

# Sistrac: Sistema de Suporte a Trabalho Cooperativo

Este exemplar corresponde a redação final da tese devidamente corrigida e defendida pelo Sr. Leirton Saraiva de Castro e aprovada pela comissão Julgadora.

Campinas, 23 de dezembro de 1991.



Prof. Dr. Rogério D. B. P. de Mello Filho †  
Orientador

Dissertação apresentada no Instituto de Matemática, Estatística e Ciências da Computação, UNICAMP, como requisito parcial para a obtenção do Título de Mestre em Ciências da Computação.

C279s

15639/BC

UNICAMP  
BIBLIOTECA CENTRAL

*A Deus, a meus pais e a Veruschka*

## Resumo

A variedade de produtos de *software* para grupo têm aumentado nos últimos anos. A evolução da tecnologia de comunicação e a proliferação das redes de computadores proporcionaram o ambiente adequado para o aparecimento destes produtos. A maioria dos sistemas cooperativos descritos na literatura requer *hardware* especial e não suporta versões simplificadas (sem imagem ou som). Este trabalho aborda um sistema de baixo custo que funciona em equipamentos padrão.

O Sistrac é o componente do A\_HAND que visa suprir a necessidade de cooperação em um ambiente de desenvolvimento de *software*. Apresenta um compartilhador de programas monousuário que possibilita a utilização de programas orientados a texto, aos quais os usuários estão habituados no seu trabalho individual, em atividades de grupo. O compartilhador de programas ainda provê um mecanismo de troca de mensagens que permite aos elementos da equipe *conversarem* durante o uso da aplicação. Além disso, foi implementado o gerenciador de sessão. Este programa distribuído responsabiliza-se pela administração das informações e operações associadas a uma sessão de cooperação e define um protocolo que facilita a incorporação gradual de novas aplicações multiusuário.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Introdução . . . . .	1
1.2	O que é <i>software</i> para grupo . . . . .	3
1.3	Motivação . . . . .	3
1.4	Sistema de Suporte a Trabalho Cooperativo: SISTRAC . . . . .	5
<b>2</b>	<b>Produtos de Grupo</b>	<b>7</b>
2.1	Taxonomia de Produtos de Grupo . . . . .	7
2.2	Colab : Suporte Computacional para Colaboração e Solução de Problemas em Encontros . . . . .	10
2.3	Capture Lab . . . . .	17
2.3.1	Experiências com o Capture Lab . . . . .	18
2.3.2	Interface do Capture Lab . . . . .	19
2.4	Sistema Rapport de Conferências Multimídia . . . . .	21
2.4.1	Sala Virtual de Reuniões . . . . .	21
2.4.2	Princípios de Projeto . . . . .	22
2.4.3	Macroestrutura do Rapport . . . . .	23
2.5	LIZA : Um Conjunto Extensível de Aplicativos de Grupo . . . . .	23
2.5.1	Objetos Ativos . . . . .	25
2.5.2	Estrutura de Aplicativos . . . . .	25
2.5.3	Estrutura do Sistema . . . . .	26
2.6	Quilt: Uma Ferramenta para Escrita Cooperativa . . . . .	27
2.6.1	Estrutura dos documentos . . . . .	28
2.6.2	Coordenação de Atividades . . . . .	29
2.7	Sistrac: Sistema de Suporte e Trabalho Cooperativo . . . . .	29
<b>3</b>	<b>Visão Geral do Sistema</b>	<b>30</b>
3.1	Requisitos para um Sistema Cooperativo . . . . .	32

<b>4</b>	<b>Módulos Componentes do Sistrac</b>	<b>36</b>
4.1	Sistema Gerenciador de Sessão . . . . .	36
4.2	Sistema de Compartilhamento de Programas Monousuário . .	40
4.2.1	Fluxo de Dados . . . . .	41
4.3	Sistractool: a interface gráfica do Sistrac . . . . .	42
4.3.1	Painel de Controle . . . . .	42
4.3.2	<i>Sistrac Console</i> . . . . .	49
<b>5</b>	<b>Implementação do Protótipo</b>	<b>50</b>
5.1	Sistema Gerenciador de Sessão . . . . .	50
5.1.1	Estrutura do Sistema Gerenciador de Sessão . . . . .	51
5.2	Sistema de Compartilhamento de Programas Monousuário - SCPM . . . . .	58
5.2.1	Problemas de Implementação . . . . .	64
5.3	Interação entre o SCPM e o SGS . . . . .	65
<b>6</b>	<b>Conclusão</b>	<b>67</b>
6.1	Expectativas e Extensões . . . . .	68
<b>A</b>	<b>Comandos do Sistrac</b>	<b>70</b>
<b>B</b>	<b>Funções Exportáveis</b>	<b>81</b>
<b>C</b>	<b>Mensagens do Sistrac</b>	<b>92</b>

# Lista de Figuras

2.1	Fase de Criação : participantes adicionam idéias ou texto explicativo. Crítica ou eliminação de itens são desencorajadas	12
2.2	Fase de Organização : a ordem das idéias é estabelecida incrementalmente	13
2.3	Fase de Organização : agrupamento de itens	14
2.4	Esquema da sala de reuniões do Capture Lab	17
2.5	Interface com o usuário do Rapport	24
2.6	Arquitetura do sistema LIZA	26
4.1	Sistractool: A Interface Gráfica do Sistrac	43
4.2	Botão <i>Session Reference</i>	44
4.3	Botão <i>Mode</i>	46
4.4	Botão <i>Message</i>	48
5.1	Estrutura do Sistema Gerenciador de Sessão	52
5.2	Sistema Compartilhador de Programas Monousuário	59
5.3	Processos Controladores: o primeiro lê caracteres do canal de comunicação com processos-usuário do executor e os envia ao processo programa compartilhado. Além disso trata as mensagens do servidor gerente da sessão. O segundo processo lê a saída do programa compartilhado e a replica para todos os processos-usuário	61
5.4	Processos-usuário: cada usuário tem dois processos-usuário associados. O primeiro exibe a saída do programa compartilhado e trata as mensagens de controle provenientes dos processos controladores. O segundo, caso o usuário seja o executor, lê o teclado e envia os caracteres lidos para processo controlador.	63
5.5	Processos dos Comandos de Controle	64

# Capítulo 1

## Introdução

### 1.1 Introdução

Um tema importante na sociedade industrial moderna é que atividades de grupo são economicamente necessárias, eficientes como meio de produção e reforçam os valores democráticos. A necessidade da atividade de grupo tem sido assegurada pelo estado industrial moderno, o qual não conseguiria sobreviver pelo esforço individual isolado [Kra 88].

Com o aparecimento e a divulgação dos computadores pessoais, concentraram-se esforços no aumento da produtividade e eficiência individual, deixando-se um pouco de lado o desenvolvimento de produtos computacionais para o suporte de trabalho de grupo [Eng 88]. A proliferação das redes locais de computadores e o desenvolvimento da tecnologia de comunicação ofereceram o ambiente adequado para o desenvolvimento de programas direcionados a atividades cooperativas.

Muitos estudos têm sido feitos sobre as variações do comportamento do homem ao trabalhar isoladamente e em grupo ([Gru 88], [Win 88], [Eng 88]). Segundo Grudin [Gru 88], o desenvolvimento de produtos de grupo nem sempre tem sido bem sucedido, não por serem pouco inteligentes ou pelo fato de terem sido mal implantados (instalação, treinamento, manutenção e manuais de operação), mas sim por tentarem mudar a maneira como as pessoas interagem num ambiente de trabalho. Alguns requisitos básicos devem ser atendidos para que produtos de grupo sejam bem aceitos. As aplicações que suportam trabalho cooperativo devem trazer benefícios a todos os usuários relevantes da equipe, embora possam acarretar trabalho extra para alguns componentes. Elas devem prever mecanismos para recuperação de erros e possibilitar a improvisação característica em atividades coletivas.

Ainda de acordo com o referido autor, a tarefa de análise de requisitos, projeto e avaliação de produtos que visam suportar trabalho de equipe é bem mais complexa do que para aplicações monousuário. O sucesso de um indivíduo utilizando uma planilha eletrônica ou um editor de texto, por exemplo, não é afetado pelo comportamento ou nível intelectual de outros membros ou até mesmo pela dinâmica de inter-relacionamentos dentro de um grupo de trabalho. Finalizando, o autor citado afirma que o projeto e desenvolvimento de produtos voltados para trabalho cooperativo deveriam ser menos orientados pela tecnologia e concentrar-se mais no entendimento do funcionamento de organizações e grupos de trabalho. A avaliação sistemática dos mecanismos que regem a interação entre membros de grupos de trabalho e dos efeitos da introdução de novas tecnologias no comportamento de indivíduos que interagem pode levar ao projeto e produção de *software e hardware*, direcionados à cooperação, mais adequados.

Atividades coletivas são caracterizadas pela comunicação verbal entre os membros de uma equipe de trabalho. Normalmente, essas atividades desenvolvem-se através de encontros entre as pessoas que compõem a equipe. Numa reunião convencional todos devem estar presentes e são obrigados a seguir as normas de comportamento definidas pelo grupo. Há um horário pré-estabelecido para seu início, nem sempre conveniente para todos os elementos da equipe. Um membro é obrigado a acompanhar os pronunciamentos de todos os outros, mesmo que indesejáveis ou insignificantes. Não há como se reproduzir discussões anteriores ou se estabelecer conversações paralelas entre membros de um subgrupo, sem prejuízo do bom andamento do encontro. A informação manipulada fica restrita ao material contido na sala de reuniões.

Ao se pensar em produtos computacionais para trabalho de grupo, o conceito de reunião assume uma outra dimensão. É possível imaginar-se encontros em que haja independência de horário. A presença de todos os participantes para início da reunião é irrelevante. A medida que vão chegando, eles podem se integrar às reuniões de sua escolha. É possível tomar parte de vários encontros ao mesmo tempo. Antes de ingressar em uma reunião, o membro pode examinar o que já foi discutido ou realizado. Um participante tem a possibilidade de *escutar* apenas as pessoas de seu interesse, além de poder permanecer no seu escritório. É possível, inclusive, que os membros da equipe estejam geograficamente distribuídos. As informações podem ser manipuladas por programas. Conversações paralelas entre elementos de subgrupos ocorrer ser mantidas.

Esta noção metafórica de reunião, com todas as suas potencialidades,



tem sido um dos alicerces conceituais para o projeto da maioria dos produtos de *software* e *hardware* para atividades cooperativas.

Esta tese tem como objetivo inicial apresentar um levantamento de alguns produtos de grupo existentes, descrevendo sua funcionalidade e algumas características de implementação. A partir dos produtos de grupo discutidos, sugere um sistema direcionado à atividade cooperativa, distribuído e extensível. Culmina com a descrição de sua implementação.

## 1.2 O que é *software* para grupo

Intuitivamente produto de *software* para grupo refere-se a programas cujo usuário é um grupo, provavelmente membros de uma equipe que interagem utilizando-se das potencialidades do computador.

Segundo Gibbs [Gib 89], produtos de grupo referem-se a sistemas de *software* que suportam dois ou mais usuários trabalhando, simultaneamente ou não, numa tarefa comum e que provê uma interface do ambiente compartilhado. A noção de ambiente compartilhado é fundamental. Segundo o autor, ambiente compartilhado é aquele no qual as ações de um usuário são visíveis aos outros. Esse conceito de ambiente compartilhado inclui sistemas gerenciadores de banco de dados como produtos de grupo, apesar de suportarem múltiplos usuários trabalhando na execução da mesma tarefa.

Neste trabalho, o conceito de ambiente ou espaço compartilhado, não se restringe à interface, mas ao conjunto de objetos (arquivos, mensagens, janelas e processos) manipulados por usuários envolvidos numa tarefa comum. Assim, um editor multiusuário, teria como ambiente compartilhado, a cópia (ou cópias) do arquivo que está sendo alterado, as janelas de entrada e saída que compõem a interface com os usuários e os processos envolvidos na edição cooperativa do mesmo. O termo espaço privado também é utilizado no decorrer deste trabalho e refere-se ao conjunto de objetos envolvidos no trabalho privado de um usuário.

## 1.3 Motivação

O desenvolvimento de *software* de grande porte, envolve o trabalho de um grande número de pessoas que atuam nas diversas fases do ciclo de vida do *software* sintetizados por Pressman [Pre 87] em:

- Análise e Engenharia de Sistemas - estabelecimento de requisitos para todos os elementos do sistema e alocação de um subconjunto para o

*software* a ser desenvolvido.

- Análise de Requisitos - os requisitos para o *software* são estabelecidos nessa fase. O engenheiro de *software* concentra seus esforços no entendimento do domínio da informação, desempenho, funcionalidade e interfaces requeridas.
- Projeto - essa fase compõe-se de múltiplos passos que focalizam três atributos distintos do *software* : estrutura de dados, arquitetura e detalhes procedurais. O projeto de *software* traduz requisitos em representações de *software* que podem ser avaliadas antes da codificação.
- Codificação - nessa fase o projeto é traduzido em código executável.
- Testes - uma vez que o código foi gerado, começa a fase de testes. O teste de *software* é o elemento crítico que assegura a sua qualidade. Representa uma revisão da especificação, projeto e codificação.
- Manutenção - alterações no *software* podem ocorrer por causa de erros, por causa de alterações no ambiente externo (por exemplo, um novo sistema operacional) ou por causa de alterações das necessidades funcionais ou de desempenho dos usuários.

As ferramentas disponíveis para suporte das fases do processo de desenvolvimento de programas, de forma geral, não possibilitam o trabalho cooperativo e não levam em consideração a necessidade de interação entre os membros da equipe de desenvolvimento. Essa interação se dá principalmente na fase de análise de requisitos e projeto onde grande parte do tempo de trabalho é gasta em reuniões para discussão de propostas e alternativas de projeto, cronogramas, alocação de pessoal, arquitetura dos sistemas, viabilidade técnica e outros temas referentes ao *software* a ser desenvolvido. Relatórios, documentos e artigos conjuntos são produzidos.

