentes também é adequada no endereçamento do problema de fundamentação do símbolo, porque a estratégia utilizando agentes para resolver os processos compostos delega aos seres humanos o papel de resolver esses problemas.

5.5 Engenharia de SAI

No capítulo 3 (seção 3.2.3), procurou-se dar uma definição inicial aos sistemas de amplificação de inteligência, sem compromisso com uma análise mais detalhada e precisa, simplesmente para que um paralelo pudesse ser traçado entre os SAI e os SSTD. Foi dito, naquela oportunidade, que SAI são capazes de lidar com as dificuldades apresentadas por problemas não estruturados em ambientes dinâmicos, quando construídos com uma filosofia correta de sistema cognitivo distribuído (Giere & Moffatt, 2003) e possuindo os atributos necessários. No entanto, que atributos necessários seriam estes? Que filosofia adequada de desenvolvimento daria a estes sistemas essas capacidades de cognição distribuída (Giere & Moffatt, 2003)?

Para garantir que a ferramenta de amplificação de inteligência a ser construída de fato amplifique a mente do agente humano usuário, enumera-se uma lista de atributos que precisam estar presentes nos SAI:

1. Interface de diálogo
2. Diálogo com agentes computacionais que possuem motores de solução
3. Fluxo cognitivo
4. Multi-objetividade
5. Simulação de diversos cenários “o que aconteceria se”
6. Responsividade¹⁰
7. Evolução

A seguir, serão discutidos os atributos necessários aos SAI, assim como uma proposta de componentização padrão para os mesmos. Essas características complementam a definição apresentada na seção 3.2.3.

5.5.1 Atributos de sistemas de amplificação de inteligência

Interface de diálogo

A interface de diálogo é a interface humano-computador dos sistemas de amplificação de inteligência. Esta interface deve possibilitar tanto aos agentes computacionais quanto aos agentes humanos construir e representarem uma solução do problema. Se fosse possível separar os atributos dos SAI e dar-lhes uma nota relacionada a sua relevância, a interface de diálogo seria certamente classificada como o atributo mais relevante. Essa interface é uma interface gráfica com o usuário.

Como já dito anteriormente na seção 5.1, um dos princípios da amplificação de inteligência é:

- Todo pensamento pode ser materialmente corporificado ou incorporado (exosomático)

Segundo Ransdell,

¹⁰Tradução destes autores para o termo original em inglês “responsiveness”

“O ponto de partida para entender a amplificação de inteligência é considerar a localização exosomática da mente no ambiente material” (Ransdell, 2003).

Este princípio está diretamente ligado à importância da interface de diálogo: a semiose (processo de interpretação dos signos, a cognição em si) precisa ser representada exosomaticamente. A interface de diálogo deve possuir uma capacidade de representação icônica, indexical e simbólica adequada a contribuir para a manipulação do conhecimento humano, ajudando-o na recuperação de informação do passado, compressão de informação, representação de uma informação futura, relacionamento entre informações passadas e futuras e formação de símbolos e conceitos para representar a informação. Mais do que isso, ele necessita ser uma poderosa mídia de representação na qual o pensamento reside, e poderosa na geração de interpretantes. Assim, não basta apenas que a interface de diálogo seja amigável ao usuário, mas precisa também fazer parte do processo cognitivo, o que significa que ela precisa ser capaz de representar as unidades de conhecimento (signos) corretamente e permitir que elas sejam manipuladas pelos agentes que atuam sobre ela. Os signos precisam ser capazes de residir na interface de diálogo, que implementa um tipo de linguagem que será utilizada entre agentes humanos e agentes computacionais para seu diálogo, em busca da solução de determinado problema de planejamento e execução dinâmica, uma **ontologia comum aos agentes**, com símbolos e atividades comuns. Quanto mais rico for a interface de diálogo para este propósito, mais amplificada será a mente dos agentes que o utilizarem. Esse poder está relacionado com a capacidade de gerar significado e cognição.

Como um exemplo simples, imagine um agente humano tentando somar dois números grandes na mente, uma tarefa a qual somos submetidos muitas vezes. Dependendo do tamanho do número, essa tarefa pode levar algum tempo. Se a mesma tarefa for realizada com a ajuda de uma simples interface de diálogo e ferramentas de amplificação, como um papel e um lápis, ele será na maioria das vezes realizado mais rapidamente, e talvez de maneira mais correta. O papel e a escrita são capazes de armazenar informações fora da mente: não é mais necessário processar todos os números porque eles já estão armazenados no papel, que nesse exemplo funciona como um tipo de memória externa. Pode-se, então, lidar com pequenos pedaços do número de cada vez. O papel faz parte do processo de cognição, armazenando símbolos externos. No entanto, pela simplicidade do exemplo, pode parecer que uma interface de diálogo não é tão relevante. Com efeito, o papel participa muito pouco no processo cognitivo. Uma interface de diálogo mais sofisticada deve ser capaz de representar conceitos complexos em formas icônicas e permitir que o agente os manipule movendo, recuperando e modificando-os de acordo com regras complexas, com rápidas respostas. Um exemplo avançado foi dado na seção 5.2.1.

Através da interface de diálogo, os agentes humanos irão incorporar suas intuições na construção da solução proposta para o problema. Dessa forma, a interface de diálogo pode também ser enxergada como uma espécie de interface entre os modelos computacionais e o mundo real: ela serve para “plugar” os agentes computacionais no mundo, pois é através dela que os agentes humanos inserem e atualizam informações dos referentes do mundo real representados como signos, resolvendo o problema de fundamentação do símbolo para que os agentes computacionais encontrem uma solução melhor do que a que encontrariam sozinhos, sem esta referência. A interface de diálogo deve ser alimentada com informações do mundo real de tal maneira que atuar nela e sentir através dela seja como fazê-los no mundo real.

Portanto, a interface de diálogo precisa apresentar os seguintes atributos para sua realização:

1. Representação icônica dos elementos mais relevantes: os conceitos mais relevantes no domínio e na construção da solução devem ser representados iconicamente, para que o agente humano perceba rapidamente seus estados no modo de operação e seja capaz de manipulá-los facilmente;
2. Visualização da situação do domínio representado e do planejamento das soluções no passado, presente e futuro em uma linha temporal: além de representar a solução em um determinado instante, a interface de diálogo deve possibilitar a visualização de planos realizados no passado e de planos de execução para horizontes de tempos futuros;
3. Mecanismos de interação do agente humano com o planejamento: o agente humano deve ser capaz de interagir com a interface de diálogo, inserindo dados manualmente e participando da construção da solução, contribuindo com suas intuições. A interface de diálogo deve ser uma ontologia comum a esses agentes, para fins de comunicação e diálogo.

É importante deixar claro, no entanto, que a melhor interface de diálogo depende do problema atacado. A melhor interface de diálogo não é aquela que representa todo o ambiente de trabalho. Se assim fosse, um controle remoto de televisão teria centenas de teclas. A melhor interface de diálogo é aquela capaz de representar todos os signos no processo de busca pela solução com a menor oposição ao fluxo de dados e informações possível. Ela deve filtrar informações desnecessárias sem perder o poder de comunicar precisamente a informação relevante para os agentes.

Comparando este conceito de interface com o conceito da abordagem da Engenharia Semiótica (de Souza, 2005), verifica-se que a visão de construção da Engenharia Semiótica apresenta um foco diferente da interface de diálogo em alguns aspectos. Conforme dito na seção 4.3, na Engenharia Semiótica, a interface é um artefato de metacomunicação, ou seja, comunica ao usuário aspectos da comunicação, e ao mesmo tempo leva mensagens diretas do projetista para o usuário.

Por outro lado, a questão fundamental da interface de diálogo é ser capaz de **exosomatizar** (externalizar) os processos de semiose (raciocínio) do agente humano (usuário), uma questão de representação, e ao mesmo tempo **ser uma ontologia** (no contexto de sistemas multi-agentes) comum aos agentes humano e computacional, para que possa haver um **diálogo** entre esses agentes, a fim de que estes possam construir uma solução para o problema não estruturado em conjunto. No entanto, como as técnicas construídas com base na Engenharia Semiótica visam especificar formalmente a interface e buscam uma melhor usabilidade, como a realizada por Leite & de Souza (1999), pode-se utilizá-las na especificação e protótipo da interface de diálogo, desde que sejam endereçados os aspectos relevantes deste tipo de interface de diálogo entre agentes.

Grande é o desafio do entendimento de como deve ser, para um problema específico, a correta interface de diálogo. Esse trabalho de projeto é muitas vezes mais complicado do que a própria implementação do sistema. Muitos fracassos de SAI podem ser causados devido à uma interface de diálogo inadequada ou deficiente. Isso se dá pelo fato da interface de diálogo ser a interface, o ponto de encontro, a fronteira entre o pensamento humano e o pensamento exosomaticamente representado. Conforme será visto na seção 6.3, a metodologia proposta neste trabalho não é capaz de prever como deve ser exatamente a interface de diálogo de um SAI. No entanto, ela será capaz de prever todos os conceitos que devem ser representados nessa interface. A forma como cada um desses conceitos deve ser representado é um trabalho que exige uma certa dose de arte, criatividade e engenhosidade. No

entanto, se for possível garantir que os conceitos necessários estão representados, torna-se o diálogo entre agente humano e computacional em busca da solução factível.

Diálogo com agentes computacionais que possuem motores de solução

A classe de problemas de otimização atacados pelos SAI representa problemas dinâmicos do mundo real. As entradas mudam constantemente à medida que o mundo muda, e desse fato advém a necessidade do diálogo entre agentes humanos e agentes computacionais, devido a capacidade única dos seres humanos de manter as informações do mundo atualizadas em sua mente, conforme discutido no problema do enquadramento, na seção 5.3.

Imagine um robô, que representa um agente planejador, possuindo uma representação interna do mundo. Esse robô tem como tarefa navegar dentro de uma sala cheia de objetos. Para tal, possui uma representação da sala internamente. Seus sensores indicam a situação atual do robô, e baseando-se no seu modelo interno, ele toma a decisão de seguir em determinada direção ao invés de uma outra, na qual bateria em um objeto, criando um plano de ação para a navegação. Agora, imagine que este robô esteja inserido em um ambiente que muda constantemente: ele seria obrigado a atualizar sua representação de mundo a todo instante. Imagine agora que este agente esteja inserido em um ambiente dinâmico o suficiente para que seja impossível captar todas as informações necessárias para atualizar sua representação: este é o ambiente de escopo dos SAI. Algumas informações só são conhecidas por agente humanos, e estes agentes só serão capazes de fornecê-las aos agentes computacionais se o diálogo entre eles estiver presente na filosofia de construção do sistema.

Como dito na seção 5.1, um dos princípios da amplificação de inteligência é:

- Todo pensamento ocorre em forma de diálogo

Dessa maneira, o correto processo de busca da solução no caso de sistemas desse tipo é um processo interativo, dando origem a um diálogo entre agentes humanos e computacionais, de maneira que o agente humano possa criticar a solução encontrada pelo(s) agente(s) computacional(ais) e vice-versa. Se o diálogo puder acontecer em tempo adequado, considerando aplicações em tempo real e as necessidades de velocidade de resposta, a sequência de feedbacks negativos dados pelo agente humano e pelo agente computacional levarão o sistema a uma solução melhor do que cada um encontraria por si só. Esse “tempo adequado”, no qual os agentes precisam retornar uma resposta para que o diálogo possa acontecer, varia entre as diversas aplicações. Uma ferramenta em tempo real precisaria da resposta em poucos segundos. Para uma ferramenta de planejamento tático, provavelmente alguns poucos minutos para completar uma sugestão de solução seria um tempo satisfatório. No entanto, se o sistema demorasse um dia inteiro para tal, com certeza seria impossível estabelecer o diálogo. Por isso torna-se importante o desenvolvimento dos motores de solução que encontrarão a solução. Pode ser que poucos minutos não sejam suficientes para retornar uma solução ótima, considerando o problema globalmente: não há problema. É melhor considerar a possibilidade de encontrar uma boa solução em pouco tempo, que pode ser complementada por outros agentes, do que gastar um tempo muito grande procurando pelo ótimo, que muitas vezes pode representar um problema computacionalmente intratável.

Para que o diálogo entre agentes computacionais e agentes humanos possa contribuir na construção de uma solução, faz-se necessário que os agentes humanos possam inserir decisões mandatórias no sistema. A criação da solução deverá ser um processo interativo no qual o objetivo é ter o agente

humano (planejador) trabalhando em equipe com o agente computacional para que cada um contribua com o melhor de si. A linguagem utilizada no “diálogo” da construção do planejamento é a interface de diálogo. Assim sendo, o agente humano “conversa” com o agente computacional através de atuações na interface de diálogo que alteram a solução sendo construída. As alterações realizadas pelo usuário podem ser consideradas como programações mandatórias que devem ser levadas em consideração pelos algoritmos durante o processo de otimização.

A figura 5.5 ilustra o papel de uma programação mandatória. Cada operação da solução fornecida pelos agentes computacionais transforma o estado da busca. Uma sequência de operações transforma o estado inicial da busca em um estado final. Para dialogar com o agente computacional, o agente humano pode interromper a sequência de operações e incluir manualmente uma nova operação ou modificar a operação existente. Os agentes computacionais irão recalcular a partir deste ponto a solução, ou seja, a sequência de operações que levam ao estado final. Na figura 5.5, o estado inicial é E1 e o estado final é o E6. Neste exemplo, o agente humano realiza uma operação mandatória sobre o estado E4, gerando o estado E4', sobre o qual o sistema recria a solução, chegando ao estado E6'. A princípio, todas as operações realizadas pelos algoritmos de otimização podem também ser realizadas pelo usuário como operação mandatória. Ele deve poder tanto criar uma nova operação quanto editar uma operação existente.

Dessa forma, SAI representam uma dualidade, fornecendo ao mesmo tempo uma interface de diálogo adequada ao agente humano para que construa e represente sua solução e sendo equipado com algoritmos de otimização e técnicas de inteligência computacional capazes de sugerir uma solução própria, para ser negociada com outros agentes. Por isso, os SAI podem ser vistos como sistemas multi-agentes, integrando mentes humanas e mentes computacionais.

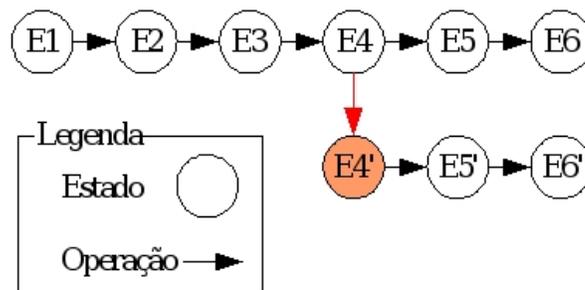


Fig. 5.5: Inserção de uma operação mandatória representando o diálogo entre agentes humanos e computacionais

Fluxo cognitivo

O fluxo cognitivo de um determinado agente é o seu processo de pensamento, cognição ou raciocínio. Se o sistema tem o objetivo de amplificar a mente do usuário na resolução de um determinado problema, ele deve ser capaz de cooperar com o fluxo cognitivo desse agente. Amplificar a inteligência é, em parte, ser capaz de desenvolver um controle crítico sobre o diálogo entre os agente computacionais e humanos de acordo com um protocolo de comunicação conhecido, eficiente e efetivo, com o objetivo de encontrar, ao fim do diálogo, uma solução melhor do que a que cada agente encontraria separadamente. Portanto, não basta ter uma linguagem e um protocolo de comunicação

adequados (interface de diálogo) e eficientes para facilitar a representação por parte de cada agente de suas unidades de conhecimento, mas faz-se necessário também estabelecer um controle crítico sobre essa comunicação, uma espécie de moderação, como um moderador faz em um debate. O objetivo do sistema de amplificação de inteligência é participar do processo de semiose, não só permitindo que ele flua através de um meio de expressão adequado, mas também conduzindo-o através de intervenções que realizam um “controle crítico” do pensamento.

Esse controle crítico pode acontecer basicamente de três maneiras diferentes:

1. Interrupção: o sistema interrompe a comunicação com a interface de diálogo ou insere signos que interrompem a tarefa cognitiva do agente para guiá-lo para uma direção diferente da que ele está tomando no momento. Exemplos são janelas abrindo, ícones piscando e sons de alarme.
2. Estímulo: o sistema apresenta signos que cooperam com a atividade cognitiva do agente. Não apenas o significado mas também o tempo em que a informação é apresentada deve ser correto. Por exemplo, alguns editores de textos tentam completar palavras à medida que o usuário as digita.
3. Não-interferência: o sistema, apesar de possuir informações importantes para apresentar, atrasa a interrupção do agente esperando o momento correto, de modo que a atividade cognitiva não seja interrompida. Esse atributo está relacionado ao fato do SAI tratar decisões seqüenciais e independentes ao mesmo tempo, com acesso a uma grande variedade de fontes, formatos e tipos de dados. Há uma necessidade de conectar essas decisões e os dados de uma maneira correta, além de relacioná-los ao fluxo cognitivo na busca pela solução de uma tarefa cognitiva específica, no sentido de amplificar a mente e não interrompê-la durante semioses importantes.

Os SAI tratam decisões seqüenciais ou interdependentes, além de proverem acesso a uma grande variedade de fontes de dados, formatos, e tipos de dados. Neste caso, essas decisões seqüenciais precisam ser corretamente conectadas e os dados fornecidos em tempo correto e da maneira correta, com relação ao fluxo cognitivo na busca da solução em determinada tarefa, no sentido de amplificar a cadeia de semiose dos agentes, ao invés de interrompê-la.

Multi-objetividade

Os problemas atacados pelos SAI, além de serem altamente dinâmicos, são também multi-objetivo, com muitos desses objetivos sendo conflitantes. Há diferentes maneiras de tratar a multi-objetividade em problemas do mundo real:

1. Considerar um único objetivo escolhido entre todos para resolver o problema, um que seja mais relevante naquele momento;
2. Ponderar os objetivos, atribuindo pesos maiores aos considerados mais importantes;
3. Popular a fronteira de Pareto, encontrando as soluções não dominadas e apresentando-as aos agentes humanos através da interface de diálogo para que, ao visualizá-las, eles sejam capazes de escolher a que mais se adequa à situação vivida naquele instante.

Popular a fronteira de Pareto e deixar a cargo do usuário que escolha uma solução é uma maneira interessante de resolver problemas multi-objetivo. Ao invés do usuário escolher a priori os pesos dos objetivos ou escolher entre os objetivos aqueles que serão de fato considerados na busca da solução, pode-se encontrar todas as soluções não dominadas na fronteira de Pareto e apresentá-las ao usuário, para que ele escolha, com a visualização da interface de diálogo, qual a solução que mais lhe serve naquela determinada circunstância. Dessa maneira, pode-se maximizar a amplificação de inteligência na busca pela solução, pois de outra maneira, seria necessário um conhecimento a priori muito grande (que quase nunca o usuário possui) para escolher apenas alguns objetivos frente a uma situação ou para atribuir pesos adequados aos objetivos. Ainda mais, os problemas possuem uma explosão combinatória muito grande, sendo quase impossível para o usuário, ainda que ele possua conhecimentos especialistas a priori, prever o que acontecerá com a solução ao alterar pesos dos objetivos. Dessa forma, é sempre melhor apresentar todas as soluções não dominadas e permitir que o usuário escolha em tempo real a que melhor resolve seus propósitos, de acordo com o contexto naquele instante.

Para ilustrar melhor o que é a fronteira de Pareto e soluções Pareto-ótimas, considere que uma solução x_1 é dita dominar uma solução x_2 se ambas as condições forem satisfeitas:

1. A solução x_1 não é pior que a solução x_2 com relação a nenhum dos objetivos m , ou seja,

$$f_m(x_1) \leq f_m(x_2), \forall m \in \{1, \dots, M\}. \quad (5.1)$$

2. A solução x_1 é estritamente melhor que a solução x_2 em pelo menos um objetivo m , ou seja,

$$f_m(x_1) < f_m(x_2), m \in \{1, \dots, M\}. \quad (5.2)$$

Uma solução Pareto-ótima é uma solução não-dominada por nenhuma outra solução factível.

O conjunto Pareto-ótimo é formado por todas as soluções não-dominadas dentre as factíveis, ou seja, por todas as soluções Pareto-ótimas.

A fronteira de Pareto é formada pelos pontos no espaço das funções-objetivo que correspondem ao conjunto Pareto-ótimo.

Vale ressaltar que, não havendo relevância entre os objetivos a serem atendidos, todos os pontos na fronteira de Pareto são qualitativamente equivalentes, sob a perspectiva de otimização. Isso implica que problemas multi-objetivo desse tipo apresentam potencialmente infinitas soluções equivalentes. Assim, o algoritmo de otimização deve ser capaz de popular a fronteira de Pareto otimamente com recursos finitos de computação, ou seja, obter soluções não-dominadas que se distribuam uniformemente sobre a fronteira de Pareto. Essas soluções são apresentadas ao usuário, que opta pela que lhe satisfaz no contexto de determinado instante, já que todas resolvem o problema da melhor maneira possível relativamente aos multi-objetivos. Heurísticas populacionais são consideradas mais adequadas para este tipo de situação, pois faz-se necessário primeiro convergir para a fronteira de Pareto, e depois manter soluções uniformemente distribuídas nela.

É interessante que o SAI apresente essas três possibilidades de tratar a multi-objetividade, principalmente se o estudo do problema não for capaz de determinar qual das três formas é a mais adequada. Dessa maneira, o agente humano pode decidir empiricamente qual a maneira que lhe fornece melhores resultados de operação.

No entanto, deve haver um compromisso entre a possibilidade de popular a fronteira de Pareto e o tempo de processamento de tantas soluções diferentes, já que muitas vezes estamos lidando com aplicações em tempo real. Algumas vezes, pode ser mais interessante não considerar esta maneira de tratar a multi-objetividade na construção de um SAI devido à restrições de tempo.

Simulação de diversos cenários “o que aconteceria se”

Para garantir que o usuário tenha diferentes intuições nas mais variadas situações possíveis de solução, o SAI deve ser capaz de simular diferentes cenários à medida que o usuário critica as soluções fornecidas pelo agente computacional em um diálogo. O sistema deve ser capaz de simular situações à medida que o usuário insere informações mandatórias.

Esse atributo é especialmente importante para sistemas de amplificação de inteligência que planejam estados futuros. Os agentes humanos têm, assim, a possibilidade de saber o que poderá acontecer com a solução dentro de alguns dias ou meses baseado numa decisão tomada neste instante, e assim torna-se possível prever problemas futuros.

Mais uma vez, precisa-se considerar restrições temporais relacionadas à aplicações em tempo real para este atributo.

Responsividade

Este atributo é apresentado por Partridge (1998) como um requisito para softwares mais poderosos no futuro. Há muitos graus e dimensões de responsividade a usuários humanos. Quando um cliente acessa sua conta no internet banking, o sistema computacional trata essa pessoa como um indivíduo: muitos desses sistemas identificam o cliente pelo seu primeiro nome e fornecem uma série de configurações personalizadas para cores e fundos de tela. O cliente possui um login e um password para as requisições, que são realizadas pelo sistema. O sistema assim, responde a cada cliente, mas de uma maneira diferente da que faz um caixa humano em uma agência bancária. O sistema computacional trata todos da mesma maneira, um tipo de ser humano médio. Um caixa humano, dotado de inteligência, é capaz de responder a individualidades diferentes, o que é muitas vezes importante para o controle do fluxo cognitivo. Há muitas diferenças que podem ser identificadas. Pessoas idosas requerem um tratamento diferenciado quando se lida com tecnologia. Idade do cliente é apenas um simples exemplo.

Um SAI deve ser capaz de construir e manter um modelo de cada usuário. O tratamento é diferenciado à medida que o usuário se torna mais “familiar” ao sistema, já que este vai aprendendo suas peculiaridades. Perceba que este atributo está fortemente relacionado com o atributo de fluxo cognitivo. Um usuário menos experiente precisará de mais ajuda, portanto mais estímulos do sistema. Um usuário mais experiente se aborrecerá com muitas interrupções.

Engelbart relaciona responsividade a sistemas de amplificação (Engelbart, 1962):

“Poderes básicos da cognição humana, como memória, inteligência ou reconhecimento de padrões podem ter um efeito de combinação grandioso. Os meios de amplificação empregados atualmente têm evoluído ao redor de grandes populações, e nenhuma tentativa tem sido feita de adaptá-los para necessidades e habilidades individuais. Cada indivíduo tende a evoluir suas próprias variações, mas não há atividades de seleção e mutação suficientes, nem há suficiente feedback de seleção, para permitir mudanças significativas.

Um sistema H-LAM/T bom e automatizado deve fornecer a oportunidade de uma adaptação significativa dos meios de amplificação para características individuais. O efeito composto dos poderes cognitivos fundamentais do ser humano sugere que sistemas projetados para efetividade máxima requerem que esses poderes sejam desenvolvidos ao máximo - através de treinamento, truques mentais especiais, linguagem especial e novas metodologias.”

Evolução

Os problemas do mundo real atacados pelos SAI têm a característica de estarem inseridos em um complexo ambiente cheio de detalhes dinâmicos. Mas não apenas os detalhes: a essência do problema é dinâmica, o que significa que o problema pode mudar com o tempo. Assim, um sistema construído hoje, para resolver o problema em questão, pode se tornar inadequado com o passar do tempo. Os SAI devem se preocupar com a adaptação de todos os atributos apresentados anteriormente a essas mudanças em função do tempo.

Em alguns casos, o usuário pode realizar as mudanças necessárias apenas modificando alguns atributos do sistema. Em outros casos, há uma necessidade de modificar a estrutura dele. A idéia é implementar maneiras do sistema monitorar sua própria performance, gerando relatórios e até mesmo sugerindo mudanças. por exemplo, se as soluções propostas pelos agentes computacionais eram antes modificadas em 10% e agora são modificadas em 30% dos casos, pode significar que uma melhoria nos algoritmos se faz necessária.

O mecanismo de evolução pode existir também fora do sistema, através de políticas de operação, suporte e evolução bem definidas.

5.5.2 Análise dos atributos de SAI

Para validar os atributos desenvolvidos para os SAI, deve-se realizar uma análise para verificar se os mesmos são compatíveis com as necessidades apresentadas por Aitken et al. (2007) para resolução de problemas de planejamento e execução dinâmicos. Por conveniência, repete-se a seguir essas necessidades e seu mapeamento nos atributos dos SAI:

1. Execução e planejamento desacoplados: ao invés de realizar o planejamento e a execução seqüencialmente, deve-se realizá-los de maneira desacoplada, ou seja, como dois processos paralelos. Dessa maneira, o processo de planejamento apenas se preocupa em receber informações atualizadas do mundo real e ajustar os planos daquele instante em diante. Assim, mudanças e eventos inesperados possuem impactos mínimos na execução, que apenas recebe sempre os planos mais atualizados e procura segui-los.

Isso ocorre, com efeito, em SAI: o usuário é um agente humano totalmente dissociado da execução do problema o qual ele resolve (planeja). O sistema recebe informações atualizadas do mundo real através da interface com sistemas externos e do próprio usuário humano, que tem a capacidade de atualizar informação dos signos representados na interface de diálogo com relação aos seus referentes no mundo real. Dessa maneira, o agente humano, juntamente com os agentes computacionais, manipula as informações sempre atualizadas na interface de diálogo de maneira rápida e ágil. Garantindo que os agentes computacionais entreguem as respostas

do diálogo em tempo hábil (este tempo varia de problema para problema), o planejamento conseguirá se manter dentro do ciclo da dinamicidade. Como o plano está em meio digital, fica fácil para os executores acessarem os planos atualizados, mesmo que estejam distribuídos geograficamente.

2. Subjetividade no planejamento: segundo Aitken et al. (2007), os planos devem ser gerados e interpretados de alguma maneira por humanos, que são determinados por suas características e personalidade, como conhecimento, treinamento, experiência, confiança, moral, agressividade e atitude diante de riscos, características que não podem ser implementadas em uma máquina.

Com a utilização de SAI, o agente humano constrói o plano, realizando a atividade inteligente juntamente com o sistema e tendo a oportunidade de inserir suas intuições e utilizar-se de suas características e personalidade para construir a solução, diferentemente de sistemas convencionais de otimização.

3. Linha temporal: segundo os autores, representações estáticas de planos não são fáceis de atualizar e costumam gastar mais tempo para tal do que está disponível.

Nos SAI, a representação dos planos é dinâmica e muito especial; a interface de diálogo possui estruturas representando signos com referentes no mundo real, que podem ser manipulados pelo usuário rapidamente para mudar os planos, construindo planos novos e mais adequados às atualizações das informações. Além disso, a interface de diálogo possui uma representação cronológica capaz de recuperar estados passados e visualizar planos futuros com estas representações.

4. Replanejamento: segundo os autores, a cada nível hierárquico ao qual o plano é submetido, modificações são realizadas devido a mudanças no ambiente antes da execução. Representações estáticas dos planos não contêm nada da lógica, interdependência e razão que existem nas considerações de alteração de um plano.

No caso dos SAI, a interface de diálogo preserva todos estes aspectos, devido a sua representação cronológica dos planos e do que foi realizado, além da possibilidade de inserir observações e anotações relevantes, associadas aos signos representados.

5. Coordenação: mudanças em planos realizadas por subordinados devem ser feitas em concordância com outros subordinados.

Um SAI, por ser um sistema computacional, facilita a comunicação de toda a equipe responsável pela montagem do plano. Basta que o sistema seja implementado em um paradigma cliente-servidor e que existam terminais nos diversos pontos responsáveis, para que as informações sejam compartilhadas e organizadas.

6. Representações semânticas no plano: segundo os autores, para que um plano possa ser mudado dinamicamente de maneira efetiva e eficiente, o planejador deve entender a lógica e o significado semântico do plano anterior, além de ser capaz de rapidamente gerenciar interdependências afetadas pelas mudanças. Para tal, faz-se necessário uma representação dinâmica e alguma forma de suporte automatizado do planejamento. A representação, segundo os autores, deve ser capaz de facilitar as trocas de informação humano-humano, humano-computador e

computador-computador, e que crítico para tal é a habilidade de capturar e representar o contexto e considerações acerca dele.

Nos SAI, um subsistema de agentes computacionais com algoritmos de otimização e inteligência computacional é desenvolvido para dar suporte ao humano na realização de atividades inteligentes, como formulações de planos de ação. Um subsistema de cálculos automáticos procura retirar do humano a carga de cálculos que podem ser feitos somente pela máquina, sem a necessidade de habilidades humanas. A representação dinâmica é realizada pela interface de diálogo, conforme dito anteriormente. Os esquemas de controle de fluxo cognitivo implementados procuram facilitar as trocas de informação entre os diversos agentes, humanos e computacionais.

Os SAI abordam todos os aspectos apresentados como relevantes para os problemas de P&ED.

A respeito de ambientes dinâmicos, pode-se ainda fazer algumas outras considerações com relação às suas características e as implicações dessas características nos SAI, como na tabela 5.1:

Característica Ambiental	Implicações nos SAI
Ambiente dinâmico. Muitas mudanças e informações confirmadas de última hora.	O tempo para se criar uma solução, não necessariamente o tempo de resposta dos agentes computacionais, deve ser pelo menos duas a três vezes menor do que o tempo médio entre as modificações do cenário operacional.
Falta de atualização em tempo real das informações: toda ou boa parte da informação é atualizada manualmente no sistema. Assim, nem sempre as informações retiradas do sistema representam a verdade, pois podem ainda não ter sido atualizadas.	O sistema deve ser capaz de tratar incertezas e imprecisões das informações de entrada e restrições.
Necessidade de simular vários cenários de discussão, para verificar o efeito que diferentes decisões a médio e longo prazo.	O tempo de resposta dos agentes computacionais deve ser pequeno o suficiente para permitir análise de vários cenários. Além disto, o sistema deve permitir a gestão de vários cenários.
Informações de muitas fontes diferentes não agrupadas e não relacionadas	O sistema deve consolidar em um único local todas as informações relevantes para a tomada de decisão

Tab. 5.1: Implicações nos SAI em função das características dos ambientes dinâmicos

5.5.3 Componentização padrão de um SAI

Para complementar a caracterização dos SAI, propõe-se uma estrutura padrão para a componentização dos SAI. Essa componentização é baseada na sugerida por Turban e Aronson para os SSTD (Turban & Aronson, 1997), apresentada na figura 3.2, e leva em consideração os atributos necessários a um SAI, conforme apresentados na seção 5.5.1.

Os componentes genéricos de um sistema de amplificação de inteligência, assim como suas possíveis interfaces, estão identificados na figura 5.6.

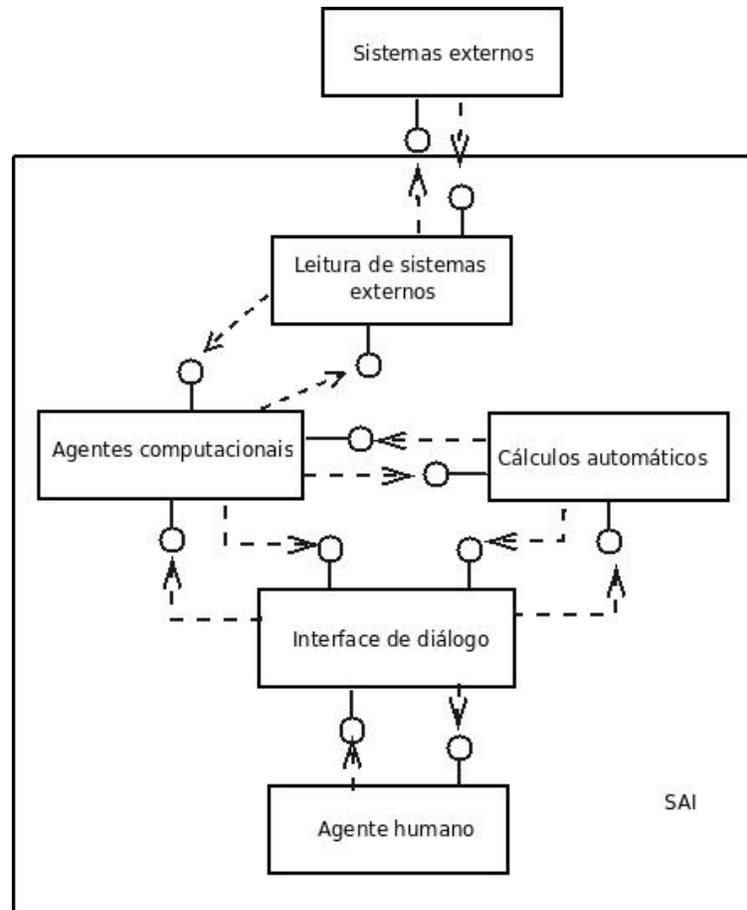


Fig. 5.6: Componentes genéricos de um SAI

Esses componentes têm o seguinte significado e propósito:

- **Leitura de sistemas externos:** este subsistema é responsável por ler todas as informações necessárias, advindas de sistemas externos, para daí serem distribuídas para os outros subsistemas do sistema.
- **Cálculos automáticos:** este subsistema será responsável por realizar cálculos computacionais que podem ser realizados de maneira transparente ao usuário, somente pelo computador, sem precisar da atuação de agentes humanos. Apesar de utilizar apenas cálculos matemáticos simples, e não incluir nenhum algoritmo de otimização, este subsistema auxilia bastante o agente

humano nas tomadas de decisão, principalmente liberando tempo para que ele realize outras tarefas e tornando praticamente inexistentes eventuais erros de cálculo.

- Agentes computacionais: este subsistema implementa agentes computacionais carregando motores de solução capazes de encontrar uma solução para o problema em questão em tempo adequado para que haja um diálogo com os agentes humanos. O tempo adequado varia de acordo com o problema em questão. Os agentes computacionais irão atuar somente nas atividades que não podem ser calculadas automaticamente, para que haja um diálogo com os agentes humanos.
- Interface de diálogo: este subsistema será a implementação da linguagem de representação dos signos necessários para formulação do domínio e da solução do problema, tanto por parte dos agentes humanos como por parte dos agentes computacionais. É por meio dessa linguagem que ambos conversarão e negociarão uma solução para o problema. É através da interface de diálogo que o agente humano se comunica com o sistema e controla algumas operações. A interface de diálogo possui representações na forma de ícones, índices e símbolos, visualizações de estados passados, presentes e futuros no planejamento, mecanismos de entrada de dados e mecanismos de interação com o usuário.
- Agente humano: o agente humano é considerado como parte do sistema, pois participa da construção da solução, contribuindo com informações valiosas sobre os referentes do mundo real dos signos representados no sistema e incorporando suas intuições na construção da solução.

5.6 Diagrama de fluxo cognitivo

Pensar na concepção de um sistema de amplificação de inteligência significa desejar amplificar a mente de agentes humanos que resolvem um determinado problema de planejamento e execução dinâmica com certa dificuldade. Isso implica que existe um fluxo cognitivo desses agentes humanos para resolver o problema, e se o intuito é desenvolver um sistema que amplifique esse repertório hierárquico (seção 5.3) de atividades cognitivas, este fluxo cognitivo não pode ser simplesmente desprezado. Com efeito, ele deve ser muito bem estudado e modelado, para que o sistema de amplificação possa se encaixar no processo sem causar oposição à aceitação de sua utilização, retirando do agente humano tarefas que possam ser realizadas automaticamente e compartilhando com ele tarefas nas quais o agente humano precisa estar presente. Faz-se necessário, portanto, uma maneira de modelar o fluxo cognitivo desses agentes, para que as tarefas de seu repertório hierárquico possam ser identificadas e exploradas.

Segundo Engelbart (1962):

“O arcabouço conceitual que buscamos deve nos orientar na direção das reais possibilidades e problemas associados com o uso da tecnologia moderna, para fornecer ajuda direta para o indivíduo na compreensão de situações complexas, isolando fatores significativos, e resolvendo problemas. Para ganhar esta orientação, nós examinamos como indivíduos alcançam seu presente nível de efetividade, e esperamos que esse exame revele possibilidades de melhoria.”

Com relação a essa modelagem do repertório hierárquico dos agentes humanos, Engelbart (1962) coloca o seguinte:

“Aumentar a efetividade do uso das capacidades básicas de um indivíduo é um problema de redesenhar partes que podem ser alteradas em um sistema. O sistema está ativamente comprometido nos processos contínuos de desenvolver compreensão (além de outros) no indivíduo e de resolver os problemas; ambos os processos estão sujeitos à motivação, propósito e desejo humano. Redesenhar as capacidades do sistema de realizar esses processos significa redesenhar todo ou parte do repertório hierárquico. Para redesenhar esta estrutura, temos que aprender o máximo possível sobre os materiais e componentes básicos à medida que são utilizados dentro da estrutura; além disso, nós temos que aprender como ver, medir, analisar e avaliar em termos do todo funcional e seus propósitos.”

“Para nossos objetivos de derivar orientação sobre as possibilidades de buscar ativamente um aumento da efetividade intelectual humana, é importante perceber que devemos estar preparados para buscar novas possibilidades ao longo de toda a cadeia de hierarquia de capacidades (o que pede uma abordagem sistemática). É também importante perceber que devemos estar orientados para a síntese de novas capacidades a partir da reorganização de outras capacidades, antigas e novas, que existem ao longo da hierarquia (o que pede uma abordagem de “engenharia de sistemas”).”

“Não se faz necessário adicionar ou modificar somente as capacidades de processo muito sofisticadas ou formais neste redesenho. Essencialmente, qualquer um dos processos utilizados por um humano hoje - os processos que ele pensa a respeito quando olha para frente e imagina seu dia de trabalho - são processos compostos do tipo que envolve composição externa e manipulação de símbolos (textos, rascunhos, diagramas, listas, etc.). Muitos dos processos de composição e manipulação externa (modificação, re-organização) servem a atividades humanas tais quais brincar com formas e com relacionamentos para desenvolver uma idéia ou listar itens para refletir a respeito e depois reorganizá-los e estendê-los à medida que os pensamentos se desenvolvem.”

Para que seja possível modelar e estudar o fluxo cognitivo dos agentes humanos que resolvem um determinado problema não estruturado, a fim de amplificar sua mente na resolução deste problema, este trabalho propõe o diagrama de fluxo cognitivo (DFC), uma ferramenta destinada a modelar o fluxo cognitivo humano na resolução de problemas de planejamento e execução dinâmicos, utilizando conceitos da semiótica de Peirce na análise de como a tomada de decisão é realizada em termos de processos cognitivos.

Os objetivos do DFC são:

1. Entender e modelar a correta sequência de atividades cognitivas;
2. Encontrar entre as atividades cognitivas representadas aquelas que devem ser realizadas exclusivamente pelo SAI e aquelas que devem ser realizadas através de um diálogo entre o agente humano e os agentes computacionais. Essas atividades serão marcadas e denominadas de “pontos de amplificação de inteligência”.

O conceito de signo, como definido por Peirce, é a base teórica das representações do DFC, pois as unidades básicas de conhecimento nele representadas são signos peirceanos.

A representação do DFC é um diagrama de atividades UML estereotipado, onde os objetos estão restritos a unidades de conhecimento ou signos, e as atividades estão restritas a atividades cognitivas ou tomadas de decisão. Na linguagem UML, estereótipo é um recurso para estender o vocabulário, permitindo a criação de novos tipos de elementos derivados de elementos existentes para um problema específico, com propriedades, semântica e notação próprias (Jacobson et al., 1999a).

Na figura 5.7, estão detalhados os principais componentes do DFC.

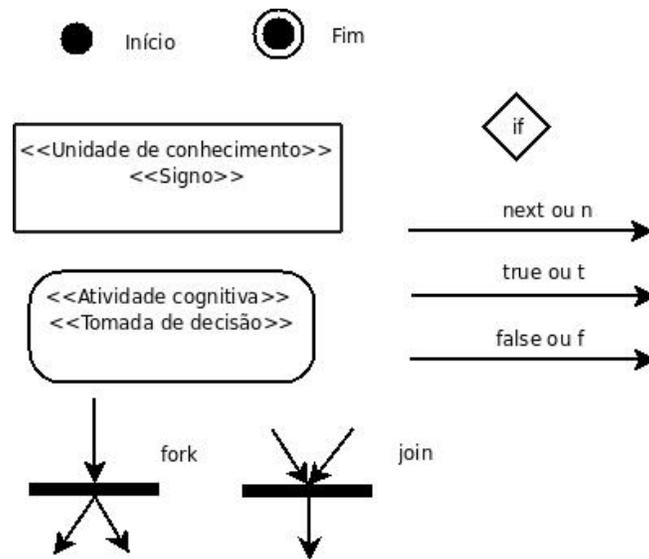


Fig. 5.7: Componentes do diagrama de fluxo cognitivo (DFC)

Poderia-se utilizar outros diagramas UML para representar o DFC, como por exemplo os diagramas de interação (diagramas de comunicação e de sequência). No entanto, para descrever uma sequência de atividades cognitivas, os diagramas de atividades são mais adequados, pelos seguintes fatores:

- Além de suportar comportamento condicional, esses diagramas são muito adequados para descrever comportamentos que possuem muitos processamentos paralelos, o que os torna diferentes de fluxogramas, que são limitados apenas a processos sequenciais. Isso é importante para modelar sequências de atividades cognitivas, pois elas muitas vezes possuem processos sequenciais desnecessários, que podem ser remodelados em processos paralelos para aumentar sua eficiência. Da mesma maneira que o diagrama de atividades é capaz de representar as *threads* em um programa e quando elas precisam ser sincronizadas, o DFC é capaz de representar as atividades cognitivas paralelas e quando elas se encontram;
- O foco do DFC não está em como as unidades de conhecimento (signos) se comunicam (colaboram), mas sim em como elas se comportam entre as várias *threads* de atividades cognitivas;
- O que se necessita representar não é a ordem cronológica de mensagens ou a organização dos objetos envolvidos na interação, mas sim uma sequência de atividades;

- O entendimento do diagrama de atividades é mais claro quando faz-se necessário representar realimentações.

Utilizando-se dos componentes do DFC, deve-se modelar a sequência de atividades cognitivas e as unidades de conhecimento utilizadas e geradas por essas atividades, desde o início do processo até a solução.

O conjunto de atividades cognitivas realizadas pelos agentes humanos na resolução do problema de planejamento e execução dinâmicos deve ser levantado utilizando a técnica de etnografia, conforme seção 6.1.1.

As regras de construção desse fluxo são as seguintes:

1. Uma atividade cognitiva dá origem a uma ou mais unidades de conhecimento, o que significa que a saída do elemento da atividade cognitiva estará ligado a um ou mais elementos de unidade de conhecimento, através da seta *next*;
2. Uma ou mais unidades de conhecimento podem ser utilizadas em uma atividade cognitiva, o que significa que na entrada de um elemento que representa uma atividade cognitiva podem estar ligados um ou mais elementos de unidades de conhecimento, através da seta *next*;
3. O elemento *if* pode ser utilizado para denotar uma condição, derivando dele as setas *true* ou *false*;
4. O elemento *fork* pode ser utilizado para indicar que uma atividade cognitiva dá origem a mais de uma unidade de conhecimento;
5. Os elementos *fork* e *join* podem ser utilizados para denotar que uma unidade de conhecimento é utilizada em paralelo em determinadas atividades cognitivas ou que várias unidades de conhecimento são necessárias para iniciar uma atividades cognitiva, respectivamente.
6. O elemento *join* seguido de um *fork* indica que mais de uma unidade de conhecimento precisa ser usada em paralelo em atividades cognitivas distintas. Para simplificar esta representação, pode-se utilizar um único elemento, que acumulará as funções *join/fork*.

Por exemplo, considere o DFC da figura 5.8, no qual pode-se observar todas as diferentes construções possíveis para representar um fluxo cognitivo através do DFC.

O fluxo cognitivo da figura 5.8 começa com a atividade cognitiva A, que dá origem a duas unidades de conhecimento, 1 e 2, o que é evidenciado pelo elemento *fork* após essa atividade (regras 1 e 4). Essas unidades de conhecimento serão utilizadas para realizar a atividade cognitiva B, que dará origem a unidade de conhecimento 3 (regras 2 e 5). Perceba que as unidades de conhecimento 1 e 2 também serão utilizadas para realizar a atividade cognitiva C, juntamente com a unidade de conhecimento 3, criada pela atividade cognitiva B. Por isso, foi necessário usar um *join/fork* (regra 6) e depois outro *join* antes da atividade C, que dará origem a unidade de conhecimento 4. Ao atingir esse conhecimento, há um nó condicional, que pode terminar o processo ou dar origem a uma atividade cognitiva D (regra 3), cuja entrada é a própria unidade de conhecimento 4. Essa atividade dá origem a uma nova unidade de conhecimento 3, o que fará com que a atividade cognitiva C seja realizada novamente, tendo as mesmas unidades de conhecimento 1 e 2 anteriores como entrada, juntamente com a nova unidade de conhecimento 3. Esse fluxo continua até que a condição leve ao fim do processo.

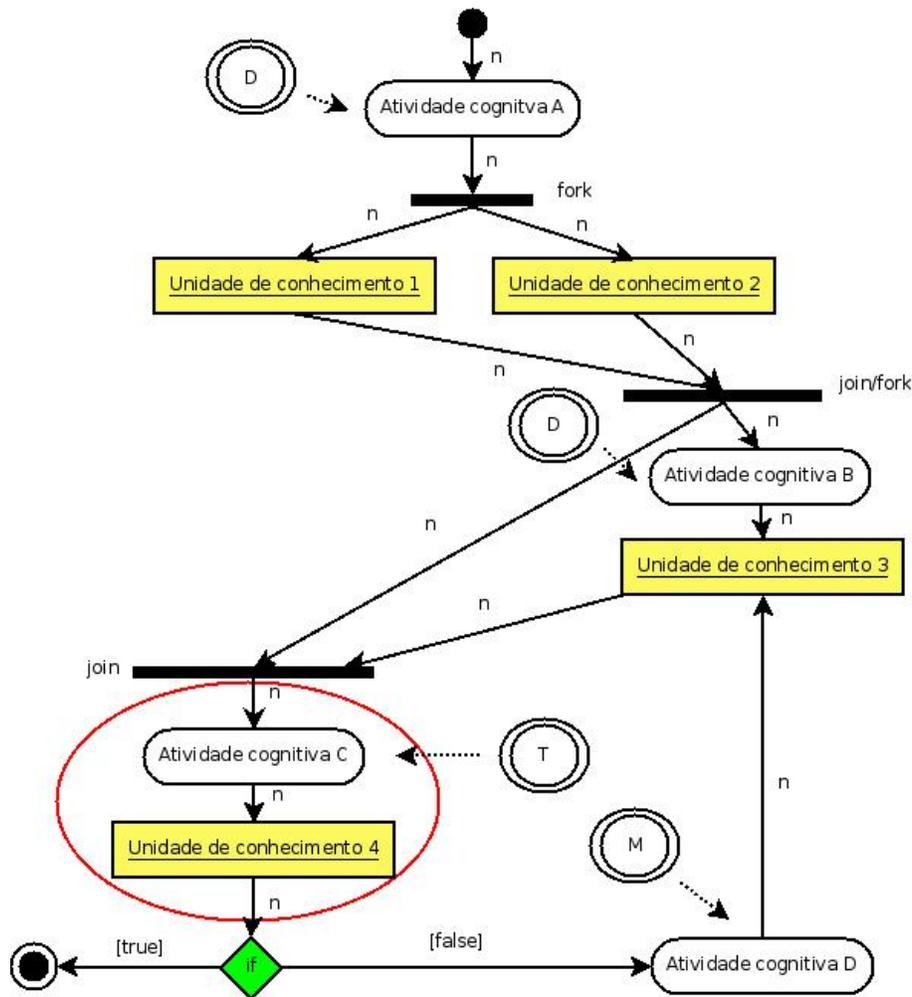


Fig. 5.8: Exemplo de construções possíveis com o DFC

Após a modelagem do fluxo cognitivo, deve-se classificar cada uma das atividades do DFC de acordo com um dos três tipos de atividades, resultantes da interpretação das categorias da teoria semiótica de Peirce no contexto de engenharia realizada neste trabalho, conforme seção 4.2.1. Os três tipos são:

- Atividades monádicas (primeiridades);
- Atividades diádicas (secundidades);
- Atividades triádicas (terceiridades).

No exemplo dado anteriormente, essa classificação pode ser observada pelas letras M (monádica), D(diádica) e T(triádica), na figura 5.8, que neste exemplo foram atribuídas arbitrariamente. Após classificar cada uma das atividades do processo em uma dessas três categorias, os pontos do processo nos quais um diálogo entre agentes computacionais e humanos deve ser aplicado devem ser identificados. Esses pontos são denominados “pontos de amplificação de inteligência”, e são escolhidos na região

das atividades triádicas. Eles são marcados com círculos, como na figura 5.8 foi marcado o único ponto do exemplo dado, ao redor da única atividade triádica marcada.

Após a análise do DFC, com o entendimento do modelo do fluxo cognitivo, é possível realizar um projeto de como será a inserção do sistema de amplificação no processo de tomada de decisão, sendo as atividades monádicas e diádicas realizadas somente pelo sistema, classificadas como “explicit-artifact process capabilities” (capítulo 5) segundo Engelbart (1962), transparente ao usuário, liberando-o para realizar apenas as atividades triádicas juntamente com os agente computacionais, classificadas como “composite process capabilities” (capítulo 5) segundo Engelbart (1962), amplificando sua capacidade de resolver o problema. No entanto, para realizar tal projeto da inserção de um sistema computacional no processo de resolução, utilizaremos Redes Semiônicas, conforme será apresentado na seção 5.7.

Com relação à modelagem do fluxo cognitivo através de um DFC, pode-se argumentar que qualquer uma das atividades cognitivas representadas no diagrama pode ser decomposta em uma série de sub-atividades, o que tornaria difícil encontrar um nível de abstração adequado para representar um determinado repertório hierárquico de capacidades utilizado para resolver um problema. Com relação a isso, Engelbart diz o seguinte (1962):

“Todo processo de raciocínio ou ação é composto de sub-processos. Deixe-nos considerar alguns exemplos como fazer um risco com lápis, escrever uma letra do alfabeto, ou fazer um plano. Alguns movimentos de músculos são organizados para fazer o rabisco com o lápis; de maneira similar, fazer traços particulares com o lápis ou fazer o plano de escrever uma carta são processos complexos em si que se tornam sub-processos para a tarefa de escrever uma letra do alfabeto.

Apesar de que todo sub-processo é um processo, que por sua vez consiste de sub-processos, não há razão para buscar o fim dessa estrutura hierárquica de processos. Parece não ser possível estar certo de que o fim aparente (processos que não podem ser sub-divididos) existe no mundo físico ou nas limitações do entendimento humano.

De qualquer forma, não é necessário começar do fim para discutir hierarquias particulares de processos. Ninguém usa um processo completamente original toda vez que resolve algo novo. Ao invés disso, o ser humano começa por um grupo básico sensorial-mental-motor de capacidades de processo, e adiciona a essas capacidades de processo as capacidades de processo de seus artefatos. Só há um número finito de capacidades humanas e de artefatos para utilizar. Além disso, mesmo processos de alto nível bem diferentes podem ter sub-processos em comum”.

Com efeito, sempre será possível expandir uma determinada atividade cognitiva representada em uma série de sub-atividades, provavelmente formando um outro DFC para representar aquela atividade. No entanto, como disse Engelbart, não há porque fazê-lo: encontrar o nível de abstração correto que represente os processos de alto nível necessário para estudar o fluxo cognitivo dos agentes é também uma tarefa do projeto.

5.7 Redes Semiônicas

O DFC é uma ferramenta adequada para o estudo do fluxo cognitivo dos agentes humanos que resolvem um determinado problema e que necessitam de amplificação de inteligência. No entanto, se o intuito é inserir um sistema para trabalhar juntamente com os agentes humanos nesse fluxo, entendê-lo é apenas a primeira etapa: posteriormente, faz-se necessário modelar a inserção do sistema de amplificação de inteligência neste contexto e ser capaz de avaliar os efeitos dessa inserção, pois a maneira como os seres humanos resolvem o problema pode ser melhorada. O que se busca nesta modelagem posterior ao DFC é a inserção do sistema computacional no fluxo de resolução e a modelagem da melhor maneira de resolver o problema. Para estudar essa inserção dos SAI no processo cognitivo analisado com o DFC, utiliza-se as Redes de Agentes Semiônicos, ou Redes Semiônicas (Gudwin, 2003), que representam uma evolução da Rede de Objetos (Gudwin & Gomide, 1998) e da Rede de Agentes (Guerrero, 2000).

As redes de agentes semiônicos são utilizadas no escopo da semiótica computacional para representação e processamento de conhecimento, podendo ser utilizadas tanto como ferramentas de análise quanto como ferramentas de síntese de processos. Essas redes são capazes de modelar sistemas dinâmicos a eventos discretos, modelando processos semióticos de manipulação de unidades de conhecimento. Essas redes também podem ser utilizadas para modelar processos organizacionais e operacionais, típicos processos que serão estudados através do DFC.

As redes de agentes semiônicos são adequadas para modelagem da inserção do sistema de amplificação no processo estudado pelo DFC, principalmente pelo fato de serem diagramas dinâmicos, em oposição ao DFC que é um diagrama estático. Um diagrama dinâmico evolui no tempo, possuindo uma parte estática, que permanece inalterada, e uma parte dinâmica, que se modifica em função do tempo. Essa peculiaridade lhe dá a capacidade de representar fenômenos dependentes do tempo, tais como sistemas dinâmicos e processos reais do mundo físico. O uso de diagramas dinâmicos pode representar de maneira mais eficiente as manipulações das unidades de conhecimento através das atividades cognitivas, modeladas no DFC, e as interações entre os agentes humanos (usuários) e computacionais (SAI). Pode-se, assim, simular várias possibilidades de inserção de um sistema computacional em diálogo com os agentes humanos para resolver o problema não estruturado.

Os agentes semiônicos são a estrutura básica de uma rede semiônica, elementos através dos quais um sistema semiótico pode ser construído. A estrutura de um agente semiônico está representada na figura 5.9 (Gudwin, 2003).

Da figura 5.9, percebe-se que um agente semiônico é composto por um conteúdo descritivo, que é uma espécie de cromossomo, responsável por determinar o comportamento do agente. Esse conteúdo descritivo pode ser dividido em cinco setores, sendo três deles estados internos (interface de entrada, estados internos e interface de saída) e os outros dois funções (transformação e avaliação). Os agentes semiônicos são classificados e agrupados em classes, que serão determinadas por essa estrutura interna do conteúdo descritivo do agente. O agente é também constituído por portas de entrada, que alimentam a interface de entrada, e portas de saída, que alimentam a interface de saída. A diferença entre agentes semiônicos e objetos em uma rede é o fato de os agentes semiônicos possuírem um ciclo de vida contínuo e uma função de avaliação que determina o comportamento dinâmico do agente. Isso significa que, ao adquirir unidades de conhecimento através de suas portas de entrada, a transformação destes em outras unidades de conhecimento nas portas de saída não será automática, como em uma rede de Petri, ou como em um objeto, em resposta a uma mensagem (Gudwin, 2003), mas

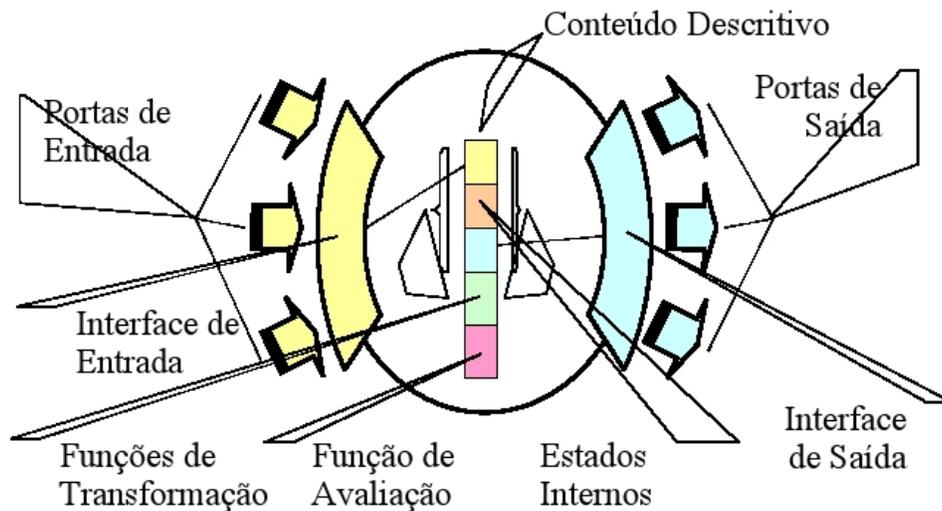


Fig. 5.9: Estrutura de um agente semiônico (Gudwin, 2003)

será uma transformação dita mediada, ou seja, por meio das funções de avaliação, que classificarão as unidades de conhecimento com relação ao interesse do agente em interagir com elas, e por meio das funções de transformação, que irão executar a transformação do conhecimento (atividade cognitiva). Ou seja, o agente semiônico irá escolher a unidade de conhecimento com a qual quer interagir e executar sua transformação, podendo destruir ou não as unidades de conhecimento utilizadas no processo (Gudwin, 2003).

O comportamento dos agentes semiônicos e sua interação com as unidades de conhecimento é dividido em 5 fases:

1. Fase de avaliação: fase na qual o agente semiônico escolhe as unidades de conhecimento com as quais deseja interagir. Através de sua função de avaliação, ele avalia cada uma das unidades de conhecimento disponíveis, dando uma nota para cada, e decide o que fará com essa unidade de conhecimento após a interação (destruí-la ou preservá-la). Essa escolha é feita para cada uma das funções de transformação, sendo testadas todas as possíveis combinações disponíveis de unidades de conhecimento com relação aos domínios de cada função de transformação. Ao fim dessa fase, cada unidade de conhecimento estará avaliada com uma nota, relativa ao interesse de interação através das variadas funções de transformação, que são atividades cognitivas;
2. Fase de atribuição: um algoritmo supervisor centralizado compila todos os interesses de cada um dos agentes semiônicos de uma rede e atribui a cada um dos agentes as unidades de interação, evitando conflitos de interesse entre os agentes. Qualquer algoritmo pode ser utilizado nesta fase. Com essa finalidade, foi desenvolvido por Guerrero et al. (Guerrero et al., 1999) um algoritmo denominado de BMSA (Best Matching Search Group), que após compilar as notas de todos os agentes do sistema dadas às unidades de conhecimento, atribui as unidades aos agente que deram a melhor pontuação para cada uma delas, evitando conflitos de interesse.
3. Fase de assimilação: o agente semiônico assimila os conteúdos das unidades de conhecimento a ele atribuídas, através das porta de entrada em sua interface de entrada.

4. Fase de transformação: a função de transformação escolhida é invocada e a atividade cognitiva é realizada, determinando os novos valores dos estados internos e as unidades de conhecimento nas portas de saída.
5. Fase de consumação: nesta última fase, a interface de saída é utilizada para construir as unidades de conhecimento nas portas de saída.

Este processo de cinco fases ocorre para todos os agentes semiônicos da rede. Caso haja uma rede de grande porte, com um grande número de agentes interagindo com todas as unidades de conhecimento, isso geraria uma explosão combinatória na fase de avaliação que seria computacionalmente intratável, pois os agentes precisam avaliar todos os outros agentes disponíveis, aumentando o custo computacional exponencialmente à medida que aumenta a população de agentes. No entanto, nem todos os agentes são adequados para interação, dependendo do seu tipo e do seu conteúdo. Para resolver tal situação, as redes semiônicas implementam “lugares”, que são espaços de representação que agrupam agentes de maneira que somente agentes que possuem interesse são sondados para formar a vizinhança de interação. Assim, os agentes são confinados em lugares, que são conectados por arcos que entram e saem das portas desses lugares. Somente agentes do mesmo tipo podem ocupar cada lugar. A figura 5.10 ilustra um exemplo de topologia da rede semiônica com agentes confinados em lugares.

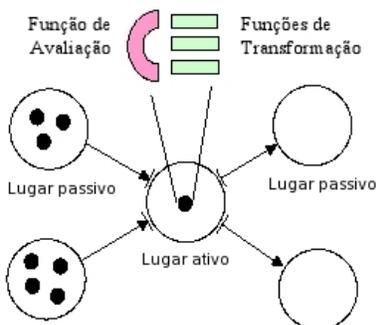


Fig. 5.10: Estrutura de uma rede semiônica com agentes semiônicos confinados a lugares (Gudwin, 2003)

Perceba na figura 5.10 que há dois tipos de lugares na rede semiônica:

- Lugares ativos: lugares onde os agentes semiônicos de uma determinada classes podem atuar. Esses agentes semiônicos (podendo ser agentes humanos ou agentes computacionais) são capazes de realizar as atividades cognitivas representadas no DFC;
- Lugares passivos: lugares que os agentes semiônicos podem ocupar mas sem atuar.

As noções de redes semiônicas apresentadas são, com efeito, abstrações bastante abrangentes em termos de modelagem, e foram aqui apresentadas com o viés do relacionamento com os conceitos do DFC. Assim sendo, os diagramas construídos com o DFC, para estudar o fluxo cognitivo dos agentes humanos que resolvem um determinado problema, podem ser modelados diretamente através de redes semiônicas, e lá modificados para estudar a inserção de SAI no contexto, sendo possível estudar as

interações entre agentes computacionais e humanos necessárias ao processo de amplificação. Assim, unidades de conhecimento e atividades cognitivas, modeladas no DFC, são distribuídas em lugares da rede, cada lugar possuindo diferentes portas e sendo conectados por arcos que representam ligações entre as unidades de conhecimento e as atividades cognitivas, realizadas por agentes humanos ou computacionais, que irão captar as unidades de conhecimento e realizar sua transformação. Assim, teremos a seguinte disposição:

- lugares passivos: unidades de conhecimento;
- lugares ativos: cálculos automáticos computacionais, representando as atividades cognitivas monádicas e diádicas;
- superlugares: agentes humanos e computacionais, representando as atividades cognitivas triádicas e a interação entre esses agentes.

Superlugar é um recurso que pode-se utilizar para permitir que exista uma hierarquia de redes semiônicas. Um superlugar em uma rede semiônica é um lugar que representa uma outra rede semiônica implementada com interfaces de entrada e saída, ou seja, a expansão do superlugar é uma subrede semiônica. Isso permite que o projetista possa modelar o processo sígnico ocorrendo internamente em um determinado lugar, reaproveitando esse projeto em outros fragmentos da rede (Gudwin, 2003).

No caso da tradução do DFC em uma rede semiônica, utiliza-se um superlugar para representar atividades triádicas, pois nessas atividades pretende-se implementar agentes computacionais que irão dialogar com agentes humanos para realização da tarefa cognitiva. Com um superlugar, pode-se implementar uma subrede dedicada a modelar somente essa interação, reaproveitando toda a estrutura do resto da rede como vizinhança. Assim, o impacto da inserção do sistema de amplificação pode ser bem estudado e compreendido, já que as atividades monádicas e diádicas serão realizadas somente pelo sistema, e as atividades triádicas, nas quais a interação faz-se necessária, poderão ser estudadas separadamente.

As redes semiônicas podem ser simuladas ou emuladas em um sistema computacional. De fato, com o intuito de implementar a teoria, foi desenvolvido um software denominado SNTToolkit (Semi-omic Network Toolkit), que é uma ferramenta de construção, desenvolvimento e execução de simulações com redes semiônicas. Essa ferramenta foi desenvolvida a partir de uma série de trabalhos, como os de Guerrero et al. (1999) e Guerrero (2000). Para ilustrar uma das aplicações, refira-se ao controle de um veículo autônomo por Ojeda et al. (2000). A ferramenta SNTToolkit está disponível na internet, no SourceForge:

- <http://sntool.sourceforge.net/>

Para ilustrar a modelagem do DFC em Redes Semiônicas e a utilização do SNTToolkit, considere o exemplo de diagrama da figura 5.8. A tradução deste DFC em uma rede semiônica está representado na figura 5.11.

Veja que as unidades de conhecimento 1, 2, 3 e 4 do DFC foram substituídas pelos lugares passivos UCo1, UCo2, UCo3 e UCo4. As atividades cognitivas A, B e D foram substituídas pelos lugares ativos ACoA, ACoB e ACoD, que serão atividades realizadas somente pelo sistema de amplificação,

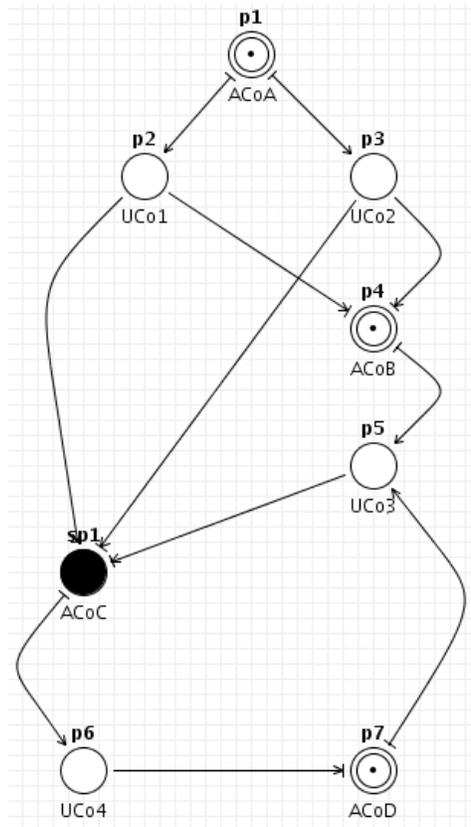


Fig. 5.11: Tradução do DFC da figura 5.8 em uma rede semiônica

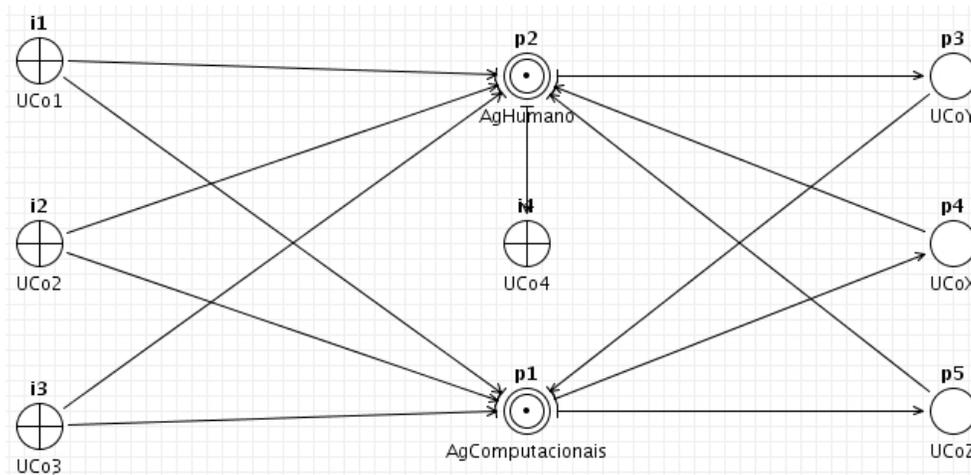


Fig. 5.12: Subrede implementada pelo superlugar ACoC da rede da figura 5.11

por serem atividades diádicas e monádicas. A atividade cognitiva C foi convenientemente substituída pelo superlugar ACoC, que implementa a subrede da figura 5.12.

Nesta subrede da figura 5.12 percebe-se quatro pseudo-lugares, i1, i2, i3 e i4. Estes pseudo-lugares funcionam como interfaces de entrada e saída do superlugar. Note que quando esta subrede é

instanciada na rede principal da figura 5.11, $i1$ se transforma em $p2$, $i2$ em $p3$, $i3$ em $p5$ e $i4$ em $p6$.

Observe como a subrede procura modelar a interação entre o agente humano e os agentes computacionais a serem implementados no sistema de amplificação: modela-se a interação desses agentes a partir da manipulação das unidades de conhecimento X (UCoX), Y (UCoY) e Z (UCoZ), que não haviam sido modeladas anteriormente no DFC. Isso significa que, neste ponto da modelagem, o projetista possui liberdade para modelar a interação a partir de unidades de conhecimento que serão manipuladas na interface de diálogo do sistema. No caso modelado, o agente humano é responsável por determinar a solução final (UCo4), após o diálogo que ocorre entre os agentes, manipulando as unidades de conhecimento UCoX, UCoY e UCoZ, utilizando para tal os conhecimentos UCo1, UCo2 e UCo3.

O diálogo traçado neste exemplo de subrede pode ser entendido da seguinte maneira: os agentes computacionais, a partir do conhecimento de UCo1, UCo2 e UCo3, propõem uma solução UCoX, como na figura 5.13. O agente humano, a partir do conhecimento de UCo1, UCo2, UCo3 e de UCoX, critica a solução UCoX, propondo UCoY, ilustrado na figura 5.14. Os agentes computacionais, agora conhecedores de UCoY, propõem a solução UCoZ, como na figura 5.15. O agente humano pode, então, acatar essa solução e gerar UCo4 ou propor um novo UCoY, para uma nova avaliação dos agentes computacionais. Após a geração da UCo4, essa unidade de conhecimento volta para a rede principal através da porta $i4$, que corresponde ao lugar $p6$. A sequência de atividades cognitivas e geração de unidades de conhecimento dessa rede está ilustrada na figura 5.16.

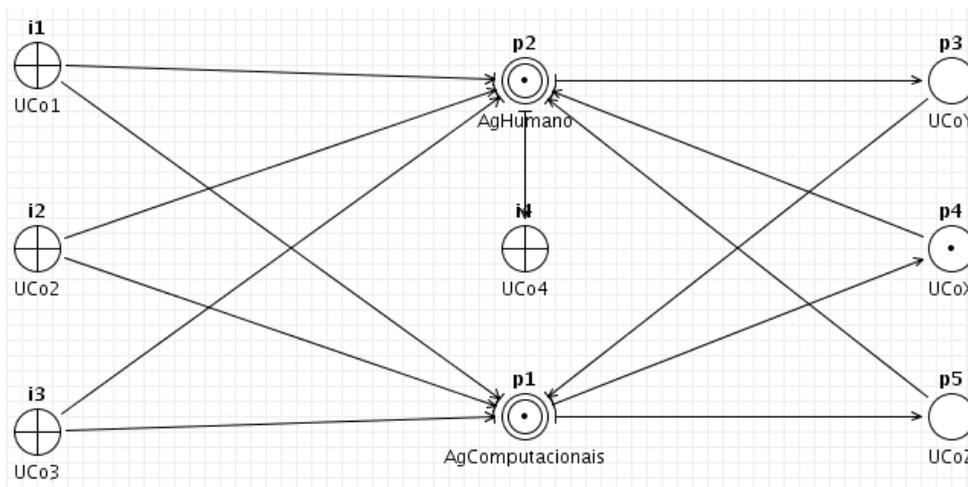


Fig. 5.13: Agentes computacionais, a partir do conhecimento de UCo1, UCo2 e UCo3, propõem uma solução UCoX

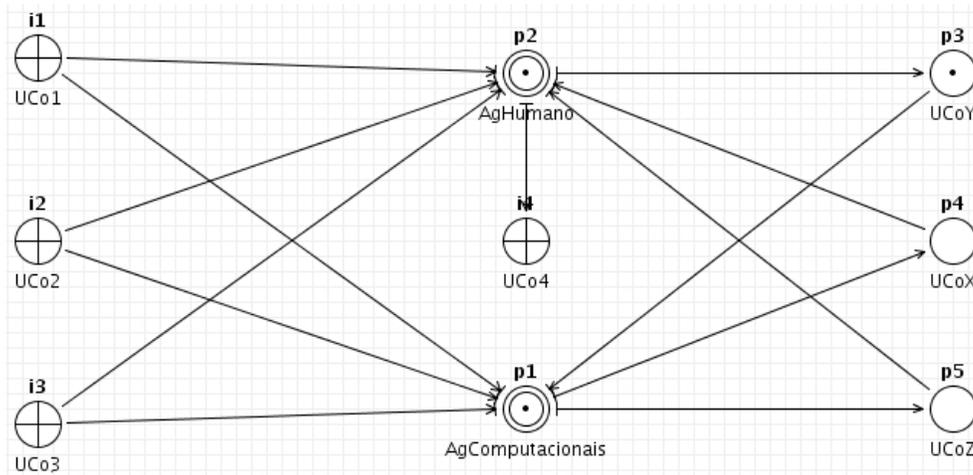


Fig. 5.14: O agente humano, a partir do conhecimento de UC01, UC02, UC03 e de UC0X, critica a solução UC0X propondo UC0Y

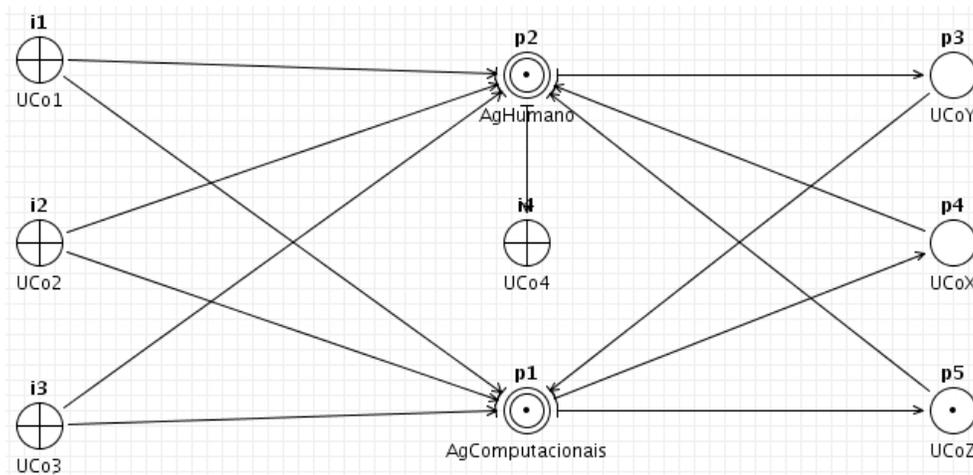


Fig. 5.15: Os agentes computacionais, levando em consideração UC0Y, propõem a solução UC0Z

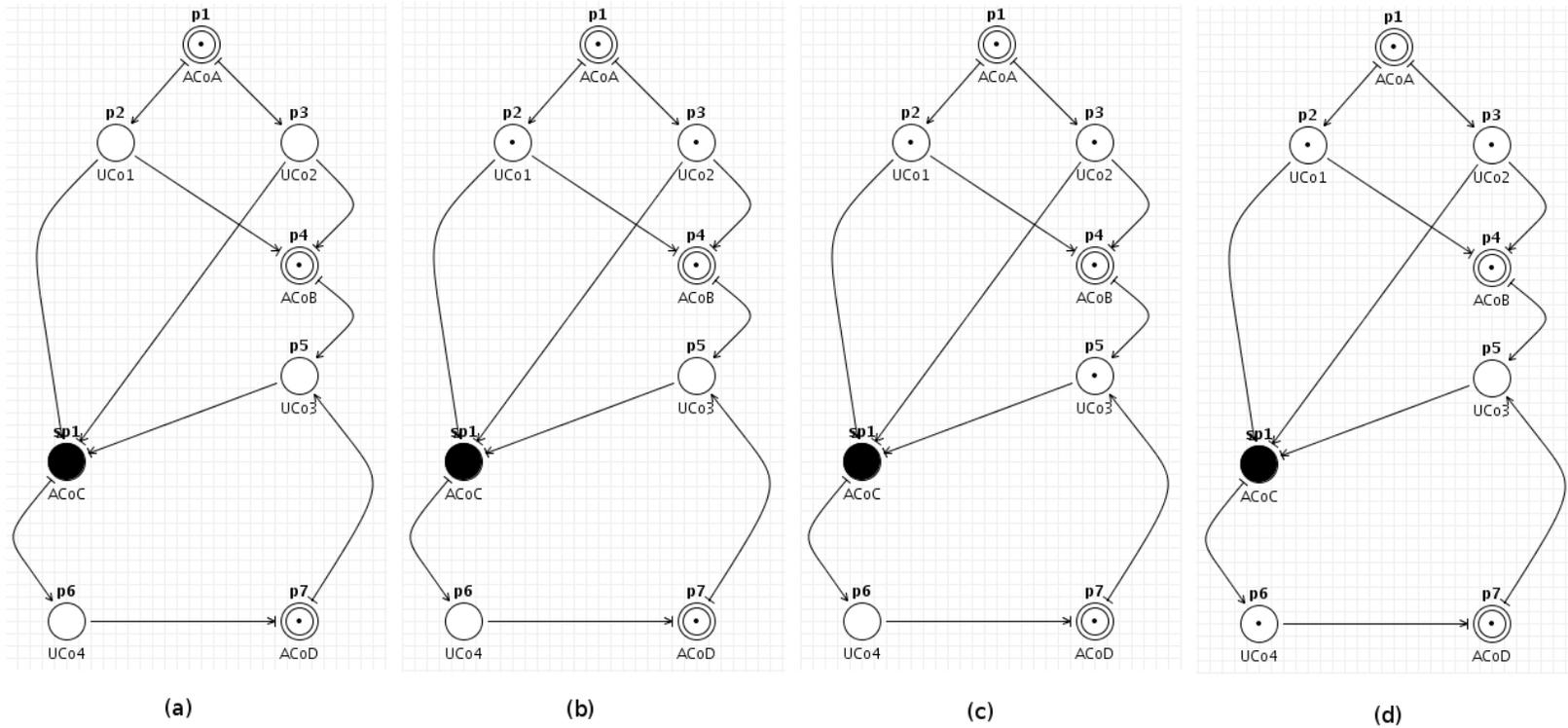


Fig. 5.16: Em (a), a rede inicia sua operação com a atividade cognitiva A, criando em (b) as unidades de conhecimento UCo1 e UCo2. Em (c), a unidade de conhecimento UCo3 é criada através da atividade cognitiva B, que se utiliza de UCo1 e UCo2 para tal. Em (d), temos a UCo4 criada depois do diálogo que acontece entre agente humano e computacional dentro do superlugar sp1, que representa a atividade cognitiva C, uma atividade inteligente. Essa rede é uma tradução do DFC da figura 5.8

Uma vantagem de utilizar a ferramenta SNTToolkit para fazer essa modelagem em Redes Semiônicas é a capacidade de simulação da ferramenta, e a oportunidade de ver as unidades de conhecimento fluindo ao longo do sistema modelado.

A ferramenta SNTToolkit gera um código para representar a rede semiônica montada. O código da rede das figuras 5.11 e 5.12 se encontra no apêndice A.1. Se este código for utilizado no SNTToolkit, é possível observar a rede e simulá-la através da ferramenta.

5.8 Resumo

Este capítulo apresentou a importância das ferramentas mentais na evolução do intelecto humano. O ser humano evoluiu seus modelos mentais ao longo da existência com a criação, utilização e refino de ferramentas mentais cada vez mais potentes e adequadas para amplificar seu intelecto, formando um sistema cognitivo distribuído (Giere & Moffatt, 2003). Neste contexto, o computador foi colocado como a principal ferramenta mental utilizada atualmente pelo ser humano, tendo o poder de amplificação que nenhuma outra ferramenta mental alcançou até hoje. Esse poder traz consigo a possibilidade de desenvolver sistemas computacionais capazes de amplificar a mente humana na resolução de problemas não estruturados do mundo real.

O capítulo também contribuiu com o estabelecimento de uma base teórica da amplificação de inteligência, com a utilização da semiótica de Peirce, e com o estabelecimento diferentes maneiras que uma ferramenta de amplificação de inteligência pode amplificar o intelecto humano, que são:

- Representar conhecimento exosomaticamente;
- Artíficos novos para expressar o pensamento;
- **Controlar o fluxo cognitivo;**
- **Realizar tarefas cognitivas.**

O capítulo fez uma revisão da história da AI, iniciando desde Engelbart em 1962 até as formas mais avançadas de amplificação de inteligência atuais, como a utilização de realidade virtual. Um importante paralelo entre a inteligência artificial e a amplificação de inteligência procurou elucidar as distinções e semelhanças entre essas duas áreas de pesquisa atuais, que são partes de um ramo denominado inteligência computacional.

Outra contribuição desde capítulo é a sugestão dos agentes computacionais autônomos inteligentes como ferramentas adequadas para realizar tarefas cognitivas compostas, juntamente com agentes humanos. Apesar de Engelbart não conhecer a teoria de agentes na época de seu manifesto sobre a amplificação de inteligência, agentes autônomos inteligentes se encaixam no arcabouço criado por ele para processos compostos.

Este capítulo apresentou os conceitos de uma Engenharia de SAI, determinando atributos necessários aos SAI e uma componentização padrão para os mesmos. No caso do atributo “interface de diálogo”, foi traçada uma comparação de seu conceito de interface com o conceito da Engenharia Semiótica, de de Souza (2005). Sugeriu-se que SAI são adequados a serem aplicados em problemas de planejamento e execução dinâmica, por endereçarem todos os requisitos necessários na resolução desses problemas inseridos em ambientes dinâmicos.

Além disso, o capítulo desenvolveu os conceitos dos Diagramas de Fluxo Cognitivo, ferramenta utilizada na análise semiótica de repertórios hierárquicos de capacidade de agentes humanos na resolução de problemas a serem amplificados, apresentando como os diagramas construídos podem ser simulados posteriormente através de Redes Semiônicas.

As possibilidades de desenvolvimento de ferramentas mentais cada vez mais avançadas são evidentes e apontam para o futuro, mas a busca por essas ferramentas necessita ser realizada no presente. Conforme Engelbart (1962):

“... as idéias apresentadas devem ser consideradas em ambos os sentidos: o sentido do direto desenvolvimento e o sentido da “arte do desenvolvimento”. Certamente, as possibilidades têm implicações de longo prazo, mas sua busca e recompensas iniciais nos esperam agora. Em nossa visão, nós não temos que esperar até aprendermos exatamente como os processos mentais humanos funcionam, nós não temos que esperar até construirmos computadores mais inteligentes, maiores ou mais rápidos, nós podemos começar a desenvolver sistemas de amplificação poderosos e economicamente viáveis baseados no que sabemos e possuímos agora. A busca por mais conhecimento básico e melhores máquinas vai continuar até o ilimitado futuro, e vai desejar estar integrada com a “arte” e com os sistemas de amplificação - mas começar agora vai fornecer não somente orientação e estímulo para essas buscas, mas nos dará efetividade na resolução de problemas que surgirão durante essas buscas.”

Capítulo 6

Uma metodologia para o desenvolvimento de Sistemas de Amplificação de Inteligência

Este capítulo apresenta uma metodologia de construção de Sistemas de Amplificação de Inteligência (SAI).

Uma premissa dos SAI é que os agentes humanos fazem parte do circuito de tomada de decisão. Assim, essa metodologia concentra-se em aspectos cognitivos dos agentes humanos que resolvem determinado problema, para que seja possível entender, em termos de processos cognitivos, como um sistema computacional pode ser inserido neste contexto para amplificar a mente de um usuário na resolução desse mesmo problema, através de um sistema de amplificação de inteligência. O intuito dessa modelagem é identificar as atividades que podem ser retiradas dos agentes humanos, sendo realizadas somente pelo sistema computacional, e quais atividades devem ser suportadas pelo sistema, atuando em conjunto com o usuário.

Os objetivos deste capítulo, que representa o núcleo deste trabalho, são:

- Entender o que é um processo de desenvolvimento de software;
- Apresentar uma proposta para um processo de desenvolvimento de software adequado aos Sistemas de Amplificação de Inteligência.

O capítulo está organizada da seguinte maneira:

- Seção 6.1: revisa conceitos importantes de engenharia de software;
- Seção 6.2: apresenta-se o Processo Unificado e seus principais conceitos;
- Seção 6.3: a metodologia é apresentada, com um diagrama e o detalhamento das atividades, fases e papéis da metodologia;
- Seção 6.4: um resumo das contribuições do capítulo é apresentado.

6.1 Engenharia de software

Engenharia de Software é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção do mesmo,

após esse ter entrado em operação (Sommerville, 2003). Para serem capazes de construir sistemas computacionais, os engenheiros de software aplicam métodos, teorias e ferramentas para solucionar problemas. A Engenharia de Software não se preocupa somente com processos técnicos de desenvolvimento, mas também com as atividades gerenciais de projeto de software e com o desenvolvimento dos métodos, teorias e ferramentas para suportar a produção.

A humanidade depende cada vez mais de sistemas de software, pois praticamente todos os produtos desenvolvidos incorporam computadores e softwares de controle. Como nesses sistemas o software representa uma porção cada vez maior do custo total, produzir softwares com boa relação custo-benefício tornou-se um dos objetivos da Engenharia de Software, essencial para um bom funcionamento dos mercados e economias mundiais. Cada nova tecnologia criada gera novas oportunidades para criação de novos sistemas computacionais, e o aparecimento de novos tipos de sistemas computacionais sugerem novas metodologias, processos de desenvolvimento de software, novas técnicas e processos de melhorias, como é caso dos sistemas de amplificação de inteligência discutidos neste trabalho.

A aplicação de metodologias para construção de sistemas de computação tornou-se indispensável a partir do fim da década de 60, quando se introduziu o hardware de terceira geração. A maior capacidade do hardware tornou possível a criação de sistemas de software muito maiores e mais complexos do que os realizados até então. No entanto, à medida que esses sistemas mais avançados começaram a ser construídos, percebeu-se que uma abordagem informal e técnicas artesanais não eram o suficientes, pois os projetos de construção sofriam atrasos excessivos e custos maiores do que os estimados. Além disso, muitas vezes os sistemas finais não desempenhavam de maneira confiável seu papel e sua manutenção era praticamente impossível, mesmo que fosse realizada pela mesma equipe que o construiu. Esse estado deu origem à chamada “crise do software” (Partridge, 1998). Enquanto os custos de hardware caíam, os de software subiam. A reação a este estado de crise foi o início da aplicação de processos de engenharia na produção de sistemas de software, dando origem à Engenharia de Software e seus métodos.

Atualmente, as técnicas construídas e evoluídas desde então são aplicadas amplamente, mas não universalmente (Sommerville, 2003). O fato de alguns ainda não aplicarem adequadamente metodologias no desenvolvimento de sistemas complexos faz com que ainda haja problemas em produzir software que atenda às necessidades do usuário, que seja entregue no prazo e dentro do custo estimados para o projeto. O fato desses problemas na construção de software ainda persistirem até os dias atuais faz com que alguns autores sugiram que a Engenharia de Software ainda está em um estado de crise (Partridge, 1998), num estado crônico de problemas.

6.1.1 Requisitos de software

Requisitos de software determinam aquilo que é requerido pelo sistema e suas restrições, ou seja, aquilo deve ser feito e em que condições deve ser realizado. Faz-se necessário desenvolver requisitos para especificar a construção de sistemas computacionais, pois os problemas enfrentados são muitas vezes complexos, sendo difícil estabelecer como o sistema deve operar. Compreender o problema e desenvolver soluções técnicas são atividades intelectuais complicadas, principalmente quando o domínio do problema é desconhecido ou pouco explorado, não existindo soluções conhecidas.

Do ponto de vista de gerenciamento de projeto, requisitos servem como base contratual do sistema, ou seja, para que o cliente aceite o sistema como resultado do contrato comercial, ele deve

realizar tudo aquilo que é especificado nos requisitos. Além disso, requisitos servem também como base de projeto do sistema, ou seja, o trabalho de construir o sistema só é considerado pronto pelos engenheiros quando todos os requisitos estão implementados. Requisitos ajudam a fechar o escopo de um projeto de software: um sistema construído sem a especificação formal de requisitos documentados tem grande chances de se tornar um projeto sem fim, de possuir dificuldades de trabalho em equipe, de ter atraso na entrega, usuários insatisfeitos e funções implementadas que nunca são de fato utilizadas. O sistema resultante torna-se de difícil manutenção e evolução quase impossível. Definir e gerenciar bem requisitos permite uma melhor comunicação entre a equipe, permite uma validação no requisito pelo cliente e até mesmo mudanças na equipe, ou equipes distribuídas geograficamente, além de mais facilidade para obter conformidade com modelos e padrões conhecidos de software. Por outro lado, definir e gerenciar requisitos representa um alto investimento, tornando o projeto mais caro e mais complexo gerencialmente. Portanto, produtos que precisam ser mantidos e evoluídos, com equipes de trabalho e conformidade com modelos e padrões, necessitam ter os requisitos desenvolvidos, documentados e mantidos.

Há uma série de características desejáveis à especificação de requisitos:

- Unicamente identificado;
- Completo;
- Atômico;
- Testável;
- Consistente com os outros requisitos;
- Modificável;
- Rastreável;
- Não ambíguo;
- Preciso.

Requisitos de usuário x Requisitos de sistema

O grau de abstração atribuído ao termo requisito é muitas vezes variável. Em alguns casos, requisitos são vistos como declarações abstratas, em alto nível, das funções e restrições do sistema. Por outro lado, algumas vezes os requisitos são vistos como descrições detalhadas, formais e até mesmo matemáticas, de uma função ou restrição. Isso se dá devido ao cenário existente na definição de um contrato para o desenvolvimento de um projeto: durante a fase de elaboração da proposta de projeto, todo o trabalho realizado pela equipe pode não resultar em ganhos futuros, caso a proposta não seja aceita. Ou seja, existe um risco nessa fase de proposta, e todo o trabalho realizado é não-remunerado, pelo menos não diretamente. No entanto, faz-se necessário desenvolver requisitos nessa fase, para que o escopo e a base contratual do projeto possam ser estabelecidos, e estimativas possam ser realizadas, baseadas nesses requisitos. A saída para essa situação é desenvolver requisitos de alto nível, que procuram expressar as necessidades dos stakeholders do projeto, sem detalhar as peculiaridades

da implementação do sistema. Por esse motivo, esses requisitos com um nível de abstração mais alto são comumente denominados de requisitos de usuário. Uma vez aceita a proposta, os engenheiros passam para uma definição mais detalhada da implementação do sistema, de modo que o cliente possa entender completamente o que o sistema fará e validar as funcionalidades. Esses requisitos mais detalhados e precisos, utilizados como base de projeto pelos engenheiros e desenvolvedores, são denominados de requisitos de sistema.

Segundo Sommerville (2003), os requisitos de usuário devem descrever os requisitos funcionais e não-funcionais de um sistema segundo a ótica de um usuário, que não possui conhecimentos técnicos detalhados. Eles devem especificar somente o comportamento externo do sistema, evitando ao máximo características do projeto e internas do sistema, ou seja, não devem referenciar características específicas da implementação das funcionalidades ou como serão implementadas as funcionalidades, mas apenas quais são as funcionalidades necessárias. Os requisitos de usuário são geralmente a base contratual de um sistema.

Os requisitos de sistema são geralmente extraídos dos requisitos de usuário, numa etapa posterior de projeto, ou seja, são descrições mais detalhadas dos requisitos de usuário. Esses requisitos devem ser uma especificação completa e consistente de todo o sistema, e devem manter uma rastreabilidade com os requisitos de usuário. Todo requisito de sistema deve ter origem em um requisito de usuário. Um requisito de usuário pode dar origem a mais de um requisito de sistema. Dessa maneira, se todos os requisitos de usuário estiverem completamente representados nos requisitos de sistema, ao implementar todos os requisitos de sistema, garante-se que todo o sistema contratado foi entregue. Por isso, requisitos de sistema são geralmente a base de projeto de um sistema computacional.

Esses diferentes níveis de abstração na definição de requisitos também são úteis para comunicar informações sobre o sistema a tipos de leitores diferentes: gerentes do cliente e fornecedores devem manipular os requisitos de usuário, por não possuírem conhecimentos técnicos, enquanto os requisitos de sistema devem ter como alvo profissionais técnicos, como engenheiros de software e desenvolvedores.

Para exemplificar as diferenças, pode-se fazer uma analogia à especificação de um projeto de um supermercado (requisitos não servem somente para especificar sistemas de software). Um dos requisitos de usuário deste projeto poderia ser o seguinte:

- RU01 - Faz-se necessário que os clientes do supermercado sejam capazes de obter rapidamente o preço das mercadorias.

Em uma fase posterior, este requisito de usuário poderia dar origem a três requisitos de sistema:

- RS01 - Faz-se necessário que as mercadorias sejam unicamente identificadas através de código de barras.
- RS02 - Faz-se necessário que as mercadorias sejam cadastradas a partir de sua identificação (código de barras) e que haja uma manutenção dos cadastros.
- RS03 - Faz-se necessário que existam máquinas para consulta de preço utilizando o código de barras das mercadorias em cada setor do supermercado.

Todos esses três requisitos de sistema são funcionais. Um requisito de sistema não-funcional para RU01 poderia ser o seguinte:

- RS04 - A consulta de preço das mercadorias através do código de barras nas máquinas de cada setor do supermercado devem ser realizadas em até 3 segundos.

Extração de requisitos

Existem atualmente algumas técnicas para obtenção e análise de requisitos bem estabelecidas. Sommerville (2003) apresenta três técnicas para este levantamento: o levantamento orientado à pontos de vista, os cenários e a etnografia. Não existe uma abordagem ideal para extração de requisitos, e geralmente faz-se necessário utilizar várias dessas abordagens para levantar e analisar completamente os requisitos de um projeto.

Levantamento orientado à pontos de vista

Normalmente, há vários tipos de stakeholders, e esses por sua vez possuem diferentes interesses nos requisitos de sistema. Por exemplo, no exemplo da especificação de requisitos de um supermercado, utilizado na seção 6.1.1, destacam-se os seguintes stakeholders:

- donos do supermercado;
- clientes do supermercado;
- representantes de supermercados concorrentes da região;
- gerentes do supermercado;
- departamento de marketing do supermercado;
- equipes de atendimento do supermercado;
- equipe de segurança do supermercado;
- administradores do supermercado;
- legislações que se apliquem ao supermercado (trabalhista, sanitária, etc).

Assim, existem muitos pontos de vista diferentes a serem considerados. As abordagens orientadas a pontos de vista reconhecem essas diferentes perspectivas e as utiliza para organizar o processo de levantamento de requisitos. Dessa maneira, pode-se mais facilmente extrair requisitos de várias perspectivas e descobrir conflitos entre as propostas dos diferentes stakeholders.

Segundo Sommerville (2003), dentre os diferentes métodos orientados a pontos de vista, destaca-se os métodos considerados como *receptores de serviços*, casos em que os pontos de vista são externos ao sistema e dele recebem serviços. A técnica resume-se a analisar os serviços recebidos por diferentes pontos de vista. As vantagens desse tipo de ponto de vista externo são as seguintes:

- os pontos de vista são externos ao sistema, o que é uma maneira natural de levantar requisitos;
- é objetiva a avaliação da validade de um ponto de vista. Para que alguma coisa seja considerada um ponto de vista, ela deve interagir com o sistema de alguma maneira;

- os pontos de vista e os serviços ajudam a estruturar os requisitos não-funcionais.

O método VORD (*viewpoint-oriented requirements definition*) é um exemplo de um arcabouço orientado à serviços, e é bem descrito por Sommerville & Kotonya (1998).

Levantamento a partir de cenários

No levantamento de requisitos a partir de cenários são descritos cenários de interação do usuário com o sistema de software, a partir dos quais os stakeholders podem compreender e criticar as interações. Cada cenário pode abordar uma ou mais interações, ajudando a acrescentar detalhes ao levantamento de requisitos. O engenheiro de requisitos trabalha com os stakeholders para identificar os cenários e captar os detalhes relevantes de cada cenário.

Segundo Sommerville (2003), estes cenários geralmente incluem:

- estado do sistema no início;
- fluxo normal de eventos;
- fluxo errado de eventos, e como lidar com isso;
- informações sobre outras atividades concomitantes;
- estado do sistema no final.

Existem diferentes tipos de cenários utilizados nesses levantamentos. Sommerville (2003) cita dois desses tipos:

1. Cenário de eventos: documenta o comportamento do sistema frente a eventos específicos e exceções que possam surgir. Há convenções diagramáticas para esse tipo de cenário;
2. Casos de uso: técnica baseada em cenários para a obtenção de requisitos, que é parte da notação UML (*Unified Modeling Language*) para descrever modelos de sistemas orientados a objetos. Um caso de uso identifica agentes envolvidos em uma interação e especifica o tipo da interação. Os agentes do processo são representados por bonecos, enquanto as classes de interação são representadas por elipses, ambos elementos nomeados. O ideal é que o conjunto de casos de uso represente todas as interações possíveis que estão representadas nos requisitos de sistema.

Levantamento com etnografia

A etnografia é uma técnica na qual um analista se insere no ambiente de trabalho no qual o sistema será utilizado para compreender os requisitos do sistema a ser desenvolvido. O trabalho diário das pessoas é observado e são identificadas as atividades em que cada um está envolvido.

A vantagem desta técnica reside na possibilidade de compreender fatores sociais e organizacionais que afetam o trabalho, mas que não são óbvios para os indivíduos que estão imersos nele, apenas para observadores externos imparciais. Satisfazer a esses requisitos sociais e organizacionais é fundamental para o sucesso do sistema, pois se esse não estiver adequado à cultura da organização, ainda que seja entregue e capaz de solucionar os problemas, pode acabar no desuso. A etnografia pode

ainda ser combinada com a prototipação, que, com um certo número de ciclos de refinamento, pode chegar a um interessante pré-sistema, validado pela cultura organizacional.

Como desvantagens, esse método não é adequado para descobrir requisitos relacionados ao domínio do problema, devido ao seu enfoque no usuário final. Portanto, a etnografia não pode ser considerada uma abordagem completa e deve ser combinada com outras técnicas, como os casos de uso (Sommerville, 2003).

Levantamento de requisitos de Sistemas de Amplificação de Inteligência

No caso de Sistemas de Amplificação de Inteligência, faz-se necessário compreender como a tomada de decisão na solução de um problema é alcançada em termos de processos cognitivos humanos, e como um sistema computacional pode ajudar na melhoria desse processo de tomada de decisão. Portanto, a utilização das técnicas de obtenção e análise de requisitos existentes não é suficiente, pois essas preocupam-se exclusivamente com o fluxo da informação através de interações e eventos. Na seção 6.3, será possível perceber que vários pontos de vista são analisados no levantamento de requisitos de Sistemas de Amplificação de Inteligência, e que a etnografia é também altamente utilizada e recomendada. No entanto, para analisar cenários de interação, ao invés de utilizar-se de casos de uso ou outros tipos de cenários diferentes, criou-se um tipo especial de diagrama denominado Diagrama de Fluxo Cognitivo (DFC), no qual será possível estudar o fluxo cognitivo humano. Esses diagramas, simulados dinamicamente através de uma ferramenta computacional, serão utilizados para compreender melhor as relações e interações homem-máquina e levantar os requisitos, denominados requisitos de amplificação de inteligência.

6.1.2 Processos de desenvolvimento de software

Processo de desenvolvimento de software é uma sequência de atividades que transforma requisitos de usuário em um sistema computacional (Jacobson et al., 1999b), como mostra a figura 6.1.

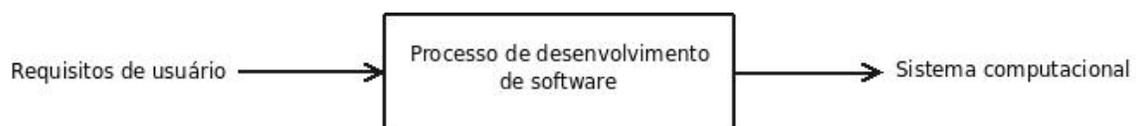


Fig. 6.1: Processo de desenvolvimento de software

As atividades de um processo formam fluxos de trabalho, que possuem diferentes tipos de papéis. Um pessoa que realiza um processo pode assumir mais de um papel, em uma série de atividades. Os fluxos de trabalho também identificam artefatos que são necessários (artefatos de entrada) e que são produzidos (artefatos de saída) em cada uma das atividades. Portanto, um processo identifica fluxos de trabalho como papéis associados a atividades, que exigem e produzem determinados artefatos.

O processo de software de uma organização é, portanto, o conjunto de atividades que as pessoas da organização realizam para desenvolver, manter e melhorar software e produtos associados (a descrição dessas atividades, não a execução delas). Há quatro atividades de processo que são fundamentais e comuns a grande parte dos processos existentes:

- Especificação do software: a operação do software e suas restrições são definidas;

- Desenvolvimento do software: o software é construído de acordo com as especificações;
- Validação do software: o software é validado para garantir que realiza as funções e objetivos especificados;
- Evolução do software: o software é evoluído para se adaptar às mudanças e novas necessidades.

Os diferentes processos organizam essas atividades de diferentes maneiras, não havendo uma maneira considerada ideal. Com efeito, a diversidade de processos é potencialmente infinita, e diferentes organizações desenvolveram diferentes abordagens para desenvolvimento de software, em parte devido ao fato dessas organizações abordarem problemas e aplicações completamente diferentes. Assim, os processos podem ser especializados, quando aplicados em um determinado contexto, escolhendo um conjunto das atividades existentes, ou estendidos, quando necessitam de fluxos de trabalho a mais para resolver problemas específicos. Os principais fatores que causam a especialização ou a extensão dos processos são (Jacobson et al., 1999b):

- Fatores organizacionais: cultura, estrutura, habilidades de gerenciamento, experiências anteriores e sistemas de software existentes em uma organização;
- Fatores do domínio: domínio da aplicação, processos do negócio e a comunidade de usuários;
- Fatores do ciclo de vida: tempo disponível para construir o sistema, tempo de vida estimado do sistema, tecnologia e habilidade das pessoas que produzem o sistema;
- Fatores técnicos: linguagem de programação utilizada, ferramentas de desenvolvimento, padrões de arquitetura, comunicações e distribuição.

A partir desses fatores, pode-se decidir por remover determinadas atividades, papéis e artefatos dos fluxos de trabalho de um determinado processo aplicado a um tipo de projeto.

Processos de desenvolvimento de software efetivos capturam as melhores práticas do estado da arte do desenvolvimento de software, reduzindo os riscos e aumentando a previsibilidade. Além disso, eles fornecem uma cultura e uma visão comum para uma comunidade que desenvolve software, tanto para os desenvolvedores como para clientes, usuários e gerentes executivos. Um processo deve ser capaz de evoluir ao longo do tempo. Ao criar um processo, os engenheiros de processo precisam equilibrar quatro pontos importantes (Jacobson et al., 1999b):

- Tecnologias: o processo deve utilizar tecnologias factíveis no presente;
- Ferramentas: processos e suas ferramentas evoluem em conjunto;
- Pessoas: a habilidade necessária para executar o processo deve ser compatível com a habilidade do seu público-alvo, para que este possa ser rapidamente treinado para utilizá-lo;
- Padrões da organização: o processo deve estar adaptado aos padrões atuais da organização onde é utilizado.

Um processo precisa amadurecer durante um certo tempo antes de atingir a estabilidade e maturidade necessárias para criar produtos comerciais de qualidade. Desenvolver produtos novos já é arriscado, utilizar um processo instável para fazê-lo torna essa atividade ainda mais arriscada.

Um processo de software comum utilizado por uma comunidade de desenvolvimento traz uma série de benefícios (Jacobson et al., 1999b):

- Todos os desenvolvedores entendem o que eles precisam fazer para desenvolver o produto;
- Desenvolvedores entendem o que os outros desenvolvedores estão fazendo;
- Gerentes entendem o que os desenvolvedores estão fazendo, mesmo que não possam ler código;
- Desenvolvedores, supervisores e gerentes podem mudar de projeto dentro de uma organização sem precisar aprender outro processo;
- Treinamentos podem ser padronizados em uma organização, podendo ser ministrados pelos próprios desenvolvedores em mini-cursos;
- O desenvolvimento de sistemas torna-se um processo que pode ser repetido, o que significa que pode ser bem estimado com relação a esforço e custo com precisão suficiente para gerenciar bem as expectativas do cliente e da equipe de desenvolvimento.

Uma organização pode ter mais de um processo, já que muitas vezes há tipos de sistemas e projetos diferentes dentro de uma mesma organização. O esperado é que uma organização possua uma espécie de "biblioteca de processos", possuindo uma série de atividades que representam técnicas de engenharia de software e um manual com regras para a escolha de um conjunto de atividades dessa biblioteca para compor um processo único para realizar um determinado projeto, de acordo com as características específicas daquele projeto.

Modelos de processo de desenvolvimento de software

Modelos de processo de software são representações simplificadas de um processo de software, apresentadas a partir de uma perspectiva específica (Sommerville, 2003). Entre os diferentes tipos de modelos, vale citar:

- Modelos de processo genéricos: modelos abstratos de processos de software, representando ciclo de vida, papéis, atividades e artefatos produzidos. Um exemplo de um modelo de processo genérico é o Processo Unificado (PU) (Jacobson et al., 1999b).
- Modelo de processo da organização: representação abstrata do processo de uma organização. Esse modelo é geralmente baseado em um modelo de processo genérico, e procura documentar o processo da organização, com o objetivo de guiar as pessoas nos desenvolvimentos, treinar novos membros da equipe ou gerenciar e melhorar continuamente o próprio processo. O ideal é que o processo descrito no documento seja igual ao praticado na organização, e utilizado para treinamento. Esse modelo de processo deve ser flexível, apropriado à cultura da organização, medido, controlado e melhorado constantemente.

- Modelos de capacidade de processo: são modelos de processo que podem ser utilizados para avaliar a qualidade do processo de software de uma organização e para estabelecer um processo de melhoria. Esses modelos são como repositórios de melhores práticas de processos de software, acumulados e desenvolvidos. Alguns exemplos mais conhecidos são os modelos CMMI, COBIT e o brasileiro MPS-br.

A figura 6.2 ilustra o relacionamento entre os tipos de modelo abordados anteriormente. O processo de software de uma organização, que são as atividades que as pessoas de uma organização realizam para produzir, manter e melhorar o software, são descritos pelo modelo de processo da organização, que costuma incorporar atividades e práticas de modelos genéricos de processo. Os modelos de capacidade de processo, verdadeiros repositórios de melhores práticas, são utilizados para avaliar o modelo do processo da organização, estabelecendo um perfil de capacidade que é utilizado para gerar uma processo de melhoria contínua do processo. A seta bi-direcional entre o processo de software da organização e o modelo de processo da organização evidencia que o modelo descreve o processo da organização, que é determinado pelo modelo. Isso significa que o modelo deve ser flexível para acompanhar as mudanças do processo da organização, que está atrelado à sua cultura, e muda constantemente. Algumas vezes, essa mudança pode até mesmo ser drástica, como acontece no caso de uma organização mudar as suas áreas de atuação no mercado. No entanto, algumas vezes pode-se querer mudar o modelo da organização para que este influencie a cultura e o processo da organização, como quando deseja-se adequar o modelo a melhores práticas de modelos de capacidade, como o CMMI ou o MPS-br, a fim de conseguir melhorias e certificações para o modelo e o processo da organização.

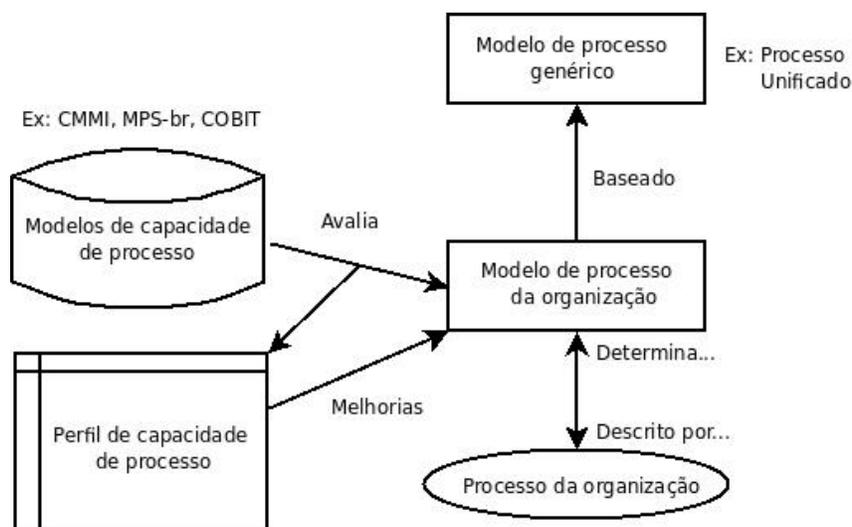


Fig. 6.2: Tipos de modelo de processo de desenvolvimento de software

No caso de sistemas de amplificação de inteligência, não há nenhum processo de desenvolvimento de software existente adequado à sua construção, pois os processos existentes não garantem que o SAI será construído com seus atributos necessários, devido ao fato de nenhum processo de software levar em consideração uma modelagem de como a tomada de decisão na solução de um problema é alcançada em termos de processos cognitivos humanos e como um sistema computacional pode ajudar

na melhoria desse processo de tomada de decisão. Dessa maneira, processos de desenvolvimento tradicionais, utilizando técnicas tradicionais como "casos de uso", são necessários mas não suficientes. Antes de aplicar as atividades tradicionais, faz-se necessário atividades capazes de modelar o processo cognitivo humano na tomada de decisão e a inserção de um sistema computacional neste processo, capaz de melhorá-lo.

6.2 O Processo Unificado de Desenvolvimento

O *Processo Unificado de Desenvolvimento* ou *Processo Unificado* (PU) é um processo de desenvolvimento de software que é resultado de trinta anos de desenvolvimento e prática (Jacobson et al., 1999b), iniciado na abordagem da Ericsson de 1967, baseada em blocos interconectados, ou seja, subsistemas implementados como componentes, passando pelo Objectory Process de 1987 a 1995, que evoluiu os conceitos de casos de uso, pelo Rational Objectory Process de 1995 a 1997, que trouxe mais ênfase à arquitetura e à iteratividade, até chegar no Rational Unified Process em 1998, que unificou tudo com outros fluxos de trabalho, como gerência de projeto e modelagem de negócios (para extrair requisitos do processo de negócio ao qual o software será aplicado) e com a Unified Modeling Language (UML) (Jacobson et al., 1999a).

O PU, além de um processo de desenvolvimento, é também um arcabouço genérico de processos que pode ser especializado para uma determinada classe de sistemas de software e para diferentes áreas de aplicação (Jacobson et al., 1999b). O PU se utiliza da UML para construir todos os modelos do software.

As principais características do PU são:

- Orientado a casos de uso: casos de uso são uma sequência de ações interativas entre usuário e sistema para entregar ao usuário algum resultado de valor (Jacobson et al., 1999b). Um sistema computacional desenvolvido com o PU é construído a partir de uma perspectiva das interações do usuário com o sistema. Esse usuário pode ser um humano ou outro sistema computacional. Essas necessidades e vontades são traduzidas em casos de uso, que capturam, dessa maneira, os requisitos funcionais do sistema. Os casos de uso são utilizados como base para guiar o processo de desenvolvimento, pois todos os fluxos de trabalho derivam deles;
- Centralizado na arquitetura: apesar dos casos de uso guiarem o PU, eles não são selecionados isoladamente, mas em harmonia com a arquitetura do sistema. Os casos de uso guiam a arquitetura do sistema, enquanto a arquitetura influencia a seleção dos casos de uso, de maneira que eles evoluem concomitantemente à medida que ocorre o ciclo de vida do processo de desenvolvimento. O papel da arquitetura de um software é similar ao papel da arquitetura de uma construção civil, uma análise de vários pontos de vista. No caso da construção, se analisa a estrutura, a rede elétrica, o sistema de calefação e resfriamento, entre outros, o que permite aos construtores uma visão do todo antes de iniciar a construção. No software, a arquitetura é representada por visões do sistema a ser construído, representando os aspectos estáticos e dinâmicos mais significativos do sistema. Entre outras coisas, a arquitetura irá capturar e tratar as necessidades dos requisitos não funcionais. Assim, ela dá uma forma ao sistema, não somente para o início do seu desenvolvimento como para sua evolução futura;

- Iterativo e incremental: o trabalho de construção do sistema é dividido em partes menores, mini-projetos, iterações que resultam em um incremento. Iterações são passagens pelos fluxos principais de trabalho, enquanto incrementos representam um crescimento do sistema. Os desenvolvedores escolhem um grupo de casos de uso que estendem as funcionalidades do sistema para ser implementados em cada iteração. Casos de uso que representam funcionalidades mais importantes e mitigam os maiores riscos do projeto são implementados nas primeiras iterações do desenvolvimento iterativo e incremental, em contraste com o modelo cascata, no qual os riscos mais graves só serão tratados no período de integração e testes do sistema, como mostra a figura 6.3. Isso torna o PU um processo de desenvolvimento orientado a riscos. Iterações seguintes complementam os artefatos produzidos a partir da situação que eles ficaram na iteração anterior. Ao fim de cada iteração, cumpridas as metas, o processo continua com a iteração seguinte. Caso contrário, as decisões são revisadas e tenta-se uma outra abordagem novamente na iteração que falhou. Essa forma de desenvolvimento reduz os riscos de gastos excessivos com retrabalho, pois se é necessário repetir a iteração, perde-se apenas o esforço de uma iteração, não o valor do projeto inteiro. Além disso, reduz-se o risco de não cumprir o cronograma, pois problemas e riscos são descobertos no início do desenvolvimento, ao invés de revelados apenas no teste do sistema completo ao fim do projeto, quando o tempo necessário para resolver os problemas será maior do que o tempo disponível. A figura 6.4 ilustra como seria este ganho de tempo do desenvolvimento iterativo e incremental com relação ao desenvolvimento em cascata. No modelo cascata, os desenvolvedores não começam a implementar antes de completar os requisitos, análise e projeto. A implementação ocorre sem muitos problemas, pois não há entregas para serem realizadas paulatinamente. No entanto, os problemas serão revelados na integração e testes todos de uma vez, acarretando um retrabalho maior, pois será necessário mudar todos os passos construídos. No modelo iterativo, a implementação começa mais cedo e as entregas frequentes revelam problemas em quantidades menores, facilitando os retrabalhos que são menores, representados pelos retrocessos da linha mais escura da figura 6.4. Além disso, desenvolver de maneira iterativa faz com que os requisitos do sistema sejam refinados nas sucessivas iterações, tornando o modo de operação mais adaptável a mudanças nestes requisitos, além de combater a falsa idéia de que requisitos podem ser completamente compreendidos desde o começo do projeto.

As figuras 6.5 e 6.6 apresentam como está organizada a sequência de ciclos que compõem o ciclo de vida do PU. Cada ciclo termina com uma entrega do produto ao cliente, e consiste de quatro fases: concepção, elaboração, construção e transição.

Em cada fase, os desenvolvedores ou gerentes de projeto podem subdividir o trabalho em iterações. Cada fase termina em um marco do projeto. Os marcos de projeto são pontos estratégicos que servem como uma espécie de divisor de águas, um ponto crítico do projeto onde deve-se analisar o que foi construído até então e o que se propõem realizar dali em diante. Além disso, os marcos de projeto servem para os gerentes monitorarem o progresso do trabalho e planejar as próximas iterações, baseados na experiência retirada da iteração que chega ao fim, por exemplo, estimando melhor os tempos das tarefas e planejando recursos especiais necessários para a equipe. Assim como antes de cada iteração há um planejamento dela, ao fim há uma avaliação dessa iteração, que servirá como base para o planejamento da próxima.

A figura 6.7 apresenta os cinco fluxos principais de trabalho do PU - requisitos, análise, projeto,

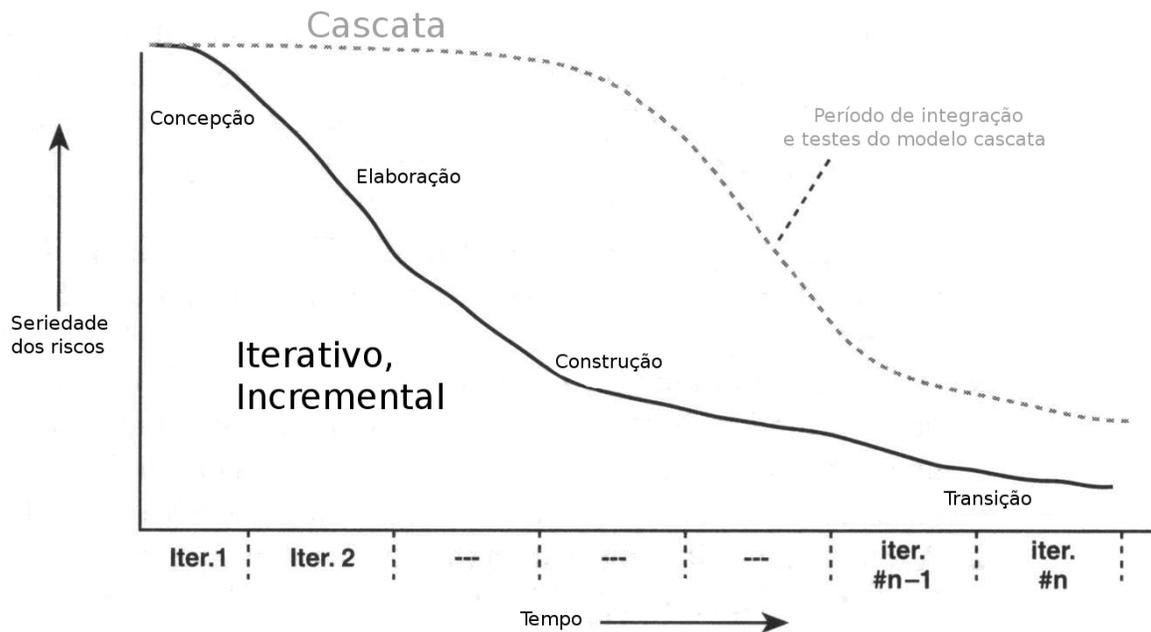


Fig. 6.3: Riscos mais sérios são reduzidos nas primeiras iterações do modelo iterativo e incremental (Jacobson et al., 1999b)

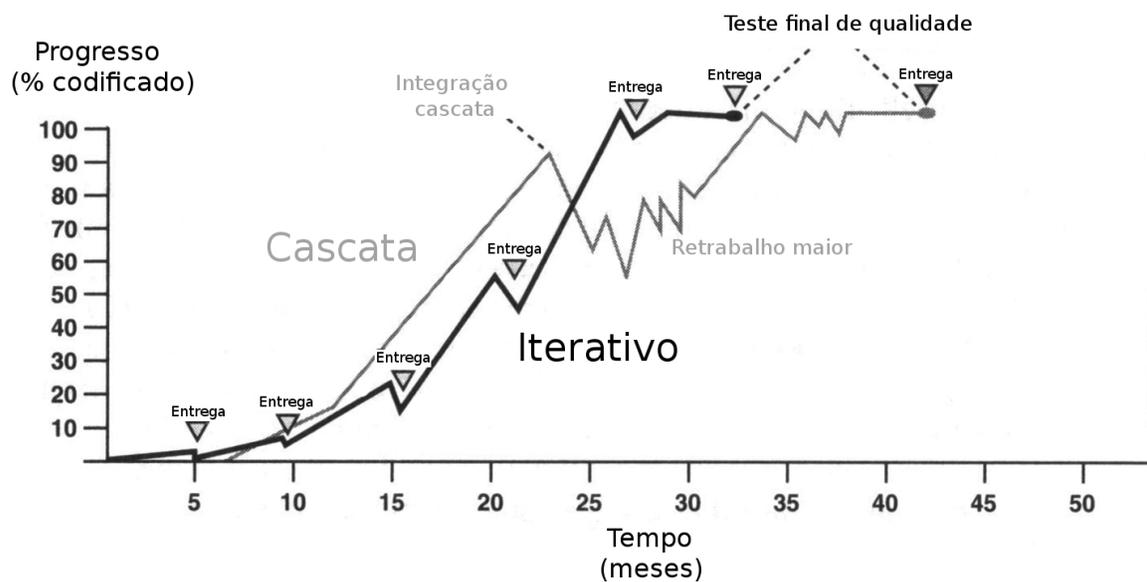


Fig. 6.4: Retrabalho nos modelos em cascata e iterativo (Jacobson et al., 1999b)

implementação e testes - acontecendo em iterações ao longo das quatro fases. As curvas são aproximações da extensão dos trabalhos de cada fluxo em cada fase. Pode-se observar por meio dessas curvas que durante a concepção e a elaboração, a ênfase se dá em capturar requisitos e realizar uma

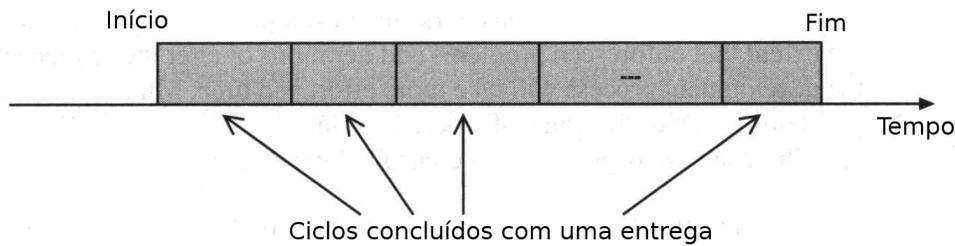


Fig. 6.5: Ciclos do PU, do início ao fim da vida do processo (Jacobson et al., 1999b)

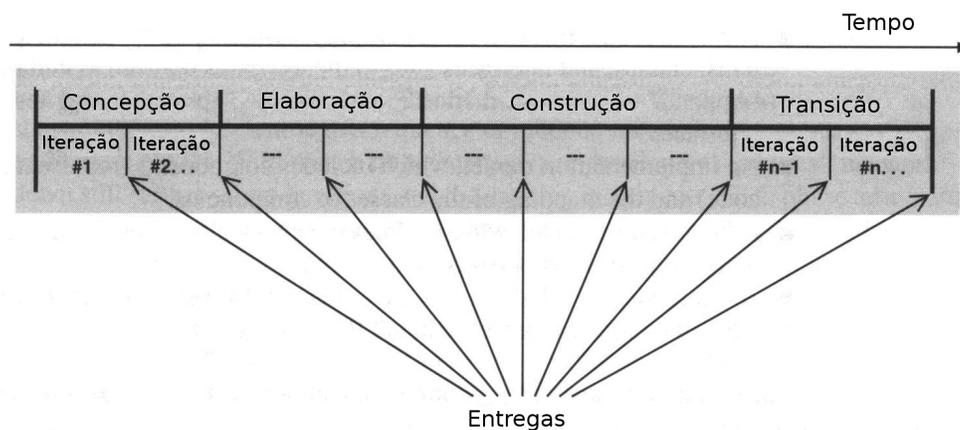


Fig. 6.6: Um ciclo do PU com suas fases e iterações (Jacobson et al., 1999b)

análise e projeto preliminares. Na fase de construção, a ênfase passa para um projeto, implementação e testes detalhado. Uma iteração passa pelos cinco fluxos principais de trabalho, como mostrado na iteração da fase de elaboração destacada na figura 6.7.

Devido a esses diferentes focos de cada uma das fases, o conteúdo das iterações muda para acomodar os objetivos da fase em questão. No entanto, o fluxo de trabalho das iterações de cada fase sempre parte de um fluxo de iteração genérico, que contém todas as atividades do PU, e está apresentado na figura 6.8, que fornece uma visão geral do processo de desenvolvimento.

Existe uma distinção entre fluxo principal de trabalho e fluxo de iteração. Os fluxos principais de trabalho do PU são: Requisitos, Análise, Projeto, Implementação e Teste. Esses fluxos principais ocorrem repetidamente em cada fluxo de iteração, diferindo em detalhes em cada uma dessas iterações.

Na figura 6.8, os papéis do processo estão listados verticalmente na esquerda e na direita, identificando as raiais, que delimitam as atividades de cada papel. O tempo avança da esquerda para a direita, caracterizando a ordem cronológica das tarefas. Na figura, as engrenagens simbolizam as atividades, enquanto as setas uma relação temporal entre as atividades. Os fluxos principais de trabalho estão circulados, destacando as atividades que eles realizam. Um engenheiro de casos de uso, por exemplo, analisa casos de uso no fluxo de análise e projeta casos de uso no fluxo de projeto. Partindo dessas atividades, monta-se o fluxo de trabalho de cada iteração, a partir dos fluxos principais de tra-

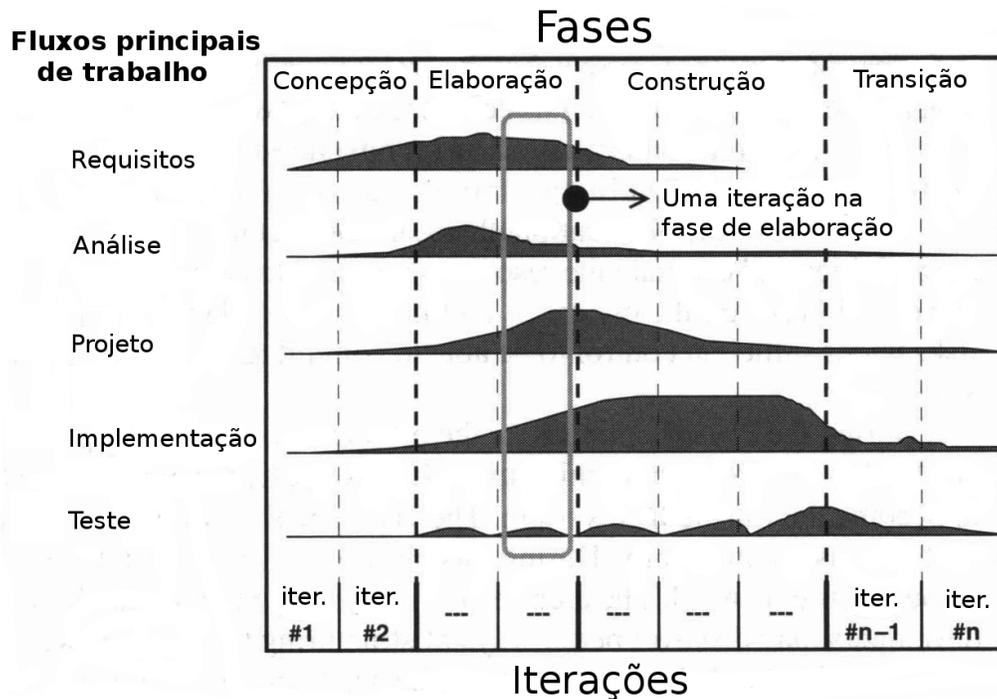


Fig. 6.7: Os cinco fluxos principais de trabalho do PU ocorrem em iterações ao longo das quatro fases (Jacobson et al., 1999b)

balho (requisitos, análise, projeto, implementação e testes), baseado nas metas estipuladas para cada iteração de uma determinada fase.

Dessa maneira, as atividades realizadas dentro do círculo de Requisitos, por exemplo, variam de acordo com a localização da iteração no processo. Na fase de concepção e elaboração, por exemplo, a atividade de prototipar interface de usuário não é realizada, sendo deixada apenas para a fase de construção. Percebe-se também que os fluxos se sobrepõem, com o Teste iniciando, por exemplo, durante o Projeto, assim que há informação suficiente para começar a projetar testes.