O SISTRAC (Sistema de Suporte a Trabalho Cooperativo) foi idealizado, projetado e em parte implementado com a finalidade de suportar a interação entre membros de uma equipe em um ambiente de desenvolvimento de *software*. Constitui-se numa das ferramentas do ambiente de desenvolvimento de *software* A\_HAND. O A\_Hand [Dru 87] é um ambiente de desenvolvimento baseado em *Hierarquias de Abstração em Níveis* que está sendo implementado no Departamento de Ciência da Computação da UNICAMP. Uma das características mais importantes do A\_Hand é que programas ou módulos são constituídos por objetos mais simples (outros módulos ou programas).

Assim, os módulos do A\_Hand correspondem a hierarquias de objetos. O ambiente se propõe a suportar o desenvolvimento de sistemas grandes e complexos. Sistemas de grande porte são desenvolvidos por uma equipe dado o volume de trabalho a ser executado. Desta forma, o ambiente deve suportar o desenvolvimento distribuído de sistemas e cooperação entre os membros da equipe de desenvolvimento. Deve apresentar facilidades específicas para integração das partes projetadas e ainda permitir a execução e testes das partes isoladas ou já integradas. Algumas ferramentas do ambiente, como a linguagem de programação Cm (C modular e polimórfica) [Fur 91], e a LegoShell [Dru 88] (interface entre o sistema operacional e o usuário que possibilita a composição de objetos mais elaborados a partir de objetos básicos) estão intrinsecamente ligados ao ambiente A\_Hand. O Sistrac enquadra-se como uma ferramenta *periférica* do ambiente, extremamente útil no desenvolvimento distribuído de *software*. Porém, é um produto cuja aplicabilidade ultrapassa os limites do ambiente, podendo ser estendido para se adequar as necessidades específicas de um grupo de usuários.

## 1.4 Sistema de Suporte a Trabalho Cooperativo: SISTRAC

O Sistrac é um sistema que possibilita a cooperação entre um conjunto de usuários localizados em uma sala de reuniões ou em seus escritórios particulares. Essa cooperação se dá através do compartilhamento de programas monousuário ou a utilização de outros programas multiusuário que venham a ser incorporados ao sistema. Uma das principais características é sua capacidade de extensão através do acoplamento de aplicativos multiusuário como um editor de texto ou planilha eletrônica de grupo. O Sistrac é composto por alguns módulos, entre os quais, o gerenciador de sessão que define uma interface a partir da qual outros programas podem ser facilmente integrados ao sistema. O sistema gerenciador de sessão encapsula uma série de operações que possibilitam a reunião e coordenação de usuários membros de uma equipe.

O *hardware* envolvido no desenvolvimento do Sistrac consistiu de estações de trabalho SUN padrão interconectadas numa rede Ethernet. Cada estação dispõe de um monitor de alta resolução, *mouse*, teclado e um canal de som. O *software* requerido compõe-se do sistema operacional Unix (SunOs) e sistema X Window.

A implementação do Sistrac envolveu conceitos de programação dis-

tribuída (paradigma cliente-servidor) e utilizou tecnologias novas como a execução remota de procedimentos<sup>1</sup> e representação externa de dados<sup>2</sup>. A primeira versão do Sistrac está operacional e disponível no Departamento de Ciência da Computação da UNICAMP.

A seguir apresentaremos uma breve descrição dos capítulos que compõem este trabalho :

O capítulo dois, intitulado Produtos de Grupo, apresenta um estudo sobre os produtos de grupo existentes. Inicialmente os produtos são agrupados em categorias de acordo com a tecnologia utilizada. Em seguida, alguns produtos são descritos, focalizando-se a sua funcionalidade, arquitetura e algumas características de implementação.

O capítulo três, intitulado Visão Geral do Sistema, descreve em linhas gerais o Sistrac. Apresenta os módulos que o compõem e discute requisitos para um sistema cooperativo.

No capítulo quatro, denominado Módulos Componentes do Sistrac, apresentam-se os módulos que são alvo principal dessa tese : Sistema Gerenciador de Sessão e Sistema Compartilhador de Programas Monousuário.

O capítulo cinco, de título Implementação do Protótipo, descreve as técnicas de implementação adotadas, o estado atual e alguns problemas de implementação.

O capítulo seis encerra a tese, apresentando as conclusões obtidas a partir da especificação, projeto, desenvolvimento e testes de um produto com uma filosofia emergente : suporte a trabalho de grupo.

O apêndice A apresenta uma tabela dos comandos do Sistrac.

O apêndice B apresenta as funções exportáveis do Sistrac.

O apêndice C apresenta as mensagens do sistema.

---

<sup>1</sup>do inglês remote procedure call

<sup>2</sup>do inglês external data representation

## Capítulo 2

# Produtos de Grupo

Neste capítulo, é apresentada uma taxonomia de produtos de grupo. Descrevem-se alguns dos produtos mais importantes, realçando-se as características dos mesmos que inspiraram o projeto Sistrac.

### 2.1 Taxonomia de Produtos de Grupo

Suporte computacional para trabalho de grupo, incluindo correio eletrônico e os primeiros sistemas de conferência por computador, foi inicialmente direcionado para interação assíncrona entre usuários. Estes sistemas eram muito úteis quando não havia grande necessidade de interação *on line* entre membros de uma equipe e permitiam que cada usuário trabalhasse em horários de sua escolha. A necessidade de ferramentas que possibilitassem o trabalho cooperativo simultâneo, fez surgir produtos que suportassem interação em tempo real entre membros de um grupo na execução de tarefas diversas.

A população destes produtos de grupo tem crescido bastante. Algumas taxonomias têm sido propostas visando agrupá-los de acordo com sua funcionalidade e área de atuação.

Kraemer & King [Kra 88] classificam aplicativos de computador e tecnologia de comunicações voltados para o trabalho de grupo (concentrando-se em ferramentas de apoio a tomada de decisão) seguindo o critério de recursos tecnológicos envolvidos. Os quatro parâmetros tomados como base para o estabelecimento da taxonomia foram:

- *Hardware*: inclui equipamentos de audiovisual, de telecomunicação e de computação. O equipamento de computação consiste de terminais

de computadores ou uma rede local de microrcomputadores ou estações de trabalho com alta capacidade de processamento e recursos gráficos.

- *Software*: a chave principal que distingue os produtos de grupo. Concentra-se basicamente no suporte de processamento da informação, modelo de decisão ou comunicações. *Software* de processamento de informação pode ser usado no trabalho individual ou de grupo e inclui sistemas gerenciadores de banco de dados, linguagens de programação de alto nível, pacotes de aplicações gráficas, análises estatísticas e planilhas eletrônicas. *Software* de modelos de decisão são aplicativos para suporte a grupos na tomada de decisão através de técnicas estruturadas de decisão. *Softwares* de comunicação incluem ferramentas para transmissão de voz, imagem e dados.
- Aspectos Organizacionais: compõem-se de dados organizacionais e processos de grupo. Alguns produtos de grupo geram seus próprios dados solicitando aos membros as suas idéias, opiniões e julgamentos. Outros produtos envolvem o uso de banco de dados da organização ou externos. Estes bancos de dados incluem planos operacionais e estratégicos, orçamentos e dados de clientes, fornecedores e do mercado. Processos de grupo variam de acordo com as relações entre membros do grupo. Eles determinam o encaminhamento de eventos e discussões, estabelecendo se há ou não um líder, como esse líder é escolhido ou como a agenda do grupo é estabelecida.
- Recursos Humanos: inclui os participantes do grupo e a equipe de suporte que facilita as atividades cooperativas.

Seguindo-se esses parâmetros, os produtos de grupo foram divididos pelo autor citado em seis categorias:

- Sala de projeção eletrônica<sup>1</sup>: é o mais elementar dos produtos de grupo e difere pouco do seu parente não eletrônico, exceto que a tecnologia audiovisual é computadorizada (computação gráfica e audiovisuais controlados por computador - projetores de *slides* e vídeo).
- Facilidades de Teleconferência de Áudio e Vídeo: têm como objetivo facilitar encontros entre grupos de duas ou mais localidades. Todas as salas de conferência são projetadas para facilitar teleconferência de

---

<sup>1</sup>do inglês electronic boardroom.

áudio e vídeo. Computadores quase nunca exercem um papel central em produtos enquadrados nessa categoria. Teleconferências por computador são tratadas no item *rede de trabalho de grupo*. O *hardware* envolvido compreende câmeras de vídeo, painéis de controle, microfones, fac-símiles, projetores e equipamento de áudio. O *software* envolvido no suporte de teleconferência de áudio e vídeo é composto basicamente por *software* de comunicação para transmissão de voz, dados e imagens. Os aspectos organizacionais compreendem planejamento para cada teleconferência, incluindo preparação de audiovisuais e protocolos de teleconferência. As pessoas envolvidas consistem de participantes e equipe de suporte.

- **Centro de Informações:** envolve recursos de processamento de dados de uma organização dedicados ao suporte de usuários de sistemas de informação baseados em computador, em atividades como geração de relatórios, manipulação, consulta e análise de dados. O *software* envolvido são programas para o gerenciamento de dados, geração de relatórios, consulta e recuperação da informação, manipulação de texto, análise estatística, modelagem matemática e simulação.
- **Conferência de decisão:** produtos que se concentram no suporte a tomada de decisão por grupos. O *hardware* envolvido consiste de uma sala de conferência com um projetor de vídeo, computadores e terminais de votação. O *software* são programas que suportam técnicas analíticas de tomada de decisão. Os aspectos organizacionais consistem de um protocolo que determina quem participa, em que base, com quais direitos de voto e quais as consequências dos resultados do processo. A maioria das conferências de decisão optam por protocolos democráticos ao invés de protocolos autoritários ou hierárquicos. A razão dessa escolha é que essas conferências de decisão geralmente envolvem pessoas de mesmo nível (cientistas ou gerentes, por exemplo) ou representantes de grupos que insistem em serem tratados como iguais.
- **Laboratório de Colaboração:** produtos que suportam reuniões face-a-face. Normalmente, envolvem o uso de modelos verbais e técnicas de manipulação de textos e imagens, que são formas muito comuns de dados utilizados em reuniões. Fisicamente é similar a conferência de decisão. Consiste de estações de trabalho que são dispostas na sala de conferência de forma a permitir o contato visual entre os membros

da reunião. Ferramentas de *software* orientados a texto, uma interface homem-máquina comum para apresentação de imagens e informação compartilhada, janelas públicas e privadas nas estações e aplicações que suportam o esboço e desenvolvimento de idéias e avaliação de planos de forma cooperativa compõem o *software* do laboratório de colaboração. Como o nome sugere, o produto da classe *laboratório de colaboração* é direcionado essencialmente para membros de grupos homogêneos. A interface multiusuário é projetada para desencorajar o controle sobre a atividade do grupo por um ou mais participantes, proporcionando direito de acesso igual a imagens de tela e dados.

- Rede de Trabalho de Grupo: originou-se das conferências por computador. Estas, porém, são assíncronas e ocorrem durante um período extenso, geralmente sobre um tema pré-definido. A rede de trabalho de grupo consiste de suporte à interação em tempo real entre membros de uma equipe. Cada participante pode estar sentado diante de sua estação de trabalho no seu escritório, e se comunicar diretamente com outros participantes do encontro por voz ou informação compartilhada presente na tela de sua estação. Cada estação tem espaços públicos e privados, agenda da equipe, aplicações compartilhadas como pacotes gráficos, processadores de texto e planilhas eletrônicas, que possibilitam a todos os participantes criar, editar ou simplesmente trocar gráficos, texto ou números. Um líder cria reuniões, convocando outros membros, conduz as mesmas gerenciando o uso de aplicativos, e as encerra.

Nas seções seguintes são apresentados em detalhes alguns sistemas de trabalho cooperativo existentes enquadrados nas categorias mais relevantes para esta tese : laboratório de colaboração e *rede de trabalho de grupo*.

## 2.2 Colab : Suporte Computacional para Colaboração e Solução de Problemas em Encontros

Encontros são usados para tarefas intelectuais que requerem a colaboração e participação de diversas pessoas. Os recursos utilizados influenciam fortemente o curso de um encontro. Um quadro negro, por exemplo, provê um espaço compartilhado e centralizado para registro das idéias, possibilitando



a manipulação de texto e figuras. O espaço de um quadro negro, porém, é limitado e itens devem ser eliminados para que haja espaço para outros itens. O rearranjo dos itens no quadro também é inconveniente. O desenho dos itens à mão livre pode ser ilegível e o armazenamento da informação para consulta posterior não é seguro. Muitas das funções para as quais o quadro negro não se adequa são implementadas facilmente por computadores. Um sistema gerenciador de janelas e aplicativos gráficos provêem a flexibilidade para o rearranjo de texto e figuras. Sistemas de arquivos possibilitam o armazenamento e a recuperação de informação de forma segura e rápida. Estações de trabalho independentes permitem que participantes de reuniões compartilhem visões, apontem para objetos sob discussão e trabalhem em diferentes aspectos de um problema simultaneamente.

Para explorar essas idéias foi desenvolvido na Xerox PARC uma sala de reunião experimental denominada Colab, que suporta processos de colaboração em reuniões face-a-face (categoria *laboratório de colaboração*). O Colab foi projetado para equipes de duas até seis pessoas usando computadores pessoais conectados através de uma rede local de computadores. O objetivo central do Colab é tornar as reuniões dos próprios cientistas da Xerox PARC mais efetivas, e com a experiência adquirida com o produto, desenvolver aplicativos mais genéricos para cooperação [Ste 87].

No Colab, cada pessoa tem uma estação de trabalho. Os computadores estão ligados numa rede local (Ethernet) que suporta uma base de dados distribuída. Além das estações, a sala de reuniões é equipada com um telão que possibilita a visão compartilhada dos itens em discussão.

Os aplicativos do Colab suportam ação simultânea ou paralela dos usuários. Para possibilitar essas atividades paralelas, as tarefas são subdivididas em operações que podem ser executadas, mais ou menos independentemente, por diferentes membros de um grupo. A habilidade de agir em paralelo em objetos compartilhados leva a conflitos em potencial. Estratégias para a solução de conflitos são necessárias em alguns casos, mas na maioria das vezes os padrões sociais de comportamento ajudam a resolvê-los.

A seguir faz-se uma breve descrição das principais ferramentas do Colab:

- Boardnoter: produto que imita a funcionalidade de um quadro negro e se adequa a reuniões informais. A característica básica deste produto é que ele apresenta uma grande área compartilhada para livre rascunho.
- Cognoter: produto utilizado para preparar esboços de palestras e artigos. É usado para organizar idéias de um grupo de pessoas. O processo suportado pelo Cognoter imita o estilo de escrita cooperativa do grupo

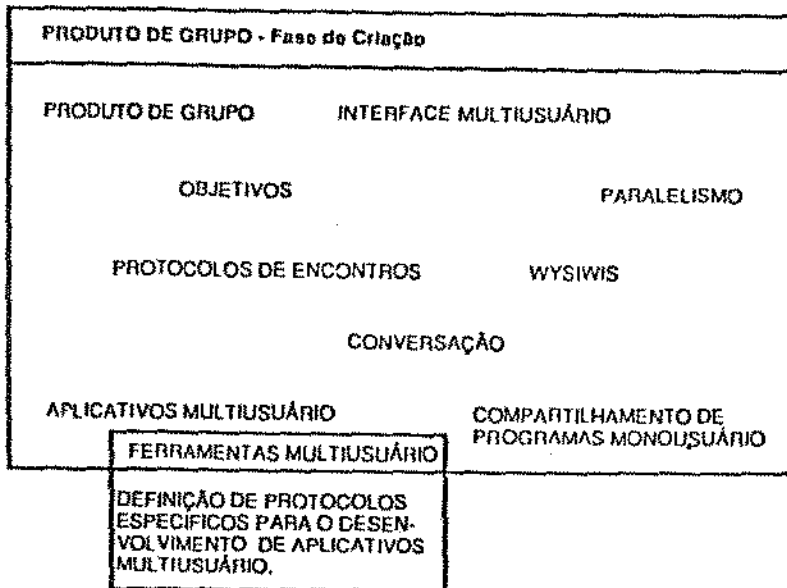


Figura 2.1: Fase de Criação : participantes adicionam idéias ou texto explicativo. Crítica ou eliminação de itens são desencorajadas

de cientistas da Xerox PARC. Cognoter organiza o encontro em três fases distintas: criação, organização e avaliação. Cada uma dessas fases enfatiza um conjunto diferente de atividades. Quando o grupo avança através das respectivas fases, o conjunto de possíveis ações é expandido. Por exemplo, a criação de idéias é explorada na primeira fase, porém ainda é possível na terceira fase. Grupos que não desejam submeter-se às restrições das primeiras fases podem saltar diretamente para a terceira, onde todas as operações são permitidas. A filosofia do Colab é encorajar a diversificação de processos e estilos de comportamento dos grupos de trabalho, através de ferramentas flexíveis, isto é, que não imponham protocolos rígidos.

- Criação(veja figura2.1): nesta fase estimula-se a criação de idéias. As idéias não são avaliadas ou eliminadas nessa fase e pouca

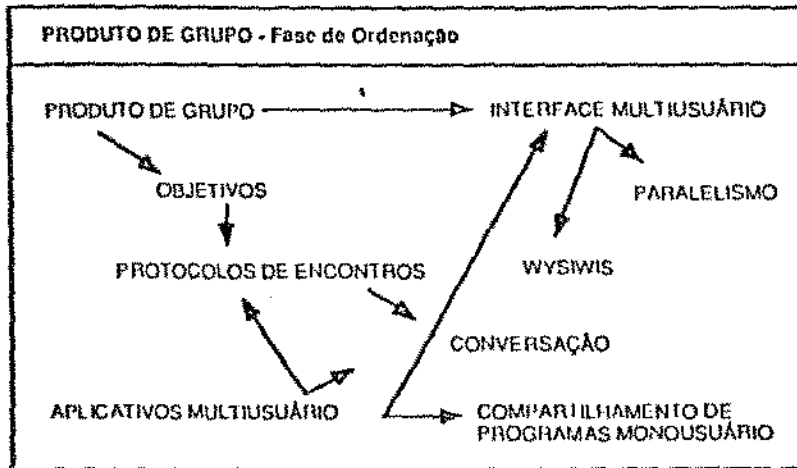


Figura 2.2: Fase de Organização : a ordem das idéias é estabelecida incrementalmente

atenção é dada a sua organização. Idéias são adicionadas pressionando-se o botão do *mouse* na janela compartilhada, digitando-se então um pequeno título ou frase representativa da mesma. Um texto pode ser inserido, usando-se um editor de texto em janela separada, para explicar idéias selecionadas com o *mouse* em maiores detalhes.

- Organização (veja figuras 2.2 e 2.3): nesta fase o grupo tenta estabelecer uma ordem para as idéias geradas na fase de criação. A operação básica é a determinação se uma idéia deve vir antes de outra, o que é indicado visualmente por vetores que ligam itens. Esta operação é acompanhada por discussão verbal. Ligações entre idéias são adicionadas ou removidas pelo acionamento do botão do *mouse* nos itens desejados. Itens geralmente têm uma ou mais ligações com outros itens. Os itens podem ser separados em grupos representativos de idéias que serão trabalhadas juntas. Cada grupo tem uma janela associada. Um grupo é nomeado quando é formado.
- Avaliação: nesta fase a forma final do esquema da palestra ou

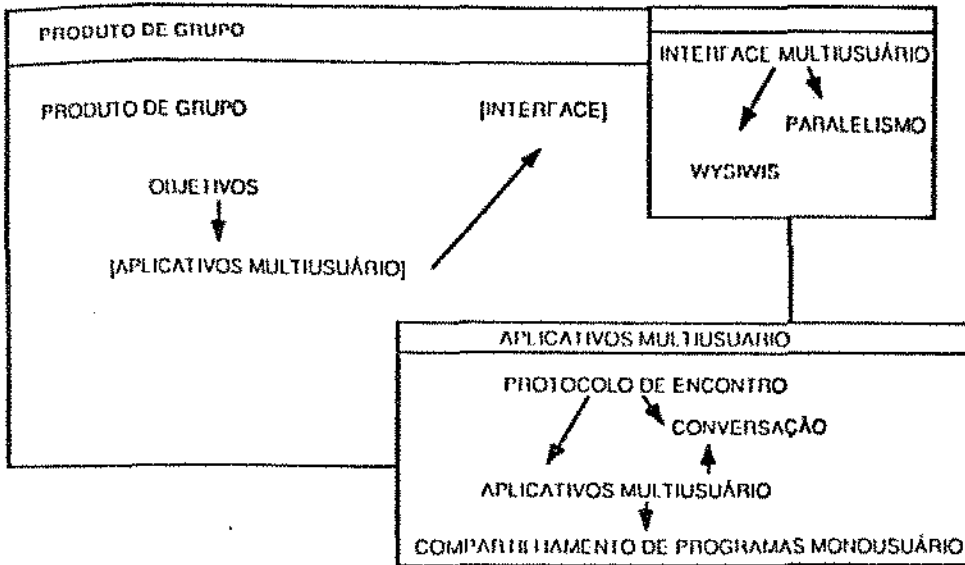


Figura 2.3: Fase de Organização : agrupamento de itens

artigo é determinada. Os participantes revêem a estrutura como um todo para reorganizar as idéias, completá-las com detalhes e eliminar aquelas que forem irrelevantes. O processo de eliminação e reorganização das idéias é deixado para a terceira fase pois nela os membros de uma equipe têm uma base mais sólida para argumentação. Além disso, o atraso da eliminação de idéias é benéfico para evitar a inibição dos participantes no processo de criação.

- **Argnoter:** aplicativo que se propõe a suportar o processo de apresentação e desenvolvimento de propostas. Segue o modelo experimentado sem suporte computacional pela própria equipe que o desenvolveu. No processo de desenvolvimento de um projeto, que é essencialmente uma dialética entre objetivos e possibilidades [Ste 87], os projetistas geralmente iniciam sem conhecer exatamente o que querem ou o que é possível. Eles exploram parte do espaço do projeto direcionados pelos seus objetivos correntes. Na essência, o Argnoter compreende três fases distintas:

– **Propostas:** nesta fase as propostas são estabelecidas explicita-

mente. A cada proposta é associada um pequeno texto descritivo. Propostas são nomeadas de acordo com suas características e funções. Elas são criadas e exibidas através de um conjunto conectado de janelas, que podem ser públicas ou privadas. As janelas de propostas públicas são compartilhadas<sup>2</sup>, enquanto que as janelas de propostas privadas aparecem apenas na estação do usuário que a controla. Janelas privadas asseguram que todo participante possa ver ou criar uma nova proposta sem ter que compartilhá-la. Novas propostas podem ser derivadas da modificação de uma ou mais já existentes.

- Argumentação: nesta fase são apresentadas as razões para a escolha ou não de propostas individuais. A estrutura do Argnoter estimula os participantes a apontarem virtudes e defeitos relativos a todas as propostas. Cada comentário é identificado como virtude ou defeito. Consiste de uma curta descrição do tipo: "muito caro". Com propósitos comparativos, é possível, nessa fase, agrupar os comentários em categorias como: custo, compatibilidade, tempo de desenvolvimento, eficiência, simplicidade e utilidade.
- Avaliação: nesta fase faz-se considerações sobre argumentos individuais. Participantes geralmente discordam sobre a validade de propostas e argumentos. Essas diferenças são modeladas pelo Argnoter através de conjuntos de *crenças*. Assim, uma pessoa pode acreditar que "seis milhões de *chips* de memória estarão disponíveis em seis meses" e outra não. Formam-se conjuntos de crenças e descrenças sobre argumentos. O Argnoter contabiliza as crenças e descrenças sem precisar entender o significado das propostas de projeto. Ele necessita apenas, diferenciar entre propostas, argumentos e crenças e computar as relações entre eles. Também, nesta fase, critérios de avaliação são estabelecidos. Critério de avaliação e crenças representam diferentes dimensões no processo de avaliação. Dois participantes podem concordar que custo é um critério primário, porém, discordar em relação ao custo de uma proposta específica. Usando-se o Argnoter, pode-se experimentar diferentes maneiras de ordenar critérios e prover mecanismos para organizar as propostas de acordo com esses critérios.

---

<sup>2</sup>WYSIWIS: *What You See Is What I See*

No Colab, cada participante tem um computador pessoal, todos ligados em rede e todos os computadores executam o mesmo *software*. Os aplicativos do Colab habilitam os usuários a compartilhar e revisar informações apresentadas nos encontros. Essas informações residem numa base de dados. Essa base de dados deve atender os seguintes requisitos:

- O tempo de recuperação e alteração da informação deve ser curto.
- A base de dados deve convergir rapidamente para um estado consistente.
- A base de dados deve ser tolerante a falhas e não pode ser vulnerável a ação acidental de um dos participantes.

Alguns modelos foram experimentados para o banco de dados:

- modelo centralizado: assegura que todos os participantes usem os mesmos dados. Há apenas uma cópia do banco de dados. Este modelo foi rejeitado devido a pouca eficiência na recuperação da informação.
- modelo centralizado com bloqueio<sup>3</sup>: o modelo distribuído corrige o problema de eficiência do modelo centralizado. Nesse modelo, cada computador tem uma cópia da base de dados, mas não pode alterar esse item até receber a propriedade sobre o mesmo. Obter propriedade significa obter o bloqueio do item através de um servidor centralizado. Essa implementação apresentou um desempenho pouco satisfatório no bloqueio de itens da base de dados.
- modelo cooperativo: nesse modelo cada máquina tem uma cópia do banco de dados e as alterações são feitas através de *broadcast* sem nenhum mecanismo de sincronização. Nessa abordagem, se dois participantes fazem alterações no mesmo dado simultaneamente, a ordem de execução dessas alterações pode ser diferente e portanto os resultados podem ser distintos em cada máquina. Dois fatores amenizam este problema: na maioria das vezes os resultados são independentes da ordem das alterações na base de dados, além do que os membros de uma reunião usam a voz para bloquear um item que vai ser alterado; o segundo fator é que quando um item está sendo alterado ele fica realçado (em negrito), havendo, porém, uma demora entre o momento em que alguém começa a trabalhar sobre um item e a propagação do

---

<sup>3</sup>do inglês lock

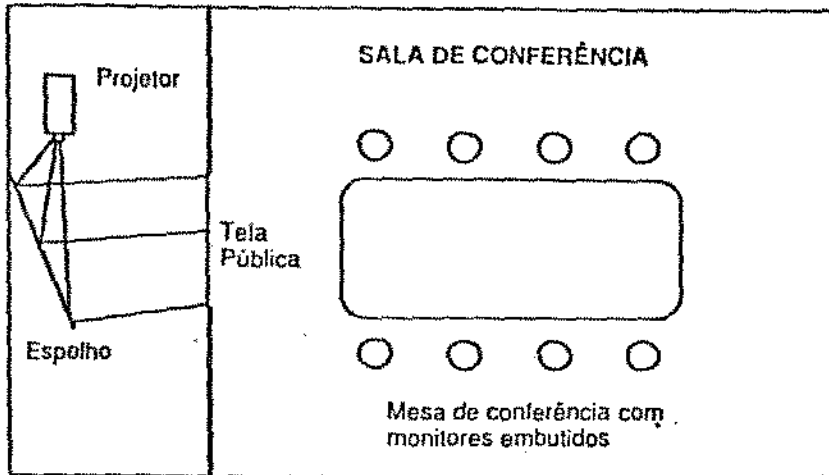


Figura 2.4: Esquema da sala de reuniões do Capture Lab

sinal de ocupado para os outros. Além dos modelos citados, outros estão em estudo.

Concentrando-se no desenvolvimento e no entendimento de sistemas cooperativos para encontros de equipes de trabalho, o projeto Colab provê um conjunto de aplicativos operacionais que aumentam a produtividade coletiva. Além disso, o Colab constitui-se num experimento que ajuda na observação e entendimento do processo de interação entre membros de um grupo e no abandono de práticas antigas para a utilização de tecnologias modernas.

## 2.3 Capture Lab

Outro sistema cooperativo importante da classe *laboratório de colaboração* é o Capture Lab [Elw 89]. A figura 2.4 ilustra o *layout* da sala de reuniões característica do Capture Lab.

A sala de reuniões inclui nove computadores *Macintosh*, uma grande tela de projeção, oito monitores, teclados e *mouses*. O telão serve como monitor do computador público. Os outros monitores e dispositivos de entrada são

de uso individual.