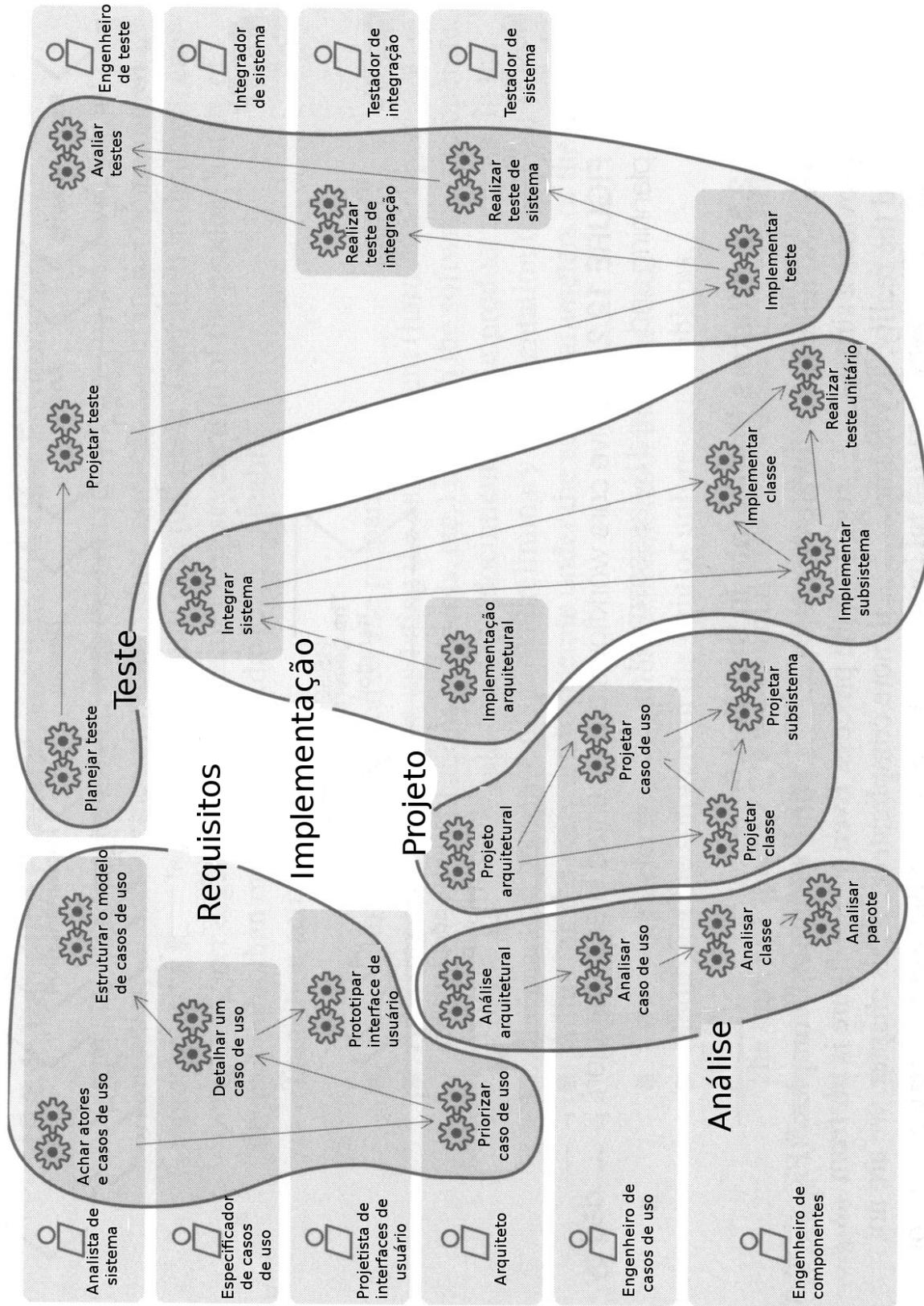


Fig. 6.8: Atividades do fluxo de iteração genérico do PU (Jacobson et al., 1999b)

Na fase de concepção é gerada uma visão do produto, determinando o escopo do que o sistema fará para os principais usuários, uma arquitetura candidata, além de um plano e custos de projeto.

A fase de elaboração especifica os casos de uso e estabelece a arquitetura.

Na fase de construção, ocorre o crescimento do sistema, e a visão transforma-se em um produto pronto para ser entregue aos usuários.

Na fase de transição, o produto entra na versão beta, acolhendo defeitos e deficiências de operação para serem rapidamente corrigidos e lançar uma versão delta, até que o produto possa ser entregue à comunidade de usuários.

Jacobson et al. (1999b) descreve cada uma das atividades da figura 6.8 em detalhes.

Um ciclo de projeto de tamanho médio distribui tempo e recursos entre as fases conforme a figura 6.9. Projetos maiores e mais complexos alteram um pouco essa distribuição, geralmente deslocando um pouco mais de tempo para as fases iniciais, que determinam o entendimento e o escopo do que se deve fazer. Por exemplo, para um sistema pequeno e um time experiente no domínio da aplicação, a fase de concepção pode ser bem curta. O time sabe que não há riscos críticos, conhece uma arquitetura que já funcionou para aquela situação e pode ser reutilizada, resultando em um escopo definido e uma arquitetura candidata identificada em poucos dias. No caso de um sistema grande, complexo e um time inexperiente, a fase de concepção e elaboração devem ser mais longas e possuir mais iterações.

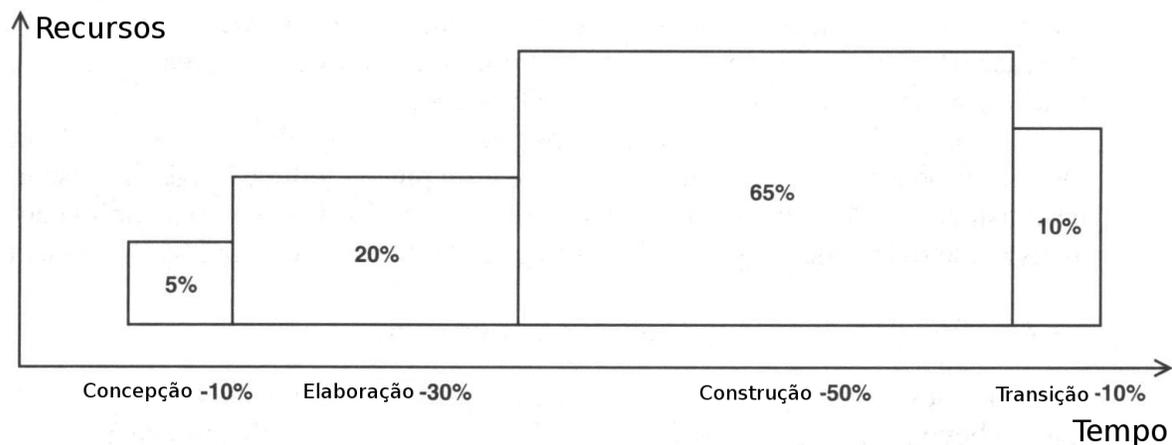


Fig. 6.9: Alocação média de recursos e tempo das fases do PU (Jacobson et al., 1999b)

Resumindo, pode-se dizer que os principais temas tratados pelo PU são (Jacobson et al., 1999b):

- Identificar os requisitos corretamente: identificar os requisitos por meio de casos de uso, análise e validações do usuário. Executar um desenvolvimento orientado a casos de uso, passando aos outros fluxos principais de trabalho a partir deles;
- Identificar a arquitetura corretamente: desenvolver o sistema centralizado na arquitetura, com partes independentes, solidifica o que deve ser feito e guia o desenvolvimento, criando um guia para futuras gerações do produto;
- Utilizar componentes reutilizáveis: as interfaces da arquitetura são elementos que tornam o desenvolvimento baseado em componentes possível. Blocos reutilizáveis reduzem o custo de desenvolvimento, o tempo de desenvolvimento e melhoram a qualidade;

- Pensar e se comunicar utilizando a UML: a UML torna o desenvolvimento uma disciplina de engenharia. Ela é uma linguagem gráfica com a qual as pessoas pensam, visualizam, analisam, comunicam e gravam informações relevantes do projeto;
- Iterar: iterações e várias entregas proporcionam vantagens por meio da utilização de pequenos grupos de trabalho, gerenciamento apertado dos riscos, vários marcos de controle e validações. Aprende-se através da experiência dos desenvolvedores e usuário nas várias entregas;
- Gerenciar riscos: identificar riscos em uma lista e mitigá-los antes que se concretizem no processo de desenvolvimento para não fracassar as estimativas de esforço e custo do projeto.

Pelo fato do PU ser um arcabouço, e não um processo rígido, ele pode ser especializado para diferentes áreas de aplicação, tamanhos de projeto, domínios de projeto e complexidades de projeto. Além disso, ele pode ser estendido, não limitando seus usuários a uma única maneira de seguir as atividades.

Para que um projeto de software tenha sucesso, além de realizar os fluxos principais de trabalho ao longo das iterações e das fases do PU, faz-se necessário também lidar com questões gerenciais do projeto, como gerência de riscos, controle de qualidade, gerência de projeto, gerência de requisitos, gerência de configuração, entre outros. O PU não descreve esses processos gerenciais, focando-se apenas nas facetas de engenharia, baseadas nas três idéias principais do processo (casos de uso, arquitetura, desenvolvimento iterativo e incremental) e em um projeto baseado em componentização. No entanto, o PU fornece um arcabouço capaz de integrar todas essas questões gerenciais (Jacobson et al., 1999b). Considera-se, portanto, que um projeto de sucesso realizará também essas atividades.

O Rational Unified Process (RUP) é um exemplo de processo de desenvolvimento de software que estende o PU, oferecendo alguns fluxos principais de trabalho adicionais para lidar com as questões de gerenciamento essenciais para um projeto de software e que não estão presentes na descrição do PU, como modelagem do negócio, gerenciamento de projeto e gerenciamento de configuração e mudanças (Jacobson et al., 1999b), como mostra a figura 6.10. O RUP é um processo proprietário, pertencente à empresa IBM Rational¹.

6.3 Processo Unificado Cognitivo: uma metodologia de desenvolvimento de SAI

Conforme dito na seção 5.5, para que um SAI seja capaz de lidar com problemas não estruturados em ambientes dinâmicos, ele precisa possuir sete atributos essenciais, com os quais irá constituir um sistema que representa uma extensão de SSTD (capítulo 3), com técnicas de amplificação de inteligência (capítulo 5), agentes computacionais (seção 5.4) carregando motores de solução, que amplificam continuamente o intelecto do usuário, aplicando essa amplificação em pontos específicos do processo cognitivo do agente humano que são ineficientes na resolução do problema, identificados como resultado de uma análise semiótica (capítulo 4) desse processo.

No entanto, não existe uma metodologia de desenvolvimento capaz de garantir que esses atributos estarão presentes em um SAI construído através dela. Necessita-se de uma metodologia que considere

¹<http://www.rational.com>

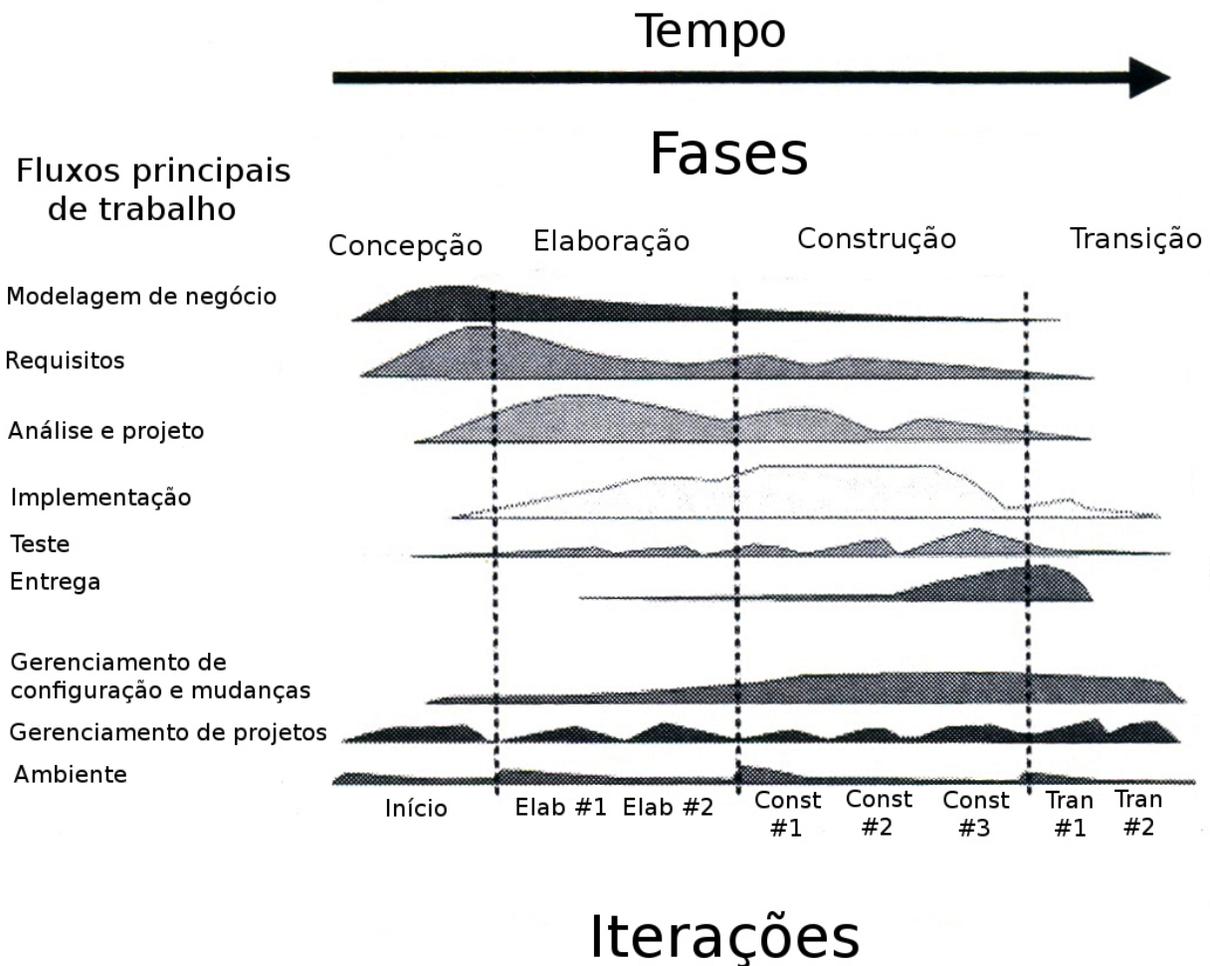


Fig. 6.10: Fluxos principais de trabalho do RUP

como a tomada de decisão é realizada em termos de processos cognitivos humanos, e como um sistema computacional pode ajudar na melhoria da inteligência dessa tomada de decisão.

Nesta seção, propõe-se uma metodologia de desenvolvimento de SAI que é um extensão do PU, adicionando a este arcabouço um fluxo principal de trabalho desenvolvido para lidar com as características peculiares dos SAI e garantir que os atributos necessários a eles estejam presentes no início da captura de requisitos em casos de uso.

O PU considera que a captura de requisitos é um processo difícil e dedica seu primeiro fluxo de trabalho a essa tarefa. Sua filosofia diz que usuários são uma fonte imperfeita de informações, que na maioria das vezes não conseguem enxergar o processo como um todo, entendendo apenas o seu único papel, e portanto não são capazes de entender como o processo de tomada de decisão poderia ser realizado de maneira mais eficiente. Além disso, prega que a abordagem tradicional, de alocar um analista que seria capaz de enxergar o processo como um todo e elicitar uma lista de requisitos de cada usuário, completa e perfeita, antes de começar o projeto, não funciona: segundo o PU, os requisitos são melhor compreendidos à medida que se constrói o software, e os próprios usuários não

entendem o que o sistema deve fazer até que ele esteja quase completo.

Quando os usuários finalmente entendem o que o sistema deve fazer, sugerem uma lista de mudanças que, apesar de benéficas, tem sérios impactos no esforço e no custo do projeto que se utiliza desta abordagem tradicional. Portanto, o PU prega que é um equívoco achar que os usuários sabem quais são os requisitos, e que só se faz necessário entrevistá-los para elicitá-los. É possível aprender com a interação com os usuário, mas antes disso o sistema precisa atingir o objetivo para o qual ele foi construído, ou seja, sua missão. Entender essa missão é uma tarefa difícil, e ainda mais complicada quando se trata de problemas não estruturados em nível operacional, onde se faz necessário planejamento e execução dinâmicos, e o ambiente muda constantemente. Por ser um processo construído com essa filosofia, e ser de fato um arcabouço capaz de ser estendido e especializado, o PU foi escolhido neste trabalho como parte da metodologia de desenvolvimento de SAI. A este processo de desenvolvimento, será adicionado apenas mais um fluxo principal de trabalho, focado nas características específicas dos SAI. Com isso, aproveita-se a experiência de mais de 30 anos embutida no PU, além de seus conceitos centralizados na arquitetura, e todos os seus benefícios trazidos com a reutilização de componentes, a iteratividade e gerenciamento de riscos, todas características desejáveis para um projeto de SAI.

O fluxo principal de trabalho adicionado ao PU é inserido antes dos fluxos principais de trabalho existentes, para tratar as peculiaridades dos SAI antes da captura dos requisitos em casos de uso, partindo das premissas corretas. Convencionou-se chamar esse fluxo principal de trabalho de *Modelagem Cognitiva*, para enfatizar a modelagem cognitiva orientada a semiótica realizada para identificar os pontos do processo de tomada de decisão onde deve haver amplificação de inteligência do usuário. Dessa maneira, dá-se origem ao *Processo Unificado Cognitivo* (PUC). A figura 6.11 apresenta um ciclo do PUC, com as quatro fases e seis fluxos principais de trabalho.

Com o fluxo principal Modelagem Cognitiva, o PUC possui uma captura de requisitos mais efetiva para os SAI do que o PU e outras metodologias existentes, pois este fluxo principal de trabalho estará focado nos atributos necessários aos SAI. Além disso, esse fluxo de trabalho irá dividir de maneira adequada a construção e evolução do SAI em um determinado número de ciclos, especificando o que deve ser implementado em cada ciclo.

O conjunto de requisitos que é desenvolvido pelo PU ficará, dessa maneira, acrescido do artefato “requisitos de amplificação de inteligência” no PUC, como pode ser visto na figura 6.12. Esses requisitos de amplificação de inteligência são requisitos de usuário funcionais e não funcionais, necessários para que o sistema implemente adequadamente os atributos necessários aos SAI, identificados na seção 5.5. Assim, o desenvolvimento do SAI parte das premissas corretas desde o início. A cada iteração esses requisitos serão refinados, mas as mudanças diminuem à medida que as fases passam no projeto, pois os requisitos tendem a se estabilizar.

No fluxo principal de trabalho de requisitos, os requisitos de amplificação de inteligência serão utilizados como artefato de entrada na captura de requisitos, influenciando a construção dos outros artefatos da figura 6.12, garantindo que os atributos necessários ao SAI serão levados em consideração depois na análise, projeto, implementação e testes.

Assim como acontece no PU, as iterações no PUC podem se sobrepor, como ilustra a figura 6.13, onde uma iteração pode começar enquanto a outra está terminando. No entanto, quando esta sobreposição ocorre, ela não é muito profunda, pois uma iteração é sempre a base para a próxima (Jacobson et al., 1999b).

Na seção 6.3.1, explica-se o fluxo principal de trabalho Modelagem Cognitiva. Os outros fluxos

Fluxos principais
de trabalho

Fases

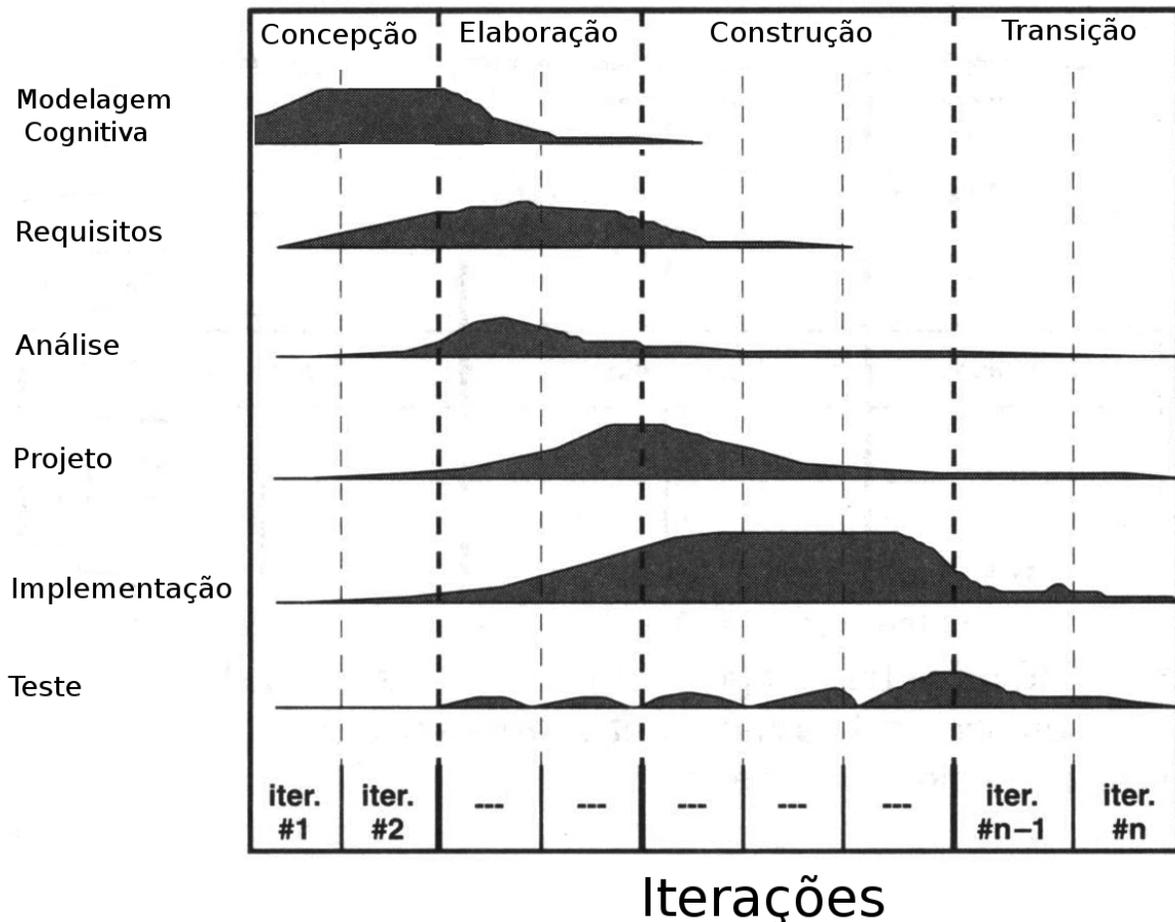


Fig. 6.11: Um ciclo do PUC, com os seis fluxos principais de trabalho - Modelagem Cognitiva, Requisitos, Análise, Projeto, Implementação e Teste - acontecendo ao longo das quatro fases: concepção, elaboração, construção e transição.

principais de trabalho do PUC foram descritos por Jacobson et al. (1999b). Como o PUC estende o PU, inserindo o fluxo principal Modelagem Cognitiva antes do fluxo principal Requisitos, esse último sofre influências daquele primeiro, e essa influência é detalhada na seção 6.3.2.

6.3.1 Modelagem Cognitiva

Introdução

Na Modelagem Cognitiva, estuda-se o fluxo cognitivo do agente humano no processo de tomada de decisão que ele realiza para resolver um determinado problema não estruturado. O propósito dessa análise do fluxo cognitivo é identificar as atividades aleatórias, mecânicas e inteligentes (seção 4.2.1

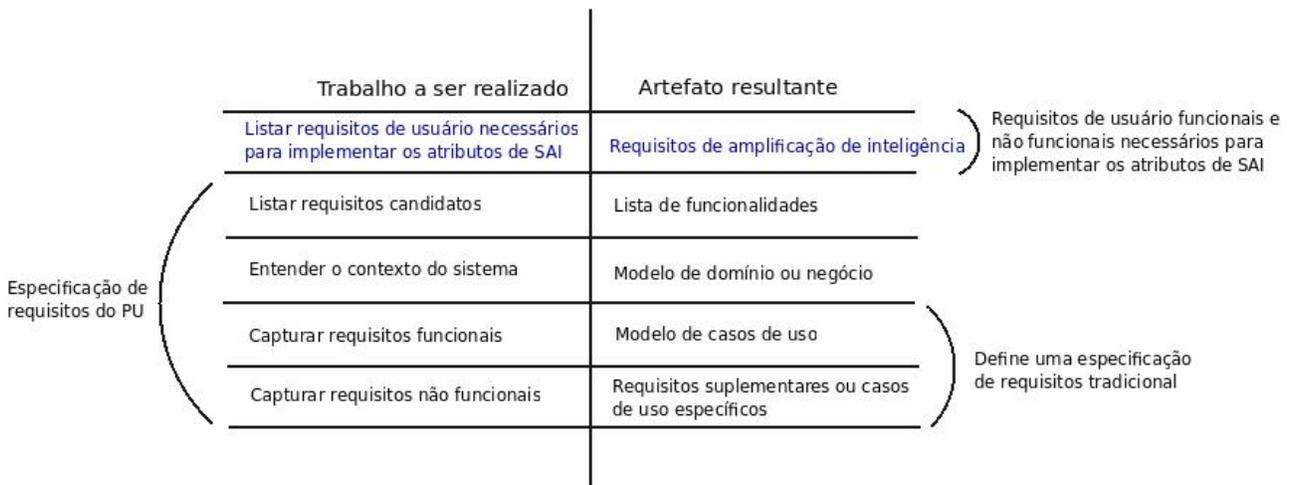


Fig. 6.12: O conjunto de requisitos fica acrescido dos requisitos de amplificação de inteligência no PUC, que procuram capturar os atributos necessários de um SAI.

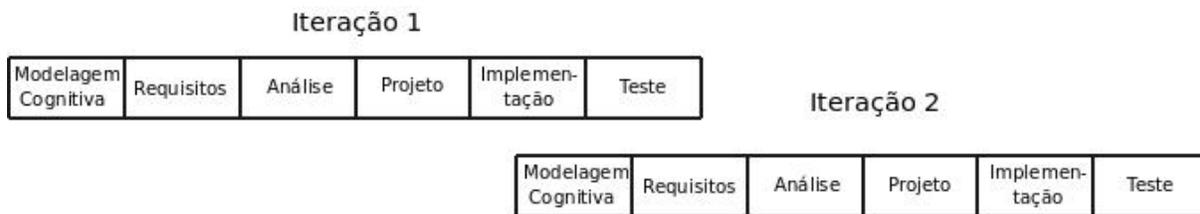


Fig. 6.13: Iterações do PUC se sobrepondo

desse fluxo, para que elas sejam corretamente distribuídas entre agente humano e computacional no SAI a ser construído, de modo a amplificar a inteligência do usuário na resolução do problema não estruturado de planejamento e execução dinâmicos. Para que essa amplificação ocorra de maneira adequada, todos os atributos necessários a um SAI (seção 5.5) devem ser identificados. Essas necessidades serão traduzidas em requisitos de amplificação de inteligência, que são requisitos de usuário funcionais e não funcionais relacionados aos casos de amplificação de inteligência que devem estar presentes no SAI (seção 5.1).

O objetivo do fluxo principal Modelagem Cognitiva é identificar o escopo da amplificação de inteligência do SAI a ser desenvolvido.

Essa análise sistemática do processo de tomada de decisão humana abre a caixa preta dos SAI através da utilização de técnicas da semiótica de Peirce e da semiótica computacional (capítulo 4). O Diagrama de Fluxo Cognitivo será utilizado para modelar o fluxo cognitivo da tomada de decisão humana e identificar os pontos de amplificação de inteligência, onde haverá diálogo entre agentes humanos e computacionais para construir a solução. Esse diagrama será simulado posteriormente em Redes Semiônica, para modelar a inserção do SAI no processo de tomada de decisão.

O fluxo Modelagem Cognitiva na fase de concepção divide a implementação dos requisitos de amplificação de inteligência em diferentes ciclos, por meio de uma estratégia de estimar resultados e

ganhos desses requisitos no processo de tomada de decisão.

A figura 6.14 ilustra o papel do fluxo Modelagem Cognitiva no ciclo de vida do Software, e como ele assume diferentes formas ao longo das diferentes fases e suas iterações:

- Durante a concepção é onde há mais Modelagem Cognitiva, capturando os requisitos de amplificação de inteligência para determinar o escopo da amplificação de inteligência do sistema. Todas as atividades do fluxo Modelagem Cognitiva são realizadas nessa fase, todo o fluxo cognitivo é modelado e todos os casos de amplificação de inteligência são identificados, para ser possível entender a missão do sistema em termos de amplificar o intelecto do agente humano. A implementação do sistema completo é dividido em um certo número de ciclos;
- Na elaboração, os modelos de inserção do SAI no fluxo cognitivo, os casos de amplificação de inteligência e o protótipo da interface de diálogo podem ser revisados para se adequarem a arquitetura definida para o sistema;
- Na construção, pouco é realizado do fluxo Modelagem Cognitiva, a não ser que haja mudanças no processo de tomada de decisão do problema não estruturado;
- Não se espera que atividades do fluxo Modelagem Cognitiva sejam realizadas durante a fase de transição.

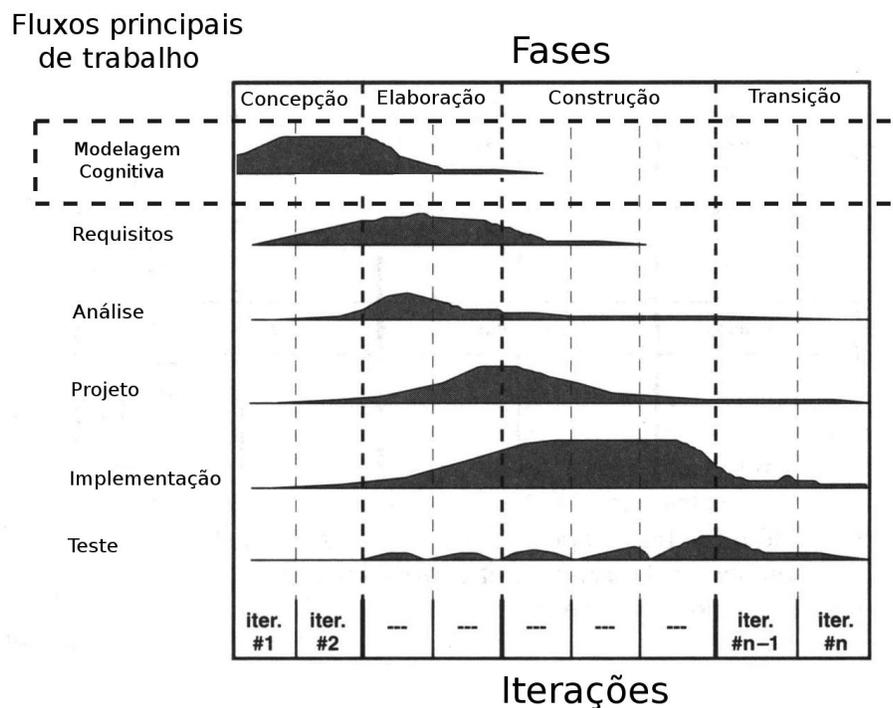


Fig. 6.14: Modelagem Cognitiva é realizada principalmente na concepção

A seguir, descreve-se o fluxo principal de trabalho Modelagem Cognitiva em três passos:

- Artefatos criados;

- Papéis que participam do fluxo de trabalho;
- O fluxo de trabalho.

Primeiramente, descreve-se os termos estáticos, artefatos e papéis, que estão apresentados na figura 6.15.

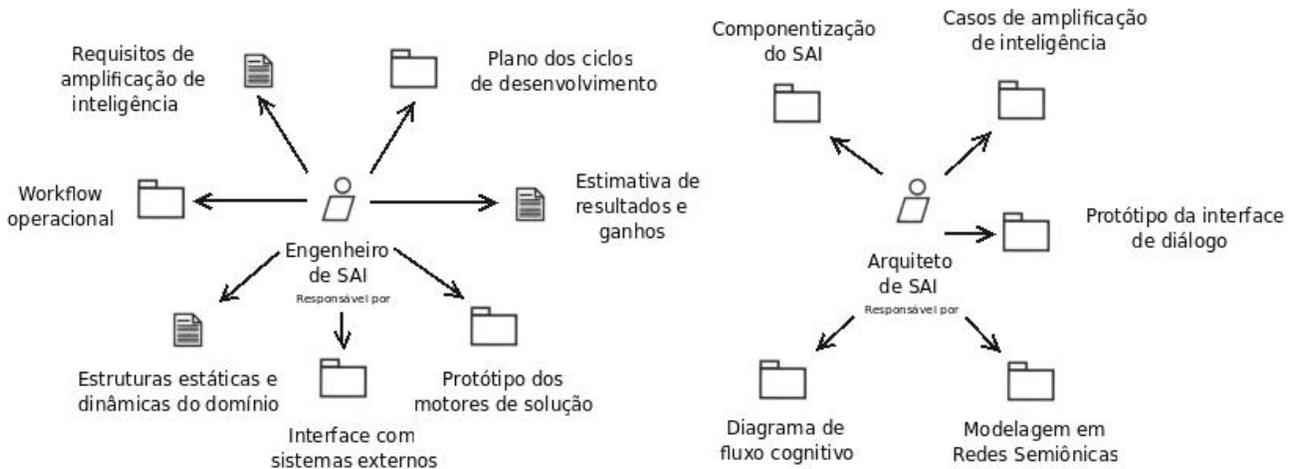


Fig. 6.15: Artefatos e papéis envolvidos no fluxo principal de trabalho Modelagem Cognitiva

Artefatos

Artefato é um termo utilizado para designar qualquer tipo de descrição ou informação criada, produzida, modificada ou utilizada por um papel (Jacobson et al., 1999b).

Os artefatos primários utilizados na Modelagem Cognitiva são os seguintes:

- **Workflow operacional:** diagrama de contexto que identifica as principais áreas que manipulam o fluxo da informação ao longo do processo de tomada de decisão. O objetivo desse levantamento é entender como o SAI se encaixaria no processo geral da tomada de decisão.
- **Estruturas estáticas e dinâmicas do domínio:** descrição das estruturas presentes no domínio do problema, dos instrumentos utilizados na tomada de decisão e do modo de operar e de interagir dessas estruturas.
- **Interface com sistemas externos:** modelo dos sistemas computacionais existentes no processo de tomada de decisão e que irão interagir com o SAI, identificando suas entradas e saídas.
- **Protótipo dos motores de solução:** protótipo conceitual dos algoritmos que serão carregados pelos agentes computacionais para sugerir uma solução para o problema não estruturado.
- **Diagrama de Fluxo Cognitivo:** modelo do fluxo cognitivo do agente humano que resolve o problema não estruturado, identificando as atividades cognitivas realizadas por ele e as unidades de conhecimento manipuladas. Este modelo terá as atividades cognitivas classificadas e os pontos de amplificação de inteligência, onde será modelado o diálogo entre agentes humanos e computacionais, identificados de maneira sistemática, conforme detalhado na seção 5.6.

- Modelagem em Redes Semiônicas: modelo de uma Rede Semiônica que traduz o DFC, validando-o, e depois modela a inserção do SAI no fluxo cognitivo, projetando o diálogo entre agentes computacionais e humanos.
- Casos de amplificação de inteligência: os casos de amplificação de inteligência representam os casos de interação entre agentes humanos e computacionais, onde há troca de informação entre eles em busca da solução. Estes casos são identificados nas Redes Semiônicas construídas nos superlugares para modelar o diálogo entre esses agentes, como foi detalhado na seção 5.7.
- Protótipo da interface de diálogo: protótipo conceitual da principal interface gráfica do sistema, utilizada para exosomatizar o pensamento dos agentes humano e computacional, representando suas soluções e simulando diferentes cenários.
- Componentização do SAI: proposta de divisão do SAI em componentes reutilizáveis, baseado na componentização padrão dos SAI e nos artefatos desenvolvidos, com as especificidades do problema não estruturado em questão.
- Estimativa de resultados e ganhos: estimativa baseada em ganhos, riscos e custos de implementação de cada um dos componentes reutilizáveis propostos para o SAI. Com essa estimativa, torna-se possível posteriormente dividir a implementação do SAI em ciclos, dando origem a projetos menores.
- Plano dos ciclos de desenvolvimento: proposta de divisão da criação do SAI em um determinado número de ciclos de desenvolvimento, sendo cada ciclo um projeto diferente. Essa divisão visa mitigar riscos com relação ao desenvolvimento de determinados componentes do SAI, e se baseará na estimativa de resultados e ganhos realizada.
- Requisitos de amplificação de inteligência: requisitos de usuário que capturam as funcionalidades e características que o sistema deve possuir para ser capaz de implementar adequadamente os atributos necessários aos SAI, identificados na seção 5.5.

O conjunto desses artefatos forma o Modelo Cognitivo, que é o produto final do fluxo principal de trabalho Modelagem Cognitiva, como ilustra a figura 6.16.

Papéis

Um papel representa uma posição que pode ser alocada a uma pessoa ou a um time de pessoas, especificando responsabilidades e habilidades necessárias (Jacobson et al., 1999b). Um papel não é idêntico a uma pessoa. Um indivíduo pode ser alocado para vários papéis em um projeto. Um papel tampouco é um cargo em uma empresa, mas sim uma abstração de um humano com habilidade necessária para realizar certas atividades do processo.

Os papéis responsáveis pelos artefatos da Modelagem Cognitiva são:

- Engenheiro de SAI: o engenheiro de SAI acumula atividades relacionadas com a solução que será fornecida pelo agente computacional, através dos cálculos automáticos e dos motores de solução que ele carrega. Dessa maneira, o engenheiro de SAI será responsável por estudar a estrutura do problema, para que as regras e objetivos possam ser simulados pelos motores de

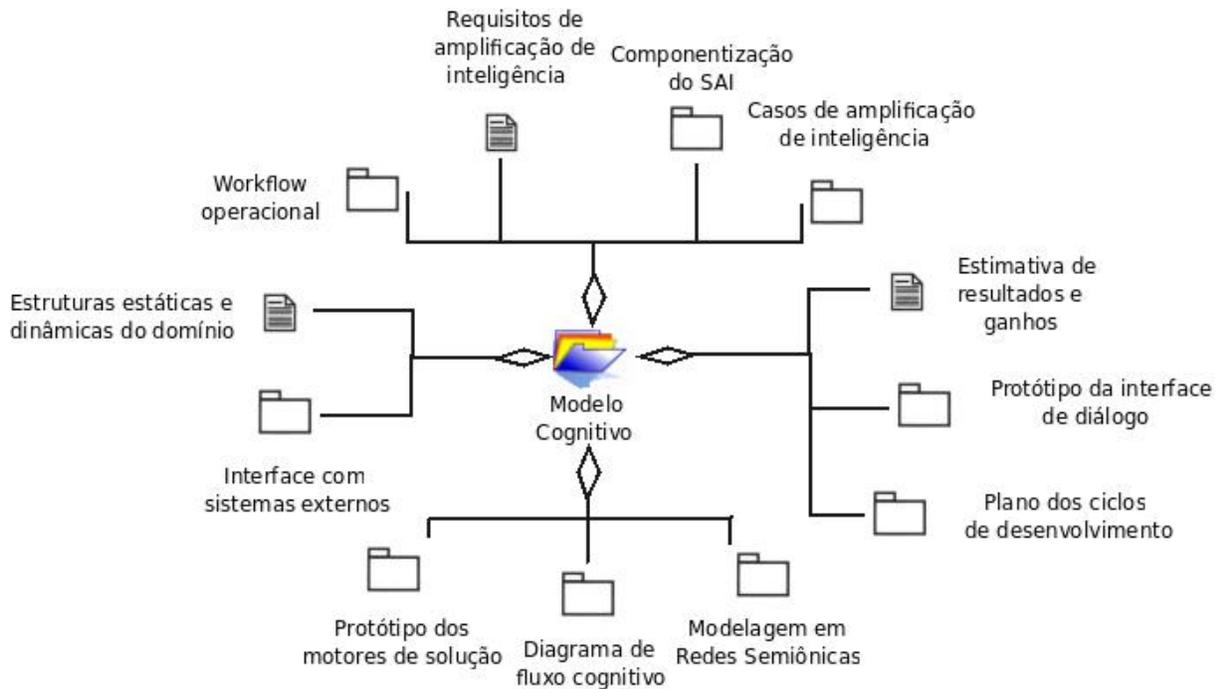


Fig. 6.16: O Modelo Cognitivo contém todos os artefatos produzidos no fluxo Modelagem Cognitiva

solução. Com efeito, ele será responsável por construir o protótipo do motor de solução, caso seja necessário, por desenvolver os requisitos de amplificação de inteligência e por dar suporte e evoluir o sistema ao longo de sua utilização.

- **Arquiteto de SAI:** o arquiteto de SAI acumula atividades relacionadas com a solução que será fornecida pelo agente humano, com a representação exosomática do conhecimento desse agente e com a interação entre os agentes humanos e computacional na construção da solução. Assim, o arquiteto de SAI é responsável por desenhar o Diagrama de Fluxo Cognitivo (que estuda os processos cognitivos do agente humano na construção da solução), modelar as interações que irão acontecer entre agentes computacionais e humanos em Redes Semiônicas e é responsável pelo protótipo da interface de diálogo. Posteriormente, o arquiteto de SAI unirá as informações dos artefatos produzidos pelo engenheiro de SAI e por ele para realizar a proposta de componentização do SAI.

Fluxo de trabalho

O fluxo de trabalho apresenta como os papéis colaboram, como o foco do trabalho muda de papel para papel e cada uma das atividades que estes realizam. As atividades representam a execução de uma operação de um papel no fluxo (Jacobson et al., 1999b).

O diagrama de atividades que apresenta o comportamento dinâmico da Modelagem Cognitiva está apresentado na figura 6.17.

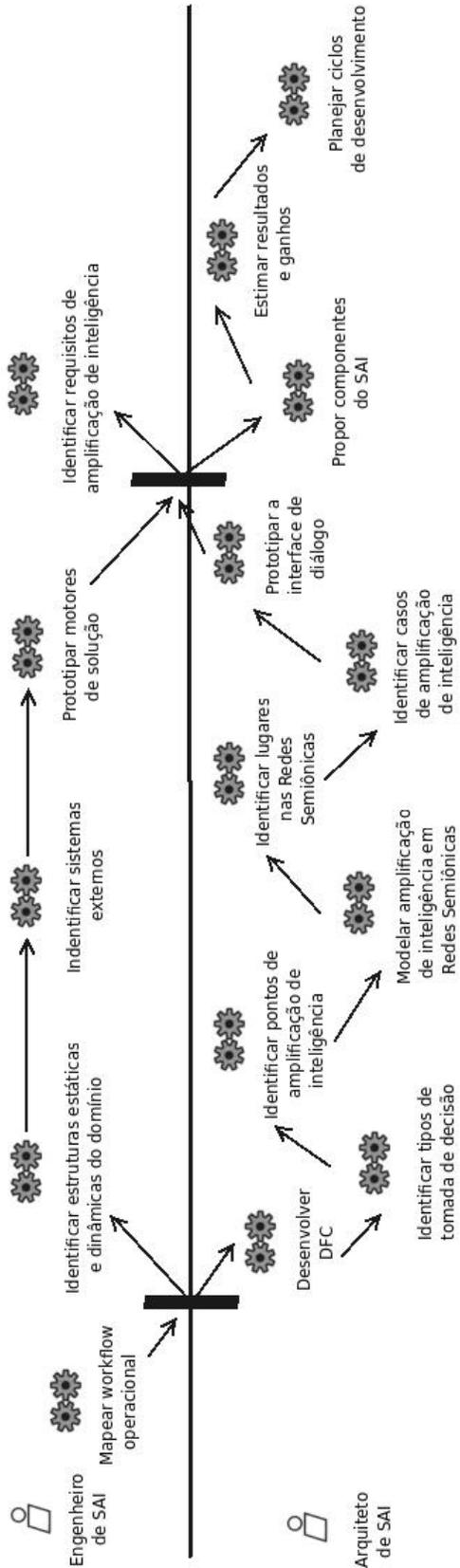


Fig. 6.17: Diagrama com as principais atividades da Modelagem Cognitiva

Apesar do paralelismo entre as atividades do Engenheiro e do Arquiteto de SAI, geralmente há uma interação e uma troca de informações entre eles, para que suas produções estejam coerentes.

As informações coletadas nas atividades da Modelagem Cognitiva são baseadas na técnica de etnografia, descrita na seção 6.1.1. Isso significa que as informações necessárias são adquiridas através da observação dos agentes humanos resolvendo o problema de planejamento e execução dinâmicos, durante o seu trabalho, *in loco*, numa visita técnica. Deve-se observar o maior número de pontos de vista (seção 6.1.1) possível nessa resolução. Ao invés de se utilizar de técnicas tradicionais de cenários (seção 6.1.1), a modelagem através do DFC e consequente Rede Semiônica trarão a possibilidade de analisar cenários de amplificação de inteligência.

Cada uma das atividades do fluxo da figura 6.17 é descrita em detalhes no apêndice A.2.

O fluxo Modelagem Cognitiva no PUC assume o mesmo papel da Modelagem de Negócio no RUP (figura 6.10), possuindo o mesmo objetivo de entender e melhorar a organização alvo do projeto e derivar requisitos para isso. No entanto, a Modelagem Cognitiva é especializada para os SAI, garantindo de forma sistemática que os requisitos derivados implementarão os atributos necessários a um SAI (seção 5.5).

6.3.2 Fluxo de iteração genérico do PUC

O fluxo de iteração genérico do PUC inclui os seis fluxos principais de trabalho: Modelagem Cognitiva, Requisitos, Análise, Projeto, Implementação e Teste. Ele é um padrão genérico utilizado como base para planejar as iterações concretas de um projeto. Dependendo da fase em que o projeto se encontra (concepção, elaboração, construção ou transição) o conteúdo da iteração muda de acordo com as metas daquela fase. Dessa maneira, o gerente de projeto pode escolher dentre as atividades do fluxo de iteração genérico para compor uma determinada iteração de uma determinada fase.

O fluxo de iteração genérico do PUC está apresentado na figura 6.18. Assim como na figura 6.8, os papéis do processo estão listados verticalmente na esquerda, identificando as raias, que delimitam as atividades de cada papel. O tempo avança da esquerda para a direita, caracterizando a ordem cronológica das tarefas. Na figura, os fluxos principais de trabalho circulam as atividades que eles realizam. A figura 6.18 foi cortada para facilitar sua visualização, e deve ser interpretada da seguinte maneira: para a direita do que foi apresentado, estão as outras atividades do fluxo de iteração genérico do PU, como na figura 6.8. Para a esquerda, estariam as outras atividades da Modelagem Cognitiva, apresentadas na figura 6.17. Portanto, esta figura 6.18 se concentrou na intersecção do fluxo principal de trabalho Modelagem Cognitiva com o fluxo principal Requisitos, delimitando quando as atividades da Modelagem Cognitiva terminam em relação ao começo das atividades do Requisitos, já que o tempo neste diagrama de atividades está da esquerda para a direita.

Em projetos de SAI, desenvolvidos para resolver problemas não estruturados de planejamento e execução dinâmicos, mais trabalho é realizado nas fases iniciais em relação às fases finais do projeto, pois o maior esforço é despendido para entender o escopo da amplificação de inteligência desse tipo de sistema. A concepção termina com a identificação dos requisitos de usuário

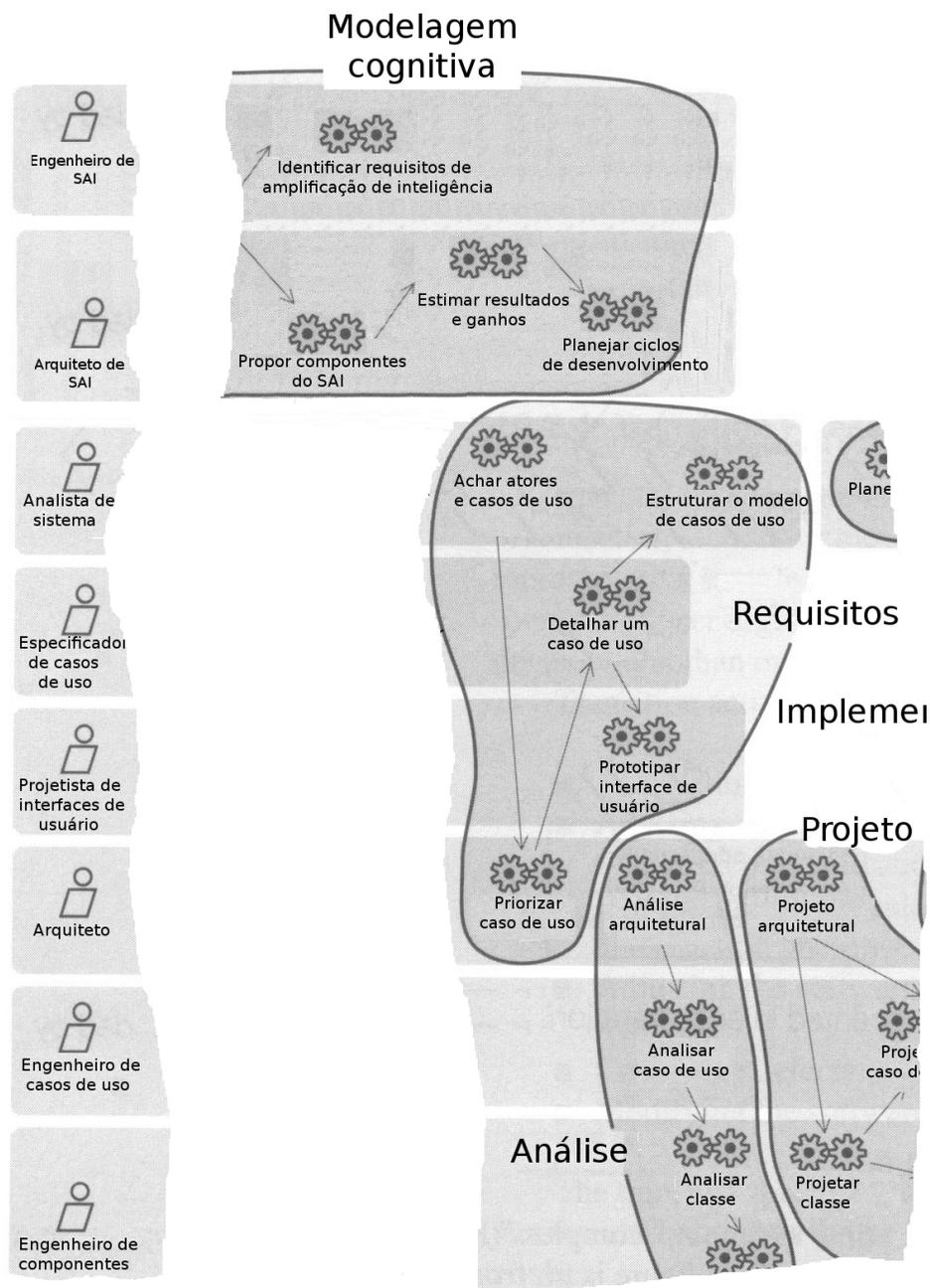


Fig. 6.18: Fluxo de iteração genérico do PUC

necessários para implementar os atributos de SAI. Em comparação à distribuição média de recursos e tempo entre as fases de um projeto médio, conforme a figura 6.9, um projeto de SAI com o PUC distribui recursos e tempo entre as fases conforme a figura 6.19.

Assim como o PU, o PUC pode ser estendido e especializado, preservando as mesmas qualidade de arcabouço que o PU possui. Isso permite ao PUC gozar dos benefícios do PU (seção 6.2) e garantir que os atributos necessários aos SAI estejam identificados em requisitos de usuário no

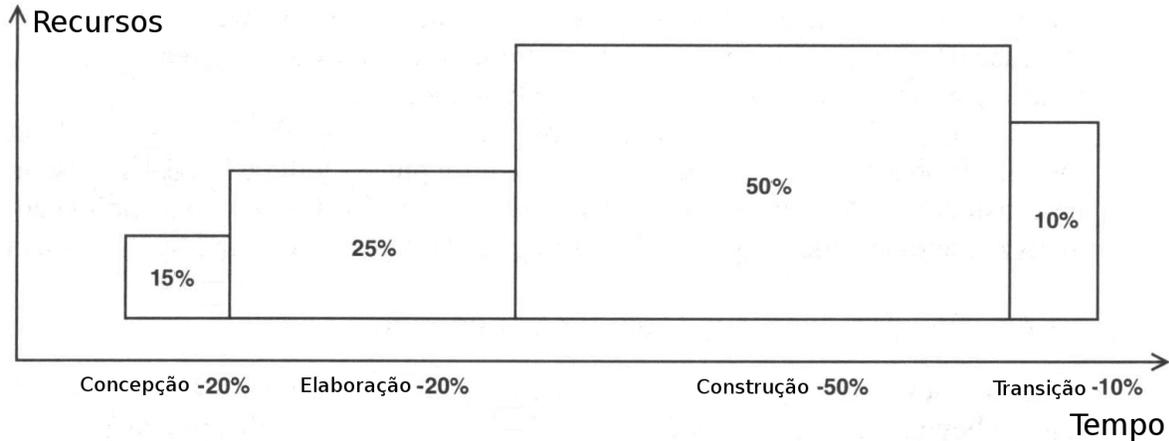


Fig. 6.19: Distribuição média de tempo e recurso entre as fases do PUC para projetos de SAI

início do fluxo principal Requisitos.

Da mesma maneira do PU, o PUC não descreve fluxos principais de trabalho gerenciais que lidam com riscos que são não técnicos, como gerência de projeto ou gerência de configuração, por exemplo, apenas se concentrando nas facetas de engenharia. No entanto, essas atividades gerenciais não técnicas são essenciais para o sucesso de um projeto de SAI. Elas apenas não são tratadas no escopo deste trabalho.

Como o fluxo principal Modelagem Cognitiva influencia o fluxo principal Requisitos

O PUC estende o PU inserindo o fluxo principal de trabalho Modelagem Cognitiva como o primeiro dos fluxos principais, antes do fluxo Requisitos. Isso é feito para garantir que a construção do SAI irá partir das premissas corretas, pois o fluxo Modelagem Cognitiva identifica requisitos de usuário necessários para implementar os atributos de SAI.

Como consequência dessa inserção, o fluxo Requisitos passará a sofrer influência do fluxo Modelagem Cognitiva, pois o resultado da Modelagem Cognitiva passa a ser uma entrada essencial para Requisitos. Pragmaticamente, essa influência se manifesta com os artefatos produzidos no fluxo Modelagem Cognitiva sendo utilizados como artefatos de entrada no fluxo Requisitos, influenciando suas atividades. Os primeiros artefatos produzidos no fluxo Requisitos serão realizados pelo próprio Engenheiro de SAI, tendo o Modelo Cognitivo como entrada, como ilustra a figura 6.20. Com efeito, todos os artefatos de Requisitos sofrerão influência do Modelo Cognitivo, principalmente o Modelo de Caso de Usos e os protótipos de interface de usuário.

6.3.3 Benefícios do fluxo principal Modelagem Cognitiva

O objetivo de estender o PU com o fluxo Modelagem Cognitiva para construir SAI é ter um fluxo principal de trabalho especializado, anterior a captura de requisitos, para garantir que os

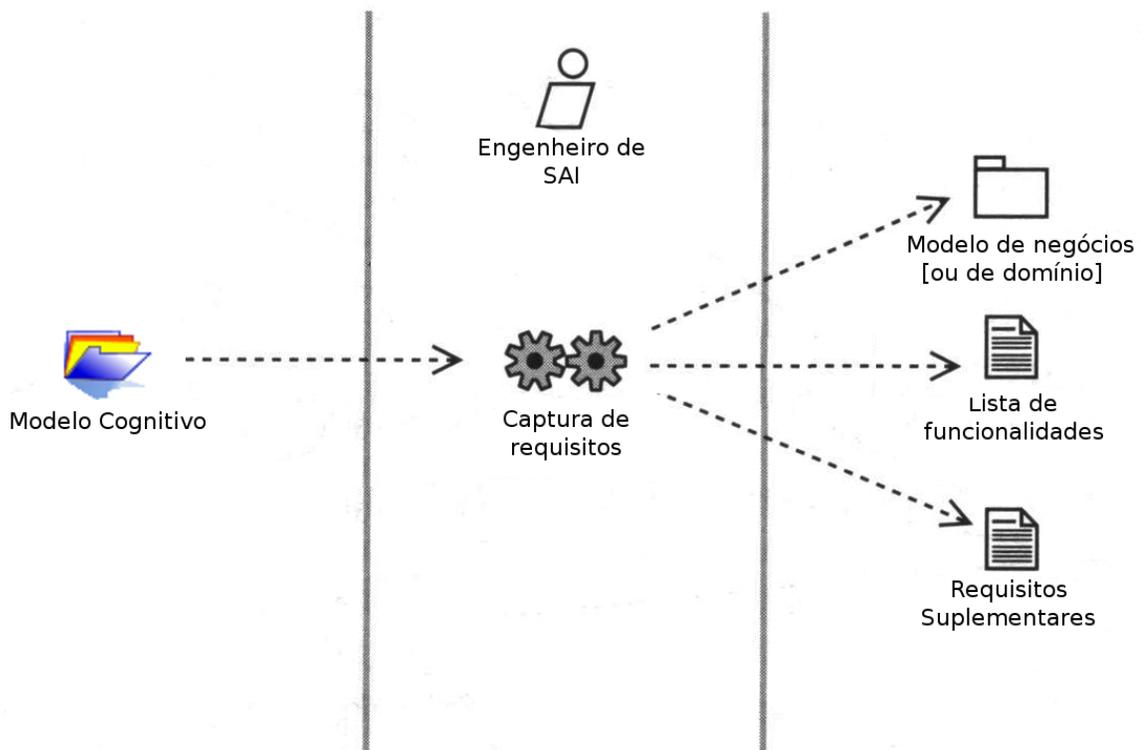


Fig. 6.20: O Modelo Cognitivo serve de artefato de entrada para a captura de requisitos.

atributos de SAI (seção 5.5) sejam implementados no sistema a ser concebido. O fluxo Modelagem Cognitiva está totalmente focado nesse objetivo, para garantir que o desenvolvimento do sistema irá partir das premissas corretas.

É possível construir um SAI somente com o PU, sem o fluxo Modelagem Cognitiva. No entanto, essa construção dependeria da competência dos profissionais que assumissem os papéis do fluxo principal de trabalho Requisitos para chegar nas premissas corretas. O fluxo Modelagem Cognitiva fornece uma maneira sistemática de chegar nas premissas, ou seja, elas estão garantidas como artefatos de saída das atividades.

Um sistema construído dessa maneira será efetivamente um SAI, que terá capacidade de incorporar os conhecimentos do agente humano usuário, que participará ativamente da construção da solução, cujas partes não estruturadas precisam ser resolvidas por ele, pois necessitam estar relacionadas com o cenário existente no mundo real.

Concluindo, o PUC é um processo de desenvolvimento que aborda os temas do PU, por ser uma extensão dele:

- Identificar os requisitos corretamente;
- Identificar a arquitetura corretamente;
- Utilizar componentes reutilizáveis;

- Pensar e se comunicar utilizando a UML;
- Iterar;
- Gerenciar riscos.

E, além disso, acrescido de um tema essencial para os SAI:

- Análise orientada a semiótica do processo de tomada de decisão: análise sistemática do processo de tomada de decisão humana para solucionar o problema não estruturado que será tratado pelo SAI, identificando nele as tarefas que são mecânicas e aleatórias, que serão atribuídas unicamente ao SAI, e as tarefas inteligentes, que serão executadas em conjunto pelo usuário e pelo SAI, através de um diálogo que amplia o poder computacional através da sinergia com a inteligência humana.

Essa análise pode inclusive mudar completamente a maneira como a tomada de decisão é realizada para torná-la mais eficiente. Pode-se também criar um processo de tomada de decisão, caso ele não exista, estruturando-o para que o usuário humano se concentre apenas nas atividades inteligentes necessárias a resolução do problema não estruturado.

6.4 Resumo

Este capítulo contribuiu sugerindo que o PUC representa um novo paradigma em termos de processo de desenvolvimento de sistemas que tem por objetivo amplificar o intelecto dos agentes humanos que resolvem um determinado problema utilizando-se de um repertório hierárquico de atividades cognitivas. Uma metodologia capaz de construir SAI é uma importante motivação para atacar aplicações do mundo real que representem problemas de planejamento e execução dinâmica, para trazer ganhos potenciais para diversos setores da economia nos níveis operacional, tático e estratégico.

Capítulo 7

Caso de estudo: desenvolvendo um sistema de amplificação de inteligência

Todas as aplicações práticas da metodologia “Processo Unificado Cognitivo” (PUC) se basearam em três anos de cooperação com a CFlex - Computação Flexível Aplicada LTDA¹, empresa que nasceu na Faculdade de Engenharia Elétrica e de Computação da Unicamp e foi incubada no projeto Softex². Durante o desenvolvimento da metodologia PUC, a CFlex já contava com mais de 9 anos de experiência no desenvolvimento de ferramentas de suporte à tomada de decisão, com experiência em atuação nos níveis operacional, tático e estratégico.

O caso de estudo apresentado neste capítulo trata da aplicação da metodologia PUC em uma empresa de mineração denominada Samarco, que apresentava um problema de P&ED relacionado à gestão do pátio de estocagem e embarque de seus produtos. A aplicação da metodologia PUC levou à especificação de requisitos de amplificação de inteligência para um sistema de gestão do pátio da Samarco, denominado MES Pátio. Este caso trata apenas da aplicação do fluxo principal Modelagem Cognitiva da metodologia PUC na primeira iteração da fase de concepção.

O capítulo está organizado da seguinte maneira:

- Seção 7.1: contextualização do exemplo apresentado, para especificação do sistema MES Pátio, apresentando um breve histórico e os objetivos do projeto;
- Seção 7.2: resultados da aplicação do fluxo principal de trabalho Modelagem Cognitiva da metodologia PUC na primeira iteração da fase de concepção, passo-a-passo, com os requisitos de amplificação de inteligência para o sistema proposto apresentados;
- Seção 7.3: apresentação de um sistema otimizante aplicado ao problema não estruturado da Samarco que falhou nos objetivos de resolução do problema, para contrastar com a concepção do SAI Mes Pátio;
- Seção 7.4: um resumo com as contribuições deste exemplo de aplicação da metodologia.

¹<http://www.cflex.com.br>

²<http://www.softex.br>

7.1 Introdução

7.1.1 Contextualização

A SAMARCO MINERAÇÃO S.A.³ é uma empresa de lavra, beneficiamento, pelotização e exportação de minério de ferro. Fundada em 1977, ocupa hoje a segunda posição no mercado transoceânico de pelotas, comercializando 100% de seus produtos para mais de 15 países na Europa, Ásia, África/Oriente Médio e Américas.

Com sede e escritório central em Belo Horizonte (MG), a Samarco mantém unidades industriais em dois estados brasileiros: Minas Gerais, nos municípios de Mariana e Ouro Preto, onde se localiza a Unidade de Germano, de mineração e beneficiamento; e Espírito Santo, no município de Anchieta, onde está a Unidade de Ponta Ubu, que compreende a pelotização e o porto. O transporte do concentrado de minério de ferro entre Germano e Ponta Ubu é feito por um mineroduto de 396 km de extensão, como na figura 7.1. A empresa também possui escritório na cidade de Vitória (ES) para operações de comércio exterior e câmbio, além de escritórios de vendas em Amsterdã e Hong Kong.



Fig. 7.1: Localização geográfica da Samarco. (Fonte: <http://www.samarco.com>)

Alguns importantes números relacionados a contingente de pessoas e vendas da Samarco estão resumidos na tabela 7.1.

Números Samarco

	Empregados	Pelotas*	Finos*	Total*	Vendas (M US\$)
2003	1286	13,7	2,7	16,4	500
2004	1320	14,1	2,2	16,2	641
2005	1325	14,1	1,4	15,5	1100
2006	1381	14,1	1,9	16	1170
2007	1762	12,8	1,7	14,5	1100

* Vendas (MT)

Tab. 7.1: Números da Samarco. (Fonte: <http://www.samarco.com>)

Na região central da figura 7.2, que representa a planta da Samarco, pode-se identificar os pátios de estocagem de produtos. Um pouco para a direita, identifica-se as estruturas da usina

³As informações específicas de localização geográfica e números de produção da Samarco apresentadas nesta seção 7.1.1 foram retiradas do site da empresa na internet - <http://www.samarco.com>

de pelotização, processo que transforma o minério-de-ferro em pequenas esferas (pelotas) com aditivos químicos, um produto pronto para ser usado como alimento para o alto forno de uma siderúrgica, por exemplo. No canto inferior esquerdo estão os dois berços do porto e a carregadora de navios, que carrega os navios com os produtos que vem do pátio através de uma esteira articulada, percorrendo alguns quilômetros.



Fig. 7.2: Planta da Samarco

7.1.2 Objetivos

A Samarco necessitava de um sistema de suporte à tomada de decisão para o pátio de estocagem e embarque de seus produtos. Havia algumas peculiaridades no que diz respeito a modelagem e a incerteza da informação. A característica marcante do problema estava no fato da pelotização ser um processo contínuo, aliado à forma como a pelota se comporta durante o processo de empilhamento. Isto faz com que haja uma imprecisão na qualidade do produto, causando problemas de previsibilidade e rastreabilidade.

Pode-se dizer que o principal objetivo do sistema a ser especificado era aumentar a previsibilidade e rastreabilidade dos produtos do pátio.