Cada usuário, sentado em torno de uma mesa, tem um teclado, um mouse e um monitor. Os monitores são embutidos na mesa para minimizar a interferência da visão que cada participante tem dos outros. Não há teclado nem mouse para o computador público. Inicialmente os dispositivos de entrada são conectados a máquina individual de cada participante. O usuário pode transferir o controle de seus dispositivos de entrada para o computador público pressionando uma chave especial de acesso. Quando um usuário deseja reconectar os dispositivos de entrada a seu computador pessoal basta pressionar a chave de acesso novamente. Apenas um usuário pode estar conectado ao computador público num certo instante. Usa-se um protocolo preemptivo. Se alguém estiver com o controle do computador público e outro usuário pressionar a chave de acesso, o primeiro usuário é reconectado ao seu computador pessoal e o segundo usuário ganha o controle do computador público.

Em adição à transferência do controle dos dispositivos de entrada, informações podem ser transferidas do computador pessoal para o computador público. No Capture Lab, isto ocorre com a transferência do controle dos dispositivos de entrada. Assim, usuários podem preparar informação no seu computador pessoal antes de apresentá-la aos demais membros da reunião. Alternativamente, um usuário pode transferir dados do computador público para o privado para trabalhá-los individualmente.

Os usuários podem executar qualquer aplicação *Macintosh* na máquina pública ou privada. Nenhuma modificação no código das aplicações é requerida.

### 2.3.1 Experiências com o Capture Lab

A sala de reuniões do Capture Lab tem sido usada por executivos e gerentes, professores e estudantes universitários. Por causa dessa diversidade de usuários não é possível definir um encontro típico suportado pelo Capture Lab. O trabalho mais comum realizado utilizando-se o Capture Lab é a revisão e escrita de documentos como artigos ou propostas em grupo. O computador público geralmente é usado para mostrar o documento no qual o grupo está trabalhando.

Grupos tendem a usar um dos três métodos para acessar a máquina pública. Em encontros interativos, participantes acessam livremente o computador público. Em encontros em que há um *escriba* pré-definido, o indivíduo escolhido como escriba é responsável pelo uso da máquina pública.



Em encontros em que o escriba é rotativo, os diversos usuários do sistema assumem alternadamente o controle da máquina pública. Foi constatado que os grupos raramente empregam apenas um desses três métodos durante o encontro.

A presença de computadores na sala de reuniões não elimina a comunicação visual e verbal característica de reuniões convencionais. As pessoas ainda gastam maior parte do tempo em discussões com pausas para gravação de informações na máquina pública e/ou privada. Alguns encontros ainda incluem períodos em que os participantes trabalham em paralelo em suas máquinas privadas para depois juntarem seus trabalhos na máquina pública.

### 2.3.2 Interface do Capture Lab

O projeto da interface do Capture Lab baseou-se em três princípios:

- Simplicidade do modelo conceitual - a distinção física entre recursos privados e públicos, um protocolo simples de acesso e transferência de informação caracterizam o modelo conceitual do Capture Lab. A sua organização física provê uma separação clara entre os recursos públicos e privados. Para cada assento em torno da mesa de conferência há um monitor e um teclado, sugerindo que eles são de uso privado. Por outro lado, o telão no fim da sala objetiva claramente mostrar informação pública. Embora outros produtos da classe *laboratório de colaboração* apresentem *layout* semelhante, a distinção entre informações públicas e privadas é obscurecida por aplicações que mostram informação compartilhada em monitores privados [Ste 87]. Em contraste com a abordagem de objetos compartilhados para a apresentação de informação pública, o Capture Lab usa a abordagem do *hardware* compartilhado.
- Funcionalidade da interface - os dois principais fatores que afetam a funcionalidade da interface do Capture Lab são a ergonomia da sala de reuniões e independência de *softwares* próprios. O ambiente de reuniões provido pelo Capture Lab é reconhecido como uma sala de reuniões e não uma sala de computadores. Isto evita a inibição dos conferencistas. Os computadores pessoais são embutidos na mesa permitindo o contato visual entre os conferencistas. Os monitores são colocados em posição angular de forma a concentrar a atenção dos conferencistas na tela pública.

A decisão de utilizar-se *softwares* não específicos para reuniões, como é o caso do Cognoter e Arguoter [Ste 87] deveu-se a dois fatores: a diversidade dos usuários em potencial do Capture Lab e portanto custo elevado de desenvolvimento de aplicativos de grupo úteis; a facilidade da transferência da atividade privada para o ambiente de encontros, já que os aplicativos projetados para suportar atividades particulares devem estar disponíveis em atividades coletivas [Eiw 89].

- Suporte para protocolos variados de encontros - No princípio do desenvolvimento do Capture Lab, experimentou-se vários protocolos para gerenciamento do acesso as máquinas pública e privadas, mas rapidamente chegou-se a um protocolo preemptivo simples por dois motivos: há uma forte tendência dos grupos estabelecerem seus próprios protocolos e o crescimento da complexidade aumenta a dificuldade da aprendizagem da utilização da ferramenta. Usando um protocolo preemptivo, o Capture Lab não interfere com os protocolos existentes, permitindo aos grupos estabelecerem seus próprios protocolos de encontro.

Algumas observações foram feitas a partir da experiência de diversos grupos que utilizaram o Capture Lab.

- Escrita de Documentos - o Capture Lab mostrou-se um produto bastante adequado para a escrita cooperativa de documentos. A possibilidade de todos os membros do grupo verem e editarem um documento permite que rapidamente chegue-se a um consenso sobre adições e alterações. Além disso, o uso dos computadores pessoais permite que cada usuário trabalhe independentemente por breves períodos, para depois copiarem o seu trabalho para máquina pública submetendo-o aos demais membros do encontro.
- Treinamento dos Usuários - Novos usuários do Capture Lab acostumados ao equipamento *Macintosh* necessitam de muito pouco treinamento.
- Qualidade das Reuniões - uma ambiente de encontros com suporte computacional não assegura reuniões de boa qualidade. Grupos mal estruturados, com objetivos mal definidos, com liderança fraca, continuarão tendo reuniões pobres independentemente dos recursos utilizados no ambiente de encontros.

## 2.4 Sistema Rapport de Conferências Multimídia

O Sistema Rapport de Conferência Multimídia [Ahu 88] é um produto da categoria *rede de trabalho de grupo*. Nesse sistema cada usuário tem a sua disposição, no seu escritório, uma estação de trabalho. Essas máquinas estão ligadas numa rede Ethernet e comunicam-se com outras máquinas remotas via linha telefônica ([Ahu 88], [Agl 86]). Cada estação inclui uma tela de alta resolução, teclado, *mouse*, ou caneta ótica, um microfone e ainda uma câmera e um monitor de vídeo ou janela na tela para mostrar as faces dos conferencistas.

O objetivo do sistema Rapport é permitir que reuniões sejam realizadas sem que seus participantes precisem sair de seus escritórios. Dessa forma, algumas limitações provocadas pelo deslocamento dos membros podem ser superadas. Quando um participante sai de seu escritório para uma reunião importante, informações podem estar sendo deixadas para trás. Essas informações podem ser parte do estado de seu computador pessoal, anotações no quadro negro, artigos, jornais ou livros. Segundo Ahuja et al. [Ahu 88] a possibilidade de se utilizar informações contidas nos escritórios torna os participantes mais acessíveis aos outros membros da reunião. Além disso, o uso de computadores em encontros provê serviços diversos como por exemplo, agenda eletrônica de equipe, programas para a gravação do roteiro da reunião, programas para votação, edição de documentos e outros.

Os membros da equipe são providos de mecanismos que auxiliam na condução de reuniões, como ponteiros de tela e sinais. Os usuários podem *navegar* livremente nas diversas reuniões que estiverem ocorrendo.

### 2.4.1 Sala Virtual de Reuniões

Rapport provê um ambiente análogo a uma sala de reuniões face-a-face. De acordo com essa metáfora, uma pessoa pode iniciar uma conferência, indo a *sala de reuniões* e convidando outros indivíduos a se juntarem a mesma. Analogamente, quando uma pessoa deseja participar de uma conferência, ele deve entrar na sala de reuniões virtual apropriada. Uma vez na *sala*, os participantes de uma conferência estão aptos a conversar com os outros integrantes. O que cada conferencista vê, depende do dispositivo através do qual ele participa do encontro. Conferencistas usando estações de trabalho executando *Unix* e *sistema X Window* estão aptos a ver um conjunto de telas compartilhadas, produzidas por diversos programas de aplicação. Cada aplicativo compartilhado produz uma tela idêntica em cada estação,

algo análogo a um quadro negro de uma reunião face-a-face. Os programas compartilhados são, porém, entidades ativas que permitem a alteração dos dados durante a reunião.

Considerando que os conferencistas estão separados, necessita-se de um mecanismo para que a informação de interesse possa ser apontada em várias estações. Capacidade de apontar para informações é natural em reuniões face-a-face. No sistema Rapport, cada conferencista tem um ponteiro associado. Além disso, o Rapport suporta um serviço de mensagem. Uma pessoa que está participando de uma reunião, pode ser notificada de convites para outras reuniões.

Finalmente, o Rapport suporta um serviço de gravação da conferência que fornece um *script* da mesma para referência posterior.

#### 2.4.2 Princípios de Projeto

Para o Sistema Rapport, a chave do trabalho cooperativo são os dados compartilhados produzidos e manipulados pelos computadores. Esses dados devem estar disponíveis durante as reuniões da mesma forma que os programas aos quais os membros dos encontros estão habituados. Portanto, o princípio mais importante para o sistema Rapport é que os participantes de uma reunião possam utilizar os mesmos *softwares* aos quais estão acostumados no seu trabalho individual. Esses programas são executados e associados a conferência. Um dos módulos do sistema Rapport responsabiliza-se pelo gerenciamento de entrada e saída dos programas de forma a permitir a utilização dos mesmos pelo conjunto de membros integrantes da sessão. Cada aplicação executa numa única estação e sua saída é replicada para estações dos outros participantes da sessão. Cada programa tem uma janela associada. O sistema suporta a noção de controle de entrada para cada aplicativo. Esse controle permite que apenas uma pessoa entre com comandos ou dados para o programa num determinado instante, enquanto todos os outros usuários podem ver a sua saída. O controle de entrada para cada programa é feito separadamente, permitindo que diferentes conferencistas tenham controle de diferentes programas.

O segundo princípio de projeto do Rapport é a simplicidade de sua utilização. Ele tem mecanismos de controle e comunicação que refletem o compartimento convencional em encontros.

### 2.4.3 Macroestrutura do Rapport

O *software* é dividido em três níveis: do sistema, servidor de conferência e interface do usuário. Cada nível provê um conjunto de serviços para clientes com níveis superiores. Um nível é definido por suas operações e eventos. Uma operação corresponde à noção comum de procedimento. Possibilita a níveis superiores solicitarem serviços de níveis inferiores. Por exemplo, um cliente no nível de interface do usuário pode requisitar que um servidor no nível de conferência tome certas ações invocando uma determinada operação. Um evento é uma mensagem que flui de um nível inferior para um nível superior. Eventos notificam clientes interessados de certas alterações, como por exemplo, a adição de um conferencista, que ocorre num nível inferior. Esses eventos ocorrem assincronamente com respeito as atividades num nível superior. O nível do sistema do Rapport oferece serviços de transmissão de informação em vários meios e gerenciamento das telas nas diversas estações de trabalho. O nível do servidor de conferência é o coração do Rapport. Seus eventos e operações definem o conceito abstrato de conferência. Provê mecanismos para comunicação entre usuários, início, condução e encerramento de encontros. Além disso, o servidor de conferência suporta a execução de programas de aplicação e mantém a consistência entre as estações dos respectivos membros. O nível de interface do usuário é o nível mais elevado do sistema. A interface com o usuário tenta implementar as abstrações de uma sala de reuniões virtual. O acesso do usuário ao sistema Rapport se dá através de uma janela de conferência (veja figura2.5). Essa janela representa todas as salas virtuais de reuniões associadas com conferencistas. Cada pessoa pode ser membro de muitas conferências concorrentemente, porém, só pode estar realmente em uma conferência. Na figura2.5, o usuário é membro de três conferências - bob.δ, sid.δ, dave.δ - mas está como membro efetivo na conferência sid.δ.

## 2.5 LIZA : Um Conjunto Extensível de Aplicativos de Grupo

LIZA é um conjunto integrado de aplicativos que apresenta características das duas classes: *rede de trabalho de grupo* e *laboratório de colaboração*. Executa numa rede local de estações de trabalho. Um grupo de participantes usando o LIZA compõem uma sessão. Cada participante deve ter sua própria estação. As sessões podem ser distribuídas (cada participante pode estar no seu escritório) ou centralizadas (todos os participantes reunidos numa sala

**CONTROL bob.d sid.d dave.d**

Participants:  
sid +  
dave +

Applications:  
xterm; ctrl sid \*  
bitmap att; ctrl dave

xterm *2		
cfint.c	handler.c	phoneutil.c
cfint.c	handler.o	phoneutil.o
cfint.o	icdl	pointers.c
...	...	...
vishnut*		

**Making xterm**

Figura 2.5: Interface com o usuário do Rapport

de encontros). Chamadas para conferências são feitas através de canais de áudio entre participantes fisicamente separados. Para sessões centralizadas deve haver uma sala especial de encontros. Esta sala deve conter várias estações e um telão.

### 2.5.1 Objetos Ativos

Um aplicativo do LIZA é um conjunto de objetos ativos. Objetos ativos são extensão do paradigma de programação orientada a objetos que permite aos objetos executarem concorrentemente. Eles são executados dentro de aplicativos onde há a necessidade de controle descentralizado e paralelismo. O grupo de aplicativos do Liza está direcionado para trabalho cooperativo e simultâneo de membros de um equipe. Isto introduz concorrência num nível básico do sistema. Por esta razão, objetos ativos são adequados para construção do LIZA[Gib 89].

### 2.5.2 Estrutura de Aplicativos

No LIZA cada grupo de aplicativos consiste de um objeto ativo conhecido como *kernel* e um conjunto de objetos chamados *objetos imagem*. Estes últimos são responsáveis pela interface com o usuário. De forma geral há um objeto imagem para cada usuário. Quando um usuário executa uma ação significativa (por exemplo, pressiona o botão do *mouse* ou uma tecla) o objeto imagem comunica um evento ao objeto *kernel*. O *kernel* pode responder ao evento enviando comandos de alteração de tela para um ou mais objetos imagem.