7.1.3 O problema em poucas palavras

Os contratos de vendas de pelotas de minério de ferro fechados entre a Samarco e seus clientes se baseiam na qualidade do produto. A qualidade das pelotas é medida por meio de sua composição física, química e metalúrgica. Esses principais itens de controle físicos, químicos e metalúrgicos são apresentados com seus valores típicos na figura 7.3.

Análise Química (% Dry Basis)		
Item de controle	Procedimento	Típico
Fe	(ISO 2597)	66.62
FeO	(K ₂ Cr ₂ O ₇ titrant)	0.29
SiO ₂	(ISO 11535)	2.1
Al ₂ O ₃	(ISO 11535)	0.50
CaO	(ISO 11535)	1.71
MgO	(ISO 11535)	0.18
P	(ISO 4687)	0.046
S	(ISO 9686)	0.004
Cu	(ISO 11535)	0.003
TiO ₂	(ISO 11535)	0.042
V ₂ O ₅	(ISO 11535)	0.007
Na ₂ O	(ISO 11535)	0.035
K ₂ O	(ISO 11535)	0.011
Mn	(ISO 11535)	0.049
Zn	(ISO 11535)	0.002
B ₂		0.80
H ₂ O	(ISO 3087)	1.7

Análise Física (Natural Basis)	
Item de controle	Típico
CCS (%)	
-16+12.5 Kg/P	291
Tumble (%)	
+6.3 mm	93.5
Abrasion (%)	
-0.5 mm	5.7
Size Distribution (%)	
-19+16 mm	7
-16+8 mm	93.0
-6.3 mm	1.4

Análise Metalúrgica	
Item de controle	Típico
L.T.D (ISO 13930) (%)	
+6.3mm (%)	85
-0.5mm (%)	11
Swelling Index (ISO 4698) (%)	15
Delta P (ISO 7992) (%)	11
dR/dt (ISO 7992) (%)	1.6
Relative Reducibility (ISO 7215) (%)	82

Fig. 7.3: Itens de controle. (Fonte: <http://www.samarco.com>)

Dessa maneira, a Samarco tem a capacidade de modificar a qualidade da pelota que produz de acordo com as necessidades do cliente.

Para controlar a produção de pelotas de qualidades diferentes, existem campanhas de produção para diferentes clientes. Quando as usinas da Samarco iniciam uma determinada campanha, que costuma durar em torno de uma semana, elas passam a produzir pelotas com a qualidade especificada naquela campanha, que está direcionada para o objetivo de atender a determinado cliente.

Como a produção nas usinas não pára, e os navios dos clientes chegam para ser embarcados em intervalos de dias, o material que sai das usinas precisa ser estocado nos pátios de pelotas, para mais tarde, quando o navio do cliente atracar, ser recuperado do pátio e embarcado no navio.

No entanto, características inerentes ao processo de produção das usinas faz com que a qualidade das pelotas sofra uma alteração em torno da meta da campanha a cada duas horas. Essa

variação bi-horária da qualidade não pode ser modelada, ou seja, não há como prever as variações da qualidade em torno da meta da campanha.

Dessa maneira, a qualidade das pelotas que são empilhadas no pátio muda a cada duas horas. Na hora de recuperar a pelota para embarcar no navio, será necessário saber onde foi empilhado cada qualidade bi-horária produzida. Este processo está ilustrado na figura 7.4. A forma como as pelotas se organizam na pilha não está reproduzido fielmente nesta figura, que apenas procura mostrar o conceito das qualidades diferentes empilhadas.

Qualidade do material empilhado muda a cada duas horas
Necessário saber a qualidade a cada baliza

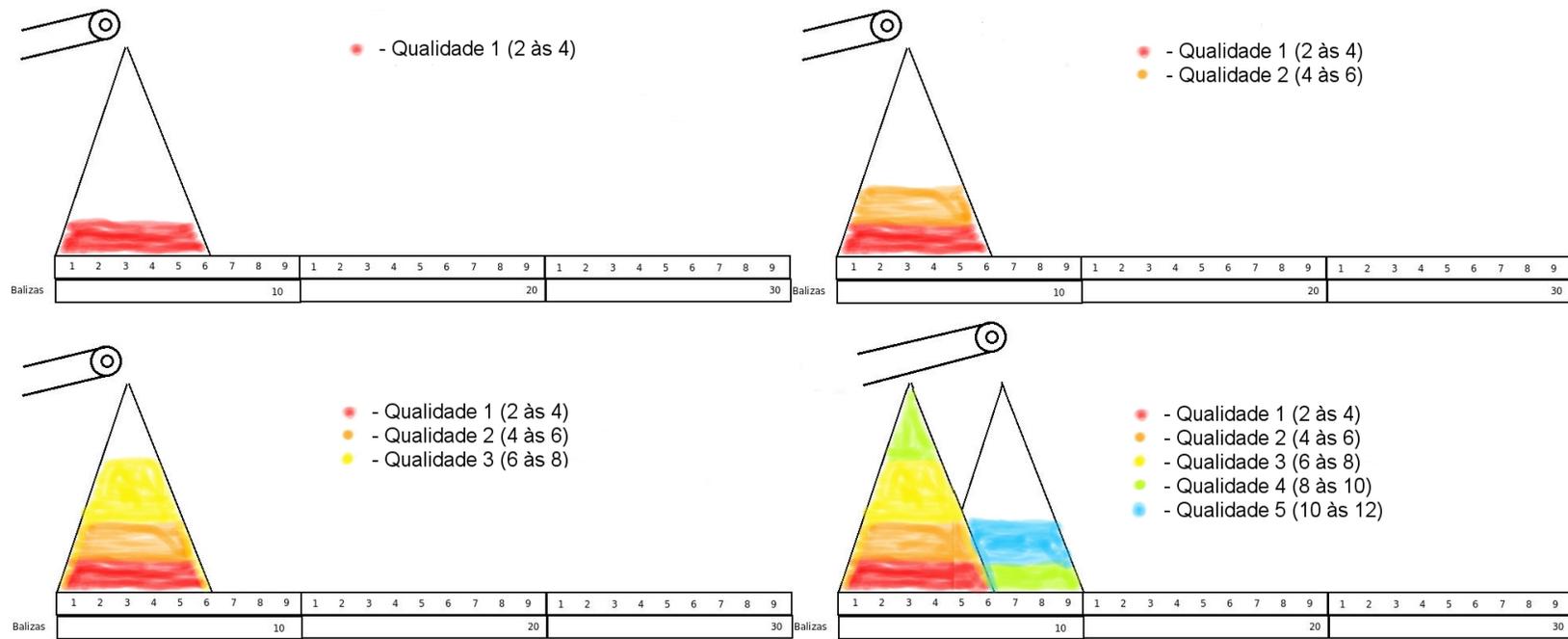


Fig. 7.4: Empilhamento de pelotas de qualidades bi-horárias diferentes. A forma como as pelotas se organizam na pilha não está reproduzido fielmente.

Quando o navio do cliente atraca, ele irá cobrar que a qualidade das pelotas embarcada seja compatível com a qualidade do contrato. Como saber se o produto da pilha do cliente possui a qualidade necessária? Isso é realizado através do casamento das medidas bi-horárias de qualidade de amostras do produto que sai das usinas com o posicionamento das máquinas. Se é possível saber em que baliza a empilhadeira estava às 2 da manhã, sabe-se que naquela baliza está empilhado o material com a qualidade da bi-horária de 2 às 4 da manhã, conforme medida na saída da usina.

Assim sendo, para montar o plano de recuperação e realizar o embarque, os engenheiros da Samarco precisam saber a qualidade média por baliza, através do casamento dessas informações de bi-horárias e posicionamento, para que seja possível saber de quais balizas deve-se recuperar para embarcar um produto de qualidades adequadas. Se em um determinado período a usina produziu um item de controle abaixo da meta, a baliza que corresponde àquele material está abaixo da meta, como ilustra a figura 7.5.

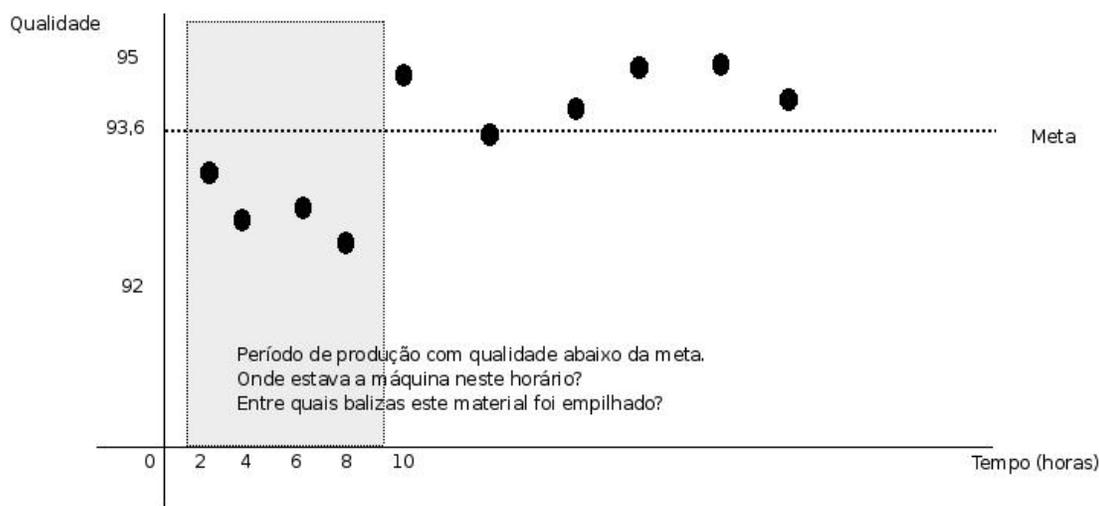


Fig. 7.5: Gráfico ilustrativo de qualidade x tempo(horas) de um possível item de controle.

Ocorre que realizar este cálculo para todos os itens de controle, para todos os produtos em cada baliza considerando todas as bi-horárias é muito custoso. Geralmente, o engenheiro não consegue fazer todos esses cálculos a tempo de tomar a decisão necessária para o embarque.

Além disso, existem outros problemas. Ao passar um determinado tempo no pátio, as pelotas sofrem um processo de degradação. Isso significa que, ao ser recuperada, a pelota não terá mais a mesma qualidade de quando foi empilhada no pátio, por conta da degradação. A umidade também é outro problema: a Samarco precisa lançar água nas pelotas ao longo da produção e no pátio por questões ambientais, para evitar particulado no ar da região. As chuvas também aumentam a umidade das pelotas que ficam empilhadas. Umidade é um dos itens de controle, como pode ser observado na figura 7.3.

Para resolver este problema, os engenheiros da Samarco precisam realizar blendagens⁴ no plano

⁴Blendagem ou Blend é o jargão utilizado na Samarco para denotar a mistura dos produtos para atingir uma qualidade média desejada.

de recuperação, ou seja, misturar pelotas de pilhas diferentes para atingir a qualidade média necessária para atender o cliente.

Faz-se necessário, portanto, tomar a decisão de quais balizas serão recuperadas para atingir a qualidade média necessária, levando em consideração as incertezas e imprecisões do processo, além de aspectos políticos e econômicos, como fornecer pelotas melhores para clientes mais importantes. Para ser capaz de conceber tal plano, faz-se necessário intervenções humanas em decisões otimizadas.

7.2 MES Pátio - Samarco

Nesta seção, serão apresentados os resultados da aplicação do fluxo principal de trabalho Modelagem Cognitiva da metodologia PUC na primeira iteração da fase de concepção para desenvolver um sistema de gestão do pátio de estocagem e embarque da Samarco. Para uma melhor compreensão da metodologia, o exemplo está separado por tópicos idênticos aos passos da Modelagem Cognitiva, apresentados na seção 6.3.

O trabalho dos operadores na sala de controle do processo de estocagem e embarque foi acompanhado, e uma série de reuniões foram conduzidas com os engenheiros de processo, equipes de manutenção, automação e projeto da Samarco. Essas atividades foram realizadas de forma a compreender o problema e levantar as informações necessárias e suficientes para a criação de um sistema de amplificação de inteligência. As próximas seções apresentarão os resultados dessa aplicação.

7.2.1 Mapear workflow operacional

O workflow apresentado nesta seção refere-se a todos os principais processos. A descrição do workflow será feita de forma bastante resumida, apenas para o entendimento de como um sistema de amplificação de inteligência para as operações no pátio se encaixaria no processo geral.

A figura 7.6 apresenta o workflow resumido das principais operações.

Tudo começa com o plano de vendas anual, onde a equipe comercial da Samarco define as quantidades de produtos que serão vendidas em uma base mensal, definindo também os clientes. A partir daí, é gerada a fila de navios que levará o produto vendido até os clientes. Cada navio tem uma data prevista de chegada, assim como as cargas (produto e quantidade) a serem embarcadas.

Para atender a esta fila de navios, é necessário que o produto esteja disponível no pátio da Estocagem e Embarque no momento da atracação dos navios. Ao chegar, o material vindo da mina deve passar pelo processo de filtragem e pelotização, e em seguida ser armazenado nas pilhas do pátio de estocagem. Para produzir as pelotas, é necessário um planejamento indicando os tipos de pelotas e as quantidades a serem produzidas. Esse planejamento é realizado periodicamente através de reuniões de “Mudança de Ata de Campanha”, definindo que as usinas de pelotização

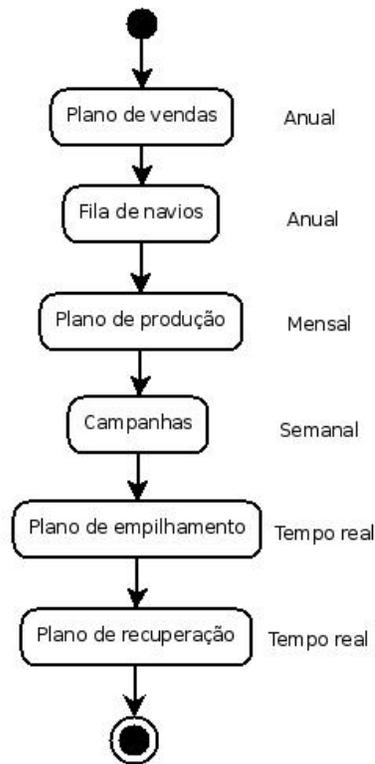


Fig. 7.6: Workflow de operação da Samarco

ficarão dedicadas à produção de um determinado produto por um período de tempo. As campanhas, elaboradas nas reuniões de Atas de Campanhas, devem levar em conta também o estoque dos produtos já armazenados no pátio, evitando, assim, a geração de produtos excedentes.

Para que seja possível produzir a pelota na usina, é necessário matéria-prima proveniente das minas. O planejamento dos tipos de material que serão enviados até as usinas pelo mineroduto, assim como as quantidades necessárias, é definido através do plano de produção da mina. É importante observar que a interação entre estes diversos processos é dinâmica e pode ser alterada caso necessário. Assim, por exemplo, se houver impossibilidade de produzir uma quantidade específica de produto na mina, para atender a um embarque, é possível que a fila de navios seja alterada de forma a encaixar as produções possíveis.

Continuando com o processo, após a produção das pelotas na usina, é necessário um plano indicando em quais localizações do pátio o produto será empilhado. Isso é feito através do plano de empilhamento e recuperação, indicando as balizas (marcos de comprimento no pátio) e quantidades empilhadas. Esse plano inicialmente é feito pelos Engenheiros de Processo da Produção (em concordância com os Engenheiros de Processo da Estocagem e Embarque), mas pode ser alterado pelo operador devido a ocorrências e imprevistos que tornam impossível a realização do plano inicial.

Finalmente, quando o navio atraca no píer e está pronto para carregar o produto vendido ao cliente, é necessário um plano indicando de quais pilhas, e quais quantidades, será retirado o produto. Para isso, é criado o “plano de empilhamento e recuperação”, que deve levar em conta

não só os tipos de produtos e as quantidades vendidas, mas também as medidas de qualidades presentes nas pilhas, de forma a atender as especificações de qualidade exigidas pelos clientes. Conforme citado no início desta seção, a descrição do workflow foi feita de forma bastante resumida, apenas para facilitar o entendimento do problema, já que o escopo do sistema de amplificação de inteligência refere-se apenas às operações da Estocagem e Embarque e, desta forma, recebe como entradas já definidas a fila de navios, as campanhas (para planejamento de operações futuras) e as produções sendo geradas nas usinas.

7.2.2 Identificar estruturas estáticas e dinâmicas do domínio

Descrição do cenário de julho de 2007

Esta seção descreve o cenário de julho de 2007 presente na Estocagem e Embarque da Samarco. Neste cenário, existem duas usinas de pelotização e três pátios de empilhamento. A descrição será feita da seguinte forma: primeiro, são apresentados os principais elementos do pátio como, por exemplo, as máquinas, pátios de empilhamento, correias, etc. Depois disso, é descrito o modo de operação do cenário atual, ou seja, a forma como os diversos elementos do pátio interagem para que o produto flua desde a usina até o embarque dos navios.

A figura 7.7 apresenta um diagrama resumido dos elementos do pátio relevantes para a implementação de um sistema de amplificação de inteligência.

A seguir, são descritos os principais elementos presentes no pátio no cenário atual:

- Usinas I e II: o cenário conta com duas usinas de pelotização responsáveis por atender às campanhas de produção de pelotas. Todo o processo se inicia nas Minas em Mariana/MG, onde é retirado e enviado o ROM (*run of mine*) ao processo de beneficiamento, sendo concentrado, ou seja, aumento do teor de Fe. O produto concentrado é então bombeado em forma de polpa pelo mineroduto até a Unidade de Ubu, em Anchieta/ES. Basicamente são obtidos três tipos de produtos:

1. Pelota: o “pellet feed”⁵ produzido pela filtragem passa pelo processo de pelotização e as pelotas são empilhadas no pátio ou enviadas diretamente para o embarque.

Em suma, o produto que é recebido em Ubu, através do mineroduto (concentrado bombeado recebido em Ubu), passa por alguns processos antes de ser pelotado, como pode ser observado na figura 7.8.

O concentrado chega via torre de recebimento (torre gravimétrica) e desta pode seguir dois caminhos: ou passa pelo processo de espessamento ou é recebido diretamente nos tanques de homogeneização, passando posteriormente para o processo de filtragem. O produto, então com a umidade reduzida para teores em torno de 10,0%, passa para o processo de prensagem num equipamento denominado roller press, que tem a finalidade de aumentar a superfície específica do pellet feed (produto da Preparação).

A partir daí, segue para o processo de pelotamento, passando antes pelas linhas de mistura onde recebe insumos: calcário, carvão e aglomerantes, e após este processo,

⁵Jargão da indústria utilizado para se referir ao material utilizado como insumo no processo de pelotização.

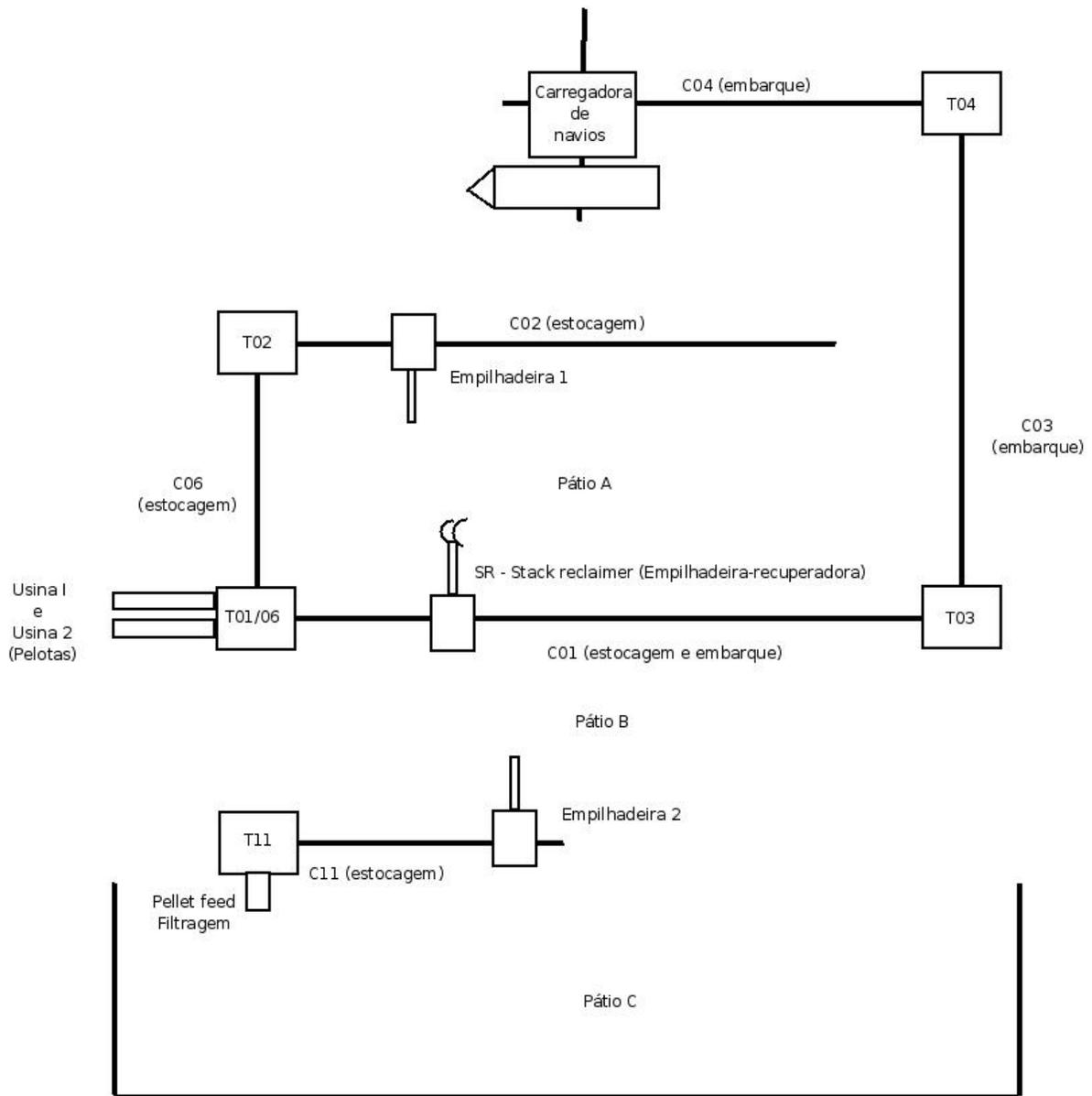


Fig. 7.7: Diagrama do cenário de julho de 2007 do pátio da Samarco

as pelotas cruas são classificadas na saída dos discos e, antes de serem alimentadas no forno de endurecimento, passam através de mesas classificadoras (mesas de rolos). Após o processo de endurecimento, todo o produto passa por classificação granulométrica (peneiramento), onde são gerados o produto final (pelotas) e o “Pellet Screening”, que é o rejeito do processo de peneiramento (ou “Sinter”, como é chamado na Samarco).

2. Pellet feed: quando a usina não consegue processar todo o produto produzido pela filtragem, este excedente é empilhado no pátio. A quantidade de pellet feed excedente produzido pela filtragem é medida na saída deste processo a cada duas horas.

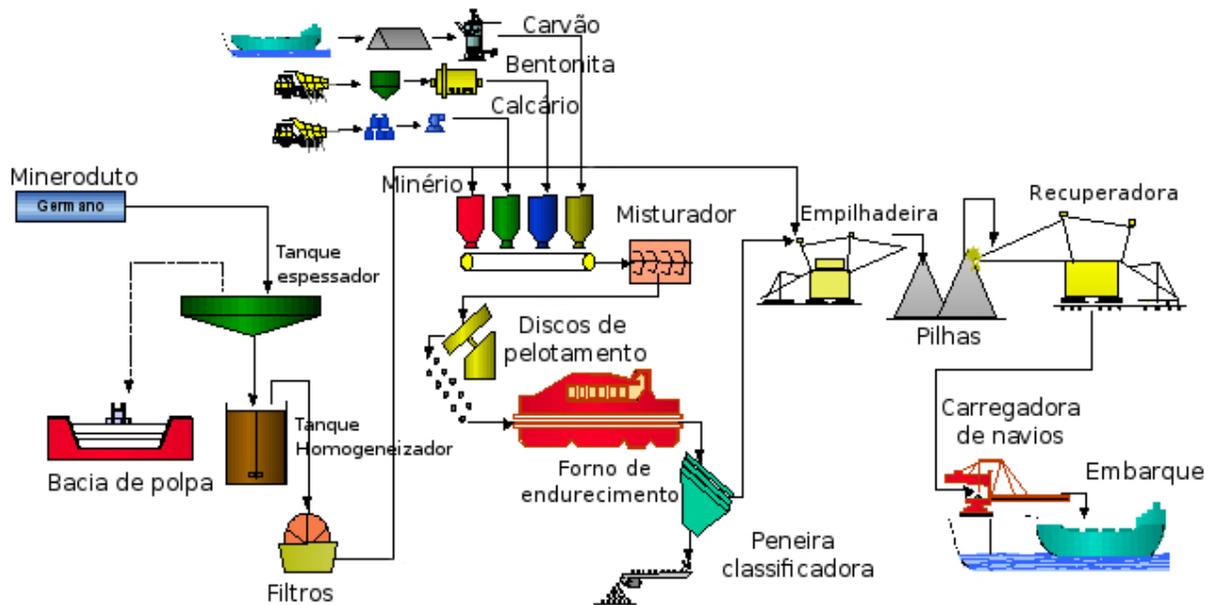


Fig. 7.8: Processos na usina

3. Sinter feed: durante o processo de pelotização, as pelotas produzidas passam por um processo de classificação granulométrica (peneiramento) na saída do forno de endurecimento. O rejeito do peneiramento é armazenado no pátio e vendido como sinter feed. Assim como no caso do pellet feed, a quantidade de sinter feed também é medida a cada duas horas. Uma boa aproximação para a taxa bi-horária de produção do sinter feed é dada pela produção bi-horária multiplicada pelo fator 0,025. Em termos da qualidade do produto produzido, é importante observar que as medidas de qualidade para as duas usinas podem ser diferentes em um determinado momento.

- Pátios de empilhamento: o cenário conta com três pátios para o armazenamento das pilhas no pátio: pátios A, B e C. Originalmente, os pátios A e B contavam com 130 balizas de 10 metros cada uma para o armazenamento do produto, mas devido às interferências resultantes das obras para instalação da terceira usina, a região utilizada para empilhamento atualmente para os dois pátios está confinada entre as balizas 25 e 130. O pátio C tem balizamento de 25 até 58.
- Torres e correias: conforme é possível observar no diagrama da figura 7.7, as correias são os elementos responsáveis pelo transporte do produto pelo pátio enquanto as torres servem como pontos de distribuição do produto e junção das correias.

O sistema atual é composto por seis torres principais:

- * Torres T01/T06: recebem os produtos das usinas I e II através das correias que saem das usinas. O produto pode então ser transferido pela correia de empilhamento C6 para a torre T02 ou pela correia de empilhamento/recuperação C1 para a SR (stack reclaimer). A amostragem bi-horária da qualidade do produto produzido nas usinas é realizada nas torres T01 e T06;

- * Torre T02: recebe o produto pela correia C6 e envia pela correia C2 para a empilhadeira I, para ser armazenado no pátio A;
 - * Torre T03: recebe o produto que será carregado no navio por duas possíveis fontes. A primeira é o produto recuperado dos pátios A ou B pela SR e enviado para a torre T03 através da correia C1. A segunda é o produto vindo diretamente das usinas, passando pelas torres T01 e T06 e caindo na correia C1. É possível haver combinações entre essas duas fontes, misturando o produto recuperado com o produto direto das usinas, podendo ainda esta variação ocorrer com uma ou outra usina independentemente. Nesta torre, são realizadas as medidas de qualidade do produto sendo embarcado;
 - * Torre T04: recebe o produto vindo da torre T03 pela correia C3 e envia para a correia C4 para a carregadora de navios;
 - * Torre T0: recebe pellet feed excedente da filtragem através da correia FCS1A e envia para a empilhadeira II através da correia C11 para ser empilhado na região inicial do pátio B ou no pátio C. O produto empilhado no pátio B é recuperado pela SR para os navios ou pode ser realimentado para o roller press. O produto empilhado no pátio C também é utilizado para realimentar o processo, contribuindo assim para aumentar a performance da produção ou compensando a produtividade.
- Pilhas: os produtos são armazenados no pátio em pilhas. Cada pilha é identificada e geralmente formada para atender um determinado cliente. O espaço ocupado por cada pilha no pátio é dado por suas balizas inicial e final. No pátio de produtos da Samarco, é desejável que cada pilha contenha apenas uma identidade e tipo de produto, excetuando-se residual da mesma família, que é considerado qualitativamente ponderando-se pela quantidade.
- Produtos: atualmente, a Samarco trabalha com seis produtos de pelotas, agrupados em duas famílias diferentes: BF (alto forno) e DR (redução direta). Estas duas famílias são resultantes dos dois tipos de pellet feed recebidos como produtos nas usinas: CNS (concentrado sílica normal) e CLS (concentrado baixa sílica). O produto CNS resulta na pelota BF e o produto CLS resulta na pelota DR. As duas categorias de pelota BF e DR são divididas em três produtos diferentes cada. Estes produtos são resultante de alterações nos aditivos acrescentados ao produto inicial de forma a proporcionar uma gama maior de produtos disponibilizados para os clientes. A tabela 7.2 resume a distribuição dos produtos de acordo com os insumos.

Insumo	Família de pelota	Produto
CNS (concentrado sílica normal)	BF (alto forno)	STD (padrão)
		HB (alta basicidade)
		MB45 (média basicidade)
CLS (concentrado baixa sílica)	DR (redução direta)	HY (hylsa)
		MG (manganês)
		MX (midrex)

Insumo	Família de pelota	Produto
--------	-------------------	---------

Tab. 7.2: Produtos

Além dos produtos de pelota, são comercializados também dois tipos de pellet feed: PFN (sílica normal) cujo insumo é o CNS e PFL (sílica baixa) cujo insumo é o CLS.

O pellet screening, resultante do peneiramento da pelota, também é comercializado como produto sinter feed (SF).

- Máquinas: a seguir, são descritas as empilhadeiras e a recuperadora presentes no pátio. Obviamente, existem outros equipamentos no pátio como, por exemplo, caminhões, pás-carregadoras, etc. No entanto, para o escopo deste trabalho, estes equipamentos não precisam ser considerados.

1. empilhadeira I: recebe o produto da torre T02 pela correia C2 e empilha no pátio A;
2. empilhadeira II: recebe o pellet feed vindo da torre T0 pela correia C11 e empilha no pátio B. Atualmente, o trilho desta empilhadeira alcança somente uma parte do pátio e, assim, o empilhamento nas regiões de balizas maiores no pátio B é feito pela SR (stack reclaimer);
3. SR (stack reclaimer - empilhadeira recuperadora): esta máquina pode tanto empilhar como recuperar. Ela empilha nos pátios A e B o produto vindo das torres T01 e T06 através da correia C1, e recupera as pilhas presentes nos pátios A e B enviando o produto pela correia C1 para a torre T03. Devido à sua estrutura, durante a recuperação, esta máquina só alcança até um pouco mais da metade do pátio (no sentido de largura) e, assim, é necessário o recheio⁶ das pilhas utilizando pás carregadoras, o que pode resultar em possíveis diminuições nas taxas de recuperação.

Para as duas empilhadeiras, assim como para a SR (stack reclaimer) e as correias, existem dados históricos de manutenções corretivas resultando em medidas de “mean time between failures”(MTBF) e “mean time to repair” (MTTR), que podem ser utilizadas como entradas do sistema de amplificação de inteligência. Além das manutenções corretivas, as máquinas também passam por manutenções preventivas geralmente duas vezes ao ano.

- Píer e carregadora de navios: o píer de carregamento de navios é formado por dois berços: leste e oeste. O berço leste tem restrições com relação ao tamanho de navios que podem ser atracados, aceitando apenas navios de até 75000 toneladas (Panamax). Além disso, este berço também não permite atracações durante a noite, por questões de segurança.

Os dois berços são atendidos por uma única carregadora de navios com taxa nominal de 9000 ton/h. A carregadora pode atender apenas um navio por vez.

Conforme pode ser observado na figura 7.7, o produto embarcado sai do pátio através da torre T03 e passa pela correia C3, torre T04 e Correia C4, até chegar à carregadora.

⁶Recheio é uma operação na qual uma parte da pilha de pelotas que foi separada e lançada para as periferias do pátio pela máquina recuperadora durante a recuperação é colocada de volta sobre a pilha, através do uso de tratores.

Depois de descritos os elementos que compõem o pátio de estocagem, será apresentado o modo de operação do pátio, ou seja, a forma como esses elementos descritos anteriormente interagem para que o produto flua pelo pátio. O fluxo de produto no pátio pode ser dividido basicamente em duas operações: empilhamento e recuperação.

- Empilhamento: Conforme visto anteriormente, são gerados três diferentes tipos de produtos na usina do pátio. Esses produtos devem ser armazenados em pilhas no pátio para posterior envio aos navios. A seguir, são descritas as formas como cada um dos três produtos é empilhado:
 1. Pellet feed: o pellet feed resultante do excedente da filtragem na usina, vindo pela correia FCS1A, é enviado para a empilhadeira II através da correia C11 e empilhado na região inicial dos pátios B ou C. O produto empilhado no pátio B é recuperado pela SR (stack reclaimer) para os navios ou pode ser retroalimentado para as usinas, em caso de falta de produtos (que pode ocorrer, por exemplo, em caso de interrupções no mineroduto). A taxa da empilhadeira II é de 1300 ton/h e o produto é empilhado em duas pilhas diferentes, conforme a família do produto: PFN (sílica normal) e PFL (sílica baixa);
 2. Sinter feed: o sinter feed, resultante do peneiramento da pelota, é enviado para uma pilha no pátio através da correia TP12. Este produto é frequentemente transferido, através de pás-carregadoras e caminhões, para uma pilha no pátio A, situada entre as balizas 25 a 34). Esta pilha é então posteriormente recuperada pela SR (stack reclaimer) para atendimento de embarques.
 3. Pelota: os produtos de pelota gerados na usina podem ser empilhados no pátio A com a empilhadeira I ou em um trecho do pátio B utilizando a SR (stack reclaimer). De maneira geral, sempre se aproveita as oportunidades para empilhar com a SR no pátio B, já que isso evita o excesso de material no pátio A e, conseqüentemente, a falta de espaço para empilhamento. A taxa de empilhamento de pelota é de aproximadamente 700 ton/h por usina, ou seja, pode alcançar 1400 ton/h quando as duas usinas estão enviando o produto para a mesma máquina. Existe ainda um tipo de pelota do peneiramento semi-móvel (PSM) que é resultante do repeneiramento de sobras colhidas nas diversas partes da usina/pátio e armazenada em uma pilha exclusiva no início do pátio B. O produto PSM não tem controle de qualidade e, desta forma, é utilizado em baixas quantidades misturadas com outras pelotas (principalmente MB45 e STD). O fino gerado neste repeneiramento é chamado de sinter PSM e empilhado em uma área no final do pátio B.

Algumas regras de empilhamento são utilizadas pela operação para gerenciar a qualidade de produto e o espaço no pátio:

- * procura-se concentrar os produtos iguais nas mesmas regiões do pátio. Assim, por exemplo, toda pelota do tipo STD é empilhada em uma região concentrada do pátio. Isso evita o deslocamento excessivo das máquinas durante o empilhamento e recuperação que ocorreria se as pilhas de STD estivessem espalhadas em várias regiões do pátio;

- * de maneira geral, procura-se empilhar o produto em pilhas já abertas. Só são criadas novas pilhas em caso de mudança de campanha, mudança de cliente ou falta de espaço para continuar empilhando em uma pilha já aberta;
 - * se sobrar produto em uma pilha reservada para um cliente, este produto pode ser utilizado como base da criação de uma nova pilha para um novo cliente, desde que as qualidades sejam compatíveis;
 - * em situações em que não é possível empilhar a pelota no pátio (falta de espaço, quebra de máquina, etc), é utilizada a área de emergência composta pelas pilhas P5 e TP17. O produto armazenado na pilha P5 deve ser transferido para a TP17 antes de voltar ao pátio. A pilha TP17 é recuperada utilizando-se pás-carregadoras e enviada para pilhas no pátio ou diretamente para o embarque de um navio;
 - * com relação à altura das pilhas criadas, se houver espaço no pátio, procura-se criar pilhas baixas (ocupando um maior número de balizas) evitando, desta forma, a necessidade de recheio durante a recuperação.
- Recuperação: as pilhas armazenadas no pátio são recuperadas pela SR (stack reclaimer) e enviadas para os navios atracados no píer. A SR (stack reclaimer) trabalha a uma taxa efetiva de 4500 ton/h para produtos de pelota, 2560 ton/h para pellet feed e 1920 ton/h para sinter feed. Devido a restrições em sua estrutura, esta máquina recupera apenas no sentido Norte - Sul (baliza 0 para baliza 130) e não alcança os pátios completamente no sentido da largura, sendo necessária a realização de recheios (produto que sobra é empurrado por pás-carregadoras para a frente dos pátios para que possa ser alcançado pela recuperadora). Devido também à sua estrutura, a SR deve recuperar trechos das pilhas de pelo menos três balizas ao mesmo tempo, tornando impossível a escolha de apenas uma baliza da pilha para recuperação.

A principal preocupação gerencial/operacional durante o carregamento é garantir que o produto embarcado atenda as especificações de qualidade exigidas pelo cliente. O plano de carregamento criado pelo engenheiro de processos procura justamente a mistura do produto das pilhas disponíveis, de forma a atender da melhor maneira possível às especificações de qualidade exigidas. O plano de carregamento contém as quantidades e balizas a serem recuperadas para as diversas pilhas. Para calcular a qualidade da mistura resultante, o engenheiro de processos faz análises da qualidade das pilhas baliza a baliza. No entanto, devido à falta de tempo para realizar os cálculos, aliado à falta de uma ferramenta computacional que facilite a tarefa, muitas vezes não é possível analisar a qualidade de cada pilha baliza a baliza, e acaba-se utilizando a qualidade média da pilha toda como entrada para o plano de carregamento, acarretando em eventuais problemas de qualidade durante o carregamento.

Durante o empilhamento, procura-se criar pilhas reservadas para determinados clientes com qualidades já definidas. No entanto, durante a recuperação, muitas vezes observa-se que a qualidade destas pilhas exclusivas não está atendendo corretamente à especificação do cliente. Isso ocorre principalmente devido à degradação do produto armazenado no pátio. Nestes casos, é necessário realizar a mistura destas pilhas com materiais de melhor qualidade para que a qualidade média final seja adequada. Nesta situação, as seguintes soluções são consideradas, nesta ordem de preferência:

1. o produto das pilhas reservadas é misturado com o produto de melhor qualidade sendo produzido na usina neste momento, desde que a campanha esteja sendo realizada para o mesmo cliente do embarque atual;
2. o produto das pilhas reservadas é misturado com o produto de outras pilhas com melhor qualidade;
3. o produto das pilhas reservadas é misturado com o produto de melhor qualidade sendo produzido na usina neste momento, mesmo que a campanha esteja sendo realizada para um cliente diferente daquele do embarque atual;
4. o produto das pilhas reservadas é descartado e o navio é carregado apenas com o produto de melhor qualidade sendo produzido na usina neste momento;
5. o produto das pilhas reservadas é descartado e o navio é carregado apenas com o produto de melhor qualidade de pilhas sendo formadas para outros clientes.

Em alguns casos, a pilha reservada para o cliente pode possuir uma qualidade acima daquela especificada. Nestes casos, o produto reservado pode ser misturado com o produto de qualidade mais baixa proveniente da usina ou de outras pilhas. Isso garante que a especificação de qualidade do cliente seja atendida e resulta em um estoque de produto de boa qualidade que pode ser utilizado para blendagens no futuro.

Descrição do cenário de janeiro de 2008

O início da seção 7.2.2 descreveu o cenário de julho de 2007 do pátio. No entanto, a implantação do sistema de amplificação de inteligência tem início de desenvolvimento previsto apenas para 2008. Nesta data, o pátio terá sofrido algumas transformações devido à instalação de uma terceira usina (o chamado projeto P3P).

Esta seção descreve este cenário de início de 2008 e segue o padrão utilizado no início da seção 7.2.2. Serão descritas aqui apenas as modificações que ocorrerão com a entrada da terceira usina e, portanto, não serão repetidos os elementos e operações no pátio já existentes no cenário atual e descritos anteriormente. Desta forma, são apresentados os novos elementos do pátio e é descrito como as operações de empilhamento e recuperação serão alteradas no cenário futuro.

A figura 7.9 apresenta o diagrama do pátio resultante da entrada da terceira usina. Esta configuração difere do cenário atual pelos seguintes elementos:

- Usina III: além das duas usinas de pelotização já existentes, será montada uma terceira usina. Esta usina tem um perfil técnico diferente das usinas já existentes, com melhor controle no processo de pelotização e, por isso, espera-se que o produto produzido seja de melhor qualidade. A terceira usina é alimentada por um mineroduto exclusivo, também acrescentado neste novo projeto. O material vindo da usina III tem acesso ao pátio através da torre 08TR001.
- Pátio C: o cenário de 2008 utilizará de forma mais ativa o pátio C. Este pátio tem o mesmo comprimento dos pátios A e B já existentes e será utilizado como reserva do pátio B no armazenamento de pelotas.

As novas estruturas construídas para o projeto P3P ocuparão as primeiras 30 balizas dos pátios A e B. Ainda não está definido se os pátios continuarão com a mesma marcação de balizas, resultante em uma área útil das balizas 30 a 130, ou se a marcação de balizas será zerada, resultando em pátios com balizas de 0 a 100.

- Novas máquinas: uma nova recuperadora será adicionada ao pátio no cenário futuro. Esta recuperadora atende os pátios B e C com uma capacidade nominal de 9330 t/h. Ela será a principal recuperadora do porto, sendo que a SR (stack reclaimer) disponível atualmente efetuará principalmente empilhamentos e servirá como auxiliar na recuperação, com capacidade de 7000 t/h.

A empilhadeira II, que no cenário atual empilha na região inicial do pátio B, será reformada e transferida para uma nova linha, passando a empilhar somente no pátio C, com capacidade de 3000 t/h.

Serão também repotenciadas a linha de embarque e a carregadora de navios para suportar uma taxa efetiva de embarque de 11200 t/h. Com a inclusão da nova recuperadora e deslocamento da empilhadeira II, o cenário futuro tem a configuração de máquinas presente na tabela 7.3:

Máquina	Função
Empilhadeira I	empilha no pátio A
SR (stack reclaimer)	empilha nos pátios A e B recupera nos pátios A e B
Recuperadora	recupera nos pátios B e C
Empilhadeira II	empilha no pátio C

Tab. 7.3: Equipamentos do pátio no cenário futuro

- Torres e correias: com a entrada da terceira usina, as torres T01/T06 vão ser apenas pontos de passagem, e serão substituídas pela torre 08TR002, onde será concentrado todo o mecanismo de chaveamento entre as diversas rotas.

No carregamento, além da torre T03, que recebe o produto recuperado pela SR (stack reclaimer), será criada uma nova torre de carregamento para receber o produto da nova recuperadora, a torre 09TR001. Esta nova torre também efetuará medidas de qualidade.

Estas medidas são independentes daquelas realizadas pela torre T03. Todo o fluxo de produto vindo da nova torre cairá na torre T03, e daí seguirá para a carregadora.

O projeto P3P prevê a instalação de balanças em todas as correias principais de empilhamento e recuperação, facilitando bastante a entrada de dados para o sistema de amplificação de inteligência.

- Peneiramento semi-móvel (PSM): no cenário de 2008, o PSM estará localizado nas balizas finais do pátio A, recebendo os resíduos do pátio e realizando um repeneiramento, que dá origem a três produtos: pelotas, sinter e lixo. O material resultante do PSM não possui controle de qualidade. O PSM atua a uma taxa de 200 t/h.
- Sistema de moegas: o sistema de moegas tem uma capacidade de 1000 t/h e serve para realizar um estorno de pellet feed do pátio para as usinas, através da 52TP52 (antiga FCS1A), que é reversível. Esta é uma forma alternativa de alimentar as usinas, além do mineroduto.
- Empilhamento de sinter feed: o sinter feed resultante do processo das usinas é empilhado no início do pátio A (usinas I e II) e do pátio C (usina III), através das saídas das torres 08TR002 e 08TR001, respectivamente.
- Emergência TP-17: o material que não pode seguir para o pátio, por algum motivo, é empilhado na emergência da TP-17. Um exemplo de motivo para o material não seguir para o pátio poderia ser uma parada nas máquinas empilhadeiras. Quando há uma oportunidade desse material seguir para o pátio, ele é recuperado por pás-carregadeiras e colocado diretamente na correia C2 através de uma moega.

Conforme citado anteriormente, a entrada da terceira usina altera alguns aspectos das operações de empilhamento e recuperação descritas no início da seção 7.2.2 para o cenário atual. A seguir, serão detalhadas essas alterações:

- Empilhamento: com relação às campanhas de produção das usinas, existem restrições em Germano para a produção de insumos diferentes. Desta forma, geralmente, o mesmo material será enviado nos dois minerodutos, fazendo com que a produção das três usinas seja da mesma família de produto. No entanto, será possível que as usinas estejam produzindo produtos diferentes dentro de uma mesma família durante um determinado momento.

A decisão de como fica a distribuição das campanhas pelas usinas, para atender as vendas, será tomada caso a caso, ou seja, não segue uma regra fixa do tipo: 50% da produção fica a cargo da usina III e 50% a cargo das usinas I e II. Os documentos de campanha por usina serão uma entrada para o sistema de amplificação de inteligência.

As pelotas produzidas nas usinas I, II e III chegam na torre principal (08TR002) e daí podem ser transferidas para o pátio B (via SR - stack reclaimer) ou para o pátio C (via empilhadeira II). A torre principal oferece flexibilidade de separar os fluxos das usinas I, II e III para as duas linhas de saída. Assim, é possível, por exemplo, empilhar a saída das usinas I e II no pátio B com a SR e, ao mesmo tempo, empilhar a saída da usina III no pátio C com a empilhadeira II. É possível também concentrar as saídas das três usinas em uma única máquina. O pátio B será o pátio preferencial para o armazenamento de

pelota fazendo com que o pátio C seja utilizado apenas quando o pátio B está cheio ou quando a SR não está disponível (em manutenção ou auxiliando a nova recuperadora em um carregamento). O excedente do pellet feed, produzido na filtragem das usinas, será empilhado no pátio A com a empilhadeira I ou com a SR (stack reclaimer). No cenário de 2008, todo o empilhamento de pellet feed será concentrado no pátio A.

O empilhamento de sinter feed das usinas I e II continua da mesma forma, mudando-se apenas a localização da pilha. A nova pilha de sinter também ficará no pátio A, em uma região não alcançada pela recuperadora, e será transferida por caminhão para uma região no pátio A alcançada pela recuperadora, como é feito no cenário de julho de 2007. O sinter feed produzido na usina III será transferido via correia e empilhado diretamente no início do pátio C.

A pilha de emergência de pelota das usinas I e II continua sendo a TP17. O material da TP17 é transferido via moega instalada na C02 para uma região no final do pátio A, onde ele poderá ser recuperado pela SR (stack reclaimer). Para a usina III, será criada a pilha de emergência TP08 no pátio C.

As seguintes formas de empilhamento são possíveis:

1. Linha 1 - Empilhamento de pellet feed: o excedente de pellet feed das usinas I, II e III é direcionado para a correia C1 e empilhado no pátio A através da SR, conforme figura 7.10.

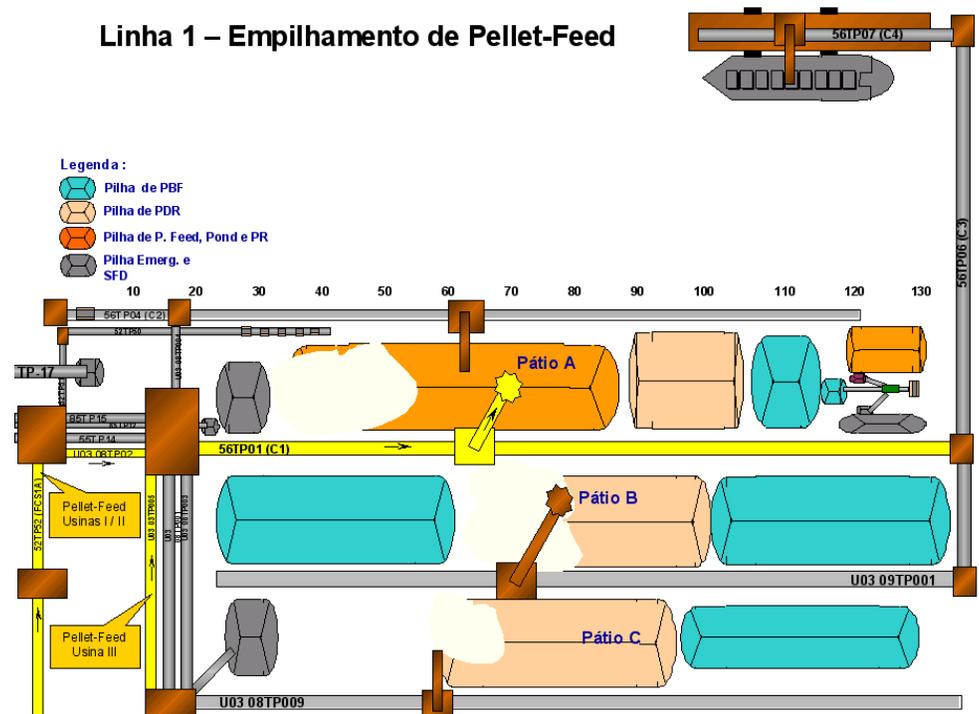


Fig. 7.10: Linha 1 - Empilhamento de pellet feed

2. Linha 2 - Empilhamento de pellet feed: o excedente de pellet feed das usinas I, II e III é direcionado para a correia C2 e empilhado no pátio a através da empilhadeira 1,

conforme figura 7.11.

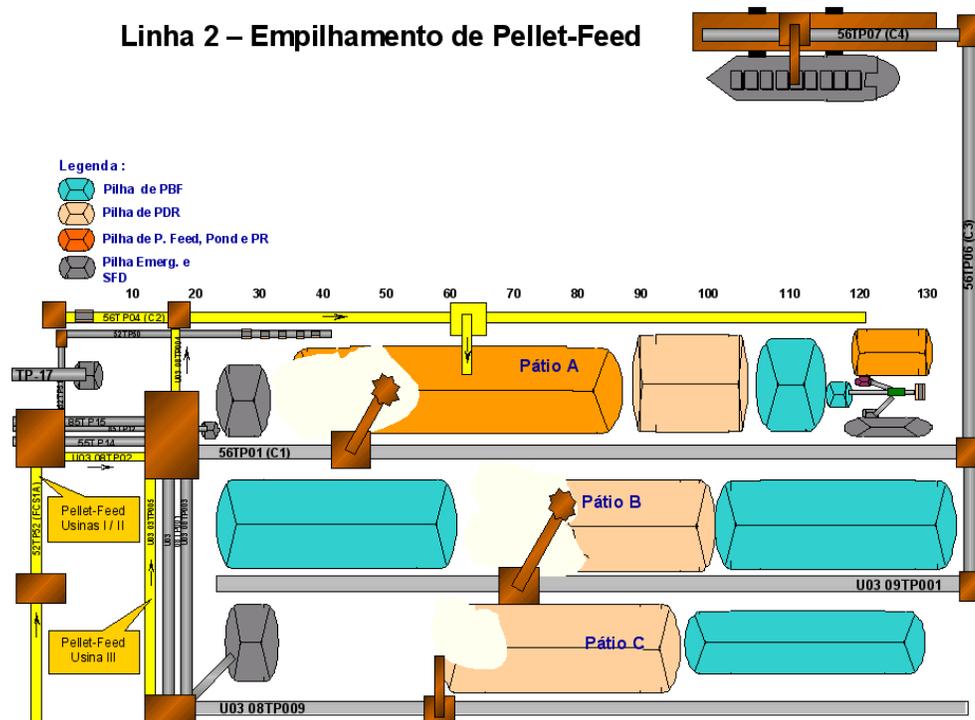


Fig. 7.11: Linha 2 - Empilhamento de pellet feed

3. Linha 1 - Empilhamento de pelotas: pelotas das usinas I, II e III seguem para a correia C1 e são empilhadas no pátio A ou B com a SR, conforme a figura 7.12.
 4. Linha 3 - Empilhamento de pelotas: pelotas das usinas I, II e III seguem para a correia 08TP009 e são empilhadas no pátio C com a empilhadeira 2, conforme a figura 7.13.
 5. Linha 2 - Empilhamento de pelotas (rota alternativa): pelotas das usinas I e II são direcionadas para a correia C2 e empilhadas no pátio A com a empilhadeira 1, conforme a figura 7.14.
 6. Linha 2 - Empilhamento de pelotas (emergência das usinas I e II): as pelotas da emergência da TP-17 retornam para a correia C2 através de alimentação por moega e são empilhadas no pátio A com a empilhadeira 1, conforme a figura 7.15.
 7. Empilhamento de pellet screening (sinter feed): o sinter feed das usinas I e II é empilhado no início do pátio A por uma saída da torre 08TR002, enquanto o sinter feed da usina III é empilhado no início do pátio C por uma saída da torre 08TR001, conforme a figura 7.16.
- Recuperação: a nova recuperadora incluída no projeto P3P será a principal fonte de saída de produto do pátio, sendo que a SR (stack reclaimer) funcionará a maior parte do tempo empilhando. A nova recuperadora poderá recuperar nos dois sentidos (baliza menor para maior e maior para menor), com capacidade efetiva estimada para 9330 ton/h. As linhas de embarque e a carregadora de navios serão repotenciadas para suportar essa nova taxa.

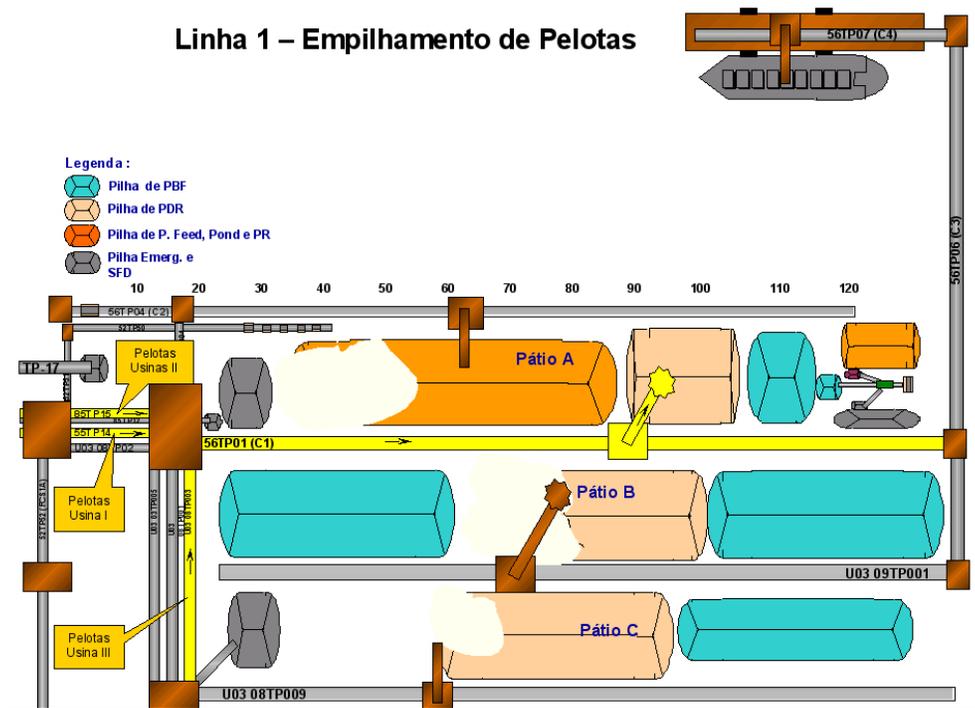


Fig. 7.12: Linha 1 - Empilhamento de pelotas

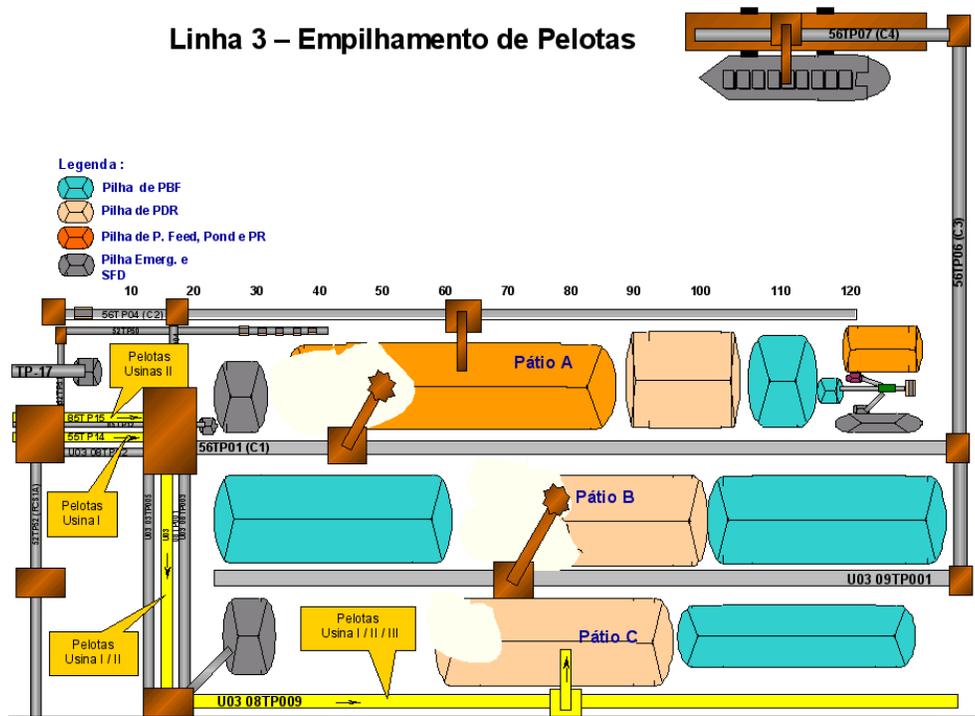


Fig. 7.13: Linha 3 - Empilhamento de pelotas

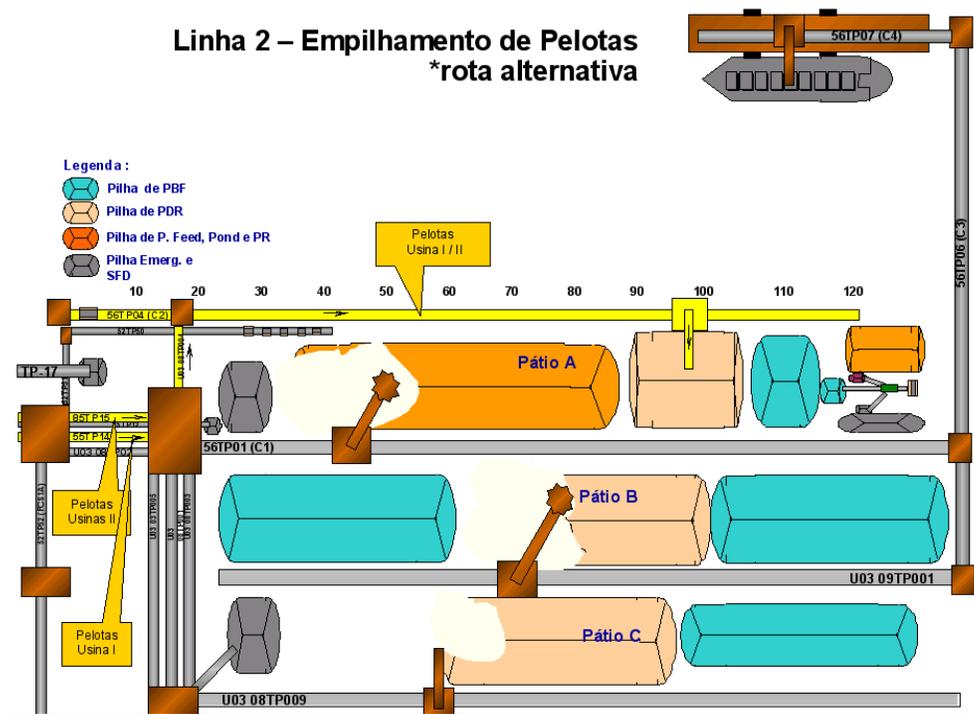


Fig. 7.14: Linha 2 - Empilhamento de pelotas (rota alternativa)

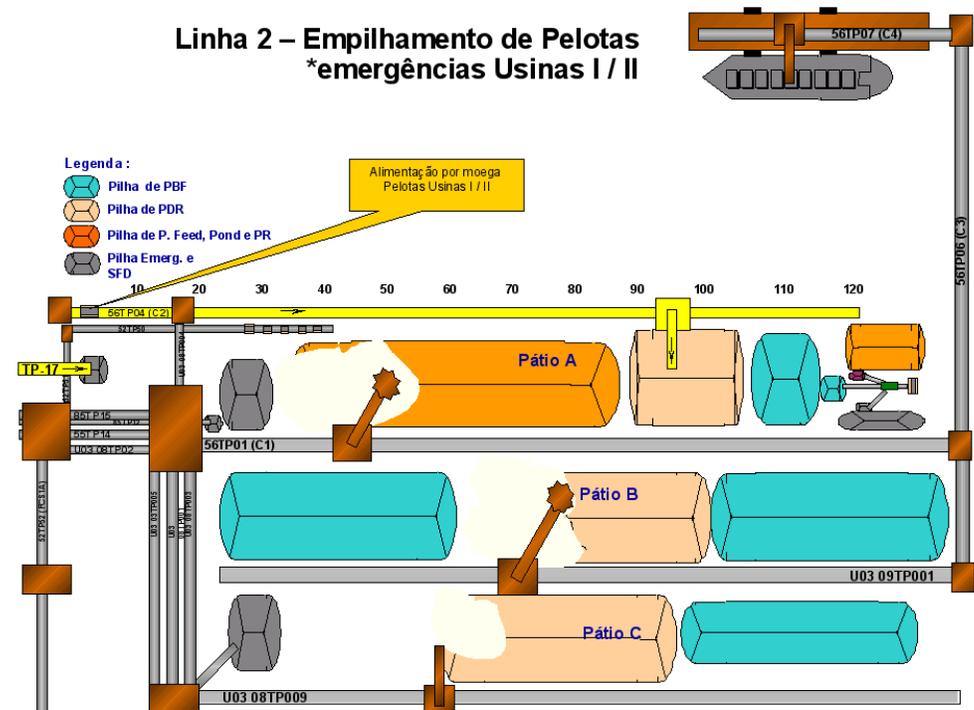


Fig. 7.15: Linha 2 - Empilhamento de pelotas (emergência das usinas I e II)

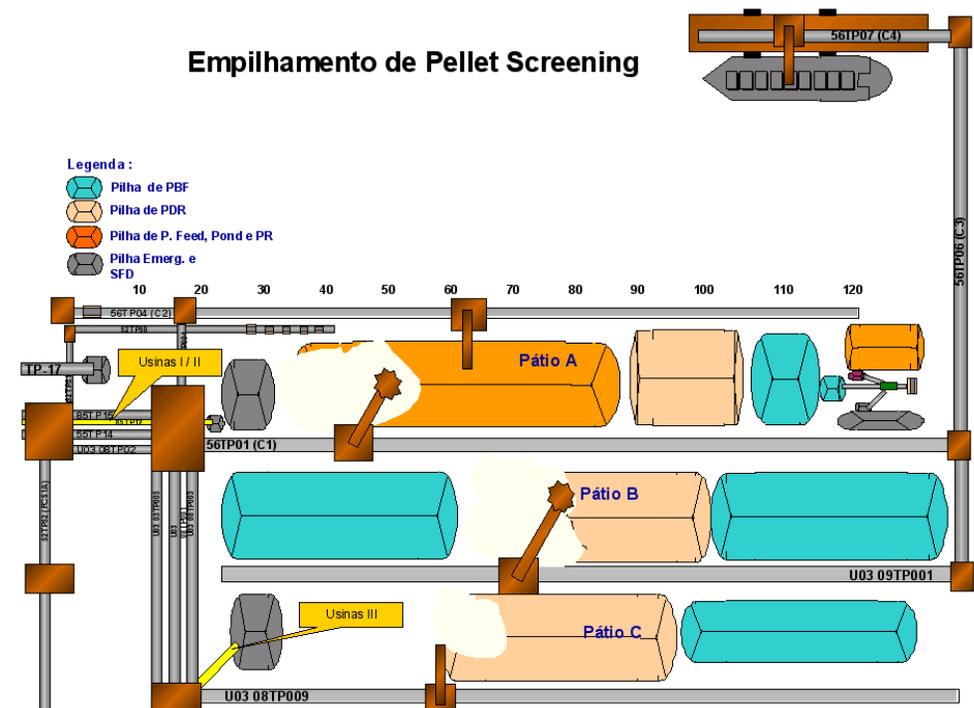


Fig. 7.16: Empilhamento de pellet screening (sinter feed)

Apesar de utilizar principalmente a nova recuperadora nos embarques, será possível trabalhar com a recuperação das duas máquinas ao mesmo tempo. Isso ocorre principalmente com as duas máquinas trabalhando na mesma pilha, com a nova recuperadora como máquina principal e a SR recuperando o que sobraria e deveria ser empurrado com pás-carregadoras (recheio). Nestes casos em que as duas recuperadoras estão trabalhando ao mesmo tempo, é necessário reduzir as taxas já que a correia de saída para embarque não suporta as duas recuperadoras operando na capacidade máxima (mesmo após o repotenciamento).

Com relação às medidas de tonelagem carregada haverá duas balanças: uma entre a nova torre de carregamento e a T03 e outra logo após a T03. Assim, a primeira balança mede a quantidade carregada pela nova recuperadora, enquanto a segunda balança mede o somatório do produto carregado pela nova recuperadora juntamente com a SR (stack reclaimers). Neste caso uma conta simples de subtração resulta no valor carregado pela SR.

As seguintes formas de recuperação são possíveis:

1. Linha de recuperação A: a SR recupera qualquer produto do pátio A, que segue pela correia C1 até a torre 3 e daí para o embarque, conforme figura 7.17.
2. Linha de recuperação B: a recuperadora retoma qualquer produto do pátio B, que segue pela correia 09TP001 até a torre 09TR001 e daí para o embarque, conforme figura 7.18.
3. Linha de recuperação A e B: é possível recuperar concomitantemente dos pátios A e B, através da SR e da recuperadora, respectivamente, conforme figura 7.19. Neste

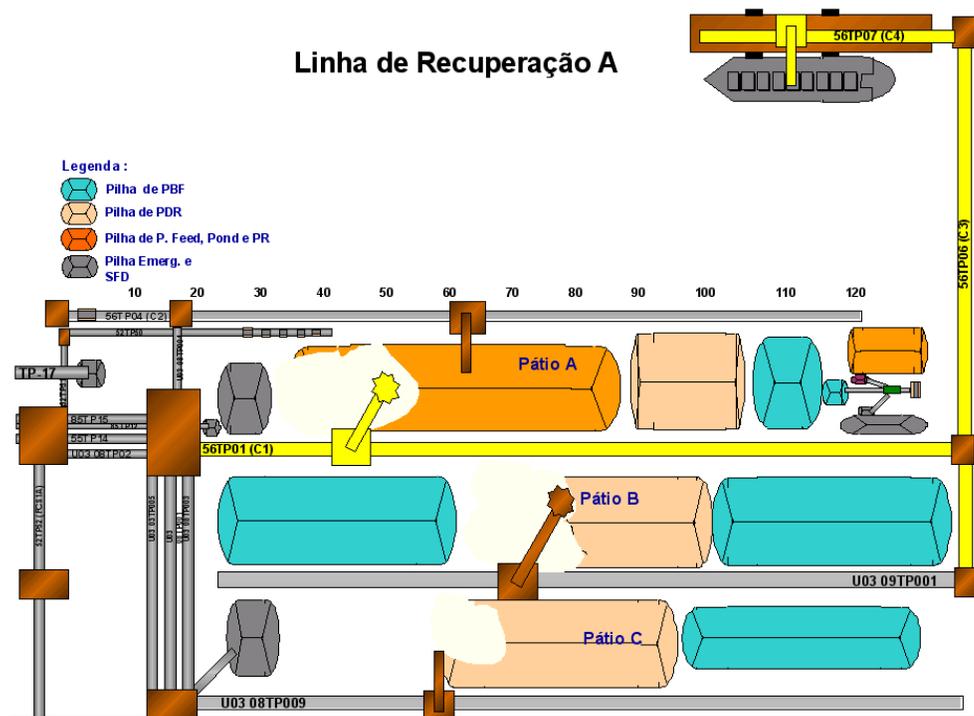


Fig. 7.17: Linha de recuperação A

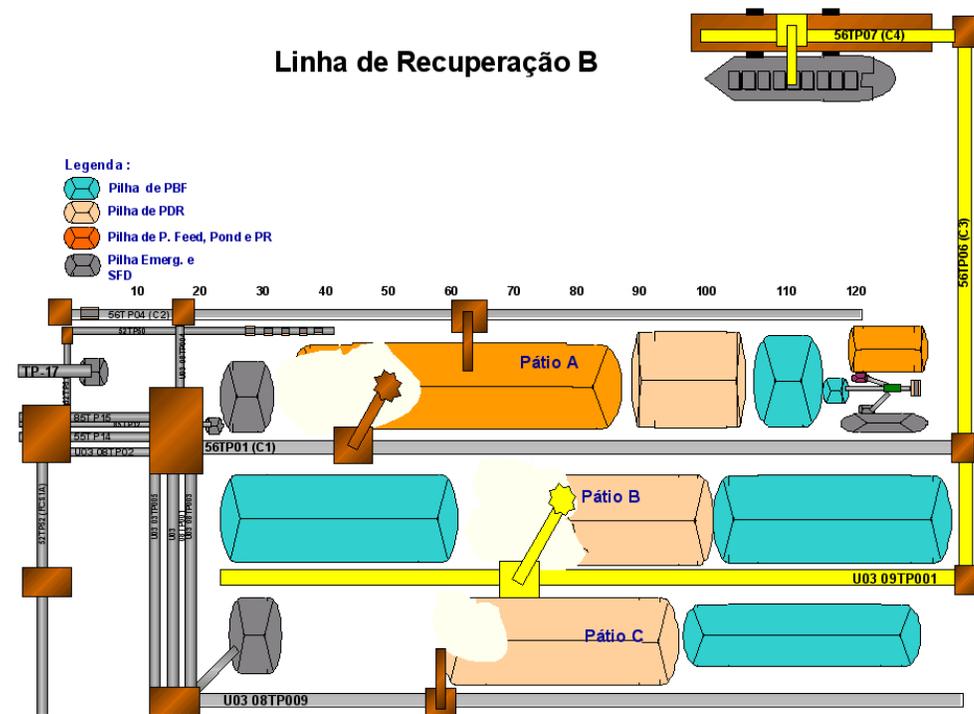


Fig. 7.18: Linha de recuperação B

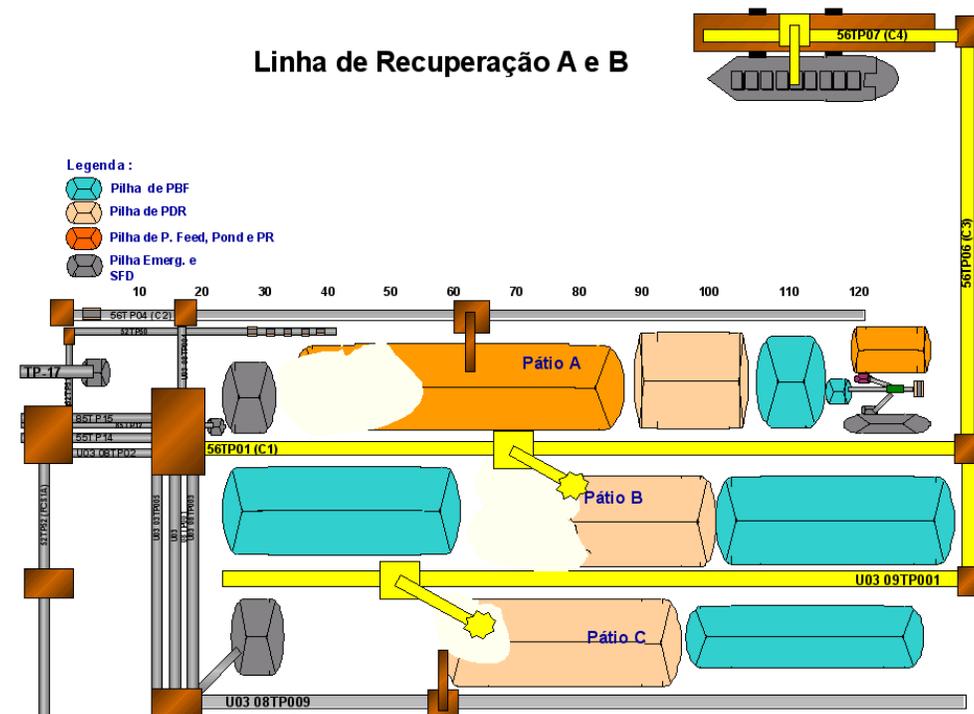


Fig. 7.19: Linha de recuperação A e B

caso, a taxa de recuperação deve ser controlada para que não exceda a taxa máxima do embarque, que é de 11200 t/h, menor do que a soma da capacidade máxima total da SR e da recuperadora juntas.

4. Linha de recuperação A + usinas: pode ser embarcado material do pátio A, recuperado pela SR, mais o material vindo diretamente das usinas I, II e III, conforme figura 7.20.
5. Retorno de pellet feed: o pellet feed pode ser estornado do pátio de volta para as usinas através de alimentação por moegas na correia 52TP50, voltando pela correia 52TP52, que é reversível, conforme figura 7.21. Essa é uma maneira alternativa de alimentar as usinas, além do pellet feed vindo do mineroduto.

Principais dificuldades no gerenciamento do pátio

Esta seção descreve as principais dificuldades encontradas atualmente no gerenciamento do pátio. Estas dificuldades são os pontos mais relevantes em que o sistema de amplificação de inteligência auxiliaria os operadores na tarefa de gerenciar o pátio de forma eficiente.

As principais dificuldades encontradas foram:

- Falta de interface gráfica computacional: atualmente, não existe uma interface gráfica computacional para representar a situação das pilhas no pátio. Existe apenas uma interface gráfica desenhada em uma lousa, apresentada na figura 7.22, composta de uma

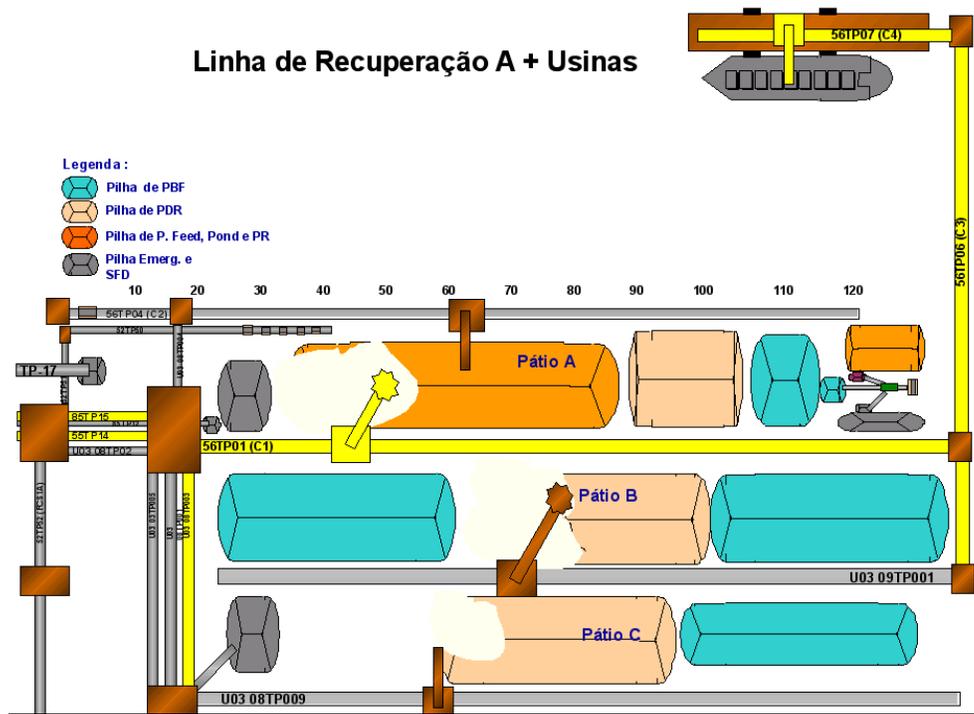


Fig. 7.20: Linha de recuperação A + usinas

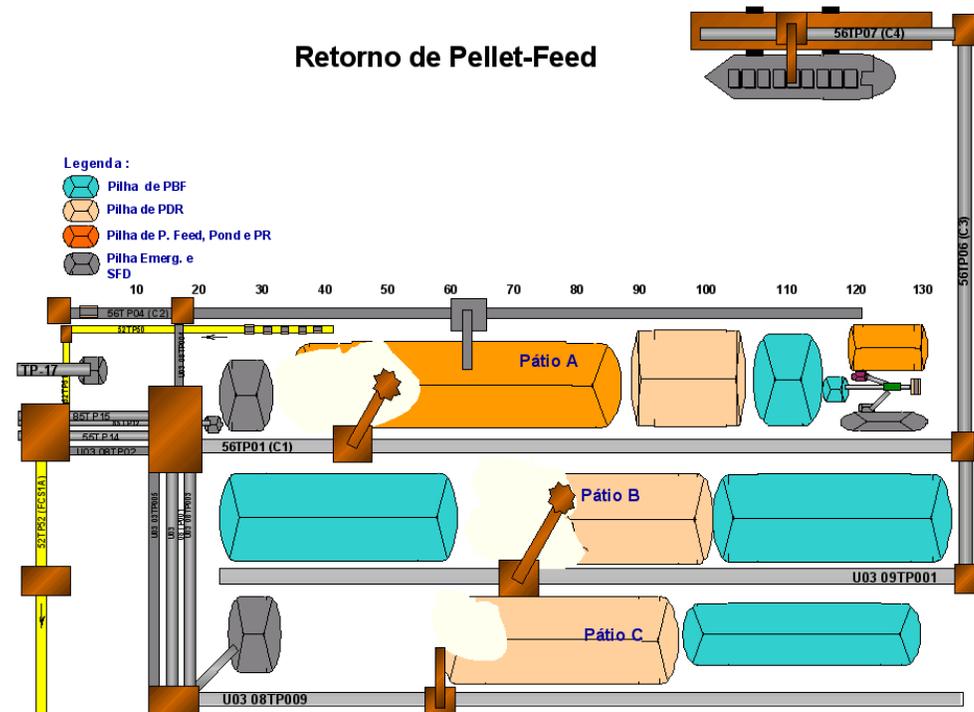


Fig. 7.21: Retorno de pellet feed

A qualidade do produto empilhado e recuperado é medida através de uma série de indicadores de características físicas, químicas e metalúrgicas. Os produtos vendidos para os clientes devem atender às especificações de qualidade pré-definidas, especificações estas compostas por um valor típico, valor máximo e valor mínimo garantidos para cada medida. O problema da rastreabilidade, no entanto, ocorre pois a qualidade do produto produzido varia de momento a momento em cada usina, assim como entre as usinas. Devido a essa variabilidade, é preciso manter a rastreabilidade do material empilhado para que haja possibilidade de tomada de decisão, em tempo hábil, em busca das metas de qualidade prevista nos contratos, por exemplo: mistura de materiais de qualidades diversas.

Atualmente, a rastreabilidade é feita através do casamento das informações das medidas bi-horárias (qualidade e quantidade) na saída das usinas juntamente com relatórios manuais de posicionamento das máquinas (dados de operação), gerando uma qualidade aproximada por baliza de cada pilha. Esta qualidade por baliza pode ser utilizada como indicativo da qualidade do produto em cada parte da pilha e utilizada em casos onde é necessário a mistura de diversas pilhas. Na operação diária, no entanto, o cálculo desta qualidade baliza por baliza é trabalhoso e demorado e, devido à inexistência de um sistema de amplificação de inteligência, acaba sendo realizado apenas em casos específicos. A figura 7.5 mostra um exemplo de análise necessária para medir a qualidade da pilha.

Cada ponto do gráfico da figura 7.5 representa uma medida bi-horária de um determinado item de controle (medida de qualidade). A região em que os valores medidos estão abaixo daqueles especificados pela orientação de embarque determina um período em que a qualidade do produto produzido não atende as especificações do cliente. Neste caso, seria necessário consultar o histórico de posicionamento da máquina para descobrir qual era seu posicionamento durante este período, e evitar recuperar das balizas onde a máquina trabalhava nestes momentos. Para uma avaliação completa da qualidade da pilha, seria necessário fazer esta análise para cada um dos diversos itens de controle. No dia a dia, muitas vezes o operador tem tempo apenas de considerar a qualidade média da pilha, resultante das qualidades diversas empilhadas durante o período de formação. Esta qualidade média nem sempre é um bom indicador, já que partes diversas da pilha podem ter qualidades diferentes.

As medidas de qualidade do produto são realizadas em dois pontos distintos:

1. Torres T01 e T06 na saída das usinas: nestas torres são realizadas as coletas de amostras a cada duas horas para o produto sendo produzido nas usinas. Devido ao tempo necessário para coletar e medir as amostras, não se sabe com certeza qual é a qualidade exata do produto que está sendo empilhado no momento, já que o resultado só estará disponível duas horas depois. É possível apenas analisar tendências na qualidade do produto, mas a incerteza é grande;
2. Torre T03: nesta torre são coletadas as amostras para o produto que está sendo embarcado nos navios. Um problema frequente ocorre quando a qualidade esperada na pilha, medida na saída da usina, é diferente daquela medida no embarque, resultando na necessidade de recuperar pilhas diferentes das planejadas de forma a resultar em uma mistura com qualidade mais próxima daquela desejada pelo cliente. Além disso, atualmente as medidas da qualidade no embarque são diferentes daquelas realizadas

na saída das usinas mesmo quando o produto é passado diretamente para o navio, indicando discrepâncias na forma como as amostras são analisadas nas torres T01/T06 e T03.

Um outro problema de difícil gerenciamento e que causa grande dificuldade na rastreabilidade da qualidade das pilhas armazenadas no pátio é o fato de que a qualidade se degrada no tempo. Alguns estudos foram realizados por equipes da Samarco para tentar prever de forma mais precisa o efeito da degradação do produto empilhado no pátio, mas nenhuma conclusão definitiva foi alcançada. O sistema de amplificação de inteligência deve considerar de alguma forma esta influência da degradação nas soluções sugeridas através de históricos de qualidade.

Um outra fonte de incertezas é o problema da umidade nas pilhas armazenadas no pátio e embarcadas nos navios. Além da umidade já natural do produto, é introduzida água no processo em diversos pontos, principalmente para diminuir a geração de particulado.

Devido à esses diversos problemas, a equipe da Estocagem e Embarque já experimentou diversos tipos de empilhamento para tentar melhorar a qualidade das pilhas armazenadas, assim como garantir a rastreabilidade da qualidade. As duas formas principais de empilhamento testadas foram:

1. Cones Alternados: a empilhadeira forma um série de pequenas pilhas ao longo do pátio. Depois de completar uma primeira sequencia de pilhas, a empilhadeira se move para trás, empilhando o material nos “gaps” restantes, minimizando a segregação. Este tipo de empilhamento resulta em uma menor segregação do produto na pilha (acúmulo de fino no “miolo” da pilha). No entanto, a rastreabilidade da qualidade baliza a baliza é praticamente impossível neste caso. Este tipo de empilhamento está representado na figura 7.23;
2. Cones Shell: a pilha é formada depositando material num cone único numa posição fixa. Quando esta pilha cônica está cheia, a empilhadeira se move um pouco para frente e um novo cone é formado ao lado do anterior. Este processo continua numa direção longitudinal até que o empilhamento esteja completo. Neste caso, a rastreabilidade da qualidade baliza a baliza é mais fácil de ser gerenciada. O problema aqui é o aumento da segregação do produto, piorando a qualidade do produto comercializado. Este tipo de empilhamento está representado na figura 7.24.



Fig. 7.23: Forma de empilhamento em cones alternados

Atualmente, o modo de empilhamento utilizado é o Cone Shell, pois uma das principais preocupações no manejo das pilhas no pátio de estocagem é a necessidade de uma

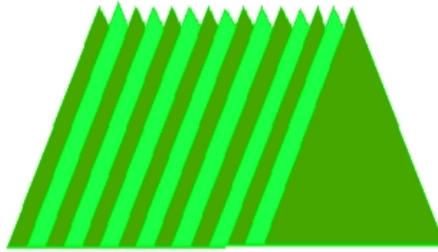


Fig. 7.24: Forma de empilhamento em cones Shell

boa rastreabilidade da qualidade das pilhas para garantir flexibilidade no atendimento das qualidades exigidas no embarque pelos clientes. No futuro, se houver uma estabilização na qualidade do produto gerado nas usinas e a rastreabilidade deixar de ser uma preocupação, será possível adotar novos modos de empilhamento como, por exemplo, o de cones alternados garantindo uma menor segregação das pilhas. Deve-se tomar como premissa para o sistema de amplificação de inteligência que, enquanto houver necessidade de rastreabilidade, o modo de empilhamento utilizado será o Cone Shell.

De forma a melhorar o entendimento dos cálculos realizados para determinar as qualidades médias das blendagens, é apresentado, nesta seção, um exemplo da forma como essa operação é realizada manualmente.

Para cada carga embarcada, existe uma planilha de rastreabilidade que concentra as informações utilizadas pelo Engenheiro de Processos na realização dos planos. Esta planilha apresenta os detalhes de cada uma das pilhas (quotas) utilizadas no embarque. Cada quota tem uma data de início e término de produção conforme apresentado na figura 7.25.

CLIENTE: NINGBO IRON		PRODUTO: PBF/MB45		NAVIO: KYZIKOS		Pag. 1	
Produto - Quota 1: Pilha 87 1 - PBFMB45		Produto - Quota 2: Pilha 87 2 - PBFMB45		Produto - Quota 3: Pilha 86 - PBFMB45		Produto - Quota 4: Pilha 76 - PBFMB45	
02/07/2007 10:00	← Início da Produção	03/07/2007 11:40	← Início da Produção	02/07/2007 00:00	← Início da Produção	22/06/2007 12:00	← Início da Produção
03/07/2007 09:40	← Term da Produção	06/07/2007 08:00	← Term da Produção	02/07/2007 10:00	← Term da Produção	23/06/2007 16:00	← Term da Produção
19023,36	← Total da usina 1	6712,8	← Total da usina 1	6002,4	← Total da usina 1	3685,44	← Total da usina 1
20608,64	← Total da usina 2	7272,2	← Total da usina 2	6602,6	← Total da usina 2	3992,56	← Total da usina 2
Total:	39.632	Total:	13.985	Total:	12.605	Total:	7.678
Qualidade Média		Qualidade Média		Qualidade Média		Qualidade Média	
Pelota - Química		Pelota - Química		Pelota - Química		Pelota - Química	
Fe	67,44 ± #DIV/0!	Fe	67,30 ± 0,06	Fe	67,59 ± #DIV/0!	Fe	67,24 ± 0,01
FeO	0,25 ± #DIV/0!	FeO	0,25 ± 0,07	FeO	0,24 ± #DIV/0!	FeO	0,32 ± 0,11
SiO2	1,79 ± 0,07	SiO2	1,96 ± 0,10	SiO2	1,81 ± 0,15	SiO2	2,09 ± 0,07
Al2O3	0,56 ± #DIV/0!	Al2O3	0,59 ± 0,01	Al2O3	0,53 ± #DIV/0!	Al2O3	0,52 ± 0,00
CaO	0,79 ± 0,05	CaO	0,87 ± 0,06	CaO	0,75 ± 0,05	CaO	0,91 ± 0,10
MgO	0,08 ± 0,01	MgO	0,08 ± 0,01	MgO	0,07 ± 0,00	MgO	0,10 ± 0,01
P	0,05 ± #DIV/0!	P	0,05 ± 0,00	P	0,05 ± #DIV/0!	P	0,05 ± 0,00
B2	0,44 ± 0,03	B2	0,45 ± 0,03	B2	0,47 ± 0,04	B2	0,44 ± 0,04
Pelota - Física		Pelota - Física		Pelota - Física		Pelota - Física	
+ 19,0 mm	0,11 ± 0,13	+ 19,0 mm	0,02 ± 0,06	+ 19,0 mm	0,08 ± 0,08	+ 19,0 mm	0,09 ± 0,13
- 19,0 + 16,0 mm	3,79 ± 1,53	- 19,0 + 16,0 mm	3,43 ± 1,36	- 19,0 + 16,0 mm	3,66 ± 0,70	- 19,0 + 16,0 mm	3,68 ± 1,38
- 16,0 + 12,5 mm	35,63 ± 4,25	- 16,0 + 12,5 mm	38,13 ± 4,98	- 16,0 + 12,5 mm	38,78 ± 3,22	- 16,0 + 12,5 mm	38,68 ± 3,79
- 12,5 + 9,0 mm	54,50 ± 4,75	- 12,5 + 9,0 mm	53,44 ± 5,16	- 12,5 + 9,0 mm	52,21 ± 3,18	- 12,5 + 9,0 mm	52,05 ± 4,24
- 9,0 + 8,0 mm	3,51 ± 0,68	- 9,0 + 8,0 mm	2,71 ± 0,71	- 9,0 + 8,0 mm	2,95 ± 0,65	- 9,0 + 8,0 mm	3,25 ± 0,67
- 8,0 + 6,3 mm	1,48 ± 0,35	- 8,0 + 6,3 mm	1,34 ± 0,36	- 8,0 + 6,3 mm	1,17 ± 0,18	- 8,0 + 6,3 mm	1,35 ± 0,26
- 6,3 mm	0,98 ± 0,21	- 6,3 mm	0,92 ± 0,22	- 6,3 mm	1,14 ± 0,24	- 6,3 mm	0,90 ± 0,23
- 16,0 + 8,0 mm	93,64 ± 1,37	- 16,0 + 8,0 mm	94,28 ± 1,18	- 16,0 + 8,0 mm	93,95 ± 0,76	- 16,0 + 8,0 mm	93,98 ± 1,28
TAMBORAMENTO	93,88 ± 0,62	TAMBORAMENTO	94,17 ± 0,56	TAMBORAMENTO	93,94 ± 0,72	TAMBORAMENTO	94,19 ± 0,48
ABRASÃO	5,19 ± 0,62	ABRASÃO	4,99 ± 0,47	ABRASÃO	5,12 ± 0,66	ABRASÃO	4,99 ± 0,42
COMPRESSÃO	297,65 ± 111,61	COMPRESSÃO	310,47 ± 111,82	COMPRESSÃO	281,90 ± 101,10	COMPRESSÃO	307,27 ± 112,72
CCS <200 kg	19,82 ± 5,42	CCS <200 kg	16,86 ± 4,68	CCS <200 kg	18,36 ± 4,27	CCS <200 kg	17,74 ± 4,07
CCS <150 kg	10,17 ± 2,77	CCS <150 kg	8,73 ± 3,46	CCS <150 kg	11,05 ± 4,48	CCS <150 kg	9,54 ± 3,37
CCS <100 kg	5,20 ± 5,38	CCS <100 kg	3,60 ± 2,21	CCS <100 kg	4,44 ± 3,05	CCS <100 kg	3,66 ± 1,87

Fig. 7.25: Quotas para embarque

A partir destas datas de início e término de produção das quotas, é realizada uma consulta no MES para recuperar as medidas bi-horárias realizadas neste período para as duas usinas. Assim, por exemplo, para a quota 1 (pilha 87-1) com data de início 10:00 - 02/07/2007 e data de término 09:40 - 03/07/2007, a consulta no MES devolve as medidas bi-horárias realizadas entre 11:30 - 02/07/2007 e 09:30 - 03/07/2007 para cada uma das usinas (obs: as bi-horárias atualmente são efetuadas nas horas 01:30, 03:30, 05:30, 07:30, 09:30, 11:30, 13:30, 15:30, 17:30, 19:30, 21:30 e 23:30).

A partir destes valores, é calculada a média de todas as bi-horárias para cada uma das usinas, conforme apresentado na figura 7.26. A primeira coluna apresenta a média dos itens de controle para as bi-horárias consideradas para a Usina I e a segunda coluna a média das bi-horárias para a Usina II. A terceira coluna mostra a média das qualidades das duas usinas ponderadas pela produção de cada uma no período considerado conforme mostra a figura 7.27.

Produto:		PBF/MB45		Início:		2/7/07 0:00		Fim:		03/07/2007 23:59	
Obs.:											
Qualidade usina 1				Qualidade usina 2				QUALIDADE MEDIA			
Pellet feed - Quimica				Pellet feed - Quimica				Pellet feed - Quimica			
Fe	66,92 ± 0,06	Fe	- ± -	Fe	66,92 ± 0,06	Fe	66,92 ± 0,06	Fe	66,92 ± 0,06	Fe	66,92 ± 0,06
FeO	0,78 ± 0,04	FeO	- ± -	FeO	0,78 ± 0,04	FeO	0,78 ± 0,04	FeO	0,78 ± 0,04	FeO	0,78 ± 0,04
SiO2	1,33 ± 0,05	SiO2	- ± -	SiO2	1,33 ± 0,05	SiO2	1,33 ± 0,05	SiO2	1,33 ± 0,05	SiO2	1,33 ± 0,05
Al2O3	0,39 ± 0,01	Al2O3	- ± -	Al2O3	0,39 ± 0,01	Al2O3	0,39 ± 0,01	Al2O3	0,39 ± 0,01	Al2O3	0,39 ± 0,01
CaO	0,09 ± 0,01	CaO	- ± -	CaO	0,09 ± 0,01	CaO	0,09 ± 0,01	CaO	0,09 ± 0,01	CaO	0,09 ± 0,01
MgO	0,02 ± 0,00	MgO	- ± -	MgO	0,02 ± 0,00	MgO	0,02 ± 0,00	MgO	0,02 ± 0,00	MgO	0,02 ± 0,00
P	0,046 ± 0,00	P	- ± -	P	0,046 ± 0,00	P	0,046 ± 0,00	P	0,046 ± 0,00	P	0,046 ± 0,00
PPC	2,33 ± 0,11	PPC	- ± -	PPC	2,33 ± 0,11	PPC	2,33 ± 0,11	PPC	2,33 ± 0,11	PPC	2,33 ± 0,11
Pelota - Quimica				Pelota - Quimica				Pelota - Quimica			
Fe	67,46 ± 0,14	Fe	67,57 ± 0,07	Fe	67,51 ± 0,11	Fe	67,51 ± 0,11	Fe	67,51 ± 0,11	Fe	67,51 ± 0,11
FeO	0,23 ± 0,01	FeO	0,26 ± 0,01	FeO	0,24 ± 0,01	FeO	0,24 ± 0,01	FeO	0,24 ± 0,01	FeO	0,24 ± 0,01
SiO2	1,82 ± 0,13	SiO2	1,72 ± 0,11	SiO2	1,77 ± 0,12	SiO2	1,77 ± 0,12	SiO2	1,77 ± 0,12	SiO2	1,77 ± 0,12
Al2O3	0,56 ± 0,04	Al2O3	0,53 ± 0,01	Al2O3	0,55 ± 0,02	Al2O3	0,55 ± 0,02	Al2O3	0,55 ± 0,02	Al2O3	0,55 ± 0,02
CaO	0,83 ± 0,07	CaO	0,77 ± 0,06	CaO	0,80 ± 0,07	CaO	0,80 ± 0,07	CaO	0,80 ± 0,07	CaO	0,80 ± 0,07
MgO	0,08 ± 0,01	MgO	0,07 ± 0,01	MgO	0,08 ± 0,01	MgO	0,08 ± 0,01	MgO	0,08 ± 0,01	MgO	0,08 ± 0,01
P	0,047 ± 0,00	P	0,05 ± 0,00	P	0,047 ± 0,00	P	0,047 ± 0,00	P	0,047 ± 0,00	P	0,047 ± 0,00
B2	0,46 ± 0,03	B2	0,45 ± 0,04	B2	0,45 ± 0,03	B2	0,45 ± 0,03	B2	0,45 ± 0,03	B2	0,45 ± 0,03
Pelota - Fisica				Pelota - Fisica				Pelota - Fisica			
+ 19,0 mm	0,0 ± 0,1	+ 19,0 mm	0,1 ± 0,1	+ 19,0 mm	0,08 ± 0,12	+ 19,0 mm	0,08 ± 0,12	+ 19,0 mm	0,08 ± 0,12	+ 19,0 mm	0,08 ± 0,12
- 19,0 + 16,0 mm	2,7 ± 1,0	- 19,0 + 16,0 mm	4,1 ± 1,5	- 19,0 + 16,0 mm	3,40 ± 1,28	- 19,0 + 16,0 mm	3,40 ± 1,28	- 19,0 + 16,0 mm	3,40 ± 1,28	- 19,0 + 16,0 mm	3,40 ± 1,28
- 16,0 + 12,5 mm	35,2 ± 5,2	- 16,0 + 12,5 mm	35,9 ± 3,4	- 16,0 + 12,5 mm	35,56 ± 4,31	- 16,0 + 12,5 mm	35,56 ± 4,31	- 16,0 + 12,5 mm	35,56 ± 4,31	- 16,0 + 12,5 mm	35,56 ± 4,31
- 12,5 + 9,0 mm	56,3 ± 5,3	- 12,5 + 9,0 mm	54,1 ± 4,0	- 12,5 + 9,0 mm	55,22 ± 4,66	- 12,5 + 9,0 mm	55,22 ± 4,66	- 12,5 + 9,0 mm	55,22 ± 4,66	- 12,5 + 9,0 mm	55,22 ± 4,66
- 9,0 + 8,0 mm	3,3 ± 0,8	- 9,0 + 8,0 mm	3,3 ± 0,6	- 9,0 + 8,0 mm	3,35 ± 0,72	- 9,0 + 8,0 mm	3,35 ± 0,72	- 9,0 + 8,0 mm	3,35 ± 0,72	- 9,0 + 8,0 mm	3,35 ± 0,72
- 8,0 + 6,3 mm	1,3 ± 0,4	- 8,0 + 6,3 mm	1,5 ± 0,4	- 8,0 + 6,3 mm	1,40 ± 0,37	- 8,0 + 6,3 mm	1,40 ± 0,37	- 8,0 + 6,3 mm	1,40 ± 0,37	- 8,0 + 6,3 mm	1,40 ± 0,37
- 6,3 mm	1,0 ± 0,2	- 6,3 mm	1,0 ± 0,2	- 6,3 mm	1,01 ± 0,21	- 6,3 mm	1,01 ± 0,21	- 6,3 mm	1,01 ± 0,21	- 6,3 mm	1,01 ± 0,21
- 16,0 + 8,0 mm	94,9 ± 1,0	- 16,0 + 8,0 mm	93,4 ± 1,5	- 16,0 + 8,0 mm	94,12 ± 1,22	- 16,0 + 8,0 mm	94,12 ± 1,22	- 16,0 + 8,0 mm	94,12 ± 1,22	- 16,0 + 8,0 mm	94,12 ± 1,22
TAMBORAMENTO	93,9 ± 0,8	TAMBORAMENTO	94,3 ± 0,7	TAMBORAMENTO	94,11 ± 0,72	TAMBORAMENTO	94,11 ± 0,72	TAMBORAMENTO	94,11 ± 0,72	TAMBORAMENTO	94,11 ± 0,72
ABRASÃO	5,2 ± 0,7	ABRASÃO	4,8 ± 0,6	ABRASÃO	5,02 ± 0,65	ABRASÃO	5,02 ± 0,65	ABRASÃO	5,02 ± 0,65	ABRASÃO	5,02 ± 0,65
COMPRESSÃO	298,2 ± 110,9	COMPRESSÃO	310,7 ± 112,4	COMPRESSÃO	304 ± 112	COMPRESSÃO	304 ± 112	COMPRESSÃO	304 ± 112	COMPRESSÃO	304 ± 112
CCS < 200 kg	19,6 ± 5,9	CCS < 200 kg	16,3 ± 4,7	CCS < 200 kg	18 ± 5,31	CCS < 200 kg	18 ± 5,31	CCS < 200 kg	18 ± 5,31	CCS < 200 kg	18 ± 5,31
CCS < 150 kg	11,1 ± 4,5	CCS < 150 kg	8,3 ± 2,8	CCS < 150 kg	10 ± 3,66	CCS < 150 kg	10 ± 3,66	CCS < 150 kg	10 ± 3,66	CCS < 150 kg	10 ± 3,66
CCS < 100 kg	4,8 ± 2,7	CCS < 100 kg	4,2 ± 6,0	CCS < 100 kg	4 ± 4,34	CCS < 100 kg	4 ± 4,34	CCS < 100 kg	4 ± 4,34	CCS < 100 kg	4 ± 4,34

Fig. 7.26: Médias das usinas I e II

Estes cálculos são efetuados para cada uma das quotas a serem embarcadas na carga, resultando nas médias de cada item de produção para cada quota. Estas médias são então ponderadas pela quantidade a ser recuperada de cada quota. Pelo exemplo da figura 7.25 serão recuperadas 39632 ton da Quota 1 e 13985 ton da Quota 2, que seriam os respectivos pesos.

A figura 7.28 mostra na coluna “Blend” os resultados das médias de cada quota ponderadas pelas quantidades recuperadas de cada uma. Este seria o valor estimado para a mistura

Início da campanha: 27/07 0:00		Término da Campanha: 31/07 23:59	
Produto: PBF/MB45		Cliente: NINGBO IRON	
Dados de produção			
Data	Usina 1	Usina 2	Produção Usina 1 =
02/07/07	20060	20990	40220
03/07/07	20160	18770	Produção Usina 2 =
			39760
			Total = 79980

Fig. 7.27: Produção das usinas

se não houvesse degradação do minério no pátio. No entanto, como esse efeito ocorre, é necessário estimar sua influência na qualidade final da mistura. Isso é feito aplicando-se um fator de correção para cada um dos itens de qualidade. Estes fatores de correção, para cada item de controle de cada produto, são levantados historicamente a cada ano e utilizados como uma aproximação aceitável, apesar de não levar em consideração diversas correlações que resultam na degradação dos produtos. A coluna “Estimado” da figura 7.28 apresenta a qualidade esperada da mistura aplicando-se os fatores de correção em cada um dos itens de controle presentes na coluna “Blend”. A coluna “Embarque” apresenta as medidas para cada item de controle efetivamente obtidas no embarque.

Navio:	KYZIKOS					
Cliente:	NINGBO IRON		Produto:	PBF/MB45		
Resumo da operação (Médias)						
Descrição	Meta	Campanha	Blend	Estimado	Embarque	Sit.
H ₂ O	2,55	x	x	x	1,27	Ok
-6,3mm	2,00	1,01	0,98	1,38	1,3	Ok
+16mm	x	3,47	3,58	x	3	x
-16+8,0mm	89,40	94,12	94,04	92,94	94,40	Ok
Compressão	279	304	304	291	300	Ok
Tamboramento	93,4	94,1	94,1	93,8	94	Ok
Abrasão	5,8	5,0	5,0	5,2	5,3	Ok
Fe	67,08	67,51	67,40	67,35	67,40	Ok
FeO	x	0,24	0,27	0,30	0,20	x
SiO ₂	x	1,77	1,83	1,82	1,82	x
Al ₂ O ₃	x	0,55	0,56	0,58	0,55	x
CaO	x	0,80	0,82	0,79	0,84	x
MgO	x	0,08	0,08	0,08	0,09	-
P	0,05	0,047	0,047	0,046	0,049	Ok
Bas2	x	0,45	0,45	0,44	0,46	x

Fig. 7.28: Qualidade da mistura

Uma outra dificuldade no gerenciamento do pátio é o fechamento do balanço de massa entre o produto empilhado no pátio e o produto embarcado nos navios. Atualmente, são feitas duas medidas principais de tonelagem: uma na saída das usinas (realizada juntamente com as medidas de qualidade) e uma realizada através do calado do navio para o produto embarcado. O problema ocorre pois essas medidas são bastantes discrepantes, tornando necessário o ajuste dos estoques no pátio de três em três meses através de

medidas topográficas.

7.2.3 Identificar sistemas externos

Nesta seção, é apresentada a interface do sistema de amplificação de inteligência, com os sistemas externos já existentes no ambiente computacional da Samarco.

Para auxiliar o operador na tomada de decisão, o sistema deve receber diversas informações sobre o estado do pátio, demanda e produção e, após processá-las, gerar as programações das operações no pátio, assim como diversos relatórios.

Serão descritas as entradas necessárias para a alimentação do sistema de amplificação de inteligência, especificando de quais sistemas externos essas entradas serão coletadas. Além disso, será descrita a forma como o sistema de amplificação de inteligência deve interagir com os sistemas externos para apresentar as programações de operações e os relatórios.

Entradas

Para o sistema de amplificação de inteligência, são necessários dados de entrada informando o estado do pátio em um determinado momento, assim como as demandas e produções previstas para o futuro.

A lista a seguir mostra as principais entradas necessárias para a alimentação do sistema:

- Demanda (entradas utilizadas pelo sistema de amplificação de inteligência para prever a demanda e atendê-la através da programação das operações de carregamento)
 1. fila de navios: cada navio da fila tem uma data estimada de chegada, assim como as cargas a serem embarcadas. Cada carga é composta de uma determinada tonelagem de produto para um determinado cliente;
 2. orientações de embarque: especifica as medidas de qualidade do produto exigidas para cada cliente.
- Produção (entradas utilizadas pelo sistema de amplificação de inteligência para prever a quantidade e tipo de produto que sai das usinas e gerar a programação das operações de empilhamento necessárias para armazenar este produto)
 1. Atas de campanha: especificam os produtos e períodos de produção que devem ser seguidos pela usina.
- Estado do pátio (entradas utilizadas pelo sistema de amplificação de inteligência para manter e atualizar a situação dos diversos elementos do pátio em um determinado momento)
 1. situação das pilhas: para cada pilha, é necessário informações sobre sua baliza inicial e final, o tipo de produto armazenado, assim como a qualidade e quantidade;
 2. situação das máquinas: para cada máquina, é necessário informações sobre seu posicionamento atual, se está operando ou em manutenção, taxa de operação, etc;

3. produção atual: as campanhas são produções desejadas para as usinas. No entanto, para que o sistema possa auxiliar o operador na tomada de decisão, é necessário informações sobre a produção real sendo realizada em um determinado momento. Essa informação é gerada através das medidas bi-horárias de qualidade e quantidade geradas na saída das usinas.

A figura 7.29 apresenta o diagrama de como estas diversas entradas são geradas e armazenadas pelos sistemas externos durante o processo de produção.

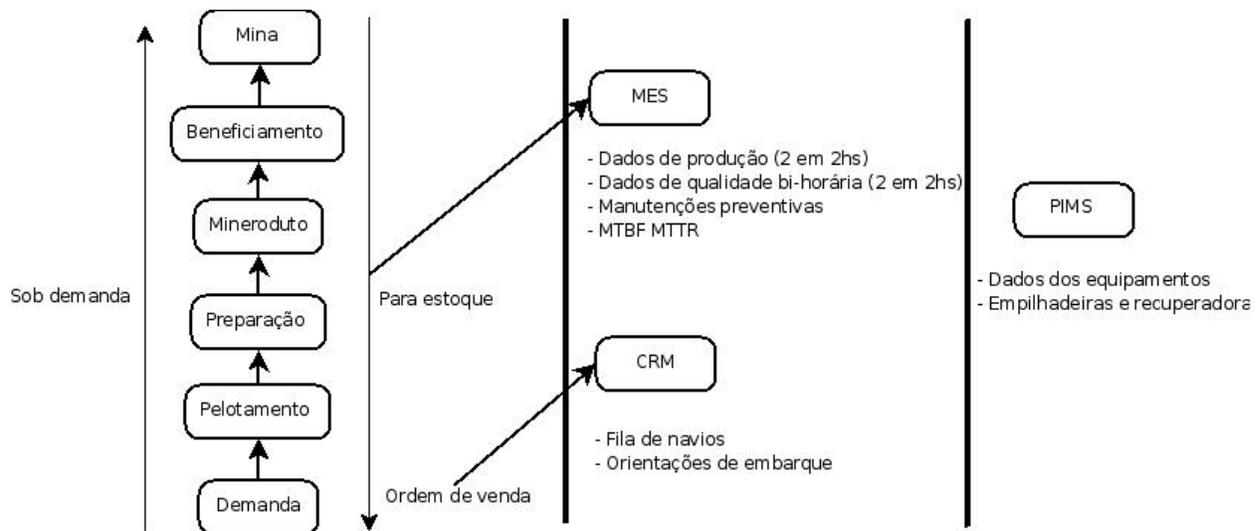


Fig. 7.29: Entradas geradas pelos sistemas externos durante o processo de produção

Existem dois fluxos principais de informação durante o processo: “sob demanda”⁷ e “para estoque”⁸. O primeiro fluxo refere-se ao pedido de produto para a mina dado as demandas da Estocagem e Embarque. O segundo fluxo refere-se às produções efetivamente realizadas pela mina e transportadas da Estocagem e Embarque. Como é possível observar, as informações são armazenadas nos sistemas externos durante o fluxo “para estoque”. Para o escopo do sistema de amplificação de inteligência, serão necessárias interfaces com três sistemas externos:

- MES: armazena as seguintes informações em base de dados Oracle:
 1. dados de produção e qualidade na saída das usinas (disponíveis a cada duas horas);
 2. dados históricos das manutenções preventivas e corretivas;
- CRM: armazena a fila de navios, assim como as orientações de embarque em base de dados Oracle;
- PIMS: armazena os dados relativos ao posicionamento e operação das máquinas no pátio.

Assim, a maioria das informações necessárias para alimentar o sistema de amplificação de inteligência está disponível em sistemas já em operação no ambiente computacional da Samarco.

⁷O jargão utilizado na Samarco é “make to order”.

⁸O jargão utilizado na Samarco é “make to stock”

A principal deficiência existe no caso da situação das pilhas: não existe nenhuma informação armazenada nos sistemas indicando a posição, tonelagem, produto ou qualidade das pilhas presentes no pátio. Além disso, problemas na calibração de aparelhos tornam pouco confiáveis as informações provenientes do PIMS sobre o posicionamento e operação das máquinas.

Devido à estes fatos, o projeto para a construção do sistema de amplificação de inteligência deve supor que todas as entradas sobre a situação das pilhas e máquinas no pátio devem ser inicialmente alimentadas manualmente. O projeto deve prever também a possibilidade de leitura automática destas informações, realizada assim que a equipe de manutenção disponibilizar dados mais confiáveis.

As campanhas deverão ser lidas, a princípio, de entradas manuais, já que estão armazenadas em arquivos em formato PDF na intranet da Samarco.

Saídas

Com relação às saídas, o sistema deverá produzir relatórios semelhantes aos relatórios existentes hoje na operação como, por exemplo, relatório de turno, movimentações de empilhamento e recuperação, qualidade das pilhas, etc. Estes relatórios serão todos gravados em uma base Oracle exclusiva do sistema de amplificação de inteligência, e não haverá nenhuma saída gravada nas bases de dados dos outros sistemas presentes no ambiente computacional já existente.

Além dos relatórios, uma saída importante do sistema é a apresentação do estado do pátio em uma interface gráfica computacional, conforme descrito na seção 7.2.2.

7.2.4 Prototipar motores de solução

Optou-se por não desenvolver um protótipo do motor de solução que resolveria o planejamento automático de recuperação e empilhamento, pelo fato de já existir um algoritmo que resolve problemas semelhantes de gerenciamento de pátio. Este algoritmo foi desenvolvido por Molck (2002), para resolver um problema similar ao da Samarco. Durante a especificação dos requisitos de sistema do projeto, faz-se necessário apenas adaptá-lo um pouco à situação encontrada na Samarco.

Neste primeiro instante de especificação dos requisitos de amplificação de inteligência, não é possível saber quais são exatamente os multi-objetivos do planejamento automático, pois nunca se operou planejando na Samarco. Esta informação deverá ser levantada durante a evolução dos requisitos de sistema, na fase de implementação, quando será possível realizar o levantamento desses objetivos, extraíndo-os de um eventual planejamento manual dos operadores.

7.2.5 Desenvolver diagrama de fluxo cognitivo

O diagrama de fluxo cognitivo (DFC) ilustra os processos cognitivos e as unidades de conhecimento que fazem parte do fluxo cognitivo do especialista humano que resolve o problema de planejamento e execução dinâmica em questão. O primeiro passo é entender este fluxo

cognitivo, para posteriormente modelar modificações para amplificá-lo. Este entendimento é alcançado analisando a operação do especialista humano que resolve o problema e aprendendo suas atividades, para que seu repertório hierárquico de capacidades seja representado na forma de atividades cognitivas.

Na Samarco, foi analisado o repertório hierárquico de capacidades dos operadores, que trabalham na sala de controle do departamento de estocagem e embarque, e dos engenheiros de processo e produção, responsáveis pelas atividades dos controladores e pelos planos de empilhamento e recuperação que são realizados por eles. Essa análise resultou num conjunto de atividades cognitivas e unidades de conhecimento que foram agrupadas na forma do diagrama de fluxo cognitivo da figura 7.30.

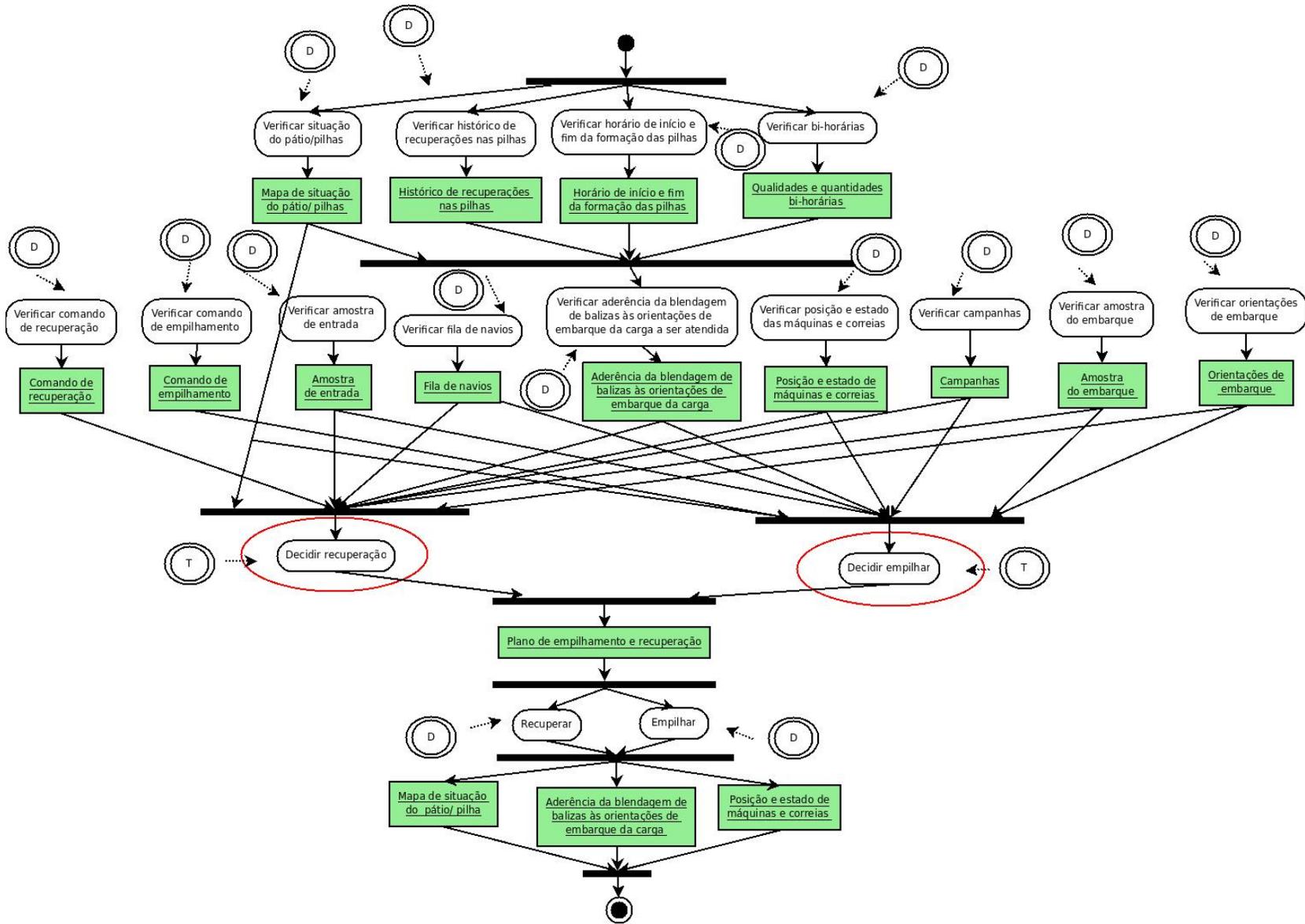


Fig. 7.30: Diagrama de fluxo cognitivo para o problema de estocagem e embarque da Samarco

Grande parte das atividades cognitivas realizadas pelo especialista são mecânicas (diádicas), e muitas delas são apenas buscas por conhecimentos que já existem em algum lugar. Essas atividades são realizadas pelo especialista para compilar todas as unidades de conhecimento que ele precisa para realizar as atividades mais importantes de criação de conhecimento, uma espécie de preparação para realizar suas atividades principais. Entre essas atividades estão:

- Verificar histórico de recuperações nas pilhas: dará origem à unidade de conhecimento “histórico de recuperações nas pilhas”. Essa unidade de conhecimento será utilizada para construir a aderência da blendagem de balizas às orientações de embarque da carga, pois a recuperação de material em uma pilha influencia na distribuição das qualidades, já que as bolas são misturadas e tiradas de posição no processo;
- Verificar horário de início e fim de formação das pilhas: dará origem à unidade de conhecimento “horário de início e fim da formação das pilhas”. Essa informação é necessária para, juntamente com as bi-horárias, determinar quais qualidades foram empilhadas em determinada pilha.
- Verificar situação do pátio e das pilhas: dará origem à unidade de conhecimento “mapa de situação do pátio e pilhas”, que é o mapa com os desenhos das pilhas informando sua posição e quantidade de material no pátio. Os controladores constroem esse mapa na parede da sala de controle, desenhando-o na mão. Ele está representado na figura 7.22.
- Verificar qualidades e quantidades bi-horárias: dará origem a unidade de conhecimento “qualidade e quantidades bi-horárias”, contendo as informações reveladas de duas em duas horas da quantidade e qualidade do material que saiu da usina.
- Verificar comando de recuperação: dará origem à unidade de conhecimento “comando de recuperação”, representando algum comando mandatário de lugares de onde se deve ou não recuperar. Pode ser uma ordem de um superior para que seja recuperado um determinado produto de uma determinada posição no pátio.
- Verificar fila de navios: dará origem à unidade de conhecimento “fila de navios”, que representa os navios a serem atendidos na sequência, de acordo com o plano de vendas da empresa.
- Verificar orientações de embarque: dará origem à unidade de conhecimento “orientações de embarque”, com as orientações de quantidade e qualidade para as cargas dos navios a serem atendidos.
- Verificar amostra de entrada: dará origem à unidade de conhecimento “amostra de entrada”, com informações de quantidade e qualidade da amostra que vem da usina para o pátio.
- Verificar amostra de embarque: dará origem à unidade de conhecimento “amostra de embarque”, com informações de quantidade e qualidade do produto que está sendo embarcado.
- Verificar posição e estado das máquinas e correias: dará origem a unidade de conhecimento “posição e estado de máquinas e correias”, com informações sobre onde se encontram dispostas no pátio as máquinas e correias e seus estados de operação (ociosa, operação ou manutenção).

- Verificar campanhas: dará origem à unidade de conhecimento “campanhas”, com informações sobre a campanha atual e sobre as futuras campanhas de produção das usinas.
- Verificar comando de empilhamento: dará origem a unidade de conhecimento “comando de empilhamento”, com informações sobre possíveis comandos mandatórios superiores relativos à empilhamento, como uma ordem superior para não empilhar em determinada área do pátio, por exemplo.
- Recuperar: atualiza as unidades de conhecimento “mapa de situação do pátio/pilhas”, “aderência da blendagem de balizas às orientações de embarque da carga” e “posição e estado das máquinas e correia”, pois uma eventual recuperação modifica seus referentes no mundo real. Essas unidades de conhecimento foram convenientemente repetidas logo após a recuperação apenas para deixar o diagrama mais limpo e claro no entendimento, mas de fato são os mesmos elementos que apareceram antes no diagrama.
- Empilhar: atualiza as unidades de conhecimento “mapa de situação do pátio/pilhas”, “aderência da blendagem de balizas às orientações de embarque da carga” e “posição e estado das máquinas e correias”, pois um eventual empilhamento modifica seus referentes no mundo real.

Há uma atividade mecânica que representa uma criação de conhecimento. Essa atividade ilustra como o fato de uma atividade ser mecânica ao invés de inteligente não significa que ela é menos importante, ou que deve ser feita de maneira menos eficiente: significa apenas que nela não se faz necessário estimar estados futuros, gerando algum tipo de plano e estados que são comparados com estes estados futuros para a tomada de decisão. Ao invés disso, atividades mecânicas envolvem simples cálculos e modelos matemáticos, como uma função que, aplicada a uma determinada entrada, fornece uma saída determinística, livre de contexto, não importando a situação do mundo. Por esse motivo, essas atividades podem ser completamente realizadas pelo SAI, liberando a mente humana da tarefa. Neste caso, a atividade é:

- Verificar aderência da blendagem de balizas às orientações de embarque da carga a ser atendida: nesta atividade, o agente procura calcular se a qualidade resultante da blendagem de diferentes grupos de balizas da mesma pilha ou de pilhas diferentes atendem a qualidade de cada um dos itens de controle da orientação de embarque de uma carga a ser atendida, para posterior tomada de decisão com relação ao empilhamento ou à recuperação. Isto é feito a princípio a partir do conhecimento das “qualidades e quantidades bi-horárias”. Com essa informação, é possível traçar um gráfico como o da figura 7.5 para cada um dos itens de controle. Perceba no gráfico que há uma meta, relativa a orientação de embarque para um determinado item de controle. Na região entre 0 e 8 horas, as qualidades desse item de controle foram ruins. Isso pode ajudar na tomada de decisão de recuperação, por exemplo, pois tendo conhecimento do “horário de início e fim de formação das pilhas”, pode-se saber que pilha foi formada com aquele material, e identificá-la no “mapa de situação do pátio e das pilhas”. O “histórico de recuperações nas pilhas” também influencia na montagem desse mapa, já que as recuperações modificam pelotas de posição e as misturam, alterando a formação da qualidade com relação aos itens de controle em cada uma das posições. O conhecimento desse mapa de qualidades auxilia

no empilhamento, evitando que material ruim vindo da usina seja empilhado e uma pilha de boa qualidade, ou que material de boa qualidade seja colocado em uma pilha ruim. A aderência é calculada a partir de uma comparação da qualidade exigida pelas orientações de embarque e a qualidade média resultante da blendagem para cada um dos itens de controle.

Há duas atividades classificadas como inteligentes (triádicas) no repertório hierárquico de capacidades dos especialistas do diagrama de fluxo cognitivo da figura 7.30:

- Decidir recuperar: esta atividade consiste em agrupar todas as informações extraídas de fontes externas e criadas na mente do especialista e gerar um plano com as posições e pilhas que devem ser recuperadas para atender as cargas dos navios atracados. É uma atividade inteligente, ou seja, que necessita de estimação de estados futuros, formulação de alternativas, avaliação das alternativas e escolha da mais adequada, conforme o contexto do mundo real no qual elas se inserem. Por isso, essa atividade necessitará ser realizada em conjunto com uma mente humana num eventual sistema de amplificação de inteligência. Juntamente com “decidir empilhar”, é responsável por gerar o plano de empilhamento e recuperação.
- Decidir empilhar: responsável por criar a unidade de conhecimento “plano de empilhamento e recuperação” juntamente com a atividade “decidir recuperar”. A atividade consiste em determinar em que região do pátio deve ser empilhado o produto que está vindo da usina, agrupando todas as informações que são necessárias a esta atividade para uma decisão inteligente. Assim como a decisão de recuperação, as decisões são altamente influenciadas pelo contexto dinâmico do mundo real na qual estão inseridas.

7.2.6 Identificar tipos de tomada de decisão

Os tipos de tomada de decisão estão identificados para cada uma das atividades do repertório hierárquico de capacidades do especialista no diagrama de fluxo cognitivo da figura 7.30. As atividades mecânicas (diádicas), estão identificadas pela letra D. As atividades inteligentes (triádicas) estão representadas pela letra T. Não há atividades tipicamente aleatórias (monádicas), que seriam representadas pela letra M.

7.2.7 Identificar pontos de amplificação de inteligência

Na identificação dos pontos de amplificação de inteligência, procura-se identificar os pontos onde se faz necessário um diálogo entre agente computacional e humano para resolução da atividade cognitiva. Todas as outras atividades serão exclusivas do SAI, sem a participação humana, deixando a mente humana livre apenas para os pontos de amplificação identificados.

No diagrama da figura 7.30, estão identificados dois pontos de amplificação de inteligência, ao redor das atividades inteligentes. Eles estão representados por círculos. Isso significa que, nestas atividades de decisão de empilhamento e decisão de recuperação, atuarão agentes computacionais presentes em um subsistema de agentes do SAI, carregando motores de solução,

capazes de encontrar a solução e estabelecer um diálogo com o especialista humano através da interface de diálogo para receber informações dos referentes do mundo real. As outras atividades, todas mecânicas, serão transparentes ao usuário humano, que receberá diretamente através da interface de diálogo as unidades de conhecimento resultantes delas.

7.2.8 Modelar amplificação de inteligência em Redes Semiônicas

Se o DFC desenhado na figura 7.30 procura representar o repertório hierárquico de capacidades do especialista na resolução do problema, a modelagem com Redes Semiônicas procura representar o repertório hierárquico de capacidades do futuro sistema de amplificação de inteligência que irá amplificar a mente do especialista, inclusive modelando o diálogo entre os agentes computacionais e o agente humano, com a vantagem de ser uma simulação dinâmica. Essa etapa serve também para validar o DFC desenhado anteriormente.

O primeiro passo é traduzir o DFC para um modelo de Rede Semiônica. As atividades mecânicas serão posicionadas em lugares ativos, as unidades de conhecimento em lugares passivos, e os pontos de amplificação de inteligência serão superlugares, nos quais se modelará posteriormente uma rede contendo agente humano e computacional, modelando a interação entre eles. A rede que traduz o DFC da figura 7.30 está representada na figura 7.31.

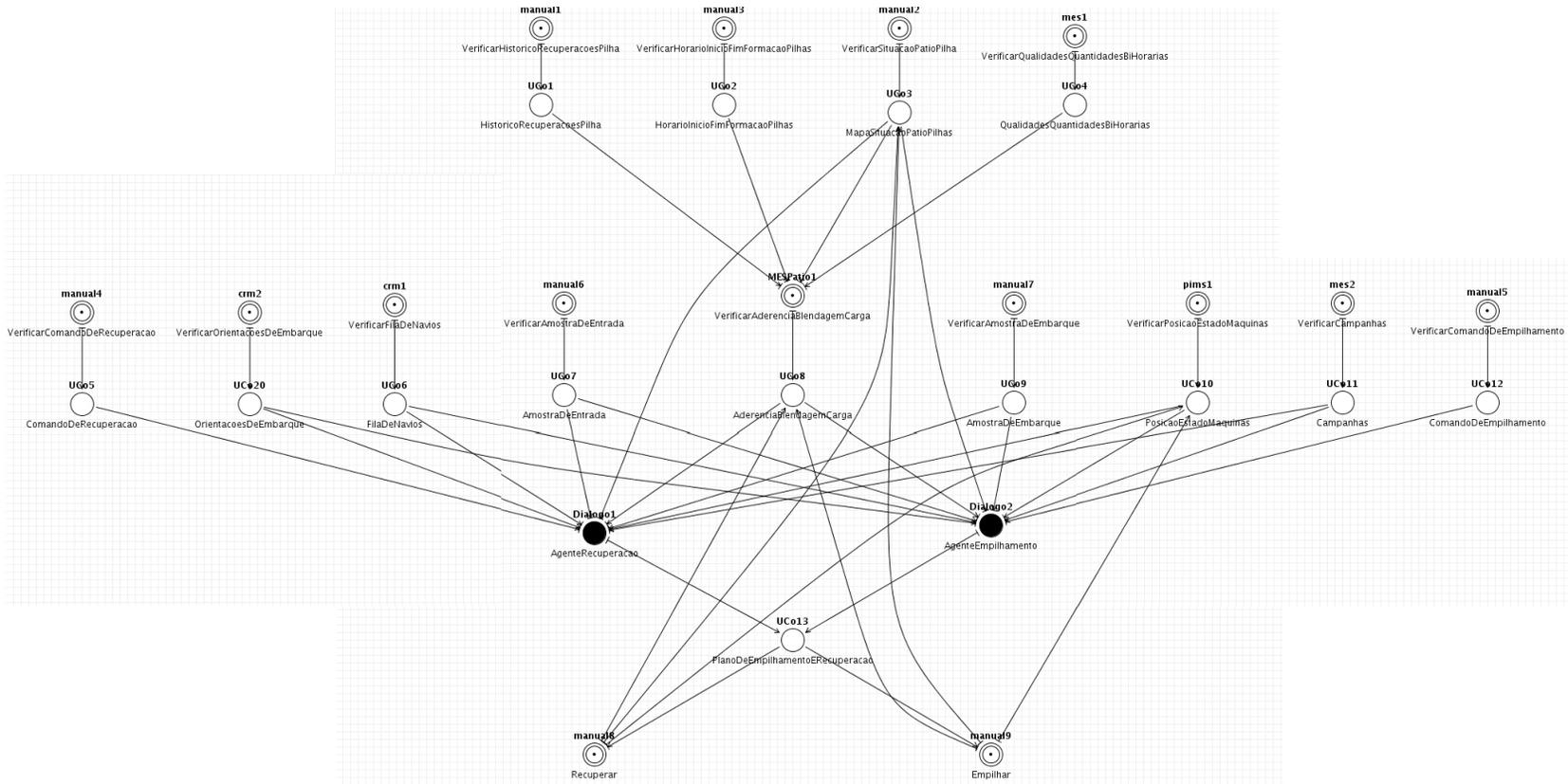


Fig. 7.31: Rede Semiônica para DFC da figura 7.30

Esta primeira rede da figura 7.31 valida o DFC e o simula dinamicamente. Dela serão extraídos, em uma etapa posterior, os tipos de lugares que representarão elementos específicos do sistema de amplificação de inteligência. Após esta primeira rede, faz-se necessário construir as redes que representam os superlugares onde atuarão os agente computacionais, estabelecendo um diálogo com o agente humano em busca da solução. A modelagem deste diálogo deve identificar as unidades de conhecimento modificadas pelos agentes humano e computacional na interface de diálogo para conseguir chegar até a solução.