Os objetos componentes de um aplicativo são instâncias de uma classe. Todas as classes de aplicativos derivam de uma superclasse padrão e portanto herdam os seus métodos (por exemplo criação e destruição de objetos imagem). Distingue-se quatro tipos de métodos:

- métodos para tratamento de eventos enviados pelos objetos imagem (usados pelo objeto *kernel*);
- métodos para tratamento de diretivas para alteração de tela (usados pelos objetos imagem);
- métodos públicos que os aplicativos importam de outros aplicativos;
- métodos privados usados internamente por cada aplicativo;

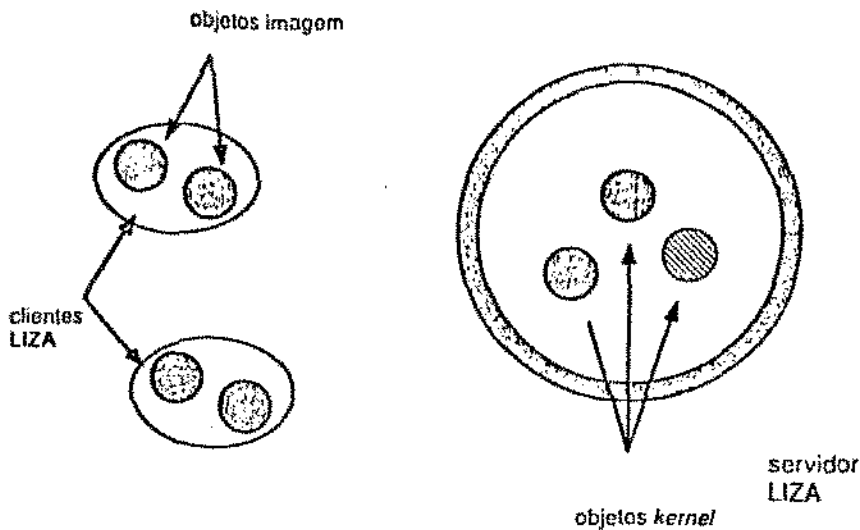


Figura 2.6: Arquitetura do sistema LIZA

### 2.5.3 Estrutura do Sistema

LIZA consiste de um certo número de clientes e um servidor único. Os objetos imagem executam como clientes enquanto o servidor executa objetos *kernel*. A comunicação entre servidor e cliente se faz via *sockets* (mecanismo de comunicação entre processos do Unix BSD). A arquitetura do sistema é mostrada na figura 2.6.

A manutenção da informação do estado global em um servidor centralizado simplifica o controle de concorrência. Os clientes podem, porém, executar operações de interface de forma autônoma.

O LIZA pode ser estendido pela adição de novos objetos aplicativos ao servidor e aos clientes. Um dos fatores que facilitam a extensão e a utilização de eventos para a comunicação entre clientes e servidores. Quando um objeto sinaliza um evento, este evento é processado pelos métodos de um objeto *kernel* que explicitamente se *interesse* pelo evento. O objeto imagem não precisa saber qual objeto *kernel* irá tratar o evento sinalizado por ele. O mesmo evento pode ser tratado por mais de um objeto *kernel*. Isso permite que o sistema seja estendido pela adição de novos objetos tratadores de eventos.

O sistema LIZA existente hoje tem oito aplicativos [Gib 89]. Abaixo apresentamos uma breve descrição dos cinco mais importantes:



- Aplicativo de Localização de Participantes - este aplicativo mostra a localização dos diversos participantes de uma sessão . O *layout* dos escritórios não correspondem a sua localização física, porém cada escritório é identificado com números das salas ou telefones.
- Aplicativo de Grafos - este aplicativo possibilita a um grupo de usuários a manipulação de um grafo compartilhado. Esta ferramenta é similar ao Cognoter, descrito anteriormente. Operações para criação, eliminação, cópia, movimentação, conexão e desconexão de nós são providas ao usuário.
- Aplicativo para Apresentação de *Slide* - este aplicativo consiste de duas janelas, uma contendo botões para seleção do *slide* e uma para sua exibição. Quando um usuário seleciona um *slide*, este é apresentado na janela de projeção de todos os usuários.
- Aplicativo do Painel de Controle - este aplicativo apresenta um painel de controle onde cada linha corresponde a participantes e cada coluna corresponde a aplicativos. O acionamento do botão do *mouse* implica na execução de um aplicativo para um usuário em particular.
- Aplicativo de Votação - suporta votação por computador.

## 2.6 Quilt: Uma Ferramenta para Escrita Cooperativa

Pessoas trabalhando juntas para escrever um documento é algo comum tanto no mundo dos negócios como no meio científico [Fis 88]. Combinando idéias de hipertexto com conhecimento de papéis sociais e padrões de comunicação envolvidos no esforço de escrita cooperativa foi projetado o sistema Quilt, que suporta alguns dos componentes do processo de escrita em grupo.

De forma geral, o Quilt permite o compartilhamento de voz, mensagens e documentos. Poderia ser enquadrado na categoria *rede de trabalho de grupo*. Ele tenta incorporar elementos sobre padrões de comunicação conhecidos e distinção de papéis entre indivíduos de uma equipe envolvidos em um trabalho cooperativo. Cada usuário do sistema Quilt tem um papel específico na escrita de um determinado documento em um certo instante. A esse papel podem ser associados direitos de acesso.

Assim, o sistema suporta diferentes direitos de acesso à informação, dependendo da interação entre o papel do usuário, da natureza da informação

e o estágio do projeto. Além disso, o acesso às informações relativas às atividades dos co-autores é facilitada para ajudar na coordenação dos trabalhos.

### 2.6.1 Estrutura dos documentos

O sistema Quilt usa metodologia de hipertexto. Um documento consiste de uma base e de nós ligados a essa base. Comentários, anotações, versões antigas, e consultas aos autores podem ser incorporadas ao texto como nós de um hipertexto.

Um documento em desenvolvimento no sistema Quilt compõem-se dos seguintes tipos de nós:

- Base do documento: texto considerado pelos autores como porção pública do seu trabalho;
- Sugestões para revisão, numa forma em que usuários com permissões apropriadas possam sobrepor parágrafos do documento;
- Comentário em texto ou voz. As permissões associadas definem três tipos de comentários:
  - Comentários privados, visíveis apenas ao criador do documento;
  - Comentários públicos, que podem ser lidos por qualquer usuário com direito da leitura do nó de nível hierárquico superior;
  - Mensagens direcionadas a um usuário ou a um grupo de usuários específicos.
- Versões antigas.

O sistema Quilt é independente de editor, de forma que editores familiares e outros programas aplicativos podem ser usados para manipular documentos. Em adição a operações de editores padrão, operações específicas do Quilt estão disponíveis para o trabalho de escrita cooperativa de documentos. As operações de edição incluem criação, eliminação, modificação, navegação, leitura, cópia de nós, pesquisa de conteúdo e atributos, impressão e outras. O usuário ao navegar por um documento, tem acesso a uma visão gráfica ou tabela de conteúdo dos nós que compõem o documento. Permissões de acesso relativas a objetos são definidos pelo papel social do usuário, tipo do objeto e estilo cooperativo estabelecido. Por exemplo, um estilo colaborativo associado com um documento pode permitir que um autor

e co-autor modifiquem o trabalho um do outro, um assistente leia correspondência entre autores, um revisor envie mensagens diretas para autores, e que todos possam ler porções públicas do documento além de fazer comentários privados. Uma mensagem direta ou comentário privado podem ser manipulados pelo criador ou receptor. Outros estilos cooperativos, geralmente informais, podem ser facilmente configurados no Quilt.

### 2.6.2 Coordenação de Atividades

De forma geral, a coordenação de atividades que envolvem o compartilhamento de informações requer que os participantes permaneçam informados sobre o que cada um está fazendo. Quilt provê dois mecanismos para troca de informações sobre atividades. Mantém um arquivo de atividades<sup>4</sup> para cada nó. Entradas nesse arquivo de atividades podem ser geradas pelos participantes, pelo sistema ou pela combinação dos dois. O segundo mecanismo é o uso de mensagens semi-estruturadas pelos participantes para comunicar suas ações. O Quilt suporta um sistema de conferência assíncrona que se constitui numa facilidade valiosa para escrita cooperativa de documentos.

## 2.7 Sistrac: Sistema de Suporte e Trabalho Cooperativo

O Sistrac é o sistema proposto e em grande parte implementado pelo autor deste trabalho. Estaria classificado na categoria *rede de trabalho de grupo* segundo Kraemer & King [Kra 88]. Provê um ambiente de cooperação que habilita usuários a compartilhar programas monousuário (característica do sistema Rapport) e possibilita a incorporação de programas multiusuário (característica do LIZA). Nos capítulos seguintes a macroestrutura do Sistrac é descrita e cada um de seus módulos detalhados.

---

<sup>4</sup>do inglês log file

## Capítulo 3

# Visão Geral do Sistema

O Sistrac (Sistema de Suporte a Trabalho Cooperativo), consiste de um conjunto extensível de ferramentas direcionadas para interação em tempo real entre usuários de um ambiente de desenvolvimento de software. O principal objetivo é permitir e motivar a utilização do computador em trabalhos em que haja a necessidade de cooperação entre membros de uma equipe.

Neste trabalho os termos programas monousuário e multiusuário são usados com frequência. Programas (aplicativos ou ferramentas) controlados por um único dispositivo de entrada são chamados monousuário. O dispositivo de entrada para aplicações interativas é normalmente o teclado. Em sistemas que dispõem de *mouse*, *track ball* ou *tablet*, considera-se o conjunto desses dispositivos ligados a uma única unidade central de processamento como elementos do dispositivo de entrada associado ao programa. Os programas normalmente produzem saída para um terminal de vídeo ou para uma janela (ou janelas) que se constituem no dispositivo de saída dos mesmos ([Ref 90],[Cou 85],[Bac 86]). Os programas monousuário normalmente estão associados a um único dispositivo de entrada e saída. Programas multiusuário são aqueles associados a vários dispositivos de entrada e saída e que interagem com múltiplos usuários.

O Sistrac compõe-se dos seguintes módulos:

- Sistema Gerenciador de Sessão (SGS) - sistema distribuído que controla as sessões de cooperação. Uma sessão de cooperação consiste de um conjunto de usuários em uma rede local de computadores interagindo através de programas multiusuário. O Gerenciador de Sessão constitui-se no *front-end* para todos os demais programas multiusuário do Sistrac.

- Sistema Compartilhador de Programas Monousuário - programa multiusuário que possibilita o compartilhamento de programas monousuário por membros de uma sessão de cooperação. O sistema compartilhador se coloca entre o programa e os usuários, associando aos primeiros múltiplos dispositivos de entrada e saída. Dessa forma, programas projetados para interagir com um único usuário podem ser utilizados em atividades cooperativas.
- Sistema de Comunicação entre Usuários (sistema de conferência por computador) - sistema cuja finalidade é suprir a necessidade de comunicação entre membros de uma equipe que realizam trabalho conjunto. Este sistema é responsável pelo gerenciamento das conversações entre os usuários componentes de uma sessão. Aproveitando o ambiente de cooperação criado e controlado pelo gerenciador de sessão, usuários da rede podem trocar mensagens. As conversações são estruturadas em falas [Pet 89]. A conferência ou conversação é ordenada e gravada em um arquivo que é usado como referência posterior. No presente trabalho, foi desenvolvido um mecanismo de comunicação ainda rudimentar com o objetivo de suprir as necessidades básicas de comunicação entre usuários que compartilham um programa. Este sistema será substituído por uma versão mais elaborada que se constitui em tema de outra tese de mestrado [Yam-91].
- Outros Programas Multiusuário - a alternativa para o utilização compartilhada de aplicativos monousuário é escrever novos programas de aplicação que possam interagir com múltiplos usuários simultaneamente. A diferença fundamental entre estas aplicações multiusuário e o compartilhamento de aplicações é que as primeiras explicitamente levam em conta a identidade de diferentes colaboradores [Sar 85]. Assim, múltiplos cursores de edição em dados compartilhados podem ser suportados; objetos estruturados manipulados por aplicações multiusuário (tais como um documento estruturado em seções e parágrafos) podem ter propriedades e direitos de acesso diferenciados que determinam quem pode executar quais operações sobre cada subestrutura do objeto. Esta abordagem requer a definição de protocolos diferentes para cada aplicação, o que gera um custo adicional de programação. O sistema gerenciador de sessão foi construído de forma a permitir o fácil acoplamento de programas multiusuário, possibilitando assim a extensão gradativa do Sistrac. Estão em fase de especificação pela equipe do projeto AHAND, um editor e uma agenda multiusuário.

além do sistema de conferência por computador.

Esta tese refere-se aos dois primeiros módulos do Sistrac, que compõem o núcleo do ambiente de cooperação e se encontram operacionais desde abril de 1991.

### 3.1 Requisitos para um Sistema Cooperativo

De que forma os computadores podem ser utilizados para suportar atividades de grupo, como por exemplo: encontros face-a-face, desenvolvimento conjunto de documentos, propostas ou projetos e estudos dirigidos por computador? Quais as características de interação entre usuários que devem ser respeitadas pelos programas? Enfim, quais requisitos devem atender? Obviamente, as respostas a essas perguntas variam de acordo com o espectro de atuação do *software* de grupo<sup>1</sup> que se pretende construir. O Sistrac foi desenvolvido em ambiente Unix, com o objetivo de ser uma ferramenta de cooperação entre membros de uma equipe de desenvolvimento de *software*. Propõe-se a ser configurável (extensível), seguindo a filosofia do Unix, de forma que novas aplicações multiusuário possam ser adicionadas ao ambiente Sistrac conforme a necessidade particular de cada equipe.

Os requisitos para o Sistrac foram estabelecidos a partir da observação das necessidades dos usuários da rede de estações de trabalho SPARC da Sun do Departamento de Ciência da Computação (UNICAMP) e de pesquisa bibliográfica sobre produtos correlatos como Colab [Ste 87], Capture Lab [Elw 89], Sistemas de Conferência Multimídia Rapport ([Ahu 88], [Agl 86]), e RTCALL [Sar 85], aplicativos para desenvolvimento cooperativo de documentos como Quilt [Fis 88] e Sistema para Colaboração entre Usuários em Tempo Real [Abd 88], o extensível conjunto de ferramentas de grupo Liza [Gib 89] e o projeto Cosmos [Wil 87].