A rede referente a atividade de decidir empilhar está representada na figura 7.32. A rede referente a atividade de decidir recuperar está representada na figura 7.33.

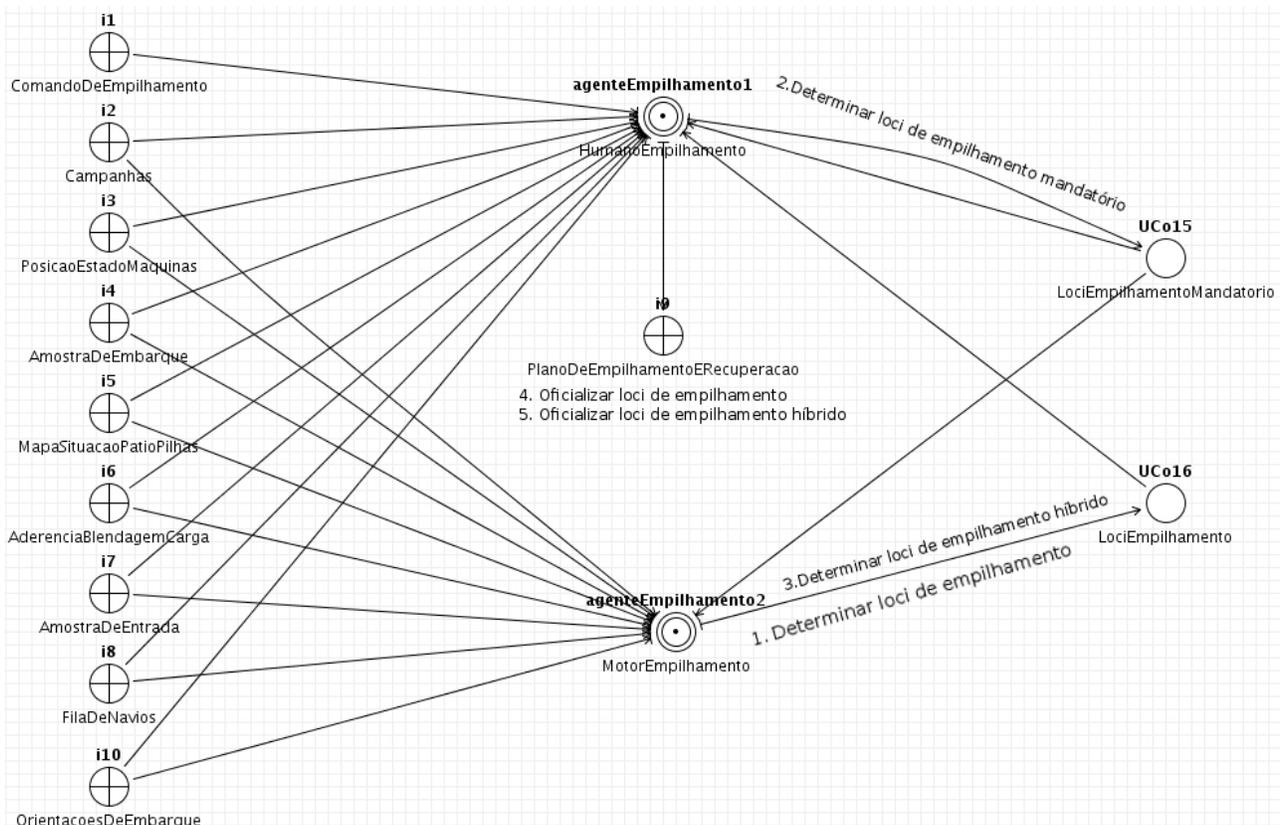


Fig. 7.32: Rede Semiônica referente a atividade de decidir empilhar

Para a rede referente a atividade de decidir empilhar, o diálogo traçado entre agente humano e computacional está representado na sequência de atividades cognitivas e geração de unidades de conhecimento dessa rede, que está ilustrada na figura 7.34.

O diálogo traçado neste exemplo de subrede, na figura 7.34, pode ser entendido da seguinte maneira: em (a), o agente computacional, a partir do conhecimento das campanhas, posição e estado das máquinas, amostra de embarque, mapa de situação das pilhas e do pátio, aderência da blendagem à carga, amostra de entrada, fila de navios e orientações de embarque (unidades de conhecimento a esquerda do diagrama que estão conectadas ao agente, ou seja, disponíveis para seu acesso), sugere loci de empilhamento, representados pela UCo16. Em (b), o agente humano

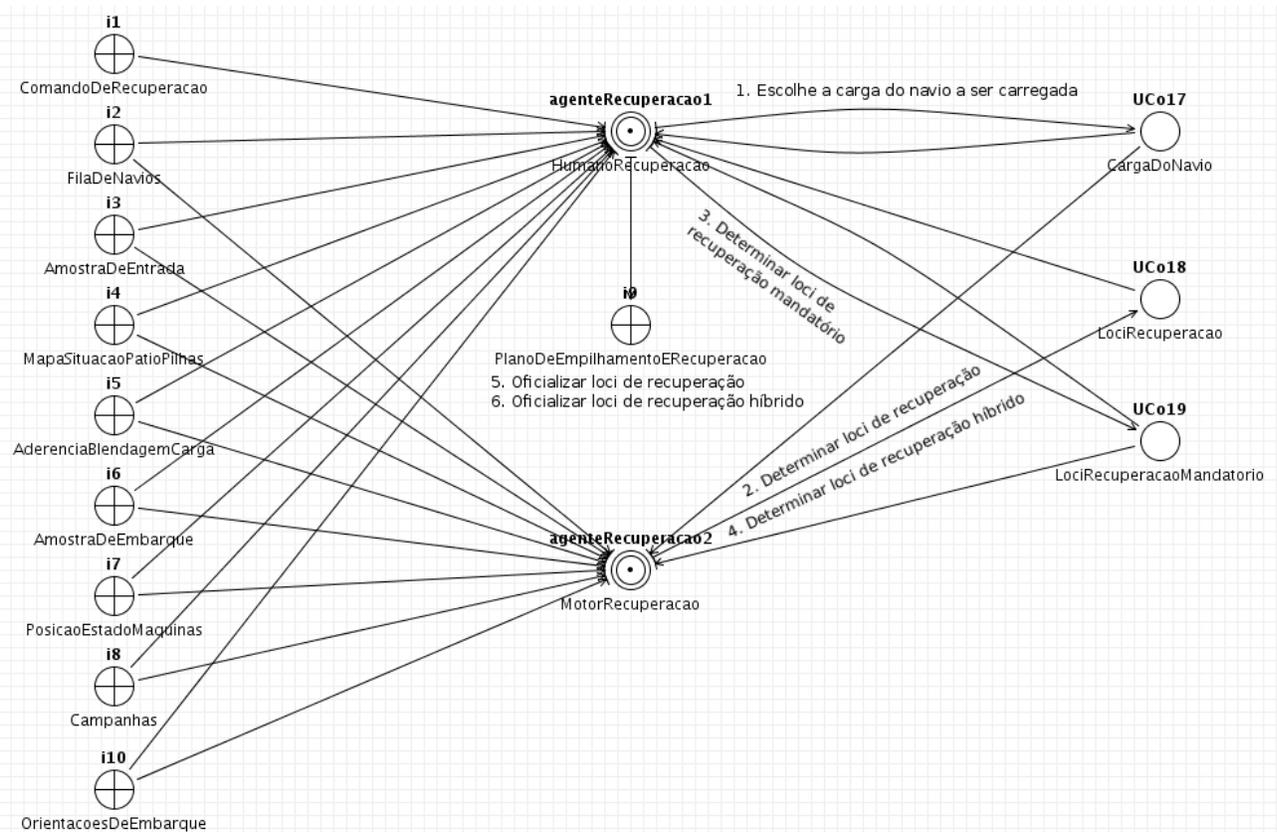


Fig. 7.33: Rede Semiônica referente a atividade de decidir recuperar

pode oficializar esses loci de empilhamento e gerar um plano de empilhamento e recuperação, ou pode sugerir loci de empilhamento mandatórios, representados pela UCo15, a partir das unidades de conhecimento que estão a sua disposição. Em (c), caso tenham sido gerados loci de empilhamento mandatórios, o agente computacional sugere loci de empilhamento híbridos. Em (d), a partir dos loci de empilhamento híbridos, o agente humano pode gerar novos loci de empilhamento mandatórios ou oficializar os lugares de empilhamento híbridos.

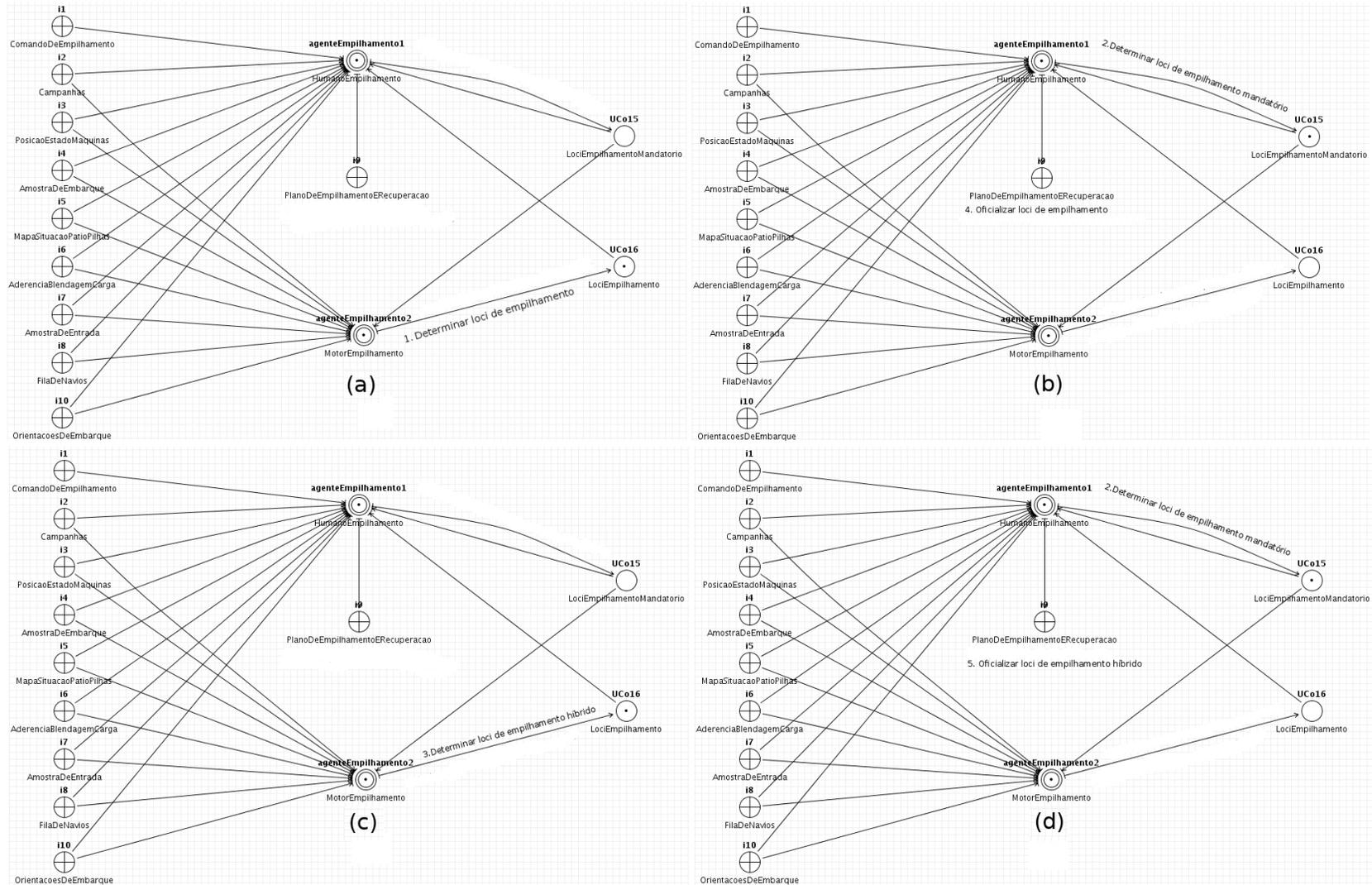


Fig. 7.34: Sequência de atividades cognitivas e geração de unidades de conhecimento da Rede Semiônica referente a atividade de decidir empilhar.

Estas redes modelam as unidades de conhecimento que os agentes computacionais e humanos compartilham até chegar na solução, que é um plano de empilhamento e recuperação. Além disso, há uma série de interações específicas que também estão identificadas, os chamados casos de amplificação de inteligência. As próximas atividades irão detalhar as unidades de conhecimento da rede e as interações entre os agentes.

A ferramenta SNToolkit gera um código para representar a rede semiônica montada. O código da rede das figuras 7.31, 7.32 e 7.33 se encontra no apêndice A.3. Se este código for utilizado no SNToolkit, é possível observar a rede e simulá-la através da ferramenta.

7.2.9 Identificar lugares nas Redes Semiônicas

Os lugares da Rede Semiônica são representações dos elementos do diagrama de fluxo cognitivo. A seguir, serão identificados os lugares da rede.

Lugares passivos

Conforme dito em A.2.9, os lugares passivos são as unidades de conhecimento do DFC, que serão representadas na interface de diálogo. São eles:

1. UCo1 - Histórico de recuperações na pilha (representação indireta).
2. UCo2 - Horário de início e fim de formação das pilhas (representação indireta).
3. UCo3 - Mapa de situação do pátio e das pilhas (representação direta).
4. UCo4 - Qualidades e quantidades bi-horárias (representação indireta).
5. UCo5 - Comando de recuperação (representação direta).
6. UCo6 - Fila de navios (representação direta).
7. UCo7 - Amostra de entrada (representação direta).
8. UCo8 - Aderência da blendagem às orientações de embarque da carga (representação direta).
9. UCo9 - Amostra de embarque (representação direta).
10. UCo10 - Posição e estado das máquinas e correias (representação direta).
11. UCo11 - Campanhas (representação direta).
12. UCo12 - Comando de empilhamento (representação direta).
13. UCo13 - Plano de empilhamento e recuperação (representação indireta).
14. UCo15 - Loci de empilhamento mandatório (representação direta).
15. UCo16 - Loci de empilhamento (representação direta).
16. UCo17 - Carga do navio (representação direta).
17. UCo18 - Loci de recuperação (representação direta).
18. UCo19 - Loci de recuperação mandatória (representação direta).
19. UCo20 - Orientações de embarque (representação direta).

Lugares ativos

Os lugares ativos armazenam os agentes semiônicos com as atividades cognitivas do DFC, funções que o sistema executará sozinho, transparentes para o usuário. Muitas dessas atividades representavam apenas buscas por conhecimentos externos, uma espécie de preparação para as atividades principais. Poupa-se este trabalho do especialista de preparação para o pensamento, buscando informações e fazendo cálculos, ao passar essas atividades para execução no SAI, liberando a mente do especialista para se concentrar nas atividades mais importantes. Essas funções que o sistema irá executar sozinho são:

1. Verificar histórico de recuperações nas pilhas (entrada de conhecimento manual)
2. Verificar horário de início e fim de formação das pilhas (entrada de conhecimento manual)
3. Verificar situação do pátio e das pilhas (entrada de conhecimento manual)
4. Verificar qualidades e quantidades bi-horárias (entrada de conhecimento do sistema externo MES)
5. Verificar comando de recuperação (entrada de conhecimento manual)
6. Verificar fila de navios (entrada de conhecimento do sistema externo CRM)
7. Verificar orientações de embarque (entrada de conhecimento do sistema externo CRM)
8. Verificar amostra de entrada (entrada de conhecimento manual)
9. Verificar aderência da blendagem de balizas às orientações de embarque da carga a ser atendida (criação de conhecimento)
10. Verificar amostra de embarque (entrada de conhecimento manual)
11. Verificar posição e estado de máquinas (entrada de conhecimento do sistema externo PIMS)
12. Verificar campanhas (entrada de conhecimento do sistema externo MES)
13. Verificar comando de empilhamento (entrada de conhecimento manual)
14. Recuperar (entrada de conhecimento manual)
15. Empilhar (entrada de conhecimento manual)

Superlugares

Os superlugares são os pontos de amplificação de inteligência do DFC. Nos pontos de superlugares, atuarão agentes computacionais do subsistema de agentes que irão dialogar com os agentes humanos para resolver atividades cognitivas inteligentes. São eles:

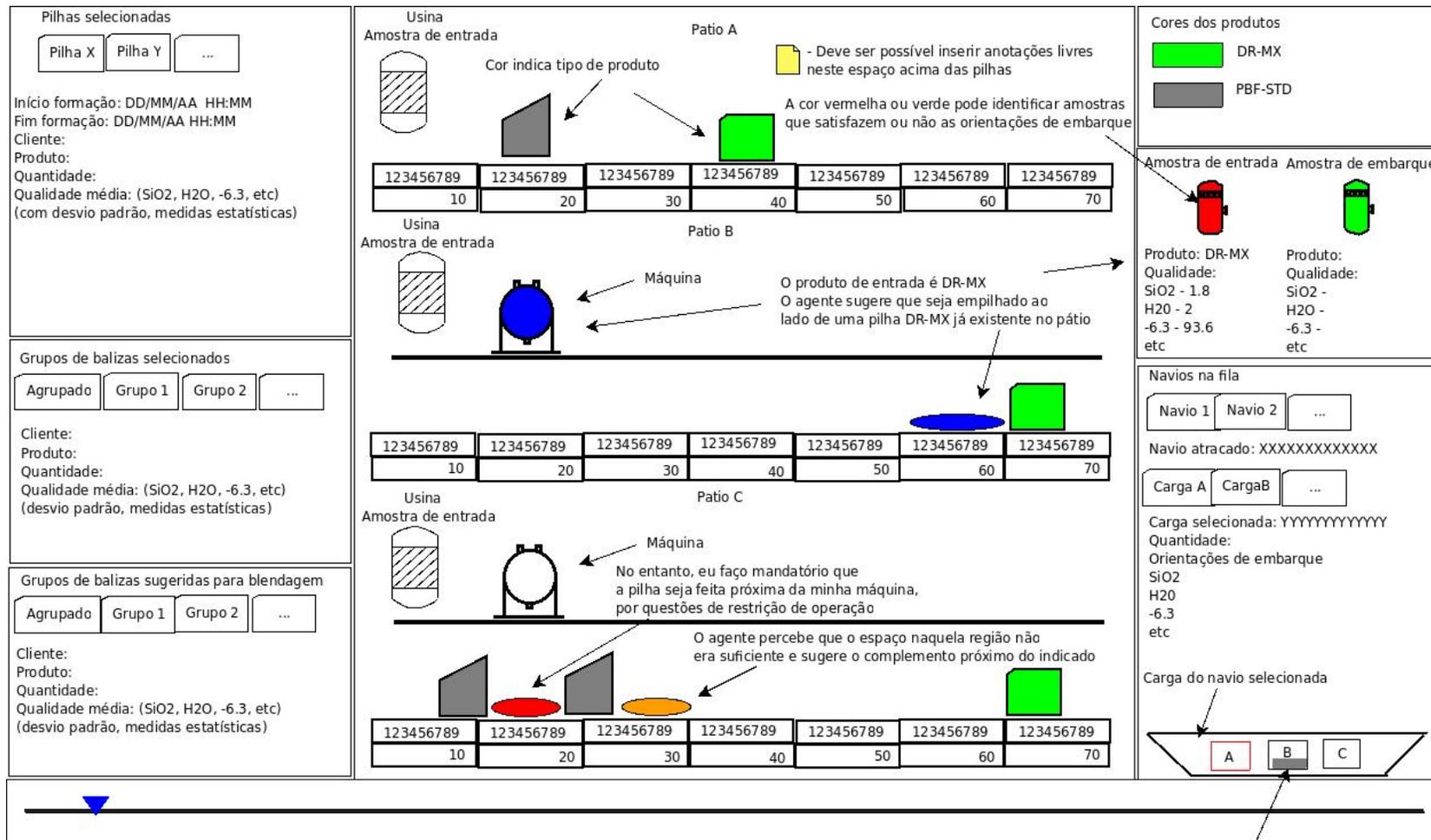
1. Decidir empilhar: motor de solução responsável por encontrar um plano de empilhamento e recuperação (UCo13).
2. Decidir recuperar: motor de responsável por encontrar um plano de empilhamento e recuperação (UCo13).

7.2.10 Identificar casos de amplificação de inteligência

Os casos de amplificação de inteligência estão numerados nas figuras 7.32 e 7.33. Eles estão detalhados no apêndice A.4.

7.2.11 Prototipar a interface de diálogo

Neste protótipo da interface de diálogo do MES Pátio, optou-se por prototipar apenas a tela principal, para melhor entendimento das unidades de conhecimento de representação direta e dos casos de amplificação de inteligência. Convencionou-se chamar essa tela principal de diagrama de situação do pátio (DSP). O protótipo foi feito esquematicamente, e não funcionalmente, ou seja, não é uma implementação de código, apenas uma diagramação. O DSP possui funcionalidades de consulta, de empilhamento e de recuperação, podendo o usuário alternar a utilização dessas funcionalidades na utilização do sistema. Primeiramente, descreve-se aqui o funcionamento do DSP referente ao empilhamento, que está relacionado aos casos de amplificação de inteligência da atividade de decidir o empilhamento, modelados na rede da figura 7.32. Nesta descrição, já será possível identificar como as unidades de conhecimento de representação direta da seção 7.2.9 estarão representadas. Esta parte do protótipo está ilustrada na figura 7.35



Obs: - as campanhas e o comando de empilhamento estarão representados indiretamente, podendo ser acessados através de menus.
 - Juntamente com a informação dos loci de empilhamento, deve ser fornecida a informação completa de máquinas, atividades e tempos para realizar o empilhamento, ou seja, todo o planejamento.
 - A oficialização será feita através de um menu.

Barra de eventos

- Cargas do navio
 - Quantidade de cada carga já carregada é mostrada

Fig. 7.35: Protótipo da interface de diálogo diagrama de situação do pátio (DSP) sendo utilizado para empilhamento

Primeiramente, procurou-se modelar a representação das unidades de conhecimento de representação direta, identificadas na seção 7.2.9. Essas unidades de conhecimento de representação direta são as unidades de conhecimento utilizadas no diálogo entre agente humano e computacional, ou seja, as unidades de conhecimento que aparecem nas redes que modelam os casos de amplificação de inteligência das figuras 7.32 e 7.33. É importante notar que, durante a prototipação da interface de diálogo, pode-se definir que algumas estruturas de representação direta acabem por ter uma representação indireta no sistema final. As unidades de conhecimento de representação direta identificadas foram:

- UCo3 - Mapa de situação do pátio e das pilhas: é o mapa desenhado pelos operadores para acompanhar a situação das pilhas no pátio, apresentado na figura 7.22. No DSP da figura 7.35, pode-se observar que essa unidade de conhecimento foi representada iconicamente da mesma maneira que os operadores desenhavam no quadro, marcando a posição das pilhas no pátio através das balizas. A altura dessas pilhas desenhadas automaticamente no DSP referenciam a quantidade de material que está contido nelas, e a cor referencia o tipo de material da pilha. A idéia é manter as mesmas cores atualmente utilizadas pelos operadores, que são mantidas já há décadas, por questões de cultura da Samarco. À medida que uma pilha for surgindo de um empilhamento ou sendo recuperada, a altura e o comprimento dela devem mudar, conforme a quantidade de material empilhado ou recuperado. Perceba que deve ser possível inserir anotações nos espaços acima das pilhas, conforme modelado no DSP da figura 7.35 por um ícone de anotação e conforme é costume dos operadores, e pode ser observado na figura 7.22. Algumas informações das pilhas são mostradas no canto superior esquerdo do DSP, uma parte delas já identificadas como unidades de conhecimento de representação indireta, como horário de início e fim da formação das pilhas, a quantidade, a qualidade média referente a cada item de controle com medidas estatísticas, entre outros.
- UCo5 - Comando de recuperação: optou-se por não representar diretamente a unidade de conhecimento comando de recuperação na confecção do protótipo, pelo fato dela aparecer de maneira espaçada, ou seja, nem sempre existir um comando de recuperação. Quando ele existe, é um comando fixo, que não muda muito em uma mesma interação de confecção de planos de empilhamento ou recuperação, sendo geralmente uma ordem superior aos operadores. Dessa maneira, não se faz necessário uma representação icônica do mesmo para que seja consultada o tempo inteiro pelo usuário. Fica, então, o comando de recuperação para ser representado em telas secundárias do sistema ou através de menus.
- UCo6 - Fila de navios: a fila de navios está representada no DSP no canto inferior direito, como pode-se observar na figura 7.35. Para cada um dos navios da fila, é apresentada uma série de informações, como suas cargas, quantidades e orientações de embarque.
- UCo7 - Amostra de entrada: a amostra de entrada é representada por um ícone no canto superior direito do DSP. Este ícone é colorido de acordo com a qualidade medida da amostra de entrada, ou seja, pode ficar vermelho caso a qualidade da amostra não satisfaça a ordem de embarque da carga selecionada. Abaixo do ícone, são apresentadas algumas informações sobre o produto da amostra de entrada e sua qualidade. Há também um ícone representando a amostra de entrada acima das pilhas, para que ela possa ser selecionada

para uma espécie de recuperação sem que o material seja empilhado, ou seja, para que o material que saia da usina seja mandado diretamente pro navio sendo carregado.

- UCo8 - Aderência da blendagem às orientações de embarque da carga: a aderência da blendagem às orientações de embarque da carga é visualizada através de um degradê de cores que pinta a carga a ser atendida. Ao selecionar a carga e um grupo de balizas, a distância da qualidade daquele grupo à meta de qualidade da carga se revela pela cor do ícone da carga e pelos valores de qualidade média de cada item de controle em comparação às metas da carga. Os itens que não atenderem às cargas devem ser destacados.

Perceba que, de acordo com as formas de empilhamento e recuperação, as pelotas podem se misturar um pouco, havendo uma fronteira razoavelmente incerta entre as qualidades empilhadas. De acordo com o tratamento matemático dado neste cálculo, a representação deve levar isto em conta.

- UCo9 - Amostra de embarque: a exemplo da amostra de entrada, a amostra de embarque está representada no canto superior direito do DSP através de um ícone, cuja cor representa o atendimento ou não às orientações de embarque. Informações de produto e qualidade da amostra de embarque ficam logo abaixo do ícone.
- UCo10 - Posição e estado das máquinas e correias: a posição e estado das máquinas e correias está representada no DSP através de ícones no meio de cada pátio, como pode ser observado nas figura 7.35. A posição das máquinas pode ser obtida através da marcação das balizas, e seu estado, assim como o das correias, através de suas cores. Uma cor amarela pode sinalizar que a máquina ou a correia está em manutenção, por exemplo.
- UCo11 - Campanhas: optou-se por não representar as campanhas diretamente, por não haver necessidade de consultar rapidamente esta informação o tempo inteiro. Para deixar o DSP mais limpo possível, as campanhas serão obtidas em telas secundárias ou através de menus.
- UCo12 - Comando de empilhamento: pelos mesmos motivos do comando de recuperação, convencionou-se durante a prototipação que a representação do comando de empilhamento seria considerada indireta.
- UCo16 - Loci de empilhamento: os loci de empilhamento sugeridos pelo sistema serão representados por elipses de uma determinada cor, que ficam piscando na posição das balizas nas quais se sugere empilhar o produto vindo para o pátio. No exemplo da figura 7.35, a elipse que representa os loci de empilhamento está de azul ou laranja. O laranja serve para loci híbridos, ou seja, considerando um mandatório do usuário humano, conforme modelado nos casos de amplificação de inteligência.
- UCo15 - Loci de empilhamento mandatório: : os loci de empilhamento mandatórios estão representados na figura 7.35 por elipses vermelhas, determinadas pelo usuário humano.
- UCo17 - Carga do navio: a unidade de conhecimento carga do navio está representada no canto inferior esquerdo por retângulos dentro da figura do navio selecionada. Pode-se selecionar uma dessas cargas por vez, que fica com uma cor diferente. À medida que esta carga vai sendo carregada no navio, uma barra de status vai sendo preenchida sobre o retângulo que representa a carga.

- UCo18 - Loci de recuperação: os loci de recuperação podem ser vistos na figura 7.36, que apresenta o DSP no modo de operação de recuperação. Os loci de recuperação sugeridos para uma determinada carga são grupos de balizas marcadas com uma determinada cor. No exemplo, estes grupos estão marcados de azul ou verde. A cor verde é usada no caso de loci de recuperação híbridos, ou seja, que consideram um mandatório do usuário, conforme modelado nos casos de amplificação de inteligência. No canto inferior esquerdo, são apresentadas informações desses agrupamentos de balizas selecionadas para recuperação.
- UCo19 - Loci de recuperação mandatória: os loci de recuperação mandatória estão representados na figura 7.36 como agrupamentos de balizas marcadas com a cor vermelha, escolhidas pelo usuário. No canto inferior esquerdo, são apresentadas informações desses agrupamentos de balizas selecionadas para recuperação.
- UCo20 - Orientações de embarque: as orientações de embarque são apresentadas no canto inferior esquerdo para uma determinada carga selecionada de um navio. Estão representadas na forma textual.

Com relação aos casos de amplificação de inteligência referentes à atividade de decidir empilhar, presentes na seção 7.2.10, estes estão modelados no DSP da seguinte maneira:

- CA1: para um determinado horizonte de tempo, o sistema sugere lugares de empilhamento representados por elipses posicionadas nas balizas sugeridas. Juntamente com essa informação, deve ser apresentada toda a informação do plano de empilhamento e recuperação, como máquinas, atividades e tempos para realizar o empilhamento. Se o plano for de vários dias, por exemplo, haverá várias posições de empilhamento sugeridas. O usuário deve ser capaz de acessar a lista de todas as posições e navegar por elas, selecionando a posição que quiser alterar em determinado momento.
- CA2: o usuário deve ser capaz de marcar como fixa e mandatória uma determinada região do pátio para que o material que seja empilhado lá. Assim, apesar do agente computacional sugerir uma determinada posição, o usuário humano pode fixar outra, baseado em informações privilegiadas que não existem no sistema, como o fato de ser impossível locomover as máquinas naquele instante, por exemplo. Isso representa uma parte do diálogo entre agentes computacionais e humanos. Esses loci de empilhamento mandatórios terão um ícone especial, como o vermelho da figura 7.35, por exemplo.
- CA3: os agentes computacionais devem ser capazes de sugerir, na interface de diálogo, loci de empilhamento híbridos, ou seja, complementos para os loci de empilhamento mandatórios do usuário, quando necessário. Esses loci híbridos deverão ter um ícone próprio, como a elipse laranja da figura 7.35, por exemplo.
- CA4: o usuário irá obter a opção de oficializar o plano de empilhamento e recuperação sugerido pelo motor de solução a partir de um menu.
- CA5: o usuário irá obter a opção de oficializar um eventual plano de empilhamento e recuperação mandatório da mesma maneira que o faz para um sugerido pelo motor de solução no item anterior, através de um menu secundário.

A parte do protótipo referente às funcionalidades de recuperação do DSP, relacionado aos casos de amplificação de inteligência da atividade de decidir recuperar, modelados na rede da figura 7.33, está apresentada na figura 7.36. É possível, assim como na parte referente às decisões de empilhamento, verificar a representação das unidades de conhecimento de representação direta. Como é a mesma interface de diálogo, as unidades de conhecimento nela representadas são as mesmas explicadas anteriormente.

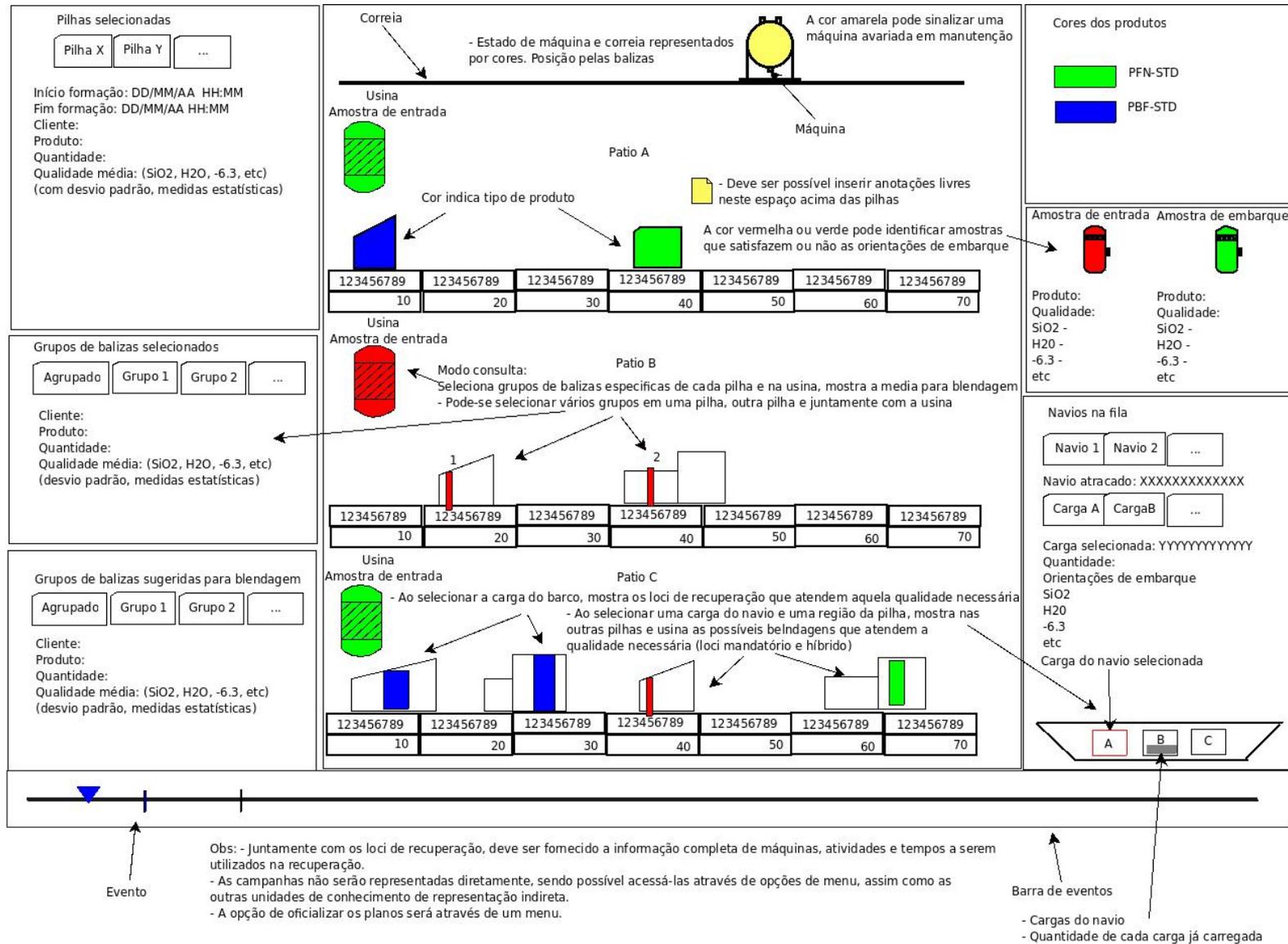


Fig. 7.36: Protótipo da interface de diálogo diagrama de situação do pátio (DSP) funcionando no modo recuperação

Com relação aos casos de amplificação de inteligência da atividade de decidir recuperar, modelados na seção 7.2.10, sua representação no DSP é da seguinte maneira:

- CA6: o usuário é capaz de selecionar uma das cargas do navio escolhido na fila no canto inferior esquerdo do DSP da figura 7.36. Todas as cargas do navio são representadas iconicamente por retângulos dentro do desenho do navio. Ao clicar na carga escolhida, ela muda sua cor para indicar que está selecionada, e informações do produto e das orientações de embarque são fornecidas.
- CA7: para cada uma das cargas selecionadas ao longo de um horizonte de tempo, o SAI sugere loci de recuperação que satisfazem a quantidade e as orientações de embarque referentes àquela carga selecionada. Esses loci são representados por agrupamentos de balizas com uma determinada cor, como o azul da figura 7.36. Pode ser sugerido que o material saindo da usina e chegando ao pátio seja enviado diretamente para o carregamento também, através do ícone de amostra de entrada que fica acima dos ícones das pilhas. No canto inferior esquerdo, aparecem informações sobre esses agrupamentos sugeridos, como produto e qualidade média da mistura sugerida. Para um plano de empilhamento e recuperação de um longo período, como vários dias, não serão mostrados na interface de diálogo todos os agrupamentos de balizas do horizonte de tempo planejado, mas somente um de cada vez. Haverá uma lista de loci de recuperação referente ao plano, e o usuário poderá escolher qual dos lugares de recuperação aparece na tela a cada instante, através da escolha da janela de tempo a ser mostrada, a partir da escala temporal. Assim, o usuário é capaz de alterar um lugar de recuperação no início do plano, mesmo que esteja no fim de sua confecção.
- CA8: o usuário consegue sugerir agrupamentos de balizas mandatórias selecionando-as em uma mesma pilha ou em pilhas diferentes. Esses agrupamentos são mostrados iconicamente com uma cor própria, como o vermelho da figura 7.36. Ele pode selecionar que o material vindo diretamente da usina entre no agrupamento para carregamento, selecionando o ícone mais a esquerda sobre os ícones das pilhas no pátio. No canto esquerdo, informações sobre os grupos de balizas escolhidas são mostradas, como produto e informações de qualidade média da mistura.
- CA9: a partir de loci de recuperação mandatórios, o sistema deve ser capaz de demonstrar no DSP loci de recuperação híbridos, ou seja, que consideram os mandatórios do usuário fixados pelo usuário. Caso as balizas escolhidas de maneira mandatória pelo usuário não sejam suficientes para a quantidade e qualidade da orientação de embarque da carga selecionada, o sistema deve complementar a escolha com outros agrupamentos, que terão uma iconização própria, por exemplo, com uma cor diferente, como o verde da figura 7.36. Mais uma vez, o material da usina pode ser selecionado para ir diretamente para o carregamento.
- CA10: o usuário deve ser capaz de oficializar o plano de empilhamento e recuperação construído até então, a partir de uma opção na interface.
- CA11: a oficialização de um plano de empilhamento e recuperação híbrido funciona da mesma maneira do item anterior.

Com relação às funcionalidade de consulta do DSP, o usuário pode observar as pilhas nas suas posições e alturas ao longo do pátio, cada uma com sua cor correspondente ao seu produto. Assim, ele é capaz de consultar as informações das pilhas e agrupamentos de balizas, selecionando-as, podendo simular vários cenários “o que aconteceria se” de empilhamento e recuperação.

No entanto, este protótipo apresentado nas figuras 7.35 e 7.36 não é a única possibilidade de interface de diálogo para o sistema, nem a melhor delas, é apenas uma proposta. A metodologia determina apenas quais conceitos e casos de amplificação de inteligência devem estar presentes na interface de diálogo, mas não determina como eles devem estar representados, o que é um trabalho que exige criatividade e engenhosidade. Uma outra proposta de interface de diálogo que apresenta todas as unidades de conhecimento e casos de amplificação de inteligência, por exemplo, é apresentada na figura 7.37.

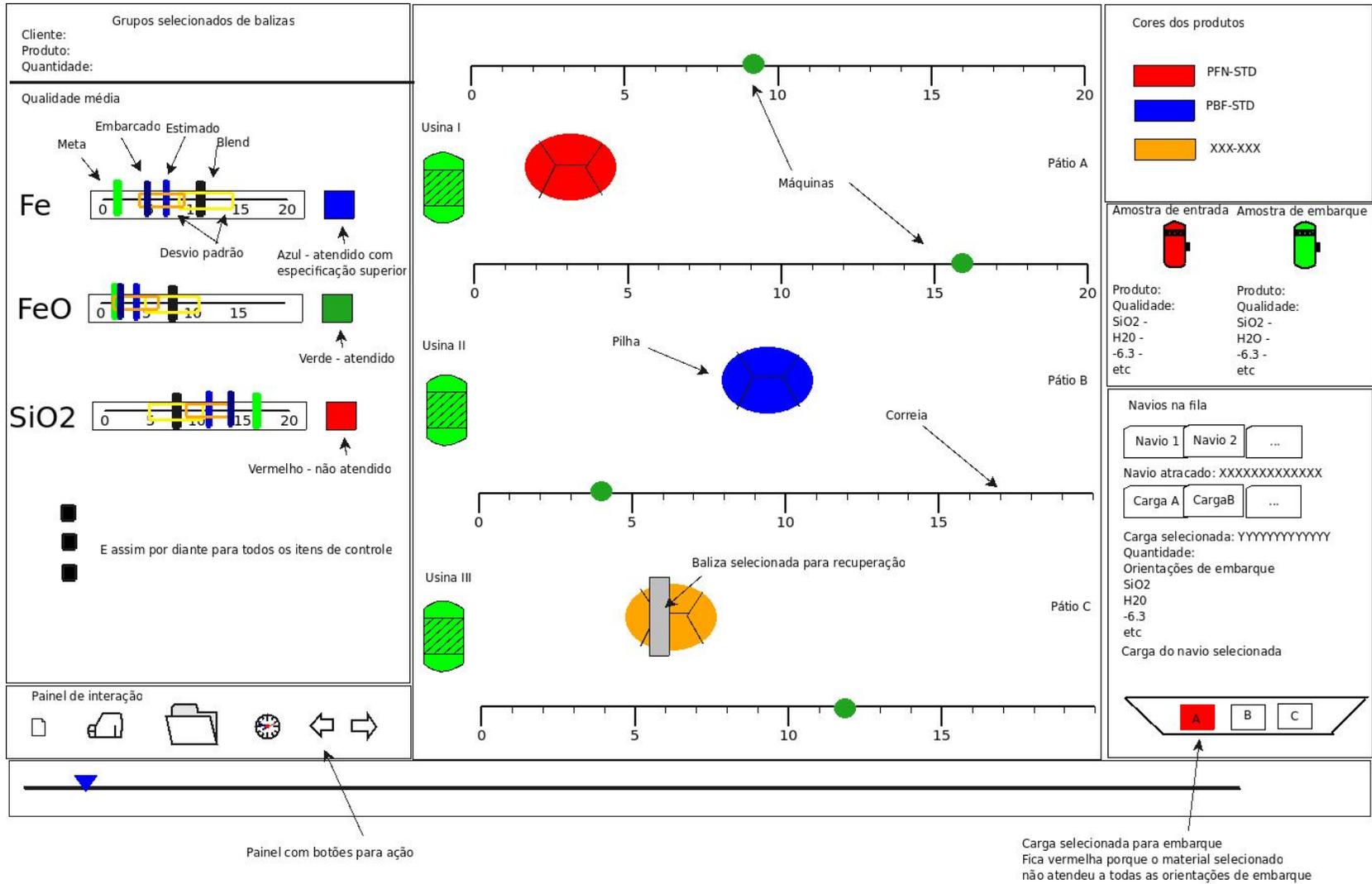


Fig. 7.37: Outra proposta de interface de diálogo representando as mesmas unidades de conhecimento de outra maneira

Nesta proposta da figura 7.37, apresenta-se uma visão aérea dos pátios, diferente da visão de perfil apresentada anteriormente nas figuras 7.35 e 7.36. Além disso, a qualidade média de cada item de controle está apresentado graficamente no canto esquerdo, ao invés de textualmente. No entanto, as mesmas unidades de conhecimento estão representadas, conforme determinado pelas atividades anteriores da metodologia, e é possível realizar os mesmos casos de amplificação de inteligência, de maneira análoga.

No entanto, essas duas propostas de interface de diálogo apresentadas anteriormente têm um ponto fraco, relacionado a um dos atributos que devem estar presentes em uma interface de diálogo, conforme mencionado na seção 5.5.1: a visualização da situação do domínio representado e do planejamento das soluções no passado, presente e futuro por meio de uma linha temporal. Nessas duas primeiras propostas, existe apenas uma barra de tempo pela qual o agente humano navega pelas situações do pátio que representam seu planejamento. Dessa maneira, o agente humano não é capaz de visualizar todos os estados do pátio em um período de tempo simultaneamente. Na proposta da figura 7.38, essa visualização simultânea de vários estados é possível.

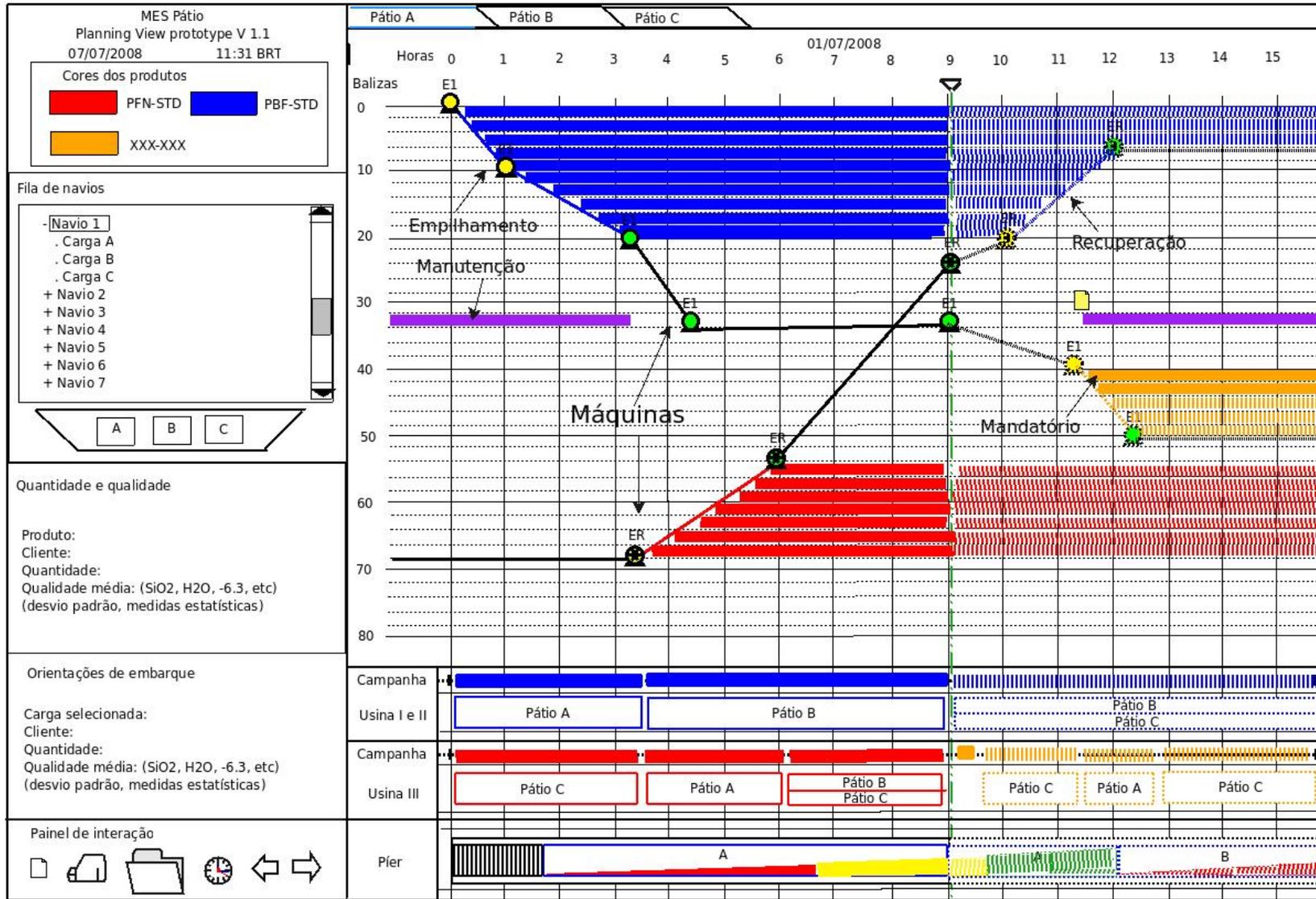


Fig. 7.38: Proposta de interface de diálogo do MES Pátio capaz de apresentar estados do pátio em um período de tempo

Portanto, a proposta de interface de diálogo da figura 7.38 apresenta mais informações simultaneamente, porque apresenta situações do pátio em vários instantes de tempo diferentes em uma única tela. Por outro lado, é uma interface a princípio menos intuitiva, por ser em forma de gráfico, e não reproduzir o formato das pilhas.

Neste protótipo, o fato de uma baliza permanecer pintada por um intervalo de tempo significa que há um produto empilhado lá ao longo deste período. As máquinas se movimentam entre as balizas ao longo do tempo, e as correias ficaram implícitas. Existe uma linha separando o que já foi realizado, ou seja, o passado, e o que está planejado, ou seja, o futuro. Na figura 7.38, esta linha está posicionada às 09:00 horas, e pode-se perceber que a representação dos estados daí para frente é diferente, tracejada, para indicar que é um planejamento futuro, em oposição às linhas cheias dos tempos anteriores às 09:00 horas, que indicam que aquelas situações já foram realizadas. Um planejamento mandatório está representado com linhas cheias no futuro, para indicar que aquela situação deve acontecer. As representações anteriores, das figuras 7.35, 7.36 e 7.37, podem ser pensadas como representações de cortes longitudinais dessa representação, pois apresentam a situação do pátio em um único instante (tente passar uma linha em cima das 03:00 horas longitudinalmente nesta representação da figura 7.38 e visualizar a situação do pátio). Um agente humano que utilizasse as representações instantâneas das figuras 7.35, 7.36 e 7.37, para ganhar entendimento das muitas situações do pátio ao longo do tempo, necessitaria viajar ao longo desses estados através da barra de tempo e montar em sua mente a concatenação das situações.

Pelo fato da interface da figura 7.38 ser uma representação menos intuitiva, o agente humano teria uma certa dificuldade inicial de planejar. Porém, provavelmente após um período de adaptação, ele teria mais facilidade de montar planos de empilhamento e recuperação. De fato, essas interfaces de diálogo têm vantagens e desvantagens, e nada impede que o sistema de amplificação de inteligência seja munido de todas elas. A intenção nesta discussão é mostrar que há diferentes maneiras de apresentar as unidades de conhecimento, e que algumas dessas maneiras podem ser mais poderosas.

O que as atividades da metodologia determinam são as unidades de conhecimento que devem estar representadas e os casos (mecanismos) de amplificação de inteligência que devem estar presentes. Inclusive, na descrição dos casos de amplificação de inteligência, determina-se quais unidades de conhecimento são usadas e quais são criadas em cada caso de amplificação. Isto deve se reproduzir nos comportamentos da interface de diálogo que reproduzem esses casos de amplificação de inteligência.

7.2.12 Propor componentes do SAI

A figura 7.39 apresenta o diagrama de componentes do sistema de amplificação de inteligência chamado de “MES Pátio”. Essa componentização foi desenvolvida a partir do diagrama de componentes padrão de SAI, proposta no capítulo 6, na figura 5.6. A seguir, cada um destes subsistemas é descrito em detalhes:

- Subsistema de qualidade e cálculos automáticos: conforme visto na seção 7.2.2, atualmente, todos os cálculos de qualidade do produto armazenado nas pilhas são feitos manu-

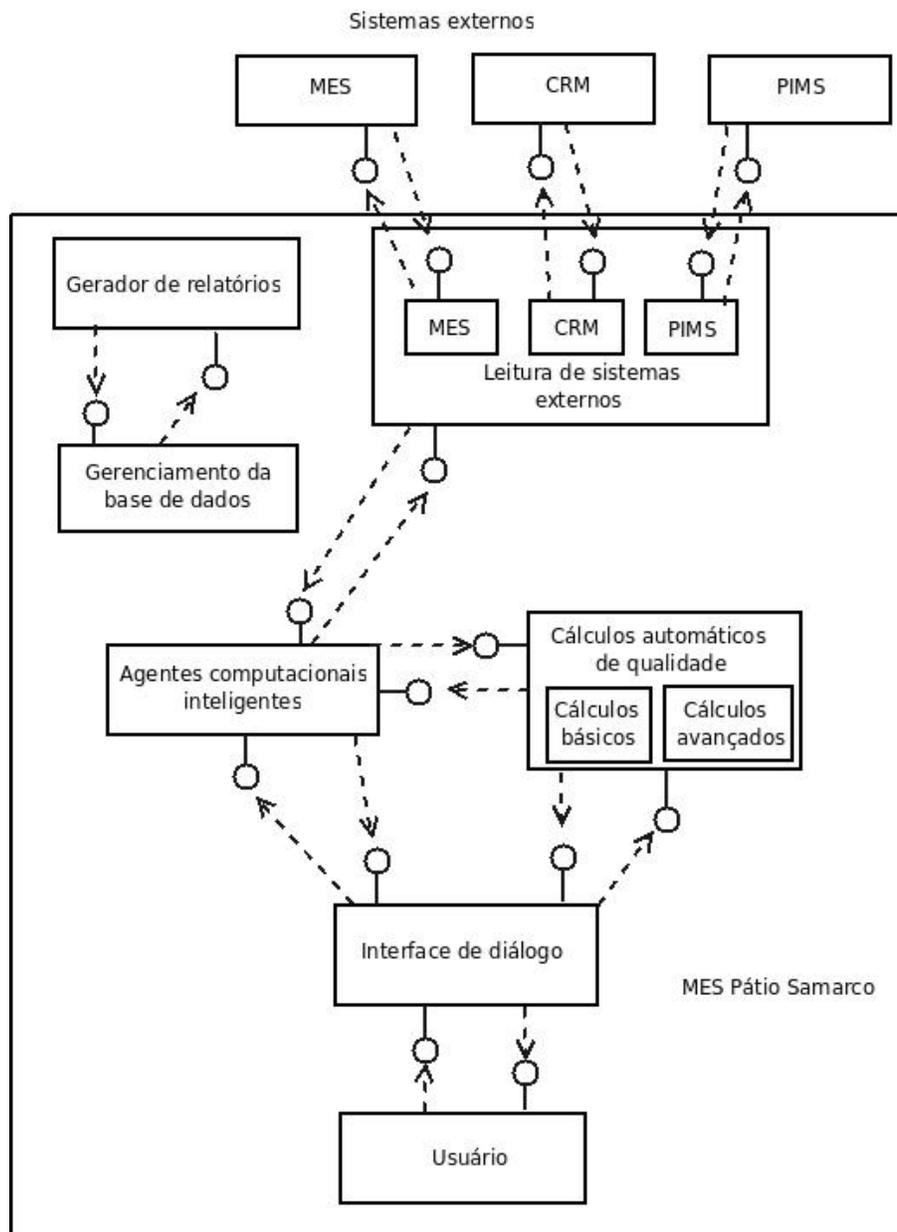


Fig. 7.39: Componentes do MES Pátio

almente, cruzando-se informações das medidas bi-horárias de qualidade e produção nas saídas das usinas com as informações provenientes dos relatórios manuais de posicionamento das máquinas. Este subsistema será responsável por efetuar estes cálculos de forma automática.

Apesar de utilizar apenas cálculos matemáticos simples e não incluir nenhum algoritmo de otimização, este subsistema auxilia bastante o usuário nas tomadas de decisão, principalmente liberando tempo para que o usuário realize outras tarefas e tornando praticamente inexistentes eventuais erros de cálculo para a qualidade.

O isolamento desta funcionalidade em um subsistema permite que a metodologia de cálculo e estimativa da qualidade real possa ser melhorada sem nenhum impacto no restante do sistema. É possível, por exemplo, trabalhar com técnicas de inteligência artificial e levar em consideração fatores como a incerteza introduzida por operações de recheio ou mesclagem do produto entre as várias balizas. Adicionalmente, pode-se sofisticar o modelo matemático da degradação da qualidade do produto no pátio. Por isso, este subsistema se subdivide em dois subsistemas independentes:

1. Cálculos básicos: esse subsistema inclui todos os cálculos automáticos que o sistema irá realizar e o cálculo da qualidade de maneira simplificada, que não levará em conta a umidade e nem as incertezas. Com relação à degradação, não se aplicará modelos matemáticos, heurísticas ou técnicas de computação inteligente para modelá-la, apenas se utilizará os coeficientes empíricos já utilizados na Samarco atualmente para prever o impacto da degradação no material do pátio.
2. Cálculos avançados: esse subsistema acrescenta os cálculos avançados de qualidade, considerando incertezas, degradação e umidade.

Durante uma primeira etapa do projeto, o sistema de amplificação de inteligência deve utilizar os fatores de correção históricos como uma estimativa do efeito da degradação, permitindo que esses valores sejam alterados pelo usuário conforme necessário. Em uma etapa posterior, deverão ser estudadas formas de se refinar estes fatores, com a inclusão do conhecimento obtido através de trabalhos acadêmicos realizados pela equipe da Samarco. Finalizando, o sistema de amplificação de inteligência deve fazer uma análise da qualidade semelhante à apresentada nesta seção, mas não somente para a pilha inteira e sim considerando baliza a baliza. Para isso, é necessário efetuar o cruzamento das informações de qualidades com as informações do posicionamento das máquinas.

- Interface de diálogo: a interface de diálogo é o subsistema responsável por realizar toda a comunicação do usuário com o sistema.

A interface deve fornecer:

1. Representação de forma icônica dos elementos do pátio: o principal componente da interface de diálogo deve ser o diagrama de situação do pátio (DSP), que representa iconicamente todos os diversos elementos do pátio modelados para uma representação direta na seção 7.2.11 como, por exemplo, o balizamento e quantidades armazenadas em cada pilha, o posicionamento e operação sendo realizada por cada máquina, a fila de navios, os navios atracados no momento, as medidas de qualidade estimadas para cada pilha, etc. O diagrama de situação do pátio deve utilizar como referência principal para seu projeto a interface desenhada atualmente à mão pela equipe de operação. Além de representar adequadamente os conceitos principais, esta interface também já faz parte da cultura da Samarco há vários anos e mantê-la reduzirá a oposição às mudanças que são introduzidas com o novo sistema.
2. Visualização da situação do pátio (no presente e no futuro), bem como visualização do plano de operações de empilhamento e recuperação: além de apresentar a situação do pátio em um determinado instante, o subsistema de interface de diálogo deve possibilitar também a visualização do plano de operações de empilhamento e recupe-

ração sugeridos pelo sistema, assim como o efeito futuro destas operações no estado do pátio.

3. Mecanismo de entrada de dados: a interface de diálogo deve prever também a entrada de informações vindas diretamente do usuário. Conforme visto nas seções 7.2.2 e 7.2.3, atualmente, uma parte dos dados necessários para alimentar o sistema está indisponível (ou não é confiável) nos sistemas externos. Dessa forma, a interface de diálogo deve permitir que esses dados sejam alimentados manualmente. No futuro, essa entrada manual de dados deve ser substituída pela leitura automática dos dados dos sistemas externos, conforme esses dados se tornem mais confiáveis. Podem ocorrer também casos em que os dados lidos através dos sistemas externos estejam inconsistentes (devido a um erro de digitação, por exemplo). O subsistema de interface de diálogo deve permitir, nestes casos, que este tipo de erro seja corrigido pelo usuário.
4. Interação do usuário com a programação: a interface deve permitir que o usuário crie suas próprias programações mandatórias de empilhamento e recuperação. Esta funcionalidade será utilizada nos casos em que as regras incluídas no sistema não sejam suficientes para resolver o problema, por exemplo, em casos de falta de espaço no pátio com necessidade de empilhamento de produtos diferentes na mesma pilha. Este tipo de operação não será sugerida pelo sistema, ficando a cargo do usuário decidir a solução a ser tomada. Como modelos computacionais não conseguem captar todos os detalhes dos problemas do mundo real, para que um sistema computacional consiga encontrar boas soluções para esses problemas de planejamento e execução dinâmicos em tempo adequado, faz-se necessário que ele seja capaz de incorporar as intuições do usuário na construção da solução. No caso do MES Pátio, o usuário será capaz de, em determinadas situações, contribuir para a construção da programação, quando julgar que alguma decisão de operação tomada pelo sistema não é adequada à situação da operação naquele instante.

Um detalhamento da interface foi apresentado na seção 7.2.11.

- Agentes computacionais inteligentes para definição e sequenciamento das operações do pátio: os subsistemas de interface de diálogo e de cálculo automático das qualidades trarão melhorias substanciais no gerenciamento do pátio. Apesar de não incluir algoritmos inteligentes em sua composição, esses subsistemas realizarão de forma muito mais eficiente tarefas hoje realizadas manualmente pelo usuário, além de tornar praticamente inexistentes erros durante o cálculo das qualidades estimadas das pilhas.

No entanto, resultados ainda melhores poderão ser alcançados com a criação de planos automáticos das operações de empilhamento e recuperação, tarefa esta realizada pelo subsistema de agentes inteligentes. Os agentes serão responsáveis por definir e sequenciar as operações do pátio para um horizonte futuro, tornando possível prever problemas e adiantar soluções de forma muito mais eficiente do que a realizada hoje no gerenciamento do pátio. Na geração das programações automáticas de empilhamento e recuperação, o principal fator de decisão é a qualidade do produto sendo empilhado ou recuperado. Como não existe atualmente nenhum modelo satisfatório para previsão da qualidade do produto gerado nas usinas, a única fonte de informação para previsão da qualidade é a especificação

contida nas campanhas. Dessa forma, o plano gerado para um período de tempo futuro a longo prazo, utilizando como previsão para a qualidade as especificações das campanhas, será apenas uma referência das operações que serão efetivamente realizadas. A partir dessa referência, é possível prever problemas de operação e adiantar soluções. Essa abordagem será bastante útil com a entrada da segunda recuperadora, pois será necessário lidar com as diferentes combinações de possíveis operações e conflitos de máquinas, paralelamente aos problemas de qualidade. Para um horizonte de planejamento de curto prazo, já se sabe a qualidade da maioria do produto a ser embarcado, e assim as programações para este período refletem de maneira mais fiel as operações a ser realizadas. Portanto, esse subsistema agregará ao sistema a capacidade de realizar planejamento e execução dinâmicos, com novos planejamento sendo realizados à medida que sejam disponibilizadas informações sobre mudanças no estado do pátio.

Devido à complexidade do problema de gerenciamento do pátio, o subsistema de agentes inteligentes deve ser projetado levando-se em consideração restrições de desempenho, memória computacional e tempo de resposta do sistema. Frequentemente, nestes casos, é preferível que o sistema sugira uma solução adequada, porém não ótima, em um tempo de processamento aceitável do que uma solução ótima em um tempo de processamento impraticável.

- Leitura de sistemas externos: conforme visto na seção 7.2.3, o sistema de amplificação de inteligência necessita de várias entradas disponíveis nas bases de dados de sistemas já existentes no ambiente computacional da Samarco. Estes sistemas são o MES, CRM e PIMS:
 - * o MES armazena os dados de produção e qualidade nas saídas das usinas, os dados históricos das manutenções preventivas e corretivas;
 - * o CRM armazena a fila de navios, assim como as orientações de embarque;
 - * o PIMS armazena os dados relativos ao posicionamento e operação das máquinas no pátio.

O subsistema de leitura de dados dos sistemas externos é responsável, desta forma, por ler todas essas informações e trazê-las para o contexto do MES Pátio.

Este subsistema de leitura de sistemas externos é subdividido em três subsistemas, cada um referente a um sistema externo, conforme figura 7.39. Os subsistemas são independentes entre si, e podem ser implementados separadamente, facilitando a implementação e diminuindo os custos e riscos envolvidos neste processo.

Os sistemas externos MES e CRM têm suas bases de dados implementadas em Oracle. Desta forma, o subsistema de leitura deve prever a implementação de um interface Oracle. Além disso, deve-se implementar também um protocolo para leitura dos dados armazenados no PIMS. Durante a fase de implementação, será necessário estudar as vantagens da compra de componentes de terceiros para o interfaceamento com o PIMS versus a implementação completa da interface.

Além da leitura dos dados, este subsistema é responsável por verificar periodicamente os sistemas externos para checar se houve mudanças no estado do pátio e demais informações. Por exemplo, quando ocorrer um deslocamento de uma máquina, este subsistema

deve perceber a mudança no estado do pátio e efetuar a leitura deste dado.

Por uma decisão da TI da Samarco, nenhum dado gerado pelo sistema de amplificação de inteligência será armazenado nas bases dos sistemas externos e assim, não é necessário a implementação de nenhum protocolo de gravação nestas bases.

- Gerenciamento da base de dados: conforme visto na seção anterior, nenhum dos dados gerados pelo sistema de amplificação de inteligência será gravado nas bases de dados dos sistemas externos. Desta forma, será necessária a criação de uma base de dados dedicada para a gravação de todas as informações pertinentes como, por exemplo, programações de operações geradas anteriormente, taxas efetivas das diversas máquinas, tempos de setup, qualidades das pilhas já existentes, etc.

Para esta tarefa, será criado o subsistema de gravação e leitura da base dedicada do sistema. Este subsistema será geralmente utilizado a cada inicialização do sistema e nos momentos em que as informações geradas devem ser consolidadas.

Devido a eventuais restrições de tamanho disponibilizadas para esta base, aliadas ao número grande de informações geradas, deve-se estudar durante a fase de projeto deste subsistema formas de manter a base com um tamanho e tempo de acesso adequados. Desta forma, deve-se implementar as operações de leitura e escrita na base de forma otimizada, evitando assim que essas operações venham a se tornar gargalos no desempenho do sistema.

- Gerador de relatórios: a geração e análise de relatórios é uma tarefa totalmente integrada à rotina de gerenciamento do pátio pela equipe de operação. Os diversos relatórios são utilizados para tarefas essenciais como, por exemplo, detalhamento dos planos de empilhamento e recuperação, resumos de medidas de qualidades das pilhas, informações para troca de turnos, etc. Os relatórios são utilizados também como material de referência em apresentações e reuniões.

Desta forma, o sistema de amplificação de inteligência deve suportar um subsistema que efetue a geração automática destes diversos relatórios, além de possivelmente implementar a geração de outros relatórios que hoje não são feitos por falta de tempo, facilitando ainda mais o trabalho da equipe de operação.

Este subsistema deve permitir a visualização dos diversos relatórios na interface gráfica no computador, assim como possibilitar sua impressão, caso desejado.

A definição de quais relatórios serão gerados, assim como a formatação utilizada em cada um, é uma tarefa a ser realizada durante a criação do documento de requisitos do sistema.

- Usuário: uma atividade essencial no projeto de um sistema inteligente é a modelagem do conhecimento dos especialistas e inclusão deste conhecimento no sistema. No entanto, mesmo que essa modelagem seja bem sucedida, é impossível que todo o conhecimento utilizado pelos especialistas seja inserido no sistema, devido principalmente à dificuldade de verbalização e estruturação do conhecimento por parte do especialista. Esta modelagem completa muitas vezes não é nem mesmo desejada, já que prejudica o desempenho do sistema com o processamento de uma quantidade enorme de variáveis a serem consideradas (problema do enquadramento, seção 5.3.3).

Desta forma, as melhores soluções para o problema são alcançadas através da interação

do usuário com o sistema. Para o caso do problema de gerenciamento do pátio, o papel do sistema é auxiliar o usuário com a automatização de tarefas como cálculos e interface de diálogo e, através de utilização de técnicas de inteligência computacional, propor planejamentos automáticos das operações de empilhamento e recuperação. Cabe ao usuário validar as soluções sugeridas pelo sistema assim como gerar suas próprias soluções em casos especiais onde o conhecimento especialista modelado no sistema não é suficiente.

7.2.13 Estimar resultados e ganhos

Toda a produção da Samarco passa pelo seu pátio de estocagem de produtos. Assim, a melhoria da performance da empresa passa, entre outras coisas, pelo aumento da eficiência da utilização do pátio. Essencialmente, há três tipos de perdas que devem ser reduzidas para aumentar a eficiência do pátio, como na figura 2.6:

- perdas de utilização: perdas de oportunidade de produção. No caso da Samarco, por exemplo, perdas de oportunidade de produção ocorreriam em situações nas quais a gestão do pátio não fosse otimizada.
- perdas de throughput: perdas de taxa de produção. Na Samarco, o problema de variabilidade de qualidade leva, por exemplo, à necessidade de se operar com taxas mais baixas de embarque para garantir a mistura correta dos produtos.
- perdas de aceitação (qualidade): casos em que o produto comercializado não atende completamente as especificações exigidas pelos clientes, causando problemas como diminuição do preço de venda, das vendas e insatisfação do cliente.

A atuação na redução das perdas pode ser feita através de, entre outras coisas, investimento em ativos, através de ferramentas de gestão e através de ferramentas de amplificação de inteligência. Esta seção apresenta as estimativas de resultados e ganhos com a implementação do sistema de amplificação de inteligência descrito nas seções anteriores.

O objetivo desta análise é balizar o plano de ciclos de desenvolvimento do software minimizando os custos e riscos e maximizando os ganhos e o retorno do investimento.

A descrição detalhada dos ganhos a serem obtidos com a implementação do sistema está no apêndice A.5.

Conforme pode ser observado na tabela A.31, os subsistemas de interface de diálogo e qualidade e cálculos automáticos (sem considerar incerteza, umidade e calculando a degradação com coeficientes empíricos) possuem a melhor relação ganho x risco, sendo assim os principais candidatos a implementação na primeira etapa do projeto. O subsistema gerador de relatórios também possui uma pontuação elevada sendo também candidato para uma primeira etapa. Os subsistemas de gerenciamento da base de dados e leitura de sistemas externos não possuem uma pontuação elevada, mas o fato de não possuírem dependências, de possuírem um baixo custo de implementação e servirem como fonte de dados para os outros subsistemas sugerem sua implementação também já em uma primeira etapa. Por outro lado, os subsistemas

de agentes computacionais inteligentes e qualidade e cálculos automáticos refinados (considerando incerteza, umidade e degradação) mesmo trazendo ganhos elevados têm um risco alto em sua implementação e propõe-se, assim, que sejam implementados após a consolidação dos subsistemas de maior pontuação. Pela tabela A.31, pode ser observado também que o subsistema de leitura de sistemas externos PIMS tem uma pontuação bastante baixa, e portanto tem uma prioridade baixa de implantação na primeira etapa. Além disso, a data de implantação deste subsistema depende fortemente da disponibilização dos dados dos equipamentos por parte das equipes de manutenção e automação da Samarco. Assim, a implementação deste subsistema deverá ser realizada na etapa 2 ou eventualmente até mesmo na etapa 3, caso os dados não sejam disponibilizados a tempo.

7.2.14 Planejar ciclos de desenvolvimento

Esta seção apresenta uma proposta de divisão da implementação do sistema especificado nas seções anteriores em diferentes ciclos, seguindo uma estratégia de dividir para conquistar. Essa divisão visa mitigar riscos envolvidos com a implementação de alguns subsistemas, considerar a dependência e os custos dos subsistemas a serem implementados, além de minimizar possíveis resistências à utilização do sistema, devido a um eventual choque cultural causado por mudanças de paradigma.

Propõe-se o plano de ciclos de desenvolvimento do software apresentado na figura 7.40 que dividirá o projeto do sistema a ser desenvolvido em três ciclos, descritos nas subseções a seguir.



Fig. 7.40: Etapas do plano de ciclos de desenvolvimento do software

Ciclo 1 - Automatização de operações manuais

A ênfase no ciclo 1 é automatizar processos que hoje são realizados manualmente pelo operador, como agrupamento de dados, desenho das pilhas e representação do pátio, cálculos de qualidade média por baliza e geração de relatórios. Além disso, serão implementadas também as funcionalidades de permissões e restrições relativas à diferentes perfis de usuários. Apesar de não contar com algoritmos de inteligência computacional, a automatização dessas tarefas permitirá uma maior eficiência no gerenciamento dos recursos do pátio e uma diminuição de eventuais erros, resultando em planejamentos manuais mais bem sucedidos em menores espaços de tempo.

Para tanto, no ciclo 1 devem ser implementados os subsistemas:

- interface de diálogo;
- leitura de sistemas externos (subsistemas MES e CRM) ;
- gerenciamento da base de dados;
- qualidade e cálculos automáticos (subsistema de cálculo simples da qualidade);
- gerador de relatórios.

Esses subsistemas foram segregados para um primeiro ciclo do projeto por se tratarem de funcionalidades com baixo risco de desenvolvimento, custo não muito elevado de implementação e grande potencial de ganho (vide tabela A.31). Se todas as funcionalidades fossem implementadas em um ciclo único, o sistema levaria um tempo grande para ser desenvolvido e começar a operar, além de aumentar a fase de familiarização do usuário com o sistema.

Com a implementação do primeiro ciclo, em pouco tempo o usuário já poderá usufruir das facilidades disponibilizadas e se adaptar ao novo paradigma enquanto o desenvolvimento de funcionalidades mais avançadas é feito em paralelo.

Ciclo 2 - Planejamento automático

No ciclo 2 será realizada a implementação do subsistema de agentes computacionais para definição e sequenciamento das operações no pátio de estocagem.

Neste ciclo, as operações de empilhamento e recuperação serão definidas e sequenciadas automaticamente utilizando algoritmos de computação inteligente. Para isso, é necessário que esses algoritmos incorporem de maneira adequada o conhecimento dos especialistas no gerenciamento do pátio de estocagem. Assim, será necessário um estudo mais detalhado das regras utilizadas pelos operadores e engenheiros de processo para as definições das operações, assim como uma definição dos critérios necessários para avaliação das soluções obtidas. Há precedentes na literatura de algoritmos utilizados no problema de gerenciamento de pátios de estocagem, como o algoritmo desenvolvido por Molck (2002). No entanto, cada situação possui suas particularidades, e pode ser necessário realizar modificações nesses algoritmos (ou mesmo desenvolver novos algoritmos), para adaptá-los ao problema específico da Samarco.

Os planejamentos gerados por este subsistema refletem de maneira fiel a realidade apenas para um horizonte de tempo futuro a curto prazo, onde a qualidade do produto utilizado já está definida. Conforme o horizonte de tempo de planejamento é estendido, aumenta-se o grau de imprecisão nos planos gerados, pois não existe nenhum modelo apropriado para prever a qualidade. No entanto, o planejamento futuro a longo prazo é uma boa referência, fornecendo importantes informações que podem ser utilizadas para auxiliar no gerenciamento efetivo dos recursos disponíveis, além de ajudar na previsão de problemas na operação. Além disso, este planejamento pode ser ajustado dinamicamente à medida que sejam disponibilizadas informações acerca da qualidade dos produtos.

Neste ciclo, será também implementado o subsistema de leitura de sistemas externos PIMS. Conforme visto anteriormente, esse subsistema será implementado posteriormente aos outros subsistemas de leitura de sistemas externos MES e CRM, pois depende da disponibilização das informações sobre o posicionamento das máquinas, que ainda estão sendo refinadas pelas equipes de manutenção e automação da Samarco.

No início da implantação deste ciclo, espera-se que o usuário já esteja familiarizado com os subsistemas implementados no primeiro ciclo e já consiga realizar seus planejamentos manuais de forma muito mais eficiente, com auxílio da ferramenta de amplificação de inteligência. Dessa maneira, a adaptação à utilização do planejamento automático ocorrerá de maneira natural. Com a entrada deste subsistema, os planejamentos para um determinado horizonte de tempo serão gerados automaticamente pelo sistema, e haverá oportunidade do usuário inserir suas próprias operações, contribuindo desta forma na construção das soluções. A geração de soluções para um horizonte de tempo mais longo possibilita a antecipação de problemas e soluções que raramente seriam contemplados sem o auxílio de uma ferramenta de amplificação de inteligência. Conforme citado anteriormente, optou-se pela implementação deste subsistema em um ciclo independente, já que o desenvolvimento e calibração dos algoritmos inteligentes é geralmente uma tarefa custosa, o que acarretaria em uma demora desnecessária para o início da operação do sistema se fosse contemplada no ciclo 1.

Ciclo 3 - Refinamento do cálculo da qualidade

Na ciclo 3, será realizado a implementação do subsistema de refinamentos no cálculo da qualidade, estudando maneiras de levar em consideração as incertezas, a umidade e calcular a degradação através de modelos matemáticos, e não somente através de coeficientes empíricos advindos de dados históricos, como é feito atualmente. Esse refinamento foi deixado para o último ciclo, pois trata-se de uma série de atividades com um nível maior de risco. No caso do tratamento da incerteza, já existem metodologias consolidadas que geralmente modelam de forma bem sucedida a influência da incerteza em diversos processos. No entanto, nada garante que essas abordagens trarão melhorias significativas para o problema específico da Samarco. Já no caso da influência da degradação e da umidade na qualidade dos produtos, será necessário desenvolver modelos e teorias adequadas com o auxílio de especialista nessas áreas. Equipes da Samarco já realizaram estudos para o tratamento desses fatores, mas nenhum resultado definitivo foi utilizado na operação. Dessa forma, o risco de que poucas melhorias sejam atingidas nesse ciclo existe.

Se esse ciclo for bem sucedido, a estimativa da qualidade nos produtos armazenados no pátio será bem mais precisa, melhorando conseqüentemente a qualidade das operações geradas automaticamente pelo sistema. Os subsistemas de agentes computacionais e interface de diálogo poderão sofrer refinamentos em consequência da geração de novos conceitos provenientes do refinamento do cálculo da qualidade. Um exemplo disso seriam diferentes representações para diferentes graus de incerteza na qualidade dos produtos armazenados. Os agentes computacionais teriam que considerar essas representações na formulação dos planejamentos automáticos.

7.2.15 Identificar requisitos de amplificação de inteligência

Os requisitos de amplificação de inteligência originados da aplicação dos passos anteriores da metodologia encontram-se no apêndice A.6.

Esses requisitos de amplificação de inteligência, juntamente com os outros artefatos produzidos, servirão de entrada para workflow de Requisitos em cada uma das iterações de cada uma das fases, com o objetivo de garantir que o SAI possua os atributos necessários, descritos na seção 5.5.1.

No caso do MES Pátio, 71 requisitos de amplificação de inteligência foram encontrados. Desses 71, 37 (52,11%) referem-se ao subsistema de interface de diálogo, 11 (15,49%) ao subsistema de agentes computacionais inteligentes, 1 (1,41%) ao usuário, 3 (4,23%) ao subsistema de qualidade e cálculos automáticos, 3 (4,23%) aos sistemas externos, 8 (11,27%) ao subsistema de base de dados e 8 (11,27%) ao subsistema gerador de relatórios. Esses números denotam o grau de importância dos atributos dos SAI, com destaque para a interface de diálogo e o diálogo com agentes computacionais inteligentes.

Esses requisitos encontrados garantem uma *interface de diálogo* capaz de representar exosomaticamente as soluções e ser uma ontologia compartilhada entre os agentes humanos e os *agentes computacionais*, capazes de sugerir soluções *multi-objetivas*, para dialogar em busca das soluções, mecanismos de cálculos automáticos que *controlam o fluxo cognitivo*, liberando os agentes humanos e computacionais para se concentrarem nas atividades inteligentes, e permitem *simulações de cenários* diferentes para a formulação de planos e tomada de decisão, diferentes *perfis de usuário* e *mecanismos de evolução internos* ao sistema, que calculam indicadores de usabilidade. Dessa maneira, ao entrar nos próximos fluxos principais de trabalho, os atributos importantes do SAI estarão garantidos.

7.3 Um sistema otimizante para a estocagem e embarque da Samarco

Antes do início do projeto do MES Pátio, a Samarco tentou utilizar um sistema otimizante para suportar a tomada de decisão do seu processo de estocagem e embarque. Este sistema falhou, tendo seu projeto abandonado.

Os objetivos deste projeto anterior eram similares aos do MES Pátio:

- Permitir um melhor gerenciamento das operações realizadas no pátio, facilitando as práticas de administração, operação e monitoramento;
- Contribuir para uma diminuição da variabilidade do material embarcado, melhor previsibilidade nos resultados das análises, melhor gerenciamento dos custos envolvidos nos equipamentos móveis e maior confiabilidade no desenvolvimento de blendagens;
- Consolidar o sistema como uma ferramenta de gestão da operação do pátio de estocagem da Samarco sem a necessidade de controles paralelos;
- Documentar e divulgar os processos de operação do pátio no contexto que estão inseridos;
- Suportar os processos operacionais do pátio de estocagem.

No entanto, ao invés de utilizar um paradigma de sistemas de amplificação de inteligência, ou seja, um sistema construído para suportar a decisão inteligente de um usuário humano, lidando com as incertezas e imprecisões inerentes ao processo, esta ferramenta procurou tratar todas as incertezas e imprecisões internamente e entregar uma solução fechada para o usuário, com os seguintes aspectos considerados nesta decisão:

- Capacidade de controle e rastreabilidade da movimentação de materiais no pátio por meio de um modelamento 3D do pátio, possibilitando visão por seções;
- Entrada de dados automatizada para o posicionamento do material em três dimensões, a partir de sensores instalados na planta.

Os principais problemas encontrados na aplicação deste sistema otimizador foram:

- Os sensores instalados no pátio, que seriam responsáveis por mapear a posição $p=(x,y,z)$ das pelotas não conseguiam trabalhar devido ao particulado (poeira) existente no pátio, que se depositava em suas superfícies. Com isso, a entrada de dados do sistema ficava prejudicada.
- Por ser um sistema otimizador, que não permitia a intervenção humana nas soluções construídas, a ferramenta passou a fornecer soluções que não faziam sentido na prática;
- Os usuários ficaram desmotivados e na maior parte do tempo não priorizavam as atividades do sistema.

O gráfico de evolução da utilização do sistema apresentado pela equipe da Samarco está na figura 7.41.

O fornecedor deste sistema alegou adequação da ferramenta em outros clientes, porém o problema de estocagem e embarque da Samarco possui especificidades, e é provável que este não tenham sido a principal causa do fracasso. A filosofia de sistemas otimizadores utilizada na ferramenta não é capaz de lidar com o problema não estruturado presente na estocagem e embarque da Samarco, devido às imprecisões e incertezas presentes no processo, que só podem ser tratadas por intervenções humanas nas decisões otimizadas.

Portanto, a principal vantagem de um sistema de amplificação de inteligência construído com a metodologia PUC é a capacidade de dialogar com o agente humano usuário, incorporando



Fig. 7.41: Evolução da utilização do sistema otimizador para a estocagem e embarque da Samarco

seus conhecimentos e motivando-o a participar ativamente da construção da solução, cujas partes não estruturadas precisam ser resolvidas por ele, pois necessitam estar relacionadas com o cenário existente no mundo real.

7.4 Resumo

O presente capítulo apresentou a aplicação do fluxo principal de trabalho Modelagem Cognitiva da metodologia PUC na primeira iteração da fase de concepção a um problema não estruturado do mundo real, inserido em um ambiente dinâmico, essencialmente um problema de planejamento e execução dinâmica (P&ED). A metodologia foi aplicada passo-a-passo, de acordo com a sequência de atividades previamente especificada no capítulo 6, que trata da metodologia, demonstrando que ela é aplicável em uma situação concreta.

O resultado desta aplicação foi validado pela equipe da Samarco. A demonstração da aplicação da metodologia em um problema de P&ED do mundo real ajudam àqueles que se interessam por amplificação de inteligência e pretendem utilizar-se da metodologia a ter uma maior confiança na sua aplicação, munidos com um exemplo de utilização da Modelagem Cognitiva.

A metodologia, portanto, sugere a utilização de DFC e de Redes Semiônicas, aplicando conceitos de semiótica computacional para tratamento do fluxo cognitivo do agente humano usuário, para que o sistema se encaixe neste fluxo, o que representa um novo paradigma no que se refere a um processo de desenvolvimento adequado a sistemas que tenham por objetivo amplificar o intelecto de seus usuários.

A aplicação da metodologia para resolução de um problema de P&ED do mundo real traz motivação para atacar problemas do mundo real e alcançar ganhos em muitos setores da economia, nos níveis estratégico, tático e principalmente operacional.

Capítulo 8

Conclusão e trabalhos futuros

8.1 Contribuições

Com o objetivo de atender à necessidade de metodologias padronizadas que auxiliem na construção sistemática de sistemas de amplificação de inteligência, capazes de resolver problemas de planejamento e execução dinâmicos, a principal contribuição deste trabalho é a proposta de uma metodologia que estende o PU (Jacobson et al., 1999b) para o desenvolvimento de sistemas de amplificação de inteligência. Assim, torna-se possível construir sistemas computacionais confiáveis e robustos que resolvam problemas dinâmicos do mundo real através da amplificação de inteligência dos seres humanos usuários.

De uma maneira mais detalhada, considera-se que as contribuições deste trabalho são as seguintes:

- apresentação da definição de problemas não estruturados em nível operacional, ou seja, problemas de planejamento e execução dinâmicos, como o tipo de problema mais adequado para a aplicação de SAI;
- apresentação do conceito e revisão bibliográfica de sistema de suporte à tomada de decisão, considerados como sendo a origem dos sistemas de amplificação de inteligência;
- utilização da semiótica computacional, baseada na semiótica de Peirce, como fundamentação para o desenvolvimento de uma metodologia de produção de sistemas de amplificação de inteligência;
- apresentação do fato que a evolução do intelecto humano se deu devido à evolução das ferramentas de amplificação de inteligência que o mesmo construiu, como a língua e a escrita, e que o computador é atualmente a ferramenta de amplificação de inteligência mais evoluída;
- revisão bibliográfica da área de amplificação de inteligência e comparação com a área de inteligência artificial, traçando paralelos, apresentando as intersecções e ortogonalidades dos conceitos;

- sugestão de que agentes computacionais são ferramentas adequadas para participar de processos de amplificação compostos, e devem constituir um sistema de amplificação de inteligência, sendo responsáveis pela solução sugerida pelo computador e pelo diálogo com agentes humanos na busca de uma solução comum;
- definição de sistemas de amplificação de inteligência e apresentação do seu diferencial, com ambos agentes humanos e computacionais participando da construção da solução para o problema;
- definição dos atributos necessários a um sistema de amplificação de inteligência;
- definição de uma componentização padrão para sistemas de amplificação de inteligência;
- desenvolvimento do diagrama de fluxo cognitivo, uma ferramenta capaz de modelar o fluxo cognitivo humano na resolução de um problema de planejamento e execução dinâmicos;
- utilização de redes semiônicas para simular o processo modelado no diagrama de fluxo cognitivo e inserir o sistema computacional no processo de resolução;
- proposição de um fluxo principal de trabalho especializado para o desenvolvimento de sistemas de amplificação de inteligência;
- estudo de caso para apresentar a aplicação da metodologia proposta e das ferramentas desenvolvidas em uma situação concreta.

A semiótica é uma teoria adequada para ser usada como *background* de uma metodologia de construção de sistemas de amplificação de inteligência pelo fato de estudar significado, representação e operações mentais. Como os SAI usam como princípio a amplificação da inteligência do ser humano (usuário) para resolver os problemas, faz-se necessário entender como este ser humano resolve o problema de P&ED em questão em termos de processos cognitivos humanos, e entender como um sistema computacional pode ajudar neste fluxo cognitivo. A semiótica mostrou-se uma ferramenta adequada para tal tarefa, tendo inspirado a construção do diagrama de fluxo cognitivo, que modela o pensamento humano na resolução do problema, e a simulação deste diagrama em Redes Semiônicas, através de agentes semiônicos. Isso possibilita modelar a inserção de um sistema computacional no processo de resolução de maneira a amplificar o intelecto do humano.

O intelecto humano evoluiu ao longo das eras que a espécie *Homo sapiens sapiens* tem habitado o planeta Terra não devido à evolução de estruturas físicas neurais, mas sim devido à evolução das ferramentas de amplificação de inteligência que os humanos desenvolveram e utilizaram ao longo desse tempo. Essas ferramentas, como a língua e a escrita, possibilitaram que os pensamentos humanos se organizassem e pudessem ser exosomatizados, ou seja, representados fora da mente, através de semioses externas. Isso caracteriza o ser humano como uma “criaturas gregorianas”, segundo as definições de tipos de mente de Dennett (1996). Atualmente, o computador é a ferramenta de amplificação de inteligência com mais recursos, o que denota a importância de construir sistemas computacionais com maior capacidade de amplificar a mente. O sistema formado pelas ferramentas de amplificação de inteligência e os humanos são considerados sistemas cognitivos distribuídos (Giere & Moffatt, 2003).

Finalmente, a proposição de uma metodologia de desenvolvimento para sistemas de amplificação de inteligência e sua aplicação através do estudo de um caso real dão início a um período de amadurecimento no desenvolvimento sistemático de sistemas de amplificação de inteligência. No entanto, apesar das diversas contribuições adicionadas por este trabalho, um longo caminho de melhorias está pela frente, o que traz um grande potencial para trabalhos futuros.

8.2 Problemas, limitações e trabalhos futuros

O desenvolvimento da metodologia presente neste trabalho é um ponto de partida para uma linha de pesquisa relacionada à Engenharia de Sistemas de Amplificação de Inteligência. Além disso, com base no estudo de caso realizado, surgiram algumas necessidades. A seguir, serão descritas algumas sugestões para trabalhos futuros:

- Utilizar a metodologia em diferentes áreas de aplicação no desenvolvimento de sistemas de amplificação de inteligência, para obter um maior entendimento e amadurecimento do método;
- Desenvolver treinamento e treinar pessoas para aplicar o método: a utilização da metodologia foi efetiva em diferentes aplicações e de fácil execução, apesar de ser um método sofisticado. No entanto, ainda não se sabe se a transmissão dos conceitos e do método em si será de fácil acesso para as pessoas, principalmente aquelas que não possuem um conhecimento técnico em desenvolvimento de software e que nunca estiveram em contato com conceitos da semiótica de Peirce;
- Desenvolver uma forma de realimentação da metodologia, com indicadores empíricos sobre como ou porque os usuários ficam mais bem servidos com sistemas de amplificação de inteligência construídos com esta metodologia do que com sistemas otimizados, construídos com outras metodologias;
- Desenvolver uma ferramenta que suportasse todo o ciclo de vida da metodologia facilitaria sua utilização e representaria um grande diferencial da metodologia proposta;
- Explorar a definição formal da interface de diálogo desenvolvendo uma linguagem específica para este fim, como por exemplo Leite & de Souza (1999) definiram uma linguagem específica para especificar a mensagem do projetista na Engenharia Semiótica de interfaces de usuário;
- Pesquisar os desenvolvimentos da área de interfaces humano-computador (IHC) para o desenvolvimento de um mecanismo de como apresentar os conceitos que precisam estar presentes na interface de diálogo do SAI;
- Evoluir a ferramenta SNTool: a ferramenta SNTool, utilizada para simular o diagrama de fluxo cognitivo em Redes Semiônicas e estudar a inserção do sistema computacional no processo, é capaz de gerar código na linguagem Java. Esta ferramenta poderia ser evoluída para que, após finalizar a rede semiônica com a inserção do sistema computacional no fluxo cognitivo, ela gerasse o código de um protótipo funcional do futuro sistema.

Como pode-se implementar qualquer código nos agentes semiônicos do SNTTool, inclusive acoplar uma interface fornecendo entradas, seria possível acoplar um protótipo da interface de diálogo e dos motores de solução no processo de fluxo cognitivo, e o protótipo gerado estaria pronto para ser validado e estudado antes de especificar os requisitos de amplificação de inteligência. No entanto, a ferramenta SNTTool possui um editor de código pouco conveniente, além de problemas de eficiência na execução de algoritmos de otimização. Para resolver tal problema, poderia ser interessante transformá-la em um plugin de alguma ferramenta de desenvolvimento Java disponível no mercado, como o Eclipse ou o Netbeans. Dessa maneira, seria possível aproveitar a interface de edição de código dessas ferramentas e suas capacidades de execução e debugging.

8.3 Implicações de médio prazo

Como implicações de médio prazo deste trabalho, têm destaque:

- A metodologia proposta neste trabalho será continuamente aplicada no cotidiano de uma empresa de desenvolvimento de SSTD, a CFlex (www.cflex.com.br);
- Pretende-se publicar um livro para disseminar a metodologia.

8.4 Considerações finais

O desenvolvimento de sistemas de amplificação de inteligência, atualmente, não é um processo sistemático. Este trabalho representa um movimento na direção do desenvolvimento sistemático de sistemas capazes de amplificar a inteligência do usuário humano que sejam robustos, confiáveis e resolvam problemas dinâmicos do mundo real, que não podem ser resolvidos por sistemas otimizantes convencionais. No entanto, ainda há muitas melhorias a serem alcançadas no desenvolvimento de sistemas de amplificação de inteligência, e os esforços atuais representam pouco frente ao potencial deste tipo de sistema para resolver problemas que antes não podiam ser atacados por sistemas computacionais. Este trabalho posiciona-se na direção desta nobre causa.

Apêndice A

Apêndices

A.1 Código da rede semiônica das figuras 5.11 e 5.12 para SNToolkit

Disponível em <https://pub.cflex.com.br/brainmerge/apendice/exemploDFC.xsn>

A.2 Atividades do fluxo principal de trabalho Modelagem Cognitiva detalhadas

Cada uma das atividades do fluxo da figura 6.17 é descrita em detalhes a seguir:

A.2.1 Mapear workflow operacional

Responsável

Engenheiro de SAI

Artefatos de entrada

Não há.

Descrição

Deve-se levantar, junto à operação dos especialistas humanos que resolvem o problema em questão, um diagrama de contexto explicitando as diferentes áreas que manipulam a informação, tanto no escopo da tomada de decisão quanto nas suas vizinhanças. Deve-se descrever:

- o fluxo da informação ao longo da operação e a periodicidade desse fluxo em cada área;
- entradas, saídas e relacionamentos entre as áreas.

O objetivo dessa atividade é entender como um sistema de amplificação de inteligência projetado para essa operação se encaixaria no processo geral.

Artefatos de saída

- Workflow operacional

Material de apoio

Não há.

A.2.2 Identificar estruturas estáticas e dinâmicas do domínio

Responsável

Engenheiro de SAI

Artefatos de entrada

- Workflow operacional

Descrição

Deve-se levantar, junto aos especialistas humanos que resolvem o problema em questão, as estruturas estáticas e dinâmicas da operação, quais os instrumentos utilizados (planilhas, sistemas, máquinas, etc) e suas características. Após a identificação dessas estruturas, deve-se identificar as interações entre elas, o resultado obtido dessas interações e quais os mecanismos que podem ser melhorados. Deve-se identificar:

- o domínio da aplicação, realizando um estudo detalhado do mesmo, especialmente nos casos de domínios desconhecidos;
- os elementos do cenário de tomada decisão;
- os modos de operação dos elementos do cenário;
- as regras de tomada de decisão;
- exceções e casos especiais das regras de tomada de decisão;
- os multi-objetivos da tomada de decisão;
- as principais dificuldades para a tomada de decisão, os pontos mais importantes que devem ser tratados pelo sistema de amplificação de inteligência a ser desenvolvido, tornando a tomada de decisão mais produtiva e eficiente.

Artefatos de saída

- Estruturas estáticas e dinâmicas do domínio

Material de apoio

Não há.

A.2.3 Identificar sistemas externos**Responsável**

Engenheiro de SAI

Artefatos de entrada

- Workflow operacional

Descrição

Deve-se detalhar os sistemas computacionais já existentes na operação de tomada de decisão que podem ser relevantes para o projeto, quais informações cada um deles contém e como elas podem ser obtidas pelo sistema de amplificação de inteligência. Deve-se identificar eventuais entradas manuais necessárias ao sistema de amplificação, que serão obtidas através de sua interface. Deve-se identificar, portanto:

- descrição dos sistemas computacionais existente relevantes para o projeto;
- entradas e sistemas de origem;
- saídas e sistemas de origem;
- entradas manuais.

Artefatos de saída

- Interface com sistemas externos

Material de apoio

Não há.

A.2.4 Prototipar motores de solução

Responsável

Engenheiro de SAI

Artefatos de entrada

- Workflow operacional
- Estruturas estáticas e dinâmicas do domínio
- Interface com sistemas externos

Descrição

Dependendo do problema identificado a ser resolvido pelo sistema de amplificação de inteligência, pode ser necessária a criação de algoritmos para o tratamento de informações que não tenham precedentes na literatura. Neste caso, faz-se necessário a implementação de um protótipo conceitual dos algoritmos para validar a abordagem e mitigar os riscos da fase de desenvolvimento.

Artefatos de saída

- Protótipo dos motores de solução

Material de apoio

Não há.

A.2.5 Desenvolver diagrama de fluxo cognitivo

Responsável

Arquiteto de SAI

Artefatos de entrada

- Workflow operacional

Descrição

Deve-se desenhar um diagrama que represente as unidades de conhecimento manipuladas pelos especialistas na tomada de decisão e as atividades cognitivas que transformam essas unidades de conhecimento, desde o início do processo de busca pela solução até a solução final, conforme explicado na seção 5.6. Esse diagrama evidencia o fluxo do conhecimento ao longo do processo de busca pela solução.

Para ser capaz de desenhar esse diagrama com a maior verossimilhança possível, sugere-se que o arquiteto de SAI desenvolva as capacidades hierárquicas dos operadores na sua tomada de decisão, ou seja, é importante realmente exercer o papel dos tomadores de decisão que realizam as atividades, sendo treinado para exercer suas funções e exercendo-as de fato. Caso isso não seja possível, deve-se procurar alternativas como a criação de exemplos didáticos para que os operadores tomem decisões. Neste caso, as conclusões deverão ser tiradas puramente da observação da tomada de decisão desses operadores, ou seja, o conjunto de atividades cognitivas realizadas pelos agentes humanos na resolução do problema de planejamento e execução dinâmicos deve ser levantado utilizando a técnica de etnografia, conforme seção 6.1.1.

Artefatos de saída

- Diagrama de fluxo cognitivo

Material de apoio

- Elementos do DFC, figura 5.7.
- Ferramenta computacional para desenhar esquemas (por exemplo DIA, disponível em <http://www.gnome.org/projects/dia/>)

A.2.6 Identificar tipos de tomada de decisão

Responsável

Arquiteto de SAI

Artefatos de entrada

- Diagrama de fluxo cognitivo

Descrição

Após a confecção do diagrama de fluxo cognitivo, deve-se classificar cada uma das atividades em tipos. Uma interpretação da teoria semiótica de Charles Sanders Peirce (Peirce, 1960) aplicada a engenharia nos leva a três tipos de atividades nas quais as atividades cognitivas podem ser classificadas, conforme explicado na seção 4.2.1:

- Atividades monádicas (primeiridades);
- Atividades diádicas (secundidades);
- Atividades triádicas (terceiridades).

Deve-se classificar cada uma das atividades do diagrama de fluxo cognitivo levantado em uma dessas três categorias.

Artefatos de saída

- Diagrama de fluxo cognitivo classificado

Material de apoio

- Elementos do DFC, figura 5.7.
- Ferramenta computacional para desenhar esquemas (por exemplo DIA, disponível em <http://www.gnome.org/projects/dia/>)

A.2.7 Identificar pontos de amplificação de inteligência

Responsável

Arquiteto de SAI

Artefatos de entrada

- Diagrama de fluxo cognitivo classificado

Descrição

No diagrama de fluxo cognitivo classificado, deve-se identificar como pontos de amplificação de inteligência as regiões de atividades triádicas. Essas regiões devem ser marcadas no diagrama com círculos ao redor das atividades.

Os pontos de amplificação de inteligência do sistema são considerados pontos de atividades inteligentes na tomada de decisão. Neste pontos, serão aplicados agentes computacionais inteligentes carregando motores de solução para solução do problema, suportados por todas as outras funções que serão realizadas automaticamente pelos SAI. Os agentes computacionais irão resolver os problemas dos pontos de amplificação em conjunto com os agentes humanos, em um diálogo cooperativo que se dará através da interface de diálogo do sistema. Assim sendo, os pontos de amplificação são também uma espécie de fronteira do processo cognitivo onde o conhecimento dos agentes humanos entra no sistema, através da interface de diálogo, num diálogo com os agentes computacionais. Todas as outras atividades classificadas como monádicas ou diádicas, por serem aleatórias ou mecânicas (aplicações de regras), serão resolvidas somente pelo SAI, transparente aos usuários humanos, liberando a mente humana somente para trabalhar juntamente com os agentes na resolução das atividades inteligentes. Dessa maneira, o SAI a ser desenvolvido amplificará a inteligência do agente humano retirando dele a necessidade de realizar atividades que não necessitam de inteligência humana para serem resolvidas, e suportando as atividades triádicas ou inteligentes com a cooperação de agentes computacionais em diálogo com o agente humano usuário.

Artefatos de saída

- Diagrama de fluxo cognitivo classificado com pontos de amplificação

Material de apoio

- Elementos do DFC, figura 5.7.
- Ferramenta computacional para desenhar esquemas (por exemplo DIA, disponível em <http://www.gnome.org/projects/dia/>)

A.2.8 Modelar amplificação de inteligência em Redes Semiônicas**Responsável**

Arquiteto de SAI

Artefatos de entrada

- Diagrama de fluxo cognitivo classificado com pontos de amplificação

Descrição

Deve-se construir uma Rede Semiônica que simule dinamicamente o diagrama de fluxo cognitivo montado para o processo de tomada de decisão. Esta rede de agentes irá traduzir o diagrama de fluxo cognitivo da seguinte maneira:

- Unidades de conhecimento do DFC: devem ser traduzidas como lugares passivos da rede
- Atividades cognitivas do DFC: devem ser traduzidas como lugares ativos da rede
- Pontos de amplificação de inteligência do DFC: devem ser traduzidas como superlugares na rede. Esses superlugares deverão modelar uma outra rede, na qual deve-se simular a interação entre os agentes humanos e os agentes computacionais, explicitando as entradas dessa sub-rede como interfaces de entrada, as unidades de conhecimento específicas que são manipuladas por ambos na tomada de decisão, e as saídas como interfaces de saída de volta para a rede principal. A modelagem deste diálogo, neste ponto, é original, e deve identificar as unidades de conhecimento modificadas pelos agente humano e computacional na interface de diálogo para conseguir chegar até a solução.

Se o DFC procura representar o repertório hierárquico de capacidades do especialista na resolução do problema, a modelagem em Redes Semiônicas procura representar o repertório hierárquico de capacidades do futuro SAI que irá amplificar a mente desse especialista, inclusive modelando o diálogo entre os agentes computacionais e o agente humano, com a vantagem de ser uma simulação dinâmica. Essa etapa serve também para validar o DFC desenhado anteriormente.

Artefatos de saída

- Modelagem em Redes Semiônicas

Material de apoio

- Ferramenta SNTTool, disponível em <http://sntool.sourceforge.net/>
- Trabalhos de Guerrero (2000), Guerrero et al. (1999) e Ojeda et al. (2000).

A.2.9 Identificar lugares nas Redes Semiônicas

Responsável

Arquiteto de SAI

Artefatos de entrada

- Modelagem em Redes Semiônicas

Descrição

Nesta atividade, os lugares da Rede Semiônica devem ser classificados e agrupados em lugares passivos, ativos e superlugares. O significado dessas estruturas é o seguinte:

- Lugares passivos: **unidades de conhecimento a serem representadas na interface de diálogo.** Existem dois tipos de lugares passivos a serem representados:
 - * lugares passivos diretos (representação direta): unidades de conhecimento presentes no diálogo entre agentes humanos e computacionais, nos superlugares. Ter uma **representação direta** significa estar representado preferencialmente de maneira icônica, de rápido acesso, e que comunique todas as informações relevantes com respeito àquela unidade de conhecimento na interface gráfica principal.
 - * lugares passivos indiretos (representação indireta): demais unidades de conhecimento que não estão presentes no diálogo entre agentes computacionais e humanos, ou seja, que não estão presentes nos superlugares. Ter uma **representação indireta** significa ter uma representação em sub-menus, abas ou outros, não necessariamente icônica e não necessariamente na interface principal do sistema.

Essa diferenciação entre as representações se dá porque as unidades de conhecimento que participam dos diálogos entre agentes humanos e computacionais necessitam estar a todo tempo disponíveis e de fácil entendimento, para que o diálogo entre agente humano e computacional possa fluir sem ruído. Já as unidades de conhecimento que servem para funções que o sistema realiza sozinho não precisam estar disponíveis todo o tempo, mas apenas para uma eventual consulta do agente humano, pois essas atividades são automáticas e transparentes para ele. As unidades de conhecimento necessárias ao diálogo entre agente humano e computacional devem ser representadas diretamente, enquanto as que servem apenas para realizar funções exclusivas do sistemas serão representadas indiretamente. Essas descrições serão utilizadas posteriormente na realização do protótipo da interface de diálogo. Neste protótipo, pode-se decidir entre mudar o status de algumas estruturas de direta para indireta e vice-versa, não sendo essa classificação das unidades de conhecimento feita aqui inicialmente a definitiva.

- Lugares ativos: **funções realizadas automaticamente e exclusivamente pelo sistema, sem a participação do agente humano.** Como essas funções foram classificadas como monádicas ou diádicas no diagrama de fluxo cognitivo, elas não necessitam de inteligência humana, podendo ser completamente realizadas pelo sistema, liberando o agente humano para se concentrar nas atividades inteligentes. Nesta fase, apenas identificamos essas funções, que serão melhor descritas nos requisitos de amplificação de inteligência. Essas atividades devem ser classificadas como entrada de conhecimento, funções que tipicamente retiram conhecimento de alguma fonte externa e incorporam ao sistema, ou como criação de conhecimento, funções que criam um conhecimento interno do sistema que não existia em nenhuma outra fonte externa.
- Superlugares: **atividades onde há cooperação entre agentes humanos e computacionais para absorver algumas unidades de conhecimento e produzir outras.** Nestes pontos, atuarão agentes computacionais presentes no subsistema de agente inteligentes

do SAI a ser construído, e essas funções serão realizadas por esse subsistema de agentes em um diálogo com o agente humano. Os papéis e as formas de interação com o agente humano do subsistema de agentes serão formalizadas nos casos de amplificação de inteligência e nos requisitos de amplificação de inteligência.

Discriminar cada uma dessas estruturas.

Artefatos de saída

- Tipos de lugares na modelagem em Redes Semiônicas

Material de apoio

- Ferramenta SNTTool, disponível em <http://sntool.sourceforge.net/>

A.2.10 Identificar casos de amplificação de inteligência

Responsável

Arquiteto de SAI

Artefatos de entrada

- Modelagem em Redes Semiônicas

Descrição

Deve-se identificar os casos de amplificação de inteligência nas redes que modelam os superlugares da modelagem dinâmica em Redes Semiônicas. Essa identificação consiste em, a partir dessas redes, numerar a sequência de manipulações de unidades de conhecimento realizadas pelo agente humano e pelo subsistema de agentes computacionais que carregam os motores de solução, nomeando as interações. Isso dará origem aos diagramas de casos de amplificação de inteligência.

A partir desses diagramas, deve-se descrever cada um dos casos de amplificação de inteligência. Essa descrição deve apresentar:

- ID
- Inputs (unidades de conhecimento necessárias)
- Outputs (unidades de conhecimento criadas)
- Eventos de disparo (eventos necessário para disparar o caso de amplificação de inteligência)

- Ator primário (agente humano ou agentes computacionais)
- Descrição textual
- Fluxo primário (passo a passo)
- Fluxos alternativos (caso existam)

Há um modelo para essa descrição no material de apoio desta atividade.

Artefatos de saída

- Casos de amplificação de inteligência

Material de apoio

- Descrição de caso de amplificação de inteligência, que segue abaixo:

NOME DO CASO DE AMPLIFICAÇÃO DE INTELIGÊNCIA

ID	CA1
Inputs	Unidades de conhecimento necessárias: 1. XXXXXXXXXXXXXXXXXXXX 2. ...
Outputs	Unidades de conhecimento criadas: 1. YYYYYYYYYYYYYYYYYY 2. ...
Eventos de disparo	Os eventos necessários ...
Ator Primario	Agente ator
Descrição	Uma descrição textual resumida, porém completa.

Fluxo primário

1	Passo	
2	Passo	

1	Passo	
3	Passo	
4	Passo	

Fluxo secundário

1	Passo	
2	Passo	
3	Passo	
4	Passo	

Tab. A.3: Tabela modelo para descrição de casos de amplificação de inteligência

A.2.11 Prototipar a interface de diálogo

Responsável

Arquiteto de SAI

Artefatos de entrada

- Diagrama de fluxo cognitivo classificado com pontos de amplificação
- Modelagem em Redes Semiônicas
- Tipos de lugares na modelagem em Redes Semiônicas
- Casos de amplificação de inteligência

Descrição

Os seguintes artefatos de entrada desta atividade fornecem as seguintes informações para a construção deste protótipo:

- Tipos de lugares na modelagem em Redes Semiônicas: possuem a identificação das **unidades conhecimento de representação direta**, ou seja, os signos que devem possuir representação icônica na interface de diálogo.
- Casos de amplificação de inteligência: trazem consigo as definições de como as **interações entre agentes humano e computacional** devem ocorrer na interface de diálogo.

Estes dois artefatos de entrada possuem nos seus campos todas as definições necessárias para montar uma interface que é uma **ontologia** comum a ambos os agentes humano e computacional.

Não se deve perder a idéia de que o que está sendo construído é um conjunto de signos para as unidades de conhecimento identificadas, e que semioses rápidas e eficientes devem ser possíveis através desses signos, que representam conceitos importantes para que o agente humano consiga cooperar na resolução do problema. Caso já haja algum tipo de interface de diálogo utilizada pelos especialistas humanos para amplificar a mente deles, ainda que primitiva, sugere-se aproveitar ao máximo a idéia, por questões de manutenção de uma cultura a qual eles já se acostumaram a utilizar no seu fluxo cognitivo. O resultado final será alcançado através de um ciclo de validações com os stakeholders no processo etnográfico.

Este protótipo da interface de diálogo não precisa ser funcional, ou seja, não precisa ser de fato uma interface obtida a partir de um código em alguma linguagem de programação. Dessa maneira, esse protótipo pode, com efeito, ser apenas um desenho da interface, detalhando os conceitos representados, seus ícones, os processos de amplificação e como eles ocorrem.

Os principais pontos que devem estar presentes na interface de diálogo são:

- Representação icônica dos elementos mais relevantes: os conceitos mais relevantes no domínio e na construção da solução devem ser representados iconicamente, para que o agente humano perceba rapidamente seus estados no modo de operação e seja capaz de manipula-los facilmente;
- Visualização da situação do domínio representado e do planejamento das soluções no passado, presente e futuro, por meio de uma linha temporal: além de representar a solução em um determinado instante, a interface de diálogo deve possibilitar a visualização de planos realizados no passado e de planos de execução para horizontes de tempos futuros;
- Mecanismos de interação do agente humano com o planejamento: o agente humano deve ser capaz de interagir com o agente computacional por meio da interface de diálogo, inserindo dados manualmente e participando da construção da solução, contribuindo com suas intuições. A interface de diálogo deve ser uma ontologia comum a esses agentes, para fins de comunicação e diálogo.

Artefatos de saída

- Protótipo da interface de diálogo

Material de apoio

Não há.

A.2.12 Propor componentes do SAI**Responsável**

Arquiteto de SAI

Artefatos de entrada

- Estruturas estáticas e dinâmicas do domínio
- Interface com sistemas externos
- Diagrama de fluxo cognitivo classificado com pontos de amplificação
- Casos de amplificação de inteligência
- Protótipo dos motores de solução
- Protótipo da interface de diálogo

Descrição

A proposta de componentização deve sugerir subsistemas para implementar o SAI. Essa atividade é importante para separar claramente as principais funções do sistema em subsistemas distintos, facilitando desta forma a implementação de cada função e eventuais manutenções no sistema. A modularização também favorece a implementação do sistema em etapas, já que diversos subsistemas podem ser implementados independentemente. Por fim, uma vantagem adicional é que o tempo para disponibilização do sistema pode ser minimizado, uma vez que ele pode entrar em produção e auxiliar nas tomadas de decisão mesmo sem a implementação completa de todos os subsistemas. Além disto, a construção dos subsistemas pode em alguns casos ser paralelizada.

A sugestão dos subsistemas dos SAI deve partir da proposta de componentização padrão dos SAI, apresentada e descrita em detalhes na figura 5.6 na seção 5.5.3. A partir dessa componentização padrão, deve-se customizar os subsistemas existentes, criar eventuais novos subsistemas, excluir algum subsistema desnecessário para o problema em questão ou sub-dividir os subsistemas em outros subsistemas, quando pertinente.

Essa proposta de componentização é apenas uma sugestão inicial, que será refinada nos fluxos principais de trabalho seguintes da metodologia.

Artefatos de saída

- Componentização do SAI

Material de apoio

- Componentização padrão dos SAI, apresentada na figura 5.6.

A.2.13 Estimar resultados e ganhos**Responsável**

Engenheiro de SAI

Artefatos de entrada

- Componentização do SAI
- Protótipo dos motores de solução
- Protótipo da interface de diálogo

Descrição

O objetivo da atividade de estimativa de resultados e ganhos é analisar a proposta de componentização sugerida para o SAI em termos de ganhos, riscos e custos de cada um dos subsistemas. Com isso, torna-se possível posteriormente dividir o projeto em etapas, decidindo a prioridade de implementação de cada um dos subsistemas de tal maneira que os subsistemas de maior ganho, menor custo e menor risco sejam implementados nos primeiros ciclos do projeto, buscando fazer com que a maior parte do ganho seja alcançado em menor parte do tempo de projeto.

Esta atividade deve avaliar as estimativas de resultados e ganhos com a implementação do sistema de amplificação de inteligência descrito nas atividades anteriores. O objetivo desta análise é balizar o plano de ciclos de desenvolvimento do software minimizando os custos e riscos e maximizando os ganhos e o retorno do investimento.

Deve-se, portanto, descrever os ganhos estimados nos três indicadores de ganhos, que são utilização, throughput e aceitação (apresentados na seção 2.2), com a implementação de cada um dos subsistemas sugeridos na proposta de componentização, analisando-os separadamente. Além de analisar os ganhos, deve-se também analisar os riscos e custos envolvidos na construção de cada subsistema, assim como a dependência entre eles. Portanto, a análise deve levar em conta:

- Sistema
 - * Ganhos
 - Utilização
 - Throughput
 - Aceitação

- * Riscos
- * Custos
- * Dependências

Essa primeira análise deve ser feita em linguagem natural, descrevendo os ganhos. Pode-se montar uma tabela como a tabela A.5, que está presente no material de apoio desta atividade.

Juntamente com essa descrição em linguagem natural, deve-se classificar os os ganhos, riscos e custos com valores lingüísticos, que serão posteriormente transformados em valores numéricos, segundo uma tabela de conversão. O objetivo é quantificar o ganho e o risco para fornecer subsídios para o plano de ciclos de desenvolvimento do software. Os valores lingüísticos dessa classificação de intensidade devem ser os seguintes:

- Não há
- Baixo
- Médio
- Alto

Além disso, deve-se classificar o ganho em:

- Direto (tangível)
- Indireto (intangível)

Assim, um determinado subsistema "A" pode ter um ganho de utilização "alto indireto", de throughput "médio direto", de aceitação "não há" e um risco baixo direto. Pode-se aproveitar a mesma tabela A.5 e fazer essa classificação logo após a descrição em linguagem natural.

Após a classificação de acordo com os valores lingüísticos anteriores, faz-se necessário converter a análise para valores numéricos. A tabela A.4 é sugerida para realizar essa conversão:

Intensidade	Tangibilidade do ganho		Tangibilidade do risco	
	Indireto ou intangível	Direto ou tangível	Indireto ou intangível	Direto ou tangível
Não há	0	0	6	8
Baixo	2	4	4	6
Médio	4	6	2	4
Alto	6	8	1	1

Tab. A.4: Fatores de conversão para os valores lingüísticos de estimativa de ganhos e riscos

Perceba que o custo não aparece na tabela de conversão. Isso se dá pelo fato de ser uma variável de menor importância na análise de prioridade, pois interessa ao implementador e não ao cliente, ficando apenas para o desempate na análise.

Perceba também que a tabela A.4 foi montada de tal maneira que o valor numérico do ganho total (ganho de utilização + ganho de throughput + ganho de aceitação) multiplicado pelo valor numérico do risco dá uma pontuação maior para os subsistema com maior ganho e menor risco. Dessa maneira, a pontuação final da análise é uma conta simples:

- Pontuação = $G \times R$, donde $G = U + T + A$. (G = Ganho total, R = risco, U = ganho de utilização, T= ganho de throughput, A = ganho de aceitação)

A tabela A.6 para esta avaliação numérica está presente no material de apoio desta atividade, preenchida com um exemplo. No exemplo em questão, o subsistema “A” tem uma pontuação maior que os outros, indicando que possui um maior ganho e um menor risco, devendo ser implementado nos primeiros ciclos do projeto.

Ao fim desse tratamento numérico, os subsistemas com maior pontuação são os de maior ganho e menor risco de implementação, estabelecendo uma estimativa de resultados e ganhos para justificar o plano de ciclos de desenvolvimento do software posterior. O custo pode ser usado como critério de desempate quando necessário, ou até mesmo como justificativa para implementação de um subsistema de baixa pontuação em fases iniciais, caso ele tenha um custo muito baixo.

Artefatos de saída

- Estimativa de resultados e ganhos

Material de apoio

- Tabela para análise de cada um dos subsistemas de acordo com ganhos, riscos, custos e dependências dos subsistemas sugeridos para a componentização do SAI. Ela está preenchida de maneira arbitrária, apenas como um exemplo.

Subsistema	Ganhos			Riscos	Custos
	Utilização	Throughput	Aceitação		

A	Descrição em linguagem natural... Valor lingüístico: alto indireto	Descrição em linguagem natural... Valor lingüístico: alto indireto	Descrição em linguagem natural... Valor lingüístico: alto direto	Descrição em linguagem natural... Valor lingüístico: baixo direto	Descrição em linguagem natural... Valor lingüístico: médio direto
B	Descrição em linguagem natural... Valor lingüístico: médio indireto	Não há	Descrição em linguagem natural... Valor lingüístico: médio indireto	Descrição em linguagem natural... Valor lingüístico: médio direto	Descrição em linguagem natural... Valor lingüístico: baixo direto
C	Descrição em linguagem natural... Valor lingüístico: médio indireto	Não há	Descrição em linguagem natural... Valor lingüístico: médio indireto	Descrição em linguagem natural... Valor lingüístico: médio direto	Descrição em linguagem natural... Valor lingüístico: baixo direto

Tab. A.5: Tabela modelo para análise de estimativas e ganhos dos subsistemas da componentização

- Tabela para avaliação numérica de ganhos e riscos. Ela está preenchida como um exemplo, e é equivalente ao exemplo da tabela A.5.

Subsistema	Ganhos			Ganho ($G = U + T + A$)	Risco (R)	Pontuação ($G \times R$)
	Utilização (U)	Throughput (T)	Aceitação (A)			
A	6	6	8	20	8	160
B	4	0	4	8	4	32

C	4	0	4	8	4	32
---	---	---	---	---	---	----

Tab. A.6: Tabela para avaliação numérica de ganhos e riscos. Seu preenchimento modelo é equivalente ao exemplo da tabela A.5

A.2.14 Planejar ciclos de desenvolvimento

Responsável

Engenheiro de SAI

Artefatos de entrada

- Componentização do SAI
- Estimativa de resultados e ganhos

Descrição

O plano de ciclos de desenvolvimento do software deve apresentar uma proposta de divisão do projeto do SAI, ou seja, a implementação do sistema em diferentes ciclos. Essa divisão visa mitigar riscos envolvidos com a implementação de alguns subsistemas, considerar a dependência e os custos dos subsistemas a serem implementados, conforme identificado na atividade de estimativa de resultados e ganhos, além de minimizar possíveis resistências e oposições à utilização do sistema, devido à um eventual choque cultural causado pela inserção do sistema completo de uma só vez no processo organizacional, procurando realizar as mudanças de paradigma por partes. O objetivo do plano de ciclos de desenvolvimento do software é garantir o maior ganho possível no menor tempo de projeto possível, ou seja, implementar os subsistemas de maior ganho e menor risco nos primeiros ciclos.

Deve-se, portanto, dividir o projeto de implementação do sistema em ciclos, atribuindo a implementação de cada um dos subsistemas para cada um dos ciclos modelados. Naturalmente, não há um número de ciclos ideal para todos os projetos, tampouco há um número de subsistemas ideal para cada um dos ciclos. Esses números devem ser atribuídos de acordo com a estimativa de resultados e ganhos de cada projeto, que é um artefato de entrada desta atividade. De acordo com o objetivo do plano de ciclos de desenvolvimento do software, deve-se selecionar os subsistemas de maior pontuação para os primeiros ciclos do plano de ciclos de desenvolvimento do software. Pode-se optar por desenvolver o projeto inteiro em apenas um ciclo. No entanto, sugere-se que seja adotada a estratégia de dividir para conquistar, justamente para minimizar o impacto da inserção do sistema na cultura do processo organizacional e mitigar riscos. Sugere-se ainda que sejam agrupados os subsistemas de acordo com suas pontuações obtidas

na estimativa de resultados e ganhos: subsistemas de pontuações próximas devem formar um grupo, que representará um ciclo distinto na implementação.

No exemplo apresentado pelas tabelas A.5 e A.6, o plano de ciclos de desenvolvimento do software poderia prever uma implementação em dois ciclos:

- Ciclo 1: subsistema A, que possui um ganho maior e um risco menor do que os outros subsistemas, tendo obtido a pontuação 160.
- Ciclo 2: subsistemas B e C, ambos com uma pontuação de 32.

Artefatos de saída

- Plano dos ciclos de desenvolvimento

Material de apoio

Não há.

A.2.15 Identificar requisitos de amplificação de inteligência

Responsável

Engenheiro de SAI

Artefatos de entrada

- Workflow operacional
- Estruturas estáticas e dinâmicas do domínio
- Interface com sistemas externos
- Protótipo dos motores de solução
- Diagrama de fluxo cognitivo classificado com pontos de amplificação
- Modelagem em Redes Semiônicas
- Tipos de lugares na modelagem em Redes Semiônicas
- Casos de amplificação de inteligência
- Protótipo da interface de diálogo

Descrição

O trabalho de levantamento desses requisitos é uma tarefa engenhosa. Com efeito, faz-se necessário reavaliar todos os artefatos produzidos e ser capaz de traduzir em linguagem natural, na forma de requisitos de usuário, o que o sistema deve fazer em acordo com o que foi modelado e diagnosticado nos artefatos anteriores. Isso não é uma tarefa simples, mas há maneiras de torná-la menos "dolorosa". Para que os requisitos sejam levantados de maneira padronizada, sugere-se que eles sejam escritos no padrão apresentado no material de apoio desta atividade. Para facilitar a sua organização, sugere-se seguir a ordem dos subsistemas levantados na atividade de proposta de componentização para o SAI, ou seja, levantar requisitos para cada um dos subsistemas separadamente. Caso um requisito se refira a mais de um subsistema, sugere-se separá-lo em dois ou mais requisitos, para que cada requisito se refira a apenas um subsistema. Dessa maneira, as decisões do plano de ciclos de desenvolvimento do software de implementar determinados subsistemas em determinadas fases do projeto poderá ser mais facilmente planejada a partir da escolha dos requisitos que se referem àqueles subsistemas planejados para uma determinada fase. Para auxiliar nesta tarefa, algumas correspondências úteis para extração dos requisitos referentes a cada um dos subsistemas sugeridos, relacionando artefatos produzidos anteriormente no processo, são sugeridos a seguir. Vale lembrar que, com efeito, pode haver requisitos de qualquer um dos subsistemas em qualquer artefato anteriormente desenvolvido, e as associações a seguir são sugeridas apenas para facilitar o trabalho de extração dos requisitos de amplificação de inteligência.

– Interface de diálogo:

- * Tipos de lugares na modelagem em Redes Semiônicas: grande parte dos requisitos referentes à este subsistema podem ser extraídos da identificação dos tipos de lugares na modelagem em Redes Semiônicas (unidades de conhecimento a serem representadas).
- * Identificação dos casos de amplificação de inteligência: mecanismos presentes na interface de diálogo para amplificar a mente.
- * Protótipo da interface de diálogo: estruturas podem ser referenciadas na descrição dos requisitos.

– Agentes computacionais:

- * Estruturas estáticas e dinâmicas do domínio: para os requisitos deste subsistema que se referem aos motores de solução, a identificação das estruturas estáticas e dinâmicas do domínio é o principal artefato, pois os problemas não estruturados representados pelos modos de operação dessas estruturas deverão ser resolvidos pelos algoritmos motores de solução. Assim sendo, uma série de características importantes para implementação destes algoritmos devem ser retiradas deste artefato, entre elas:
 - Funções objetivo com multi-objetivos
 - Regras de operação
 - Restrições do problema

- * Identificação dos pontos de amplificação de inteligência: a identificação dos pontos de amplificação no diagrama de fluxo cognitivo mostrará em quais pontos do processo cognitivo o subsistema de agentes deverá agir, e o diagrama evidencia os mecanismos que os agentes terão que realizar, ou seja, quais unidades de conhecimento eles terão como entrada e quais unidades de conhecimento deverão ser criadas.
 - * Modelagem dinâmica em Redes Semiônicas: a modelagem em Redes Semiônicas detalha a interação entre os agentes computacionais e humanos, evidenciando ainda mais os mecanismos a serem extraídos em forma de requisitos.
 - * Identificação dos casos de amplificação de inteligência: a identificação dos casos de amplificação de inteligência é o artefato mais utilizado no levantamento dos requisitos do subsistema de agentes computacionais, pois detalha bastante os casos de interação entre agentes computacionais e humanos, presentes nos diagramas de amplificação de inteligência advindos da modelagem em Redes Semiônicas.
 - * Protótipo dos motores de solução e protótipo da interface de diálogo: ambos os protótipos podem fornecer requisitos para este subsistema, advindos de necessidades especiais constatadas apenas na implementação.
- Leitura de sistemas externos:
- * Identificação das interfaces com sistemas externos: praticamente todos os requisitos referentes a este subsistema serão extraídos da identificação das interfaces com sistemas externos.
- Cálculos automáticos:
- * Diagrama de fluxo cognitivo completo: o diagrama de fluxo cognitivo completo, classificado e com os pontos de amplificação fornece grande parte da base para extração dos requisitos deste subsistema. As atividades classificadas como diádicas (mecânicas) deverão ser realizadas pelo sistema de maneira transparente para o usuário, para que este fique livre para realizar as atividades triádicas juntamente com os agentes.
 - * Modelagem dinâmica em Redes Semiônicas: complementa o DFC, pois simula o processo mapeado dinamicamente, ajudando na compreensão do mesmo.

Ao desenvolver os requisitos de amplificação de inteligência, deve-se mapear os atributos de um SAI, descritos na seção 5.5.1.

A descrição de cada um dos requisitos deve ter o seguinte conteúdo mínimo:

- Sumário: o título do requisito em forma de uma frase em linguagem natural que resuma sua descrição
- ID: identificador único para cada requisito
- Descrição: descrição completa do requisito
- Tipo: funcional (que dá origem a uma funcionalidade do sistema) ou não-funcional (que descreve características que o sistema deve possuir, mas não mapeia diretamente uma funcionalidade)
- Subsistema: subsistema sugerido para o SAI ao qual o requisito pertence.

- Prioridade: prioridade de implementação do requisito do projeto. Sugere-se que seja classificada a prioridade numa escala de 1 (maior prioridade) a 5 (menor prioridade). Essa prioridade está relacionada com o plano de ciclos de desenvolvimento do software, que se baseia na estimativa de ganhos para decidir quais subsistemas serão implementados nas consecutivas fases do projeto. Por exemplo, se o subsistema A foi escolhido para a primeira fase porque tinha a maior estimativa de ganho do projeto, provavelmente os requisitos que pertencem a este subsistema terão prioridade 1.
- Dependência: outros requisitos com os quais este possui uma relação de dependência, ou seja, requisitos que afetam e que são afetados por este.
- Data de criação: data que o requisito foi desenvolvido.