Elwart-Keys et. al. [Elw 89] enumeram alguns requisitos para produtos de *software* que suportam reuniões face-a-face e suas interfaces. Alguns desses requisitos são comuns ao Sistrac, que pode ser usado tanto em reuniões face-a-face como em reuniões nas quais cada usuário está em seu escritório diante de sua estação interagindo com sua equipe. Os requisitos são comentados abaixo:

- Transição entre Trabalho Individual e Trabalho de Grupo - programas de computador que visam suportar reuniões face-a-face devem

---

<sup>1</sup>do inglês *groupware*

permitir dois níveis de transição: trabalho individual no escritório particular para o trabalho coletivo em encontros, e do trabalho individual dentro da reunião para o trabalho coletivo. O Sistrac possibilita a utilização dos mesmos programas aos quais os usuários estão habituados no seu trabalho individual, em trabalhos cooperativos. Assim, editores de texto, *shells* ou qualquer programa interativo não gráfico pode ser compartilhado através do Sistrac. Seguindo a filosofia do RT-CALL [Sar 85] e do LIZA [Gib 89], o Sistrac também permite a adição de ferramentas multiusuário. A transição do trabalho individual para trabalho de grupo é facilitada pelo compartilhamento de programas monousuário, enquanto atividades específicas de grupo podem ser suportadas por aplicações projetadas para tal fim.

- **Privacidade de Espaços para Trabalho Individual:** é natural que o projeto de sistemas cooperativos seja direcionado para as atividades coletivas. É necessário, no entanto, que haja espaço para trabalho individual, isto é, a possibilidade da utilização de janelas ou espaços privados. Desta forma, um participante de uma atividade coletiva pode realizar trabalhos preliminares antes de expor suas idéias à equipe. O Sistrac apresenta uma interface onde há janelas (e processos) compartilhadas (espaço compartilhado) e janelas (e processos) privadas (espaço privado). Um usuário membro de uma sessão de cooperação pode solicitar a execução de programas monousuário compartilhados ou programas multiusuário que comporão o seu espaço compartilhado. Ao mesmo tempo, pode de forma independente dos outros membros, abrir janelas privadas onde executam programas que constituirão o espaço privado.
- **Interface WYSIWIS<sup>2</sup>** - em ambientes em que cada usuário tem um computador pessoal, a informação compartilhada pode ser mostrada seguindo-se um dos dois critérios:
  - A informação compartilhada é mostrada na tela de cada computador pessoal e permite-se ao usuário controlar as características das janelas compartilhadas (localização e dimensão), isto é, configurar a sua visão do espaço compartilhado;
  - A informação compartilhada é apresentada da mesma maneira para todos os usuários. Essa abordagem facilita a referência às

---

<sup>2</sup>What You See Is What I See

informações compartilhadas durante as conversações. Em ferramentas para suporte de encontros face-a-face como o Capture Lab a interface segue o paradigma WYSIWIS. O Sistrac, assim como o Colab [Sar 85], usa uma versão relaxada do paradigma WYSIWIS. Janelas privadas são permitidas, além da configuração particular das janelas do espaço compartilhado;

- Desempenho capaz de permitir o acompanhamento das conversações: este é um requisito fundamental para ferramentas de suporte a reuniões face-a-face. O Sistrac, no que diz respeito aos módulos de gerenciamento de sessões e compartilhamento de programas monousuário responde rapidamente aos comandos do usuário, possibilitando sua utilização em reuniões face-a-face;
- Suporte para protocolos e estruturas sociais: um sistema de suporte a trabalho cooperativo não deve impor uma estrutura particular de cooperação, o que pode romper com o modelo familiar de interação, diminuindo a produtividade da equipe. O Sistrac define alguns modos de operação para uma sessão de cooperação que definem protocolos diferentes para o gerenciamento de atividade conjunta:
  - \* democrático: qualquer usuário pode solicitar a execução de programas no espaço compartilhado;
  - \* autocrático: somente o gerente da sessão pode executar programas no espaço compartilhado;
  - \* público: a sessão está aberta para entrada de novos membros (usuários não convidados) e as informações referentes às mesmas estão disponíveis a todos os usuários da rede;
  - \* secreto: é vedada a entrada de usuários não convidados na sessão de cooperação, e as informações relativas às sessões só estão disponíveis para seus membros;

A estrutura social definida compõe-se dos seguintes elementos:

- \* Gerente da sessão: geralmente o usuário que solicitou criação da sessão. Tem privilégios na execução de algumas operações associadas a sessão.
- \* Membro efetivo: usuário que participa da sessão de cooperação.
- \* Membro convidado: usuário que foi convidado para participar da sessão de cooperação e ainda não se integrou a mesma.



- **Executor de programa monousuário:** o sistema de compartilhamento de programas monousuário opera usando um protocolo não preemptivo de controle de concorrência. Somente um usuário tem direito de escrita num certo instante (executor). Todos os outros têm direito de leitura (vêem a saída do programa compartilhado). Somente quando o executor libera o direito de escrita, outro usuário pode assumir a condição de executor. Cada ferramenta multiusuário que for acoplada ao ambiente Sistrac pode estabelecer protocolos variados e uma estrutura social compatível com o enfoque do aplicativo.
- **Mínimo treinamento requerido:** o Sistrac atende a este requisito, permitindo o compartilhamento de programas aos quais os usuários estão habituados no seu trabalho individual.

## Capítulo 4

# Módulos Componentes do Sistrac

Neste capítulo são apresentados em detalhes os dois módulos que são focalizados nesta tese - Sistema Gerenciador de Sessão (SGS) e Sistema de Compartilhamento de Programas Monousuário (SCPM) - dando-se ênfase a funcionalidade e ao fluxo de dados entre eles. No final do capítulo, a interface gráfica do Sistrac é discutida.

### 4.1 Sistema Gerenciador de Sessão

O sistema gerenciador de sessão é o coração do Sistrac. É através dele que as pessoas que desejam colaborar são reunidas e programas aplicativos são executados.

Usuários que pretendem trabalhar de forma cooperativa, compartilhando programas monousuário ou utilizando programas multiusuário e mantendo conversações através de uma rede de computadores, compõem uma sessão de cooperação. A cada sessão são associadas informações e operações. As informações consistem dos seguintes itens:

- Lista de usuários convidados: lista de usuários que devem ser localizados e convidados a participar da sessão de cooperação;
- Lista de membros efetivos: um membro é efetivado quando aceita participar da sessão de cooperação. Associado a cada membro efetivo tem-se o seu identificador e o de sua estação;

- Gerente de sessão: A gerência é um cargo que é atribuído a um dos membros da sessão. Este membro pode ser especificado no momento da criação da sessão. Senão, o usuário que solicita a criação da sessão de cooperação passa automaticamente a ser o seu gerente. Como tal, tem algumas atribuições especiais. O convite de usuários, a determinação do modo de operação da sessão, a expulsão de membros desobedientes, a aceitação de novos membros e o encerramento da sessão são privilégios do gerente;
- Lista de programas: numa sessão de cooperação, pode-se executar programas multiusuário ou programas monousuário compartilhados. Associado a cada programa há um conjunto de informações:
  - Identificador da estação na qual executa o programa;
  - Linha de comando referente à execução do programa;
  - Tipo de programa: monousuário ou multiusuário;
  - Descritor do canal de comunicação entre o SGS e o programa;
- Identificador da sessão de cooperação;
- Data e horário do início da sessão;
- Título da sessão;
- Modos de operação (democrático ou autocrático, público ou secreto);

O conjunto de operações habilita um componente de uma sessão de cooperação ou um mero usuário da rede a acessar as informações referentes às sessões correntes ou solicitar a execução de serviços que vão desde a criação ao encerramento de sessões. As operações associadas as sessões de cooperação foram agrupadas em três categorias:

- Operações gerais: dizem respeito ao sistema como um todo.
  - Criação de sessão - quando um usuário deseja trabalhar em equipe, solicita a criação de uma sessão de cooperação. Esta operação consiste na determinação da lista de convidados, modos de operação, gerente e título da sessão, localização e o envio de convites para todos os membros e iniciação das estruturas de dados referentes a sessão;

- Listagem de convites e sessões correntes - quando um usuário se conecta à rede, pode solicitar a listagem de todas as sessões correntes (excluindo-se as sessões secretas) e de todos os convites a ele endereçados;
- Operações específicas: referem-se a uma sessão em particular.
  - Entrada de Membros - um usuário, convidado ou não, pode solicitar sua integração a uma sessão através dessa operação. Se estiver na lista de convidados, sua integração é automática. As janelas e processos referentes ao espaço compartilhado são criados. Se o usuário, porém, não estiver convidado e a sessão tiver modo de operação público, a solicitação é encaminhada ao gerente que aceita ou não a entrada do novo membro;
  - Convite de usuário - mesmo depois de iniciada uma sessão de cooperação, é permitido ao gerente convidar novos membros. Os usuários são localizados e convites são exibidos na seus terminais. Se o usuário estiver utilizando um terminal gráfico, o convite é apresentado na forma de ícone. Senão, uma mensagem é exibida na sua tela;
  - Execução de programas - como já foi dito, a cooperação entre usuários se dá através de programas que compõem o espaço compartilhado. Se a sessão tiver modo de operação democrático, qualquer membro pode executar programas. Caso o modo seja autocrático, somente o gerente está autorizado a executar programas;
  - Obtenção de informações - se a sessão for pública, todos os usuários têm acesso a suas informações. Se for secreta, somente membros têm permissão para leitura das informações;
  - Alteração dos modos de operação - essa operação é privilégio do gerente. Consiste da alteração dos modos da sessão anteriormente descritos;
  - Eliminação de membro - possibilita ao gerente a remoção de membros da sessão;
  - Saída de membro - um usuário que não mais deseja participar de uma sessão pode solicitar sua exclusão da mesma. Os processos e janelas componentes do espaço compartilhado referentes ao membro são encerrados e o usuário retirado da lista de membros

efetivos. Se o gerente desejar sair da sessão, ele pode especificar o seu sucessor. Há, porém, a possibilidade de ter-se uma sessão sem gerente. Nesse caso, nenhuma das operações que são privilégio do gerente podem ser executadas e o encerramento da sessão ocorre automaticamente com a saída de todos os membros.

- Alteração do gerente - o gerente pode passar os direitos de gerência para outro membro da sessão através desta operação;
  - Encerramento da sessão - esta operação é privilégio do gerente. Todos os usuários são informados do fim da sessão. Os processos e janelas referentes ao espaço compartilhado são encerrados e as estruturas de dados da sessão reinicializadas. Uma sessão termina automaticamente quando o último membro sai da sessão.
- Operações referentes aos programas: estas operações estão ligadas aos programas executados na sessão. O conjunto de operações desta categoria, pode ser estendido com a incorporação de novos programas multiusuário. Abaixo são descritas as operações referentes aos programas monousuário compartilhados.
    - Alteração de executor: um programa monousuário compartilhado permite que apenas um usuário tenha direito de escrita num certo instante. Essa operação consiste na requisição ou liberação do direito de escrita por um participante, o que implica na alteração do executor de uma determinada aplicação compartilhada.
    - Envio de mensagens: Essa operação possibilita a um usuário enviar mensagens para todos os membros ou um subgrupo dos componentes de uma sessão

As operações acima descritas são mapeadas em serviços. Estes serviços são realizados por servidores de sessão (excetuando-se as operações referentes aos programas) que executam em cada máquina (veja capítulo sobre implementação). A execução dos serviços podem ser solicitadas interativamente por linha de comandos, por interface gráfica (*Sistractool* - em desenvolvimento) ou por programas através de funções de biblioteca do *Sistrac*.

## 4.2 Sistema de Compartilhamento de Programas Monousuário

Elementos de um grupo que utilizam programas de aplicação particulares (por exemplo, um editor de texto) como parte de seu trabalho cotidiano, terão tendência natural de querer continuar a usar aquelas ferramentas ao trabalharem em equipe. Seria necessário, apenas, aprender os novos comandos de controle indispensáveis ao gerenciamento do compartilhamento da aplicação.

Visto que os programas aplicativos existentes tipicamente interagem com um terminal do usuário via entrada e saída de fluxo de caracteres, tais programas podem ser compartilhados sem modificação, como segue ([Sar 85], [Bac 86]):

- Ao invés de interagir diretamente com o terminal do usuário, o programa interage com um *processo controlador* que exerce o papel de um terminal virtual;
- O processo controlador se comunica com as estações de trabalho dos participantes (ou processos referentes aos mesmos), e multiplexa a entrada e replica a saída do programa de forma a produzir a funcionalidade desejada. Isto é, o controlador envia a saída do aplicativo para todas as estações, e recebe os comandos de entrada do terminal do executor;

Apesar das vantagens de simplicidade e baixo custo (tanto de implementação como de treinamento) do compartilhamento de programas monousuário, há limitações de funcionalidade decorrentes do fato destes aplicativos terem sido projetados para interação com um único usuário [Sar 85]:

- Múltiplos contextos concorrentes (tais como a edição de parágrafos distintos por diferentes usuários) no mesmo espaço compartilhado não podem ser suportados;
- Não é possível obter visões privadas de informações compartilhadas (por exemplo, informações relativas a um usuário realçadas na sua tela);
- Não podem ser dadas permissões de acesso e ambientes diferenciados aos usuários que compartilham a aplicação. Todas as operações executadas por um programa compartilhado serão baseadas em privilégios e

ambiente de um usuário específico (o usuário que solicitou a execução do mesmo), que serão em geral diferentes daqueles referentes aos outros usuários. Portanto, um usuário poderá tentar ler ou escrever em um arquivo ao qual normalmente teria acesso e não conseguir pelo simples fato de que o usuário de referência não tem o acesso requerido. Por outro lado, um participante da sessão poderia inadvertidamente escrever num arquivo dentro do diretório do usuário de referência, ao invés de escrever no seu.

#### 4.2.1 Fluxo de Dados

É a partir do sistema gerenciador de sessão que são executados os aplicativos do espaço compartilhado (componentes dos outros três módulos). Canais de comunicação são estabelecidos para que processos componentes dos módulos possam *conversar*.

O fluxo de informação entre o sistema gerente de sessão e os outros módulos ocorre em função de algumas operações. A ação resultante do envio de mensagem do sistema gerente de sessão para processos dos outros módulos depende exclusivamente desses processos. Abaixo, relacionamos as operações que produzem a troca de mensagens na direção do módulo gerenciador de sessão para demais módulos:

- Execução de programa - quando um programa é executado, algumas informações referentes a sessão são passadas como parâmetro para que ele reconheça os membros da sessão. Essas informações compõem-se de:
  - membros da sessão (identificadores do membro e de suas estações);
  - geometria da janela de entrada e saída;
  - executor inicial (no caso específico de programas monousuário compartilhados);
  - ambiente do usuário solicitante;
- Entrada e saída de membro - essa operação deve produzir a execução ou encerramento (respectivamente) de processos e janelas componentes do espaço compartilhado referentes ao membro. A informação consiste de:
  - identificador do membro;
  - identificador da estação do membro;

- informação de controle identificando a operação;
- Encerramento de sessão - esta operação deve implicar na morte de processos e janelas do espaço compartilhado. A mensagem que flui para outros módulos compõe-se somente de informação de controle.

O fluxo de dados na direção programas-SCS ocorre em consequência do encerramento de um programa. Quando um programa multiusuário ou programa compartilhado termina, um sinal é enviado para o sistema gerenciador de sessão. Esse sinal é tratado, atualizando-se as estruturas de dados referentes a sessão da qual faz parte o programa.

### 4.3 Sistractool: a interface gráfica do Sistrac

O *Sistractool* corresponde à interface gráfica que possibilita ao usuário solicitar serviços do sistema gerente de sessão. A interface compõe-se de um painel de controle com diversos botões e uma janela a parte intitulada *Sistrac Console* (veja figura 4.1).

#### 4.3.1 Painel de Controle

No painel de controle, cada botão corresponde a uma operação. O painel está dividido em seções. A primeira seção refere-se a operações gerais e é intitulada *General Operations*. Ao lado do título aparece o identificador do usuário da *console* da estação. Os botões dessa seção são descritos abaixo:

- *Create*: esse botão refere-se a operação de criação de sessão. A seleção deste botão com o *mouse* faz com que o sistema apresente uma janela através da qual solicita do usuário a digitação dos seguintes dados:
  - Título da sessão
  - Nome do gerente da sessão (se nada for digitado assume-se que o gerente é o usuário que solicitou a criação da sessão);
  - Lista de convidados no formato: usuário1 usuário2...;
  - Modo de operação da sessão;
- *Current Sessions*: quando este botão é ativado, informações concisas sobre as sessões correntes são apresentadas. Estas informações compõem-se dos seguintes itens:



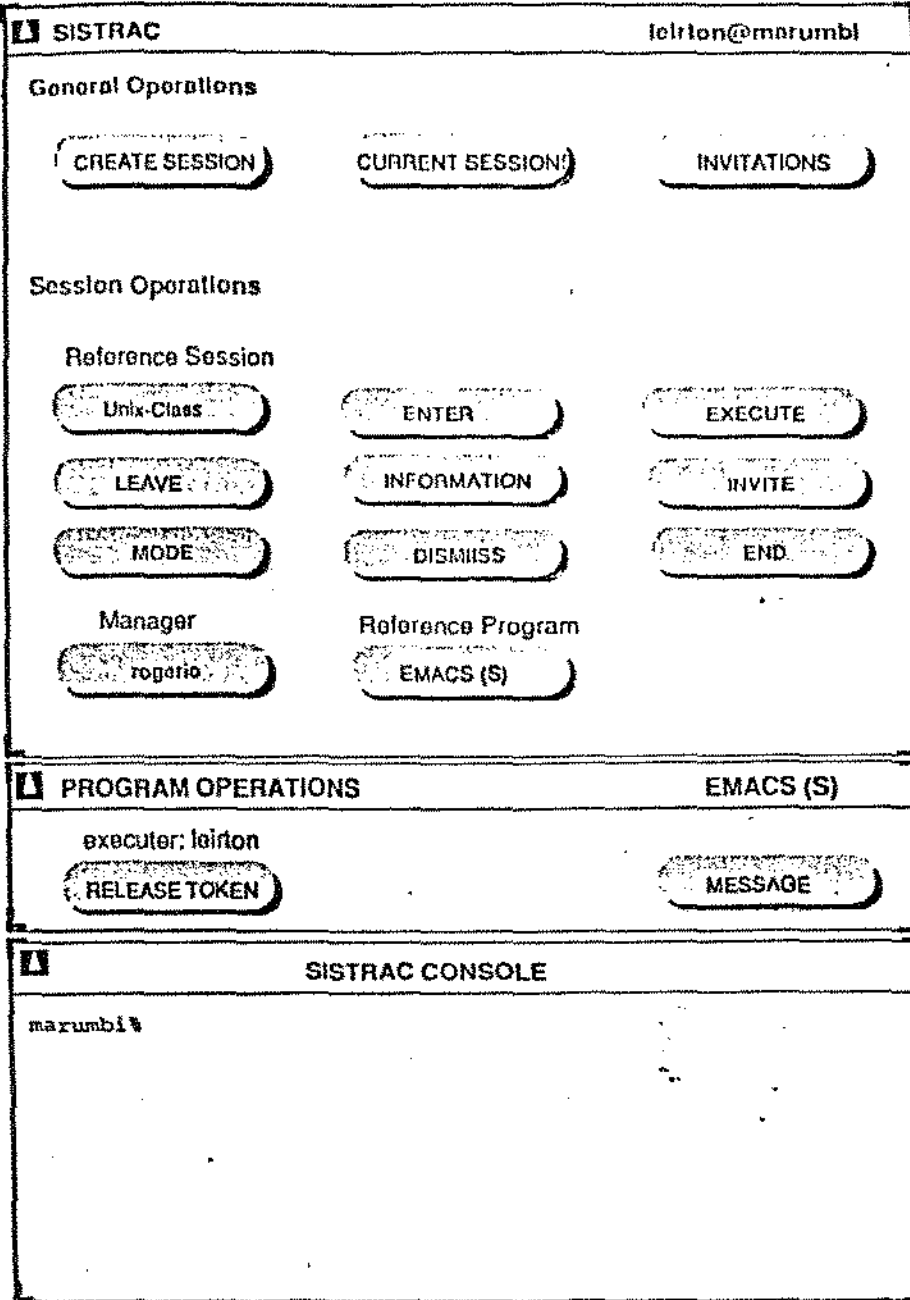


Figura 4.1: Sistractool: A Interface Gráfica do Sistrac

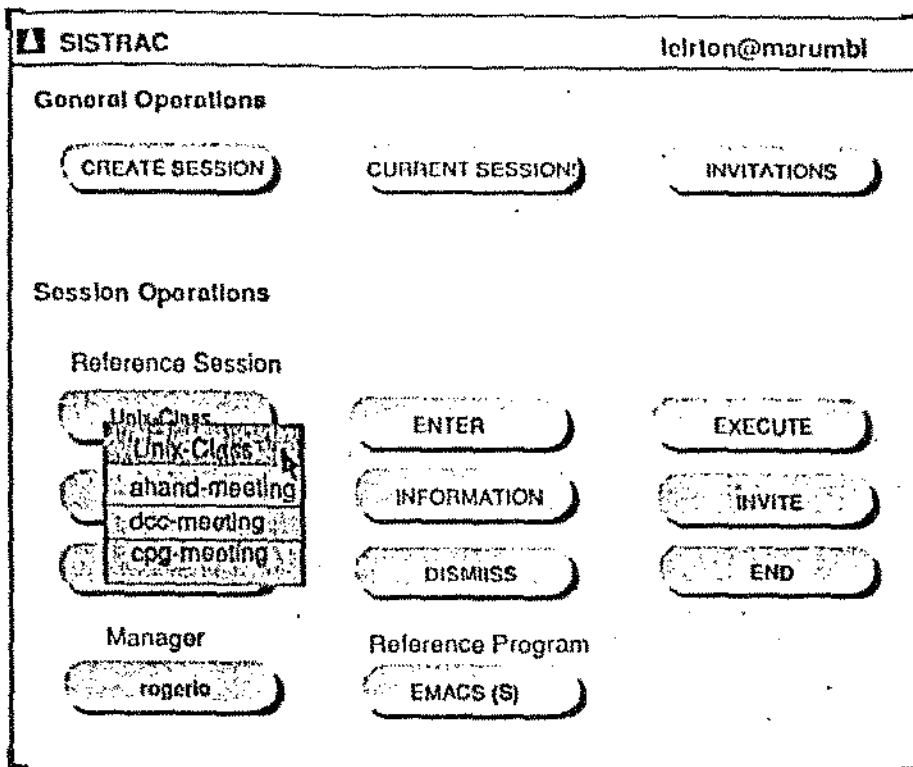


Figura 4.2: Botão *Session Reference*

- Título da sessão
  - Nome e estação do gerente
  - Lista de convidados
- *Invitations*: quando este botão é selecionado, os convites enviados ao usuário são apresentados em uma janela sobreposta ao painel de controle. Para cada convite são apresentadas as seguintes informações:
    - Identificador da sessão
    - Título da sessão
    - Nome do gerente

A próxima seção do painel do *Sistractool* corresponde as operações específicas de uma sessão e é intitulada *Session Operations*. A seguir, são descritos os botões desta seção:

- *Referenc* *Sessiom*: o título deste botão é o título da sessão de referência. Todas as outras operações desta seção do painel de controle aplicam-se a esta sessão. Quando este botão é ativado, mostra-se na tela um menu com o título de todas as sessões correntes (veja figura 4.2). O usuário pode selecionar uma delas que passa a ser a sessão de referência;
- *Enter*: a seleção deste botão implica na solicitação de entrada na sessão de referência. Este botão refere-se a operação de entrada de membro.
- *Execute*: este botão possibilita ao usuário a execução de programas dentro de uma sessão de cooperação. A seleção deste botão implica na apresentação de uma janela com a solicitação para a digitação dos seguintes dados:
  - Tipo do programa: monousuário compartilhado ou multiusuário;
  - Geometria da janela de entrada e saída do programa compartilhado;
  - Identificador do executor inicial no caso de programa monousuário;
  - Linha de comando para execução do programa;

Este botão refere-se a operação de execução de programas.

- *Mode*: quando este botão é ativado, é exibida uma janela com os dois modos correntes da sessão (autocrático ou democrático e público ou secreto). O usuário, caso seja gerente da sessão, pode alterar os modos pressionando o botão do *mouse* sobre o modo que deseja modificar.
- *Information*: essa opção permite ao usuário obter informações completas sobre a sessão de referência. Essa informação é apresentada em uma janela sobreposta ao painel com os seguintes itens:
  - Título da sessão
  - Nome e estação do gerente
  - Nome e estação de cada membro efetivo
  - Linha de comando e tipo e tipo de cada programa do espaço compartilhado;
  - Data e horário do início da sessão;
  - Nome e estação de usuários que solicitaram entrada na sessão e ainda não foram convidados;

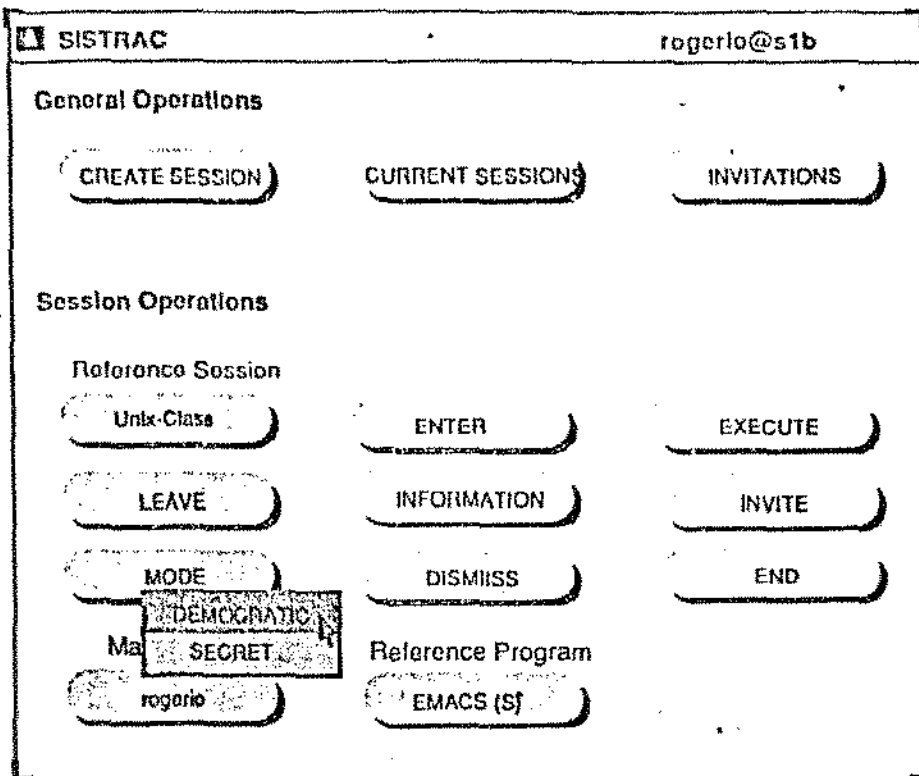


Figura 4.3: Botão *Mode*

– Modos da sessão de cooperação;

Este botão corresponde a operação de obtenção de informações sobre sessões.

- *Invite*: este botão possibilita ao gerente da sessão o convite de novos usuários para participarem de uma sessão já iniciada (no caso, a sessão de referência). A seleção deste botão faz com que o sistema apresente uma janela onde deve ser digitada a lista de novos convidados. Este botão corresponde a operação de convite de usuários.
- *Manager*: quando o gerente da sessão de referência quiser ser substituído, deve ativar este botão e digitar o nome de um dos membros efetivos numa janela sobreposta ao painel de controle. O nome do gerente atual é o título do botão. Corresponde a operação de alteração de gerente.
- *Leave*: este botão corresponde a operação de saída da sessão. Quando um usuário deseja se *retirar* da sessão deve ativar este botão. No caso do usuário ser o gerente, o sistema solicita a digitação do nome do sucessor em janela sobreposta ao painel de controle.
- *Dismiss*: refere-se a operação de eliminação de membro. Quando o gerente da sessão de referência deseja eliminar um participante, deve

selecionar este botão. Uma janela é sobreposta ao painel de controle para que o usuário solicitante digite o nome do participante a ser eliminado.