Artefatos de saída

- Requisitos de amplificação de inteligência

Material de apoio

Modelo de requisito de amplificação de inteligência, que segue abaixo:

Sumário	Título, uma frase que resuma o requisito
ID	Identificador único para o requisito
Descrição	Descrição completa em linguagem natural
Tipo	Funcional ou não-funcional
Subsistema	Subsistema sugerido na componentização ao qual este requisito pertence
Prioridade	Escala de 1(maior prioridade) a 5 (menor prioridade), relacionado com as decisões do plano de ferramentalização
Dependências	Quando um caso de uso A depende de um caso de uso B, uma mudança em B tem impacto em A.
Data de criação	

Tab. A.7: Tabela modelo para descrição dos requisitos de amplificação de inteligência

A.3 Código da rede semiônica das figuras das figuras 7.31, 7.32 e 7.33 para SNToolkit

Disponível em <https://pub.cflex.com.br/brainmerge/apendice/dfcSamarcoAgentes.xsn>

A.4 Descrição dos casos de amplificação de inteligência do MES Pátio Samarco

A seguir, apresenta-se a descrição dos casos de amplificação de inteligência identificados nas figuras 7.32 e 7.33. Primeiramente, os casos de amplificação relativos ao empilhamento:

1. Determinar Loci de Empilhamento

ID	CA1
Inputs	Unidades de conhecimento necessárias: <ol style="list-style-type: none"> 1. Posição e estado das máquinas 2. Mapa de situação dos pátios e pilhas 3. Amostra de entrada 4. Aderência da blendagem às orientações de embarque da carga 5. Fila de navios 6. Orientações de embarque 7. Campanhas
Outputs	Unidades de conhecimento criadas: <ol style="list-style-type: none"> 1. Loci de empilhamento
Eventos de disparo	O evento necessário para que os loci de empilhamento sejam determinados é material chegando para ser empilhado, vindo da usina, determinado na campanha atual.
Ator Primario	Motor de solução de empilhamento

ID	CAI
Descrição	<p>O agente computacional deverá, a partir dos inputs discriminados neste caso de amplificação, sugerir para o usuário humano os melhores loci de empilhamento para cobrir a quantidade de material vindo da usina, determinado na campanha atual. Esses loci devem otimizar o gerenciamento do pátio. A informação de máquinas, atividades e tempos de empilhamento fazem parte dessa sugestão de loci.</p> <p>As unidades de conhecimento necessárias e criadas devem estar representadas no meio de representação de maneira única. Os loci de empilhamento criados devem ser representados de maneira icônica para o usuário humano.</p>

Fluxo primário

1	Passo	Agente busca as unidades de conhecimento necessárias
2	Passo	Agente escolhe os loci de empilhamento baseado na otimização dos dados possuídos
3	Passo	Agente representa o melhores loci de empilhamento no meio de representação, para que o agente humano possa se utilizar da informação

2. Determinar Loci de Empilhamento Mandatório

ID	CA2
Inputs	<p>Unidades de conhecimento necessárias:</p> <ol style="list-style-type: none"> 1. Comando de empilhamento 2. Campanhas 3. Posição e estado das máquinas 4. Amostra de embarque 5. Mapa de situação das pilhas 6. Aderência da blendagem às orientações de embarque da carga 7. Amostra de entrada 8. Fila de navios 9. Orientações de embarque 10. Loci de empilhamento
Outputs	<p>Unidades de conhecimento criadas:</p> <ol style="list-style-type: none"> 1. Loci de empilhamento mandatório
Eventos de disparo	<p>Um locus de empilhamento sugerido pelo agente computacional e uma necessidade diferente do agente humano para determinar o locus de empilhamento oficial.</p>
Ator Primario	<p>Usuário humano</p>
Descrição	<p>O agente humano de empilhamento deverá ter a capacidade de sugerir loci de empilhamento mandatórios a partir do loci de empilhamento sugerido pelo agente computacional. A informação de máquinas, atividades e tempos da operação deve poder ser modificada também, pois faz parte da unidade de conhecimento loci de empilhamento. Todos os inputs desse caso de amplificação são capazes de influenciar nessa decisão, conforme explícito na descrição das estruturas estáticas e dinâmicas do domínio.</p> <p>Estes loci de empilhamento mandatório devem ser representados de maneira diferenciada no meio de representação, já que é uma nova unidade de conhecimento criada.</p>

Fluxo primário

1	Passo	Agente busca as unidades de conhecimento necessárias
2	Passo	Agente humano percebe a necessidade de empilhar em um locus diferente daquele sugerido pelo agente computacional
3	Passo	Agente humano modifica o meio de representação de maneira a identificar os loci de empilhamento mandatário

3.Determinar Loci de Empilhamento Híbrido

ID	CA3
Inputs	<p>Unidades de conhecimento necessárias:</p> <ol style="list-style-type: none"> 1. Posição e estado das máquinas 2. Mapa de situação do pátio e das pilhas 3. Aderência da blendagem às orientações de embarque da carga 4. Amostra de entrada 5. Fila de navios 6. Campanhas 7. Orientações de embarque 8. loci de empilhamento mandatário
Outputs	<p>Unidades de conhecimento criadas:</p> <ol style="list-style-type: none"> 1. Loci de empilhamento
Eventos de disparo	O evento necessário para que o locus de empilhamento híbrido seja determinado é a determinação de locus de empilhamento mandatário pelo agente humano
Ator Primario	Motor de solução de empilhamento

ID	CA3
Descrição	O agente computacional deverá, a partir dos loci de empilhamento mandatório estipulados pelo usuário humano, sugerir loci de empilhamento. Isso significa que o agente computacional deve considerar fixo os loci de empilhamento mandatório com as atividades e tempos das máquinas estipulados pelo usuário humano e calcular os loci faltantes para cobrir o material que chega para ser empilhado.

Fluxo primário

1	Passo	Agente detecta loci de empilhamento mandatório
2	Passo	Agente busca conhecimentos necessários
3	Passo	Agente calcula loci de empilhamento levando em consideração os loci mandatórios do agente humano.
4	Passo	Agente representa o melhores loci de empilhamento híbridos no meio de representação, para que o agente humano possa se utilizar da informação

4. Oficializar Loci de Empilhamento

ID	CA4
Inputs	<p>Unidades de conhecimento necessárias:</p> <ol style="list-style-type: none"> 1. Comando de empilhamento 2. Campanhas 3. Posição e estado das máquinas 4. Amostras de embarque 5. Mapa de situação e estado do pátio 6. Aderência da blendagem às orientações de embarque da carga 7. Amostras de entrada 8. Fila de navios 9. Orientações de embarque 10. Loci de empilhamento
Outputs	<p>Unidades de conhecimento criadas:</p> <ol style="list-style-type: none"> 1. Plano de empilhamento e recuperação
Eventos de disparo	Necessidade do agente humano de oficializar um plano para empilhar
Ator Primario	Usuário humano
Descrição	O agente humano, a partir da análise dos inputs desse caso de amplificação de inteligência, decide que os loci de empilhamento sugeridos atualmente são adequados e os oficializa como um plano de empilhamento e recuperação

Fluxo primário

1	Passo	Agente humano percebe os loci de empilhamento sugeridos pelo agente computacional
2	Passo	Agente busca conhecimentos necessários
3	Passo	Agente decide que os loci de empilhamento são adequados.
4	Passo	Agente oficializa os loci de empilhamento como um plano de empilhamento e recuperação oficial.

5. Oficializar Loci de Empilhamento híbrido

ID	CA5
Inputs	Unidades de conhecimento necessárias: <ol style="list-style-type: none"> 1. Comando de empilhamento 2. Campanhas 3. Posição e estado das máquinas 4. Amostras de embarque 5. Mapa de situação e estado do pátio 6. Aderência da blendagem às orientações de embarque da carga 7. Amostras de entrada 8. Fila de navios 9. Orientações de embarque 10. Loci de empilhamento mandatório
Outputs	Unidades de conhecimento criadas: <ol style="list-style-type: none"> 1. Plano de empilhamento e recuperação
Eventos de disparo	Informação relacionada aos inputs que torna os loci de empilhamento sugeridos pelo agente computacional não-adequados ou pouco adequados.
Ator Primario	Usuário humano
Descrição	O agente humano, a partir da análise dos inputs desse caso de amplificação de inteligência, decide que os loci de empilhamento mandatórios atualmente são adequados e os oficializa como um plano de empilhamento e recuperação

Fluxo primário

1	Passo	Agente humano percebe os loci de empilhamento mandatórios
2	Passo	Agente busca conhecimentos necessários
3	Passo	Agente decide que os loci de empilhamento são adequados.

1	Passo	Agente humano percebe os loci de empilhamento mandatórios
4	Passo	Agente oficializa os loci de empilhamento como um plano de empilhamento e recuperação oficial.

Com relação às atividades de recuperação, tem-se os seguintes casos de amplificação:

1. Escolher a carga do navio a ser carregada

ID	CA6
Inputs	Unidades de conhecimento necessárias: 1. Fila de navios
Outputs	Unidades de conhecimento criadas: 1. Carga do navio
Eventos de disparo	Navio atracado para carregamento
Ator Primario	Usuário humano
Descrição	O agente humano deve selecionar uma das cargas do navio atracado para ser carregada. Essa carga passa a ser carga considerada para formar o plano de empilhamento e recuperação

Fluxo primário

1	Passo	Navio atracado para carregamento com suas cargas discriminadas
2	Passo	Agente humano seleciona a carga a ser carregada
3	Passo	A carga escolhida passa a ser considerada para formar o plano de recuperação

2. Determinar Loci de Recuperação

ID	CA7
Inputs	<p>Unidades de conhecimento necessárias:</p> <ol style="list-style-type: none"> 1. Fila de navios 2. Amostra de entrada 3. Mapa de situação do pátio e das pilhas 4. Aderência da blendagem às orientações de embarque da carga 5. Posição e estado das máquinas 6. Orientações de embarque 7. Carga do navio
Outputs	<p>Unidades de conhecimento criadas:</p> <ol style="list-style-type: none"> 1. Loci de recuperação
Eventos de disparo	Carga do navio atracado escolhida para recuperação
Ator Primario	Motor de solução de recuperação
Descrição	O agente computacional de carregamento deve, a partir das unidades de conhecimento dos inputs, determinar os loci de recuperação que atendem a qualidade e quantidade determinadas pela carga selecionada. As informações de atividades, tempos e máquinas fazem parte da unidade de conhecimento loci de recuperação

Fluxo primário

1	Passo	Agente percebe a carga selecionada, sua qualidade e quantidade
2	Passo	Agente analisa os inputs e decide os loci de recuperação.
3	Passo	Agente sugeri loci de recuperação, marcando-os no meio de representação.

3. Determinar Loci de Recuperação Mandatórios

ID	CA8
Inputs	<p>Unidades de conhecimento necessárias:</p> <ol style="list-style-type: none"> 1. Comando de recuperação 2. Fila de navios 3. Amostra de entrada 4. Mapa de situação o pátio e das pilhas 5. Aderência da blendagem às orientações de embarque da carga 6. Amostra de embarque 7. Posição e estado das máquinas 8. Campanhas 9. Orientações de embarque 10. Loci de recuperação
Outputs	<p>Unidades de conhecimento criadas:</p> <ol style="list-style-type: none"> 1. Loci de recuperação mandatória
Eventos de disparo	Informação relacionada aos inputs que torna os loci de recuperação sugeridos pelo agente computacional não-adequados ou pouco adequados.
Ator Primario	Usuário humano
Descrição	O agente humano deve ser capaz de, a partir dos loci de recuperação sugeridos pelo agente computacional, determinar loci de recuperação mandatórios, que serão fixados. Deve ser capaz de modificar também informações de tempos, atividades e máquinas, que fazem parte da unidade de conhecimento loci de recuperação.

Fluxo primário

1	Passo	Agente humano percebe os loci de recuperação sugeridos pelo agente computacional
2	Passo	Agente humano percebe os conhecimentos necessários
3	Passo	Agente humano fixa loci de recuperação mandatórios

4. Determinar Loci de Recuperação Híbridos

ID	CA9
Inputs	Unidades de conhecimento necessárias: <ol style="list-style-type: none"> 1. Fila de navios 2. Amostra de entrada 3. Mapa de situação do pátio e das pilhas 4. Aderência da blendagem às orientações de embarque da carga 5. Posição e estado das máquinas 6. Campanhas 7. Orientações de embarque 8. Loci de recuperação mandatórios
Outputs	Unidades de conhecimento criadas: <ol style="list-style-type: none"> 1. Loci de recuperação
Eventos de disparo	Determinação de loci de recuperação mandatórios
Ator Primario	Motor de solução de recuperação
Descrição	O agente computacional deve captar os loci de recuperação mandatórios e ser capaz de sugerir loci de recuperação híbridos, considerando fixos os mandatórios do agente humano

Fluxo primário

1	Passo	Agente computacional percebe os loci de recuperação mandatórios
2	Passo	Agente computacional percebe os conhecimentos necessários
3	Passo	Agente computacional sugeri loci de recuperação híbridos, considerando fixos os loci mandatórios

5. Oficializar Loci de Recuperação

ID	CA10
Inputs	<p>Unidades de conhecimento necessárias:</p> <ol style="list-style-type: none"> 1. Fila de navios 2. Amostra de entrada 3. Mapa de situação do pátio e das pilhas 4. Aderência da blendagem às orientações de embarque da carga 5. amostra de embarque 6. Posição e estado das máquinas 7. Orientações de embarque 8. Loci de recuperação
Outputs	<p>Unidades de conhecimento criadas:</p> <ol style="list-style-type: none"> 1. Plano de empilhamento e recuperação
Eventos de disparo	Necessidade do agente humano de oficializar um plano para recuperar
Ator Primario	Usuário humano
Descrição	O agente humano deve ser capaz de tornar os loci de recuperação sugeridos pelo agente computacional um plano oficial de recuperação

Fluxo primário

1	Passo	Usuário humano percebe uma necessidade de oficializar um plano de recuperação
2	Passo	Usuário humano relaciona os inputs
3	Passo	Usuário humano percebe os loci de recuperação sugeridos pelo agente computacional como adequados
4	Passo	Usuário oficializa os loci, criando um plano de empilhamento e recuperação

6. Oficializar Loci de Recuperação híbridos

ID	CA11
Inputs	<p>Unidades de conhecimento necessárias:</p> <ol style="list-style-type: none"> 1. Fila de navios 2. Amostra de entrada 3. Mapa de situação do pátio e das pilhas 4. Aderência da blendagem às orientações de embarque da carga 5. amostra de embarque 6. Posição e estado das máquinas 7. Loci de recuperação mandatórios 8. Orientações de embarque
Outputs	<p>Unidades de conhecimento criadas:</p> <ol style="list-style-type: none"> 1. Plano de empilhamento e recuperação
Eventos de disparo	Necessidade do agente humano de oficializar um plano para recuperar
Ator Primario	Usuário humano
Descrição	O agente humano deve ser capaz de tornar os loci de recuperação mandatórios um plano oficial de recuperação

Fluxo primário

1	Passo	Usuário humano percebe uma necessidade de oficializar um plano de recuperação
2	Passo	Usuário humano relaciona os inputs
3	Passo	Usuário humano percebe os loci de recuperação mandatórios como adequados
4	Passo	Usuário oficializa os loci, criando um plano de empilhamento e recuperação

A.5 Estimativas de resultados e ganhos do MES Pátio Samarco

A seguir, são descritos os ganhos estimados obtidos com a implementação do sistema, apresentados de acordo com a distribuição dos subsistemas especificados na seção 7.2.12. Conforme visto, os ganhos são descritos através dos critérios de utilização, throughput e aceitação. Além da especificação dos ganhos, são descritos também os riscos e custos envolvidos na construção de cada subsistema, o que auxiliará na construção do plano de ciclos de desenvolvimento do software, apresentado na seção 7.2.14.

Subsistemas:

– Interface de diálogo

* Ganhos

- Utilização: agilidade da programação permite que o usuário reaja mais rápido a rupturas operacionais. Evita perdas de oportunidades.
- Throughput: planos melhores resultam em maiores taxas.
- Aceitação: a interface gráfica permite ao usuário explorar mais opções para os planos de empilhamento e recuperação levando a uma possível melhora na qualidade do produto empilhado e recuperado. Além disso, a interface computacional diminui a possibilidade de erros por automatizar a representação do pátio. Um operador, por exemplo, poderia se equivocar no balizamento de uma pilha e efetuar um planejamento inválido, o que demandaria um retrabalho.

* Riscos: baixo, pois a implementação da interface gráfica computacional não oferece nenhum risco considerável. A especificação para este subsistema está bem definida e a tecnologia necessária para seu desenvolvimento é bastante difundida. Com relação à aceitação da interface computacional por parte dos usuários, provavelmente haverá pouca resistência já que a interface computacional é bastante semelhante à interface manual já utilizada e os ganhos obtidos com sua utilização são facilmente percebidos.

* Custos: o custo para a implementação da interface gráfica é médio. Mesmo com o fato de já ter sua funcionalidade totalmente especificada, a implementação de interfaces gráficas com manipulação de conceitos geralmente é tarefa demorada e trabalhosa.

– Leitura de sistemas externos

* Ganhos

- Utilização: intangível. A concentração das informações dos diversos sistemas externos em um único sistema facilita a distribuição do conhecimento e leva a uma melhor utilização dos recursos do pátio.
- Throughput: não há.
- Aceitação: atualmente, grande parte das informações dos sistemas externos utilizadas são copiadas manualmente em relatórios e formulários. Com a automação dessas tarefas, aumentará a confiabilidade destas informações tornando

possível, além disso, sua validação automática com checagens de consistência. Com a menor possibilidade de erros, diminui também a possibilidade da confecção de planos de empilhamento e recuperação que resultam em produtos com qualidades abaixo das especificadas pelos clientes.

- * Riscos: do ponto de vista computacional, o risco de desenvolvimento deste subsistema é baixo, já que trata-se basicamente da implementação de “queries” para leitura das tabelas existentes nos sistemas externos. No entanto, há um risco alto de que a qualidade dos dados obtidos não seja confiável. Conforme visto na seção 7.2.3, a confiabilidade das informações obtidas nos sistemas CRM e MES é sólida, mas ainda não é possível confiar totalmente nos dados armazenados no PIMS para a definição da localização e operação das máquinas. Dessa forma, o sistema deve ser projetado para aceitar entradas manuais. O risco, neste caso, é o de que o excesso de entradas manuais cause um desgaste no usuário, devido a uma grande oposição à aceitação na utilização do sistema.
 - * Custos: o custo de implementação deste subsistema é baixo. Existe a possibilidade, no entanto, de um custo alto para melhorar a qualidade dos dados armazenados nos sistemas externos, resultando em despesas com instrumentação, manutenção, etc, mas isso não se relaciona diretamente ao custo do sistema. A especificação dessas melhorias na qualidade das informações armazenadas nos sistemas externos, assim como uma estimativa de seu custo, estão fora do escopo deste trabalho.
- Gerenciamento da base de dados
- * Ganhos
 - Utilização: não há.
 - Throughput: não há.
 - Aceitação: a existência de uma base de dados dedicada permite o arquivamento e recuperação do histórico das operações e estados do pátio, tornando possível a geração de diversos relatórios. Pode-se, através da análise destes relatórios dos históricos, melhorar o entendimento do problema e estudar casos específicos de melhorias no processo levando, possivelmente, a melhorias nas qualidades dos produtos comercializados.
 - * Riscos: a implementação da base de dados dedicada é uma tarefa de baixo risco.
 - * Custos: a implementação da base de dados dedicada é uma tarefa de custo baixo pois trata-se de uma base de dimensões pequenas, com poucas tabelas e uma arquitetura bastante simples.
- Gerador de relatórios
- * Ganhos
 - Utilização: com a implementação deste subsistema aumenta a agilidade na geração dos relatórios utilizados atualmente pela equipe de Estocagem e Embarque. Além disso, existe a possibilidade de geração de relatórios diferentes dos já existentes e que não são produzidos atualmente por falta de tempo ou outros recursos necessários no levantamento e organização dos dados.

- Throughput: a geração automática dos relatórios facilita a comunicação interna da equipe de Estocagem e Embarque, assim como a comunicação entre diferentes departamentos.
 - Aceitação: a geração dos relatórios facilita o estudo dos histórico das operações de empilhamento e recuperação resultando em uma análise mais produtiva dos erros e tendências comuns presentes no processo de gerenciamento do pátio.
 - * Riscos: a implementação do subsistema para geração dos relatórios é uma tarefa de baixo risco.
 - * Custos: o custo de implementação deste subsistema depende diretamente do número de relatórios a serem implementados de acordo com a solicitação da Samarco, podendo variar de um custo bastante baixo até um custo médio se o número de relatórios for muito alto.
- Agentes computacionais inteligentes
- * Ganhos
 - Utilização: alto ganho, pois o planejamento automático leva a uma série de ganhos na utilização dos recursos disponíveis no pátio - 1) minimização da movimentação de máquinas, 2) é possível estudar as melhores oportunidades para a realização das manutenções dos diversos equipamentos, 3) com o planejamento automático, os planos são efetuados considerando não só os efeitos das operações no presente, mas também suas consequências em um período futuro de planejamento, levando a uma antecipação de eventuais problemas como falta ou subutilização de recursos.
 - Throughput: baixo/intangível, aumento de produtividade dos usuários.
 - Aceitação: o planejamento automático torna mais fácil a consideração de objetivos diversos, e muitas vezes conflitantes, no gerenciamento do pátio. Além disso, aumenta a capacidade do usuário de explorar diferentes soluções e simular diferentes cenários, levando a planejamentos com de melhor qualidade.
 - * Riscos: o risco da implementação deste subsistema depende bastante do conhecimento da equipe responsável pelo desenvolvimento do sistema, assim como do sucesso de se incluir no sistema o conhecimento dos especialistas da equipe de Estocagem e Embarque. Há precedentes na literatura na implantação bem sucedida de algoritmos semelhantes e na filosofia de suporte à tomada de decisão.
 - * Custos: o custo na implantação deste subsistema é alto. Apesar de existirem soluções já desenvolvidas para problemas semelhantes, existe a necessidade de estudar as particularidades do problema e testar variações da solução proposta até a obtenção de resultados satisfatórios. Essas tarefas geralmente consomem bastante tempo e alocação de pessoal com alto conhecimento do problema estudado, assim como das possíveis técnicas a serem aplicadas na sua solução.
- Qualidade e cálculos automáticos
- * Ganhos

- **Utilização:** este subsistema realiza os cálculos médios de qualidade da pilha, assim como as qualidades estimadas baliza a baliza. Isso torna muito mais eficiente a rastreabilidade da qualidade nas pilhas armazenadas no pátio e recuperadas.
 - **Throughput:** atualmente, o estudo detalhado das medidas de qualidade nos embarques ocorre apenas em situações mais críticas, pois é praticamente impossível esta análise para todos os embarques, já que se trata de uma tarefa trabalhosa e demorada. Com a implementação do sistema, este cálculo pode ser feito para todos os navios.
 - **Aceitação:** com a implementação deste subsistema torna-se possível completar, juntamente com outros sistemas já existentes, a cadeia de rastreabilidade da qualidade dos produtos desde a mina até o embarque nos navios. Além disso, a equipe de Estocagem e Embarque terá mais tempo livre para simular diferentes cenários e explorar melhores resultados. Existe também a possibilidade, em um ciclo futuro (no subsistema de cálculos avançados) de considerar a influência dos problemas da degradação (com modelos matemáticos) e umidade na qualidade dos produtos, assim como a incerteza envolvida nos cálculos de qualidade. Para finalizar, a automatização reduz eventuais erros nos cálculos de qualidade feitos atualmente de forma manual.
- * **Riscos:** a implementação do subsistema de cálculo de qualidade como uma automatização da forma com é feito hoje tem um risco baixo, pois tratam-se de cálculos matemáticos com fórmulas já conhecidas. No entanto, o refinamento do cálculo da rastreabilidade através da consideração da incerteza e da influência da degradação e umidade no produto tem um risco alto, já que não se conhece métodos para realizá-lo, estando o sucesso dependente de estudos futuros. É importante observar que mesmo sem esses eventuais refinamentos, o sistema deve gerar resultados satisfatórios, já que a forma como os cálculos são feitos atualmente representam uma boa aproximação da qualidade real medida.
- * **Custos:** o custo para a automatização dos cálculos feitos hoje manualmente é médio; é uma tarefa trabalhosa, mas a princípio não é necessário nenhum estudo adicional do problema. O custo para refinar este cálculo da qualidade e considerar fatores como degradação (com modelos matemáticos), umidade e incerteza é alto pois depende do sucesso em representar adequadamente todas as diversas variáveis que influenciam estes fatores, assim como a interrelação entre essas diversas variáveis.

A tabela A.30 consolida os ganhos, riscos e custos descritos anteriormente.

Subsistema	Ganhos			Riscos	Custos
	Utilização	Throughput	Aceitação		
Interface de diálogo	Alto Intangível	Alto Intangível	Alto Tangível	Baixo	Médio

Subsistema	Ganhos			Riscos	Custos
	Utilização	Throughput	Aceitação		
Leitura de sistemas externos (MES)	Médio Indireto	Não há	Médio Indireto	Médio	Baixo
Leitura de sistemas externos (CRM)	Médio Indireto	Não há	Médio Indireto	Médio	Baixo
Leitura de sistemas externos (PIMS)	Médio Indireto	Não há	Médio Indireto	Alto	Baixo
Gerenciamento da base de dados	Não há	Não há	Médio	Baixo	Baixo
Gerador de relatórios	Médio	Médio	Médio	Baixo	Baixo ou médio (depende da quantidade de relatórios)
Agentes computacionais inteligentes	Alto	Alto	Alto	Médio	Alto
Qualidade e cálculos automáticos (cálculos básicos, sem considerar incerteza, umidade e degradação)	Alto	Alto	Alto	Baixo	Médio
Qualidade e cálculos automáticos refinados (cálculos avançados, considerando incerteza, umidade e degradação)	Alto	Alto	Alto	Alto	Alto

Tab. A.30: Consolidação dos ganhos, riscos e custos

A tabela A.31 mostra os ganhos e riscos convertidos em valores numéricos, de acordo com os fatores de conversão da tabela A.4. Calculou-se também uma pontuação total que reflete a prioridade e a importância de cada subsistema.

Subsistema	Ganhos			Ganho U+T+A	Risco	Pontuação G x R
	Utilização (U)	Throughput (T)	Aceitação (A)			
Interface de diálogo	6	6	8	20	8	160
Leitura de sistemas externos (MES)	4	0	4	8	4	32
Leitura de sistemas externos (CRM)	4	0	4	8	4	32
Leitura de sistemas externos (PIMS)	4	0	4	8	1	8
Gerenciamento da base de dados	0	0	4	4	6	24
Gerador de relatórios	4	4	4	12	6	72
Agentes computacionais inteligentes	6	6	6	18	2	36
Qualidade e cálculos automáticos (cálculos básicos, sem considerar incerteza, umidade e degradação calculada com coeficientes empíricos)	6	6	6	18	8	144

Subsistema	Ganhos			Ganho U+T+A	Risco	Pontuação G x R
	Utilização (U)	Throughput (T)	Aceitação (A)			
Qualidade e cálculos automáticos refinados (cálculos avançados, considerando incerteza, umidade e degradação calculada com modelos matemáticos)	6	6	6	18	1	18

Tab. A.31: Avaliação de ganhos e riscos

A.6 Requisitos de amplificação de inteligência do MES Pátio Samarco

Sumário	Diagrama de situação do pátio (DSP)
ID	RA1
Descrição	Deve haver uma interface de diálogo que represente todos os conceitos e casos de amplificação de inteligência (seção 7.2.10) necessários para resolver os problemas de estocagem e embarque na Samarco. Essa interface deve aproveitar ao máximo e usar como principal referência a idéia de representação já utilizada na Samarco, o mapa de situação do pátio/pilhas da figura 7.22, uma interface desenhada atualmente à mão pela equipe de operação. Deve-se automatizar o processo de construção e uso dessa representação e melhorá-la para suportar ainda mais a tomada de decisão da equipe. As estruturas devem ser representadas conforme a configuração do cenário de 2008, descrito na seção 7.2.2.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1-35

Sumário	Diagrama de situação do pátio (DSP)
Data de criação	02/08/2007

Sumário	Representação dos pátios de empilhamento no DSP
ID	RA2
Descrição	Deve-se apresentar graficamente os pátios de empilhamento balizados, ou seja, com a marcação das balizas do início ao fim do pátio, como é feito na Samarco e pode ser observado na figura 7.22.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação da posição e estado de máquinas e correias no DSP
ID	RA3
Descrição	As máquinas e correias devem ter suas posições e estados representados iconicamente no DSP. A posição das máquinas pode ser obtida através da marcação das balizas, e seu estado, assim como o das correias, através de um esquema de cores configurável. Os possíveis estados são: operação, ociosa ou em manutenção.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Anotações no DSP
ID	RA4

Sumário	Anotações no DSP
Descrição	Deve ser possível inserir anotações no DSP. As anotações devem ser representadas por um ícone de anotação e devem possuir um texto livre como seu conteúdo.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;2
Data de criação	02/08/2007

Sumário	Representação das informações das pilhas no DSP
ID	RA5
Descrição	Ao selecionar uma pilha no DSP, algumas informações devem ser mostradas como horário de início e fim da formação das pilhas, a quantidade, a qualidade média da pilha referente a cada item de controle com medidas estatísticas, cliente e produto da pilha.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;2
Data de criação	02/08/2007

Sumário	Representação do mapa de situação do pátio e das pilhas no DSP
ID	RA6

Sumário	Representação do mapa de situação do pátio e das pilhas no DSP
Descrição	Deve-se apresentar graficamente cada pilha, incluindo informações como: baliza inicial e final, quantidade empilhada em cada baliza, quantidade total da pilha e tipo de produto (representado por uma cor específica), para os três produtos produzidos pela Samarco (pelota, sinter feed e pellet feed). Os produtos produzidos pela Samarco estão especificados na seção 7.2.2, e especificamente os produtos de pelota na tabela 7.2. O tamanho dessas pilhas deve ser desenhado automaticamente no DSP, referenciando a quantidade de material que está contido nelas, e a cor referencia o tipo de material da pilha. As mesmas cores atualmente utilizadas pelos operadores devem ser utilizadas, pois são mantidas já há décadas, por questões de cultura da Samarco. À medida que uma pilha for surgindo de um empilhamento ou sendo recuperada, o comprimento dela deve mudar, conforme a quantidade de material empilhado ou recuperado.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;2
Data de criação	02/08/2007

Sumário	Fila de navios representada no DSP
ID	RA7
Descrição	A fila de navios deverá ser representada no DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação das informações dos navios da fila no DSP
ID	RA8
Descrição	Para cada um dos navios da fila, deve-se apresentar as suas informações no DSP, como nome ou identificador, data estimada de chegada e suas cargas.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação das cargas do navio atracado no DSP
ID	RA9
Descrição	As cargas dos navios atendidos nos berços existentes devem ser representadas no DSP. Deve ser possível selecionar uma dessas cargas por vez, que fica com uma cor diferente. À medida que esta carga vai sendo carregada no navio, uma barra de status deve ser preenchida sobre o retângulo que representa a carga.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação das informações das cargas dos navios na fila do DSP
ID	RA10
Descrição	Para cada uma das cargas dos navios da fila de navios, deverão ser apresentadas informações sobre quantidades a ser embarcada e orientações de embarque.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação das informações no DSP sobre o material que sai das usinas
ID	RA11
Descrição	Informações do produto que sai da usina e da estimativa de sua qualidade média devem ser apresentadas no DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação no DSP da escolha do material que sai das usinas para ser embarcado.
ID	RA12
Descrição	Deve haver uma representação no DSP que torne possível selecionar o material que sai das usinas para ser embarcado diretamente.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação da qualidade resultante de uma blendagem em relação a uma orientação de embarque de determinada carga
ID	RA13
Descrição	Ao analisar uma determinada carga a ser atendida e diferentes grupos de balizas de uma mesma pilha ou de pilhas diferentes para blendagem, deve ser possível visualizar através de um esquema de degradê de cores pintadas na carga se aquela blendagem atende ou não às orientações de embarque. Além disso, deve ser evidenciado o atendimento ou não de cada item de controle às orientações de embarque, também através de um esquema de degradê de cores.
Tipo	Funcional

Sumário	Representação da qualidade resultante de uma blendagem em relação a uma orientação de embarque de determinada carga
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Informações sobre grupos de balizas selecionados no DSP
ID	RA14
Descrição	Ao selecionar diferentes grupos de balizas de uma mesma pilha ou de pilhas diferentes, deve ser disponibilizado no DSP informações sobre o produto selecionado como quantidade total de produto nas balizas selecionadas e qualidade média de cada item de controle, como no canto esquerdo da figura 7.35. Só deve ser possível selecionar no DSP diferentes grupos de balizas que correspondam a um mesmo produto.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação do material embarcado no DSP
ID	RA15
Descrição	Deve ser possível verificar se o material embarcado atendeu ou não às orientações de embarque da carga do navio sendo carregado, através de um esquema de cores. Informações do produto embarcado, quantidade e qualidade média de cada item de controle, ponderada pelo peso de cada qualidade, devem ser apresentadas.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação dos lugares de empilhamento sugeridos pelos agentes computacionais no DSP
ID	RA16
Descrição	Os lugares de empilhamento sugeridos pelos agentes computacionais, posições do pátio onde serão empilhados os materiais que chegam no pátio em determinado instante, devem ser representados no DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação de lugares de empilhamento mandatórios, sugeridos pelo usuário, no DSP
ID	RA17
Descrição	Os lugares de empilhamento mandatórios, determinadas pelo usuário, deverão ser representados no DSP de maneira diferenciada dos lugares de empilhamento sugeridos pelos agentes computacionais.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação dos lugares de recuperação sugeridos pelos agentes computacionais no DSP
ID	RA18
Descrição	Os lugares de recuperação sugeridos pelos agentes computacionais, grupos de balizas nas pilhas do pátio de onde se deve recuperar o produto para atender a carga do navio, devem ser representados no DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1

Sumário	Representação dos lugares de recuperação sugeridos pelos agentes computacionais no DSP
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação de lugares de recuperação mandatórios, sugeridos pelo usuário, no DSP
ID	RA19
Descrição	Os lugares de recuperação mandatórios, grupos de balizas determinadas pelo usuário para serem recuperadas e atenderem a uma determinada carga de um determinado navio, devem ser representadas no DSP de maneira diferenciada dos lugares de recuperação sugeridos pelos agentes computacionais.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação das campanhas no DSP
ID	RA20
Descrição	Deve-se apresentar graficamente as campanhas previstas para o período desejado de programação, assim como o estado da campanha sendo realizada neste instante. Essa representação não acontece diretamente no DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Representação das quantidades e qualidades bi-horárias no DSP
ID	RA21

Sumário	Representação das quantidades e qualidades bi-horárias no DSP
Descrição	Deve-se apresentar graficamente as quantidades e qualidades bi-horárias para um determinado grupo de balizas em um período desejado. Essa representação não acontece diretamente no DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Edição manual para quantidades e qualidades bi-horárias no DSP
ID	RA22
Descrição	Deve ser possível inserir, editar e remover manualmente as informações de quantidades e qualidades bi-horárias.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1,53
Data de criação	02/08/2007

Sumário	Atualização do pátio através de edição manual no DSP
ID	RA23
Descrição	Deve ser possível inserir, editar e remover manualmente as informações sobre a situação de cada elemento do pátio, atualizando o pátio em um determinado instante através de edição manual.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Entrada manual para detalhes de empilhamento no DSP
ID	RA24
Descrição	Deve ser possível inserir manualmente os valores referentes a qual produto será empilhado, seus dados de quantidade e qualidade e aonde será empilhado.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;41;42;43
Data de criação	02/08/2007

Sumário	Entrada manual para detalhes de recuperação no DSP
ID	RA25
Descrição	Deve ser possível inserir manualmente os valores referentes a qual produto será recuperado, seus dados de quantidade e qualidade e de onde será recuperado.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;44;45;46
Data de criação	02/08/2007

Sumário	Edição manual para cliente atribuído a uma pilha no DSP
ID	RA26
Descrição	Deve ser possível editar manualmente o cliente atribuído a uma determinada pilha no pátio através do DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Edição manual para posição e estado das máquinas
ID	RA27

Sumário	Edição manual para posição e estado das máquinas
Descrição	Deve ser possível alterar manualmente posição e o estado das máquinas no pátio.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;55
Data de criação	02/08/2007

Sumário	Edição manual para fila de navios no DSP
ID	RA28
Descrição	Deve ser possível editar manualmente a fila de navios no pátio através do DSP, podendo-se criar, editar ou apagar navios na fila.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;53
Data de criação	02/08/2007

Sumário	Edição manual para cargas dos navios no DSP
ID	RA29
Descrição	Deve ser possível editar manualmente as cargas dos navios no pátio através do DSP, podendo-se criar, editar ou apagá-las.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;53
Data de criação	02/08/2007

Sumário	Edição manual para orientações de embarque no DSP
ID	RA30
Descrição	Deve ser possível editar manualmente as orientações de embarque das cargas dos navios no pátio através do DSP, especificando as medidas de qualidade do produto exigidas para cada carga.
Tipo	Funcional

Sumário	Edição manual para orientações de embarque no DSP
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;53
Data de criação	02/08/2007

Sumário	Entrada manual para atas de campanhas no DSP
ID	RA31
Descrição	Deve ser possível criar e alterar manualmente as atas de campanhas, especificando os produtos e períodos de produção que devem ser seguidos pelas usinas.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Edição manual para produção atual
ID	RA32
Descrição	Deve ser possível alterar manualmente informações sobre a produção atual.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1;54
Data de criação	02/08/2007

Sumário	Edição manual para informações do produto sendo embarcado
ID	RA33
Descrição	Deve ser possível inserir manualmente informações sobre o material sendo embarcado em determinado instante.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Evolução do estado do pátio no tempo
ID	RA34
Descrição	Deve haver uma representação temporal, uma escala de tempo, para representar graficamente a evolução do estado do pátio no tempo conforme as operações são realizadas no horizonte de planejamento. Por exemplo, as pilhas vão sendo consumidas conforme recuperações são realizadas e as máquinas se deslocam entre as operações no tempo. Deve ser possível percorrer o tempo observando o estado do que foi realizado e do planejado no horizonte de tempo escolhido.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Parametrização dos elementos do DSP
ID	RA35
Descrição	Todos os elementos que compõem a planta devem ser parametrizados para suportar uma evolução da mesma planta, e o DSP deve suportar esta parametrização.
Tipo	Não funcional
Subsistema	Interface de diálogo
Prioridade	1
Dependências	RAs: 1
Data de criação	02/08/2007

Sumário	Planejamento automático de operações
ID	RA36
Descrição	Deve ser possível planejar automaticamente as operações de empilhamento e recuperação, criando e sequenciando as operações de empilhamento e recuperação dentro de um horizonte de tempo escolhido determinado, com a utilização das máquinas, atividades e tempos discriminados, de forma a maximizar os critérios de otimização definidos para o gerenciamento do pátio.

Sumário	Planejamento automático de operações
Tipo	Funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 37
Data de criação	02/08/2007

Sumário	Crêterios de avaliaço do planejamento
ID	RA37
Descriço	Os crêterios de avaliaço das soluçoes obtidas no planejamento automtico devem ser: 1.Maximizar o atendimento da quantidade e das orientaçoes de embarque de cada carga de cliente 2.Minimizar o atraso no atendimento dos navios 3.Minimizar o deslocamento das mquinas durante o empilhamento e a recuperaço 4.Minimizar o nmero de pilhas diferentes para atender um embarque 5.Minimizar o nmero de pilhas no ptio 6.Minimizar a necessidade de recheio na recuperaço, utilizando pilhas baixas
Tipo	No funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36
Data de criaço	02/08/2007

Sumrio	Regras de empilhamento
ID	RA38

Sumário	Regras de empilhamento
Descrição	<p>Algumas regras de empilhamento são utilizadas pela operação para gerenciar a qualidade de produto e o espaço no pátio, e devem ser consideradas na formulação dos planos:</p> <p>1 - procura-se concentrar os produtos iguais nas mesmas regiões do pátio. Assim, por exemplo, toda pelota do tipo STD é empilhada em uma região concentrada do pátio. Isso evita o deslocamento excessivo das máquinas durante o empilhamento e recuperação que ocorreria se as pilhas de STD estivessem espalhadas em várias regiões do pátio;</p> <p>2 - de maneira geral procura-se empilhar o produto em pilhas já abertas. Só são criadas novas pilhas em caso de mudança de campanha, mudança de cliente ou falta de espaço para continuar empilhando em uma pilha já aberta;</p> <p>3 - se sobrar produto em uma pilha reservada para um cliente, este produto pode ser utilizado como base da criação de uma nova pilha para um novo cliente, desde que as qualidades sejam compatíveis;</p> <p>4 - em situações em que não é possível empilhar a pelota no pátio (falta de espaço, quebra de máquina, etc) é utilizada a área de emergência composta pelas pilhas P5 e TP17. O produto armazenado na pilha P5 deve ser transferido para a TP17 antes de voltar ao pátio. A pilha TP17 é recuperada utilizando-se pás-carregadoras e enviada para pilhas no pátio ou diretamente para o embarque de um navio;</p> <p>5 - com relação à altura das pilhas criadas, se houver espaço no pátio procura-se criar pilhas baixas (ocupando um maior número de balizas) evitando, desta forma, a necessidade de recheio durante a recuperação.</p>
Tipo	Não funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36
Data de criação	02/08/2007

Sumário	Regras de recuperação
ID	RA39
Descrição	<p>Nos casos em que a qualidade do produto disponível para o cliente não satisfaz as orientações de embarque da carga, é necessário realizar a blendagem destas pilhas com materiais de melhor qualidade para que a qualidade média final seja adequada. Nesta situação, as seguintes soluções são consideradas, nesta ordem de preferência:</p> <ol style="list-style-type: none"> 1 - o produto das pilhas reservadas é blendado com o produto de melhor qualidade sendo produzido na usina neste momento, desde que a campanha esteja sendo realizada para o mesmo cliente do embarque atual; 2 - o produto das pilhas reservadas é blendado com o produto de outras pilhas com melhor qualidade; 3 - o produto das pilhas reservadas é blendado com o produto de melhor qualidade sendo produzido na usina neste momento, mesmo que a campanha esteja sendo realizada para um cliente diferente daquele do embarque atual; 4 - o produto das pilhas reservadas é descartado e o navio é carregado apenas com o produto de melhor qualidade sendo produzido na usina neste momento; 5 - o produto das pilhas reservadas é descartado e o navio é carregado apenas com o produto de melhor qualidade de pilhas sendo formadas para outros clientes. <p>Em alguns casos pode acontecer também que a qualidade da pilha reservada para o cliente esteja com uma qualidade muito acima daquela especificada. Nestes casos, o produto reservado pode ser blendado com o produto de qualidade mais baixa proveniente da usina ou de outras pilhas. Isso garante que a especificação de qualidade do cliente seja atendida e resulta em um estoque de produto de excelente qualidade que pode ser utilizado para blendagens no futuro.</p>
Tipo	Não funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36

Sumário	Regras de recuperação
Data de criação	02/08/2007

Sumário	Consideração de estimativa de estados futuros nas avaliações
ID	RA40
Descrição	Os agentes devem considerar nas suas avaliações não somente a avaliação do estado atual, mas também uma estimativa da influência da decisão atual nos estados futuros do pátio.
Tipo	Não funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36
Data de criação	02/08/2007

Sumário	Planejamento automático de lugares de empilhamento
ID	RA41
Descrição	Para um determinado horizonte de tempo determinado pelo usuário, deve-se planejar automaticamente lugares de empilhamento de forma a maximizar os critérios de otimização definidos para o gerenciamento do pátio. Sugerir um lugar de empilhamento implica em programar também as atividades de máquinas e outros recursos necessários, assim como estimar seus tempos.
Tipo	Funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36;37;38
Data de criação	02/08/2007

Sumário	Determinação mandatória de lugares de empilhamento
ID	RA42

Sumário	Determinação mandatória de lugares de empilhamento
Descrição	O usuário deve ser capaz de criar ou editar lugares de empilhamento ao longo do horizonte de tempo de programação, marcando esses lugares como mandatórios, ou seja, fixos, imutáveis e que devem fazer parte do planejamento final. Sugerir um lugar de empilhamento implica em programar também as atividades de máquinas e outros recursos necessários, assim como estimar seus tempos.
Tipo	Funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36;37;38
Data de criação	02/08/2007

Sumário	Planejamento automático de lugares de empilhamento considerando lugares mandatórios
ID	RA43
Descrição	Deve ser possível planejar automaticamente lugares de empilhamento em um determinado horizonte de tempo de programação considerando lugares de empilhamento mandatórios criados pelo usuário, que deverão estar incluídos no planejamento final, gerando lugares de empilhamento híbridos, ou seja, em parte construídos pelo agente humano, em parte construídos pelo agente computacional. Sugerir um lugar de empilhamento implica em programar também as atividades de máquinas e outros recursos necessários, assim como estimar seus tempos.
Tipo	Funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36;37;38
Data de criação	02/08/2007

Sumário	Planejamento automático de lugares de recuperação
ID	RA44

Sumário	Planejamento automático de lugares de recuperação
Descrição	Para um determinado horizonte de tempo determinado pelo usuário, deve ser possível planejar automaticamente lugares de recuperação de forma a maximizar os critérios de otimização definidos para o gerenciamento do pátio. Sugerir um lugar de recuperação implica em programar também as atividades de máquinas e outros recursos necessários, assim como estimar seus tempos.
Tipo	Funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36;37;39
Data de criação	02/08/2007

Sumário	Determinação mandatória de lugares de recuperação
ID	RA45
Descrição	O usuário deve ser capaz de criar ou editar lugares de recuperação ao longo do horizonte de tempo de programação, marcando esses lugares como mandatórios, ou seja, fixos, imutáveis e que devem fazer parte do planejamento final. Sugerir um lugar de recuperação implica em programar também as atividades de máquinas e outros recursos necessários, assim como estimar seus tempos.
Tipo	Funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36;37;39
Data de criação	02/08/2007

Sumário	Planejamento automático de lugares de recuperação considerando lugares mandatórios
ID	RA46

Sumário	Planejamento automático de lugares de recuperação considerando lugares mandatórios
Descrição	Deve ser possível planejar automaticamente lugares de recuperação em um determinado horizonte de tempo de programação considerando lugares de recuperação mandatórios criados pelo usuário, que deverão estar incluídos no planejamento final, gerando lugares de recuperação híbridos, ou seja, em parte construídos pelo agente humano, em parte construídos pelo agente computacional. Sugerir um lugar de recuperação implica em programar também as atividades de máquinas e outros recursos necessários, assim como estimar seus tempos.
Tipo	Funcional
Subsistema	Agentes computacionais inteligentes
Prioridade	4
Dependências	RAs: 36;37;39
Data de criação	02/08/2007

Sumário	Consolidação dos planos de empilhamento e recuperação
ID	RA47
Descrição	Deve ser possível consolidar os planos de empilhamento e recuperação construídos através de uma atividade de oficialização. Essa consolidação salva o plano construído.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	2
Dependências	RAs: 36;41-46
Data de criação	02/08/2007

Sumário	Criação do estado inicial
ID	RA48

Sumário	Criação do estado inicial
Descrição	Deve ser possível gerar o estado inicial do sistema através da leitura de informações de sistemas externos, da base dedicada do sistema e de entradas manuais do usuário através da interface de diálogo. Uma vez construído, o estado inicial deve ser apresentado no DSP.
Tipo	Funcional
Subsistema	Interface de diálogo
Prioridade	2
Dependências	–
Data de criação	02/08/2007

Sumário	Perfis de usuários
ID	RA49
Descrição	Deve haver restrições de acesso de determinados usuários a determinadas funções baseado nos seus perfis.
Tipo	Funcional
Subsistema	Usuário
Prioridade	2
Dependências	–
Data de criação	02/08/2007

Sumário	Cálculos básicos da qualidade por baliza
ID	RA50
Descrição	Deve-se calcular automaticamente a estimativa da qualidade por baliza de cada pilha utilizando como dados de entrada as qualidades e quantidades medidas a cada duas horas nas saídas das usinas juntamente com os dados de posicionamento e operação das máquinas. Esse cálculo deve ser realizado de maneira básica, sem considerar incerteza, umidade, e fórmulas avançadas de degradação, somente com médias ponderadas e coeficientes de ajuste simplificados de degradação.
Tipo	Funcional
Subsistema	Qualidade e cálculos automáticos (cálculos básicos)
Prioridade	2
Dependências	–

Sumário	Cálculos básicos da qualidade por baliza
Data de criação	02/08/2007

Sumário	Cálculos avançados da qualidade por baliza
ID	RA51
Descrição	Deve-se calcular automaticamente a estimativa da qualidade por baliza de cada pilha utilizando como dados de entrada as qualidades e quantidades medidas a cada duas horas nas saídas das usinas juntamente com os dados de posicionamento e operação das máquinas. Esse cálculo deve ser realizado de maneira avançada, modelando e levando em consideração a incerteza, degradação e umidade.
Tipo	Funcional
Subsistema	Qualidade e cálculos automáticos (cálculos avançados)
Prioridade	5
Dependências	–
Data de criação	02/08/2007

Sumário	Cálculo de média da qualidade para combinações de balizas
ID	RA52
Descrição	Quando grupos de balizas são selecionados na interface, deve-se calcular a quantidade, qualidade média e desvio padrão da média do produtos desses agrupamentos.
Tipo	Funcional
Subsistema	Qualidade e cálculos automáticos
Prioridade	2
Dependências	RAs: 50
Data de criação	02/08/2007

Sumário	Interface com o sistema externo CRM
ID	RA53
Descrição	Deve-se ler as informações de navios, programações de embarque e embarques realizados do sistema externo CRM.

Sumário	Interface com o sistema externo CRM
Tipo	Funcional
Subsistema	Sistemas externos (CRM)
Prioridade	3
Dependências	–
Data de criação	02/08/2007

Sumário	Interface com o sistema MES
ID	RA54
Descrição	Deve-se ler as informações dos dados de produção e qualidade bi-horárias nas saídas das usinas do sistema externo MES.
Tipo	Funcional
Subsistema	Sistemas externos (MES)
Prioridade	3
Dependências	–
Data de criação	02/08/2007

Sumário	Interface com o sistema externo PIMS
ID	RA55
Descrição	Deve-se ler as informações de estado e posicionamento das máquinas do pátio do sistema externo PIMS.
Tipo	Funcional
Subsistema	Sistemas externos (PIMS)
Prioridade	4
Dependências	–
Data de criação	02/08/2007

Sumário	Armazenamento de usuários e perfis
ID	RA56
Descrição	Deve-se armazenar os usuários cadastrados, perfil de cada usuário e as características de cada perfil.
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	RAs: 49
Data de criação	02/08/2007

Sumário	Armazenamento dos parâmetros da planta
ID	RA57
Descrição	Deve-se armazenar os valores de parametrização da planta, para todos os seus elementos.
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	RAs: 35
Data de criação	02/08/2007

Sumário	Armazenamento dos critérios de avaliação
ID	RA58
Descrição	Deve-se armazenar os critérios de avaliação para soluções obtidas pelos agentes computacionais assim como o peso atribuído a cada um desses critérios.
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	–
Data de criação	02/08/2007

Sumário	Armazenamento dos planos de empilhamento e recuperação consolidados
ID	RA59
Descrição	Deve-se armazenar os planos de empilhamento e recuperação consolidados.
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	RAs: 47
Data de criação	02/08/2007

Sumário	Armazenamento de informações sobre situação das pilhas no pátio
ID	RA60
Descrição	Deve-se armazenar informações históricas sobre a situação das pilhas no pátio.

Sumário	Armazenamento de informações sobre situação das pilhas no pátio
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	–
Data de criação	02/08/2007

Sumário	Armazenamento de informações sobre posição e estado das máquinas
ID	RA61
Descrição	Deve-se armazenar os dados históricos de posição e estado das máquinas.
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	RAs: 55
Data de criação	02/08/2007

Sumário	Armazenamento de informações sobre atas de campanha
ID	RA62
Descrição	Deve-se armazenar os dados históricos de atas de campanha.
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	RAs: 31
Data de criação	02/08/2007

Sumário	Armazenamento de relatórios gerados
ID	RA63
Descrição	Deve-se armazenar os relatórios gerados
Tipo	Funcional
Subsistema	Gerenciamento da base de dados
Prioridade	2
Dependências	RAs: 64-69
Data de criação	02/08/2007

Sumário	Relatório de troca de turno
ID	RA64
Descrição	Deve ser possível gerar um relatório de troca de turno. Esse relatório deve ser tanto apresentado na tela como enviado para impressão.
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências fracas	RAs: 69
Data de criação	02/08/2007

Sumário	Relatório de plano de empilhamento e recuperação
ID	RA65
Descrição	Deve ser possível gerar um relatório de plano de empilhamento e recuperação. Esse relatório deve ser tanto apresentado na tela como enviado para impressão.
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências	RAs: 69;59
Data de criação	02/08/2007

Sumário	Relatório de campanhas
ID	RA66
Descrição	Deve ser possível gerar um relatório das campanhas. Esse relatório deve ser tanto apresentado na tela como enviado para impressão.
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências	RAs: 69;62
Data de criação	02/08/2007

Sumário	Relatório de fila de navios
ID	RA67
Descrição	Deve ser possível gerar um relatório da fila de navios. Esse relatório deve ser tanto apresentado na tela como enviado para impressão.

Sumário	Relatório de fila de navios
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências	RAs: 69;53
Data de criação	02/08/2007

Sumário	Relatório de situação do pátio
ID	RA68
Descrição	Deve ser possível gerar um relatório de situação do pátio, uma foto do DSP em um determinado instante.
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências	RAs: 69;60;2
Data de criação	02/08/2007

Sumário	Formato e detalhamento dos relatórios
ID	RA69
Descrição	Deve ser possível configurar um padrão de formato e detalhamento dos relatórios.
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências	–
Data de criação	02/08/2007

Sumário	Apresentar os indicadores de utilização do sistema
ID	RA70
Descrição	Deve ser possível observar indicadores de utilização do sistema. Esses indicadores servirão para identificar se o sistema está sendo corretamente utilizado, além de apontar possibilidades de evolução dos mecanismos de amplificação e diálogo.
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências	–
Data de criação	02/08/2007

Sumário	Apresentar tela de indicadores táticos.
ID	RA71
Descrição	Deve ser possível avaliar o planejamento através de indicadores táticos.
Tipo	Funcional
Subsistema	Gerador de relatórios
Prioridade	2
Dependências	RAs: 36;41-47
Data de criação	02/08/2007

Referências Bibliográficas

- Aitken, Andy; Humiston, Todd; Patel, Jitu (2007). Dynamic planning & execution. In *International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2007. KIMAS 2007.*, pages 401–406. Waltham, MA, USA.
- Alter, S.L. (1980). *Decision Support Systems: Current Practices and Continuing Challenges*. Addison-Wesley.
- Anthony, R.N. (1965). *Planning and Control Systems: A Framework for Analysis*. Harvard University Graduate School of Business Administration.
- Beynon, Meurig; Rasmeyuan, Suwanna; Russ, Steve (2002). A new paradigm for computer-based decision support. *Decision Support Systems*, 33:127–142.
- Bhargava, H.K.; Sridhar, S.; Herrick, C. (1999). Beyond spreadsheets: Tools for building decision support systems. *IEEE Computer*, 32(3):31–39.
- Biocca, Frank (1997). The cyborg's dilemma: Embodiment in virtual environments. In *Proceedings of the 2nd International Conference on Cognitive Technology (CT '97)*.
- Bonczek, R.H. (1980). The evolving roles of models in decision support systems. *Decision Sciences*, 11(2).
- Bratman, Michael E. (1987). *Intention, Plans, and Practical Reason*. Center for the Study of Language and Information.
- Brooks, Rodney A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1&2):3–15.
- Burks, Arthur; Weiss, Paul (1945). Peirce's sixty-six signs. *The Journal of Philosophy*, Vol. 42, No. 14, pp. 383-388.
- Bush, Vannevar (1945). As we may think. *Atlantic Monthly*.
- Chen, Jim Q.; Lee, Sang M. (2003). An exploratory cognitive dss for strategic decision making. *Decision Support Systems*, 36:147–160.
- Courtney, J.F. (2001). Decision making and knowledge management in inquiring organizations: Toward a new decision-making paradigm for dss. *Decision Support Systems*, 31:17–38.

- Damasio, Antônio R. (1994). *Descartes' Error: Emotion, Reason, and the Human Brain*. Grosset/Putnam.
- de Souza, Clarisse Sieckenius (1993). The semiotic engineering of user interface languages. *International Journal of Man-Machine Studies*, 39:753–773.
- de Souza, Clarisse Sieckenius (2005). *The Semiotic Engineering of Human-Computer Interaction*. The MIT Press.
- Dennett, Daniel C. (1996). *Kinds of Minds: Towards an Understanding of Consciousness*. Weidenfeld & Nicolson.
- Dutta, Amitava (1996). Integrating ai and optimization for decision support: A survey. *Decision Support Systems*, 18:217–226.
- Eco, Umberto (1976). *Tratado Geral de Semiótica*. Editora Perspectiva.
- Einstein, Albert (1922). *Sidelights on Relativity*. Courier Dover Publications.
- Engelbart, Douglas C. (1962). Augmenting human intellect: A conceptual framework. Summary report, Stanford Research Institute.
- Franklin, Stan; Graesser, Art (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–36. Budapest, Hungary.
- Gardner, Howard (2003). *A Nova Ciência da Mente: uma História da Revolução Cognitiva*. EDUSP.
- Geoffrion, A.; Maturana, S. (1995). Generating optimization-based decision support systems. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*. Hawaii.
- Giere, Ronald N.; Moffatt, Barton (2003). Distributed cognition: Where the cognitive and the social merge. *Social Studies of Science*, 33(2):301–310.
- Gorry, G.A.; Morton, M.S. Scott (1971). *A Framework for Management Information Systems*. Massachusetts Institute of Technology.
- Gudwin, Ricardo; Gomide, Fernando (1998). Object networks - a modelling tool. In *Proceedings of FUZZ-IEEE98, WCCI'98 - IEEE World Congress on Computational Intelligence*, pages 77–82. Anchorage, Alaska, USA.
- Gudwin, Ricardo R.; Queiroz, João (2007). Towards machine understanding: Some considerations regarding mathematical semiosis. In *Proceedings of the 2007 IEEE KIMAS'07*, pages 247–252. Waltham, MA.

- Gudwin, Ricardo Ribeiro (1999). From semiotics to computational semiotics. In *9th International Congress of the German Society for Semiotic Studies/ 7th International Congress of the International Association for Semiotic Studies (IASS/AIS)*. Dresden, Germany.
- Gudwin, Ricardo Ribeiro (2003). *Semiônica: Uma proposta de contribuição à semiótica computacional*. Tese de livre-docência, DCA-FEEC-UNICAMP.
- Guerrero, J.A.S. (2000). *Rede de agentes: Uma ferramenta para o projeto de sistemas inteligentes*. Tese de mestrado, DCA-FEEC-UNICAMP.
- Guerrero, J.A.S.; Gomes, A.S.R.; Gudwin, R.R. (1999). A computational tool to model intelligent systems. In *Anais do IV Simpósio Brasileiro de Automação Inteligente - SBAI99*, pages 227–232. São Paulo, SP, Brasil.
- Harnad, Stevan (1990). Symbol grounding problem. *Physica D*, 42:335–346.
- Henderson, Kathryn (1998). The role of material objects in the design process: A comparison of two design cultures and how they contend with automation. *Science, Technology & Human Values*, 23(2):139–174.
- Hessen, Johannes (1980). *Teoria do Conhecimento*. Armenio Amado.
- Hoch, Stephen J.; Schkade, David A. (1996). A psychological approach to decision support systems. *Management Science*, 42(1):51–64.
- Holsapple, C.W.; Whinston, A.B. (1996). *Decision Support Systems: A Knowledge-based Approach*. West Publishing Company.
- Hsu, Wen-Ling; et al. (1993). A mixed-initiative scheduling workbench: Integrating ai, or and hci. *Decision Support Systems*, 9(3):245–257.
- Jacobson, Ivar; Booch, Grady; Rumbaugh, James (1999a). *The Unified Modeling Language User Guide*. Addison-Wesley.
- Jacobson, Ivar; Booch, Grady; Rumbaugh, James (1999b). *The Unified Software Development Process*. Addison-Wesley.
- Keen, P.G.W. (1980). Adaptive design for decision support systems. *Data Base*, 12(1 e 2).
- Lang, A. Scheffer; Jr., Paul O. Roberts; Sammon, John P. (2000). Car scheduling: Why bother? *Railway Age*, 12.
- Leite, Jair Cavalcanti; de Souza, Clarisse Sieckenius (1999). Uma linguagem de especificação para a engenharia semiótica de interfaces de usuário. In *IHC'99 Proceedings*. Campinas, SP, Brazil.
- Licklider, J. C. R. (1960). Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1(2):4–11.

- Little, J.D.C. (1970). Models and managers: The concept of a decision calculus. *Management Sciences*, 16(8).
- Makowski, Marek (1994). Design and implementation of model-based decision support system. Working paper, International Institute for Applied Systems Analysis (IIASA).
- Maturana, S.; Ferrer, J.C.; Baraño, F. (2004). Design and implementation of an Optimization-Based Decision Support System. *European Journal of Operational Research*, 154:170–183.
- Merrel, Floyd (1997). *Peirce, Signs and Meaning*. University of Toronto Press.
- Molck, Paulo (2002). Métodos heurísticos em programação de atividades em pátios de estocagem. Tese de mestrado, DCA-FEEC-UNICAMP.
- Moore, J.H.; Chang, M.G. (1980). Design of decision support systems. *Data Base*, 12(1 e 2).
- Neumann, J. Von (1958). *The Computer and the Brain*. Yale University Press.
- Neumann, J. Von (1969). Self-reproducing automata. Technical report, University of Michigan. Dept. of Computer and Communication Sciences.
- Nöth, Winfried (1995). *Handbook of Semiotics*. Indiana University Press.
- Ojeda, D.; Liñero, L.; Gudwin, R. (2000). An autonomous vehicle controller using agent networks. In *IV Industry Applications Conference - INDUSCON'2000*, pages 519–524. Porto Alegre, RG - Brasil.
- Partridge, Derek (1998). *Artificial Intelligence and Software Engineering: Understanding the Promise of the Future*. AMACOM.
- Peirce, Charles Sanders (1960). *Collected Papers of Charles Sanders Peirce*. Harvard University Press - Cambridge, Massachusetts.
- Penrose, Roger (1989). *The Emperor's New Mind*. Oxford University Press.
- Queiroz, João (2004). *Semiótica Segundo C.S. Peirce*. EDUC.
- Ransdell, Joseph (2003). The relevance of peircean semiotic to computational intelligence augmentation. *SEED Journal (Semiotics, Evolution, Energy and Development)*, 3(3).
- Rittel, H.W.J.; Webber, M.M. (1973). Dilemmas in a general theory of planning. *Policy Sciences*, 4:155–169.
- Rosenblueth, A.; Wiener, N; Bigelow, J (1943). Behavior, purpose and teleology. *Philosophy of Science*, 10(1):18–24.
- Russel, Steven; Norvig, Peter (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

- Searle, John (1980). Minds, brains and programs. *The Behavioral and Brain Sciences*, 3:417–457.
- Searle, John (1983). *Intentionality: An Essay in the Philosophy of Mind*. Cambridge University Press.
- Shanahan, Murray (2004). The frame problem. *Stanford Encyclopedia of Philosophy*.
- Shim, J.P.; et al. (2002). Past, present and future of decision support technology. *Decision Support Systems*, 33(2):111–126.
- Simon, H.A. (1960). *The New Science of Management Decision*. Harper Brothers.
- Skagestad, Peter (1993). Thinking with machines: Intelligence augmentation, evolutionary epistemology, and semiotic. *Journal of Social and Evolutionary Systems*, 16(2).
- Skagestad, Peter (1996). The mind's machines: the turing machine, the memex, and the personal computer. *Semiotica*, 111(3/4):217–243.
- Skagestad, Peter (1998). Peirce, virtuality and semiotic. In *Proceedings of the 20th World Congress of Philosophy (WCP)*. Boston, Massachusetts.
- Sommerville, Ian (2003). *Engenharia de Software*. Addison Wesley.
- Sommerville, Ian; Kotonya, Gerald (1998). *Requirements Engineering: Processes and Techniques*. John Wiley & Sons, Inc.
- Taddeo, Mariarosaria; Floridi, Luciano (2005). Solving the symbol grounding problem: a critical review of fifteen years of research. *Journal of Experimental & Theoretical Artificial Intelligence*.
- The Standish Group (1995). Chaos report. Survey on it project failure, The Standish Group.
- Turban, Efraim; Aronson, Jay E. (1997). *Decision Support Systems and Intelligent Systems*. Prentice Hall PTR Upper Saddle River.
- Turing, Alan M. (1936). On computable numbers, with an application to the entscheidungsproblem. Report, London Mathematical Society.
- Whorf, Benjamin Lee (1964). *Language, Thought and Reality: Selected Writings of Benjamin Lee Whorf*. The MIT Press.
- Wierzbicki, A.; Makowski, M.; Wessels, J. (2000). *Model-Based Decision Support Methodology with Environmental Applications*. Springer.
- Zadeh, Lofti A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions. Systems, Man and Cybernetics*, 3:28–44.