- *End*: quando este botão é selecionado a sessão de referência é encerrada. Refere-se a operação de encerramento de sessão.
- *Reference Program*: este botão é semelhante ao botão *Reference Session*. A sua ativação implica na apresentação de um menu com todos os programas do espaço compartilhado da sessão de referência. O menu apresenta a linha de comando de cada programa com as letras *S* ou *P* entre parênteses indicando programa compartilhado ou programa multiusuário respectivamente. A seleção de uma opção do menu faz com que seja alterado o programa de referência para as opções da próxima seção do painel. O programa de referência é apresentado como do título do botão.

A última seção do *Sistructool*, intitulada *Program Operations*, apresenta os botões relacionados com operações sobre programas do espaço compartilhado. Planeja-se que o *Sistructool* seja uma interface configurável, de forma que cada programa multiusuário (incluindo-se o compartilhador de programas multiusuário) acrescente os subpainéis que necessite. O subpainel referente ao compartilhador de programas monusuário apresenta dois botões:

- *Token*: este botão refere-se a operação de alteração de executor. Acima do botão é exibido o nome do executor atual do programa de referência. Se o usuário é o executor, o título do botão é *Release Token* indicando que o usuário pode liberar o direito de escrita ativando o botão. Senão, o título do botão é *Get Token* e sua ativação implica na requisição do direito de escrita. No primeiro caso, o usuário poderá indicar seu sucessor numa janela sobreposta ao painel de controle. Se o sucessor não for indicado, o primeiro da lista de espera associado ao programa assume o papel de executor.
- *Message*: este botão permite, quando ativado, que o usuário solicitante leia ou envie mensagens para todos ou um subconjunto dos usuários que compartilham um programa (veja figura 4.4). Quando o usuário recebe uma mensagem o botão aparece realçado. Ativando-se o botão um menu é exibido com duas opções:

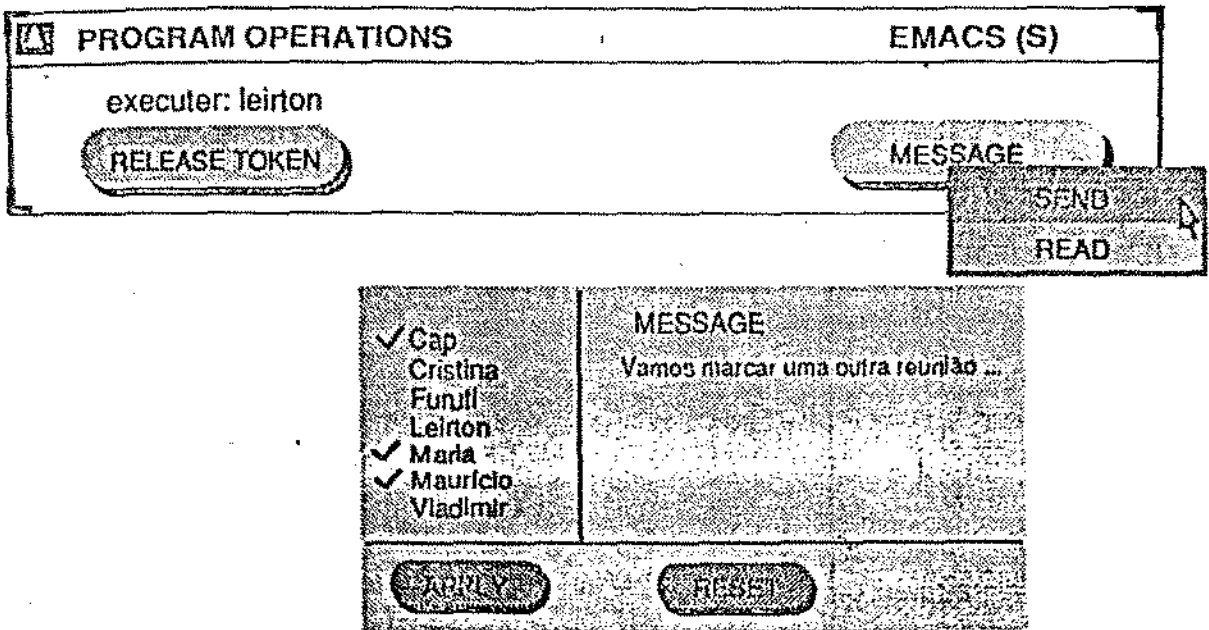


Figura 4.4: Botão *Message*

- *Recall*: a escolha desta opção faz com sejam mostradas numa janela sobreposta ao painel todas as mensagens que ainda não foram lidas pelo usuário;
- *Send*: a seleção desta opção produz a apresentação de uma janela com todos os usuários que compartilham o programa, além de espaço para digitação da mensagem. O usuário, deve selecionar aqueles que receberão a mensagem, digitá-la e depois ativar o botão *Apply* para enviá-la.

Não se pretende com este sistema simples de troca de mensagens, implementado pelo SCPM, substituir o Sistema de Conferência por Computador em desenvolvimento por outros membros da equipe AHAND.

#### 4.3.2 *Sistrac Console*

O *Sistrac Console* é a janela onde são apresentadas as mensagens do sistema gerenciador de sessão. Um dos serviços do sistema implementa o *Sistrac Console*. Quando uma mensagem deve ser apresentada e ainda não há a janela do *Sistrac Console*, ela é criada e a mensagem é impressa. Uma *shell* executa no *Sistrac Console* de forma que os comandos do sistema podem ser executados a partir desta janela. As mensagens do sistema são listadas no apêndice C.

## Capítulo 5

# Implementação do Protótipo

Este capítulo aborda alguns tópicos de implementação do protótipo do Sistrac. Enfoca técnicas utilizadas, descreve problemas e soluções encontradas. Os sistemas gerenciador de sessão e compartilhador de programas monousuário são discutidos separadamente.

### 5.1 Sistema Gerenciador de Sessão

Todas as operações associadas as sessões de cooperação, excetuando-se as operações referentes a programas, são mapeadas em serviços. Cada estação executa um processo criado em tempo de inicialização da máquina<sup>1</sup> denominado de servidor de sessão. Esse servidor é responsável pelo armazenamento e manipulação das informações referentes à sessão, além da execução de serviços relativos às operações descritas anteriormente.

O sistema gerenciador de sessão segue o paradigma mais comumente utilizado na construção de aplicativos distribuídos: modelo cliente-servidor ([Net 90], [Man 89]). Nesse modelo, processos clientes solicitam a execução de serviços a processos servidores. Isto implica no estabelecimento de um protocolo de comunicação entre estes processos. O protocolo entre cliente e servidor no sistema gerenciador de sessão é assimétrico, isto é, o servidor recebe requisições e devolve resultados aos processos clientes. O servidor nunca assume papel de cliente e vice-versa.

Na implementação do sistema gerenciador de sessão utilizou-se chamada remota de procedimentos<sup>2</sup> ([Net 90], [Man 89]). Cada sessão tem um ser-

---

<sup>1</sup>do inglês boot time

<sup>2</sup>do inglês remote procedure call



vidor gerente, que executa na estação do membro designado inicialmente como gerente (caso haja alteração posterior do membro gerente, o servidor gerente permanece o mesmo). Todas as informações e operações associadas a uma sessão são administradas pelo servidor gerente. Há um arquivo visível a todos os servidores que centraliza informações concisas sobre as sessões correntes. Essas informações compõem-se de:

- Identificador da sessão;
- Identificador do usuário gerente, geralmente o usuário que solicitou a criação da sessão;
- Identificador da estação do servidor gerente, inicialmente a estação do usuário gerente da sessão;
- Lista de convidados;
- Título da sessão;

Os processos clientes executados pelos usuários da rede (através de linha de comando ou interface gráfica) localizam o servidor gerente consultando o arquivo de sessões correntes. Solicitam, então, o serviço através de uma chamada remota de procedimento. O processo servidor recebe a requisição, realiza o serviço e retorna os resultados para processo cliente.

A representação dos dados no sistema gerenciador de sessão está no formato padrão de *representação externa de dados XDR*<sup>3</sup>. Dessa forma, os processos servidor e cliente podem executar em máquinas com arquiteturas distintas.

### 5.1.1 Estrutura do Sistema Gerenciador de Sessão

Estruturalmente, o Sistema Gerenciador de Sessão encontra-se dividido conforme figura 5.1. Os módulos estão organizados em duas unidades: cliente e servidor. Os módulos *file* e *sistrac\_xdr* são compartilhados por ambos. A seguir apresentamos uma descrição detalhada de cada módulo:

- *Módulo parser*

O *parser* é o módulo encarregado de analisar a linha de comando e da interface com o usuário. Este módulo faz a chamada das funções do

---

<sup>3</sup>External Data Representation proposta pela Sun Microsystems e disponível em suas estações junto com o RPC

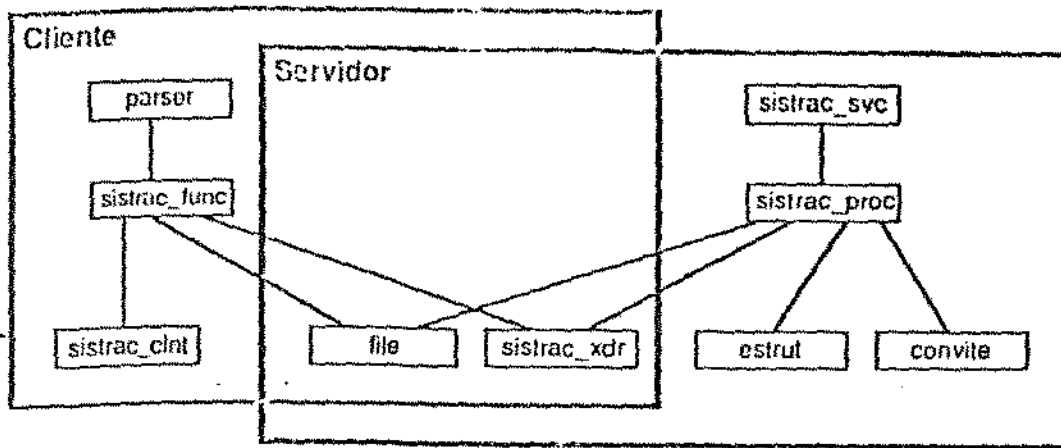


Figura 5.1: Estrutura do Sistema Gerenciador de Sessão

módulo *sistrac\_func* e no seu retorno exibe mensagens para o usuário. O *parser* ocupa estruturalmente a mesma posição do módulo de interface gráfica do Sistrac descrita no capítulo 4. O apêndice A apresenta a sintaxe de todos os comandos reconhecidos pelo *parser* e o apêndice C as mensagens exibidas pelo sistema.

- Módulo *sistrac\_func*

Este módulo contém quase todas as funções exportáveis do Sistrac descritas no apêndice B (exceto a função *interface()*). A maioria das funções deste módulo corresponde a um serviço no módulo *sistrac\_proc* da unidade servidor. Estas funções são chamadas pelo *parser* (ou módulo de interface gráfica) com parâmetros correspondentes aos argumentos das opções da linha de comando ou informações solicitadas ao usuário. O arquivo de sessões é normalmente consultado para a localização do servidor gerente da sessão e/ou, no caso de operações de direito exclusivo do gerente, para verificação se usuário solicitante é o gerente.

O retorno das funções deste módulo corresponde em geral, ao resultado da execução remota de procedimentos pelo servidor gerente da sessão. As operações de listagem de convites e sessões correntes, porém, não são mapeadas em serviços, pois consistem de simples consultas ao ar-

quivo de sessões.

As funções deste módulo não se responsabilizam pela exibição dos resultados.

- Módulos *sistrac\_clnt*, *sistrac\_svr* e *sistrac\_xdr*

O modelo de chamada remota de procedimentos trata a troca de mensagens de maneira similar as chamadas de procedimentos tradicionais ([Man 89], [Net 90]). O processo designado cliente, envia uma requisição para um outro processo chamado servidor. A requisição, dentre outras coisas, compõe-se dos parâmetros do procedimento chamado. O cliente fica bloqueado até que o servidor envie uma resposta. Um dos componentes desta resposta é o resultado da execução do procedimento remoto.

Na implementação do Sistrac, usou-se o compilador *rpcgen*. Ele aceita como entrada a definição de uma interface do programa remoto escrita na linguagem RPC, muito similar a C. A saída do *rpcgen* compõe-se de *stubs* do cliente e servidor e um módulo contendo rotinas XDR. Os *stubs* tornam praticamente transparente para o usuário a chamada e execução remota de procedimentos. Ao desenvolver o programa o usuário faz as chamadas remotas de procedimento praticamente como se fossem locais. O *stub* do cliente responsabiliza-se pelo *empacotamento* dos parâmetros, pelo envio da requisição e o *desempacotamento* dos resultados devolvidos pelo servidor. O *stub* do servidor desempacota os parâmetros e faz a chamada local do procedimento desejado. Este procedimento, por sua vez, trata a chamada como se fosse local e devolve os resultados ao *stub* do servidor. Os resultados são empacotados e enviados de volta para o *stub* do cliente. As rotinas XDR são usadas para o *empacotamento* e *desempacotamento* dos parâmetros e resultados num formato independente da arquitetura dos computadores envolvidos.

Os módulos *sistrac\_clnt*, *sistrac\_svr* e *sistrac\_xdr* correspondem respectivamente aos *stubs* do cliente e servidor e às rotinas XDR.

- Módulo *file*

Neste módulo estão todas as rotinas de manipulação do arquivo de sessões. Sempre que o arquivo é consultado para posterior atualização, o registro é bloqueado. Dessa forma, se dois servidores concorrem pelo

mesmo registro, um deles ficará bloqueado até que o registro seja liberado. O formato do registro do arquivo de sessão é o seguinte:

*sid:título:gerente:servidor:convidados*

onde

*sid* - valor numérico que identifica unicamente uma sessão;

*título* - cadeia de caracteres correspondente ao título da sessão. Se a sessão é secreta, este campo é nulo;

*gerente* - cadeia de caracteres que identifica o usuário gerente;

*servidor* - cadeia de caracteres que identifica a estação servidor gerente;

*convidados* - cadeia de caracteres que identifica os usuários convidados;

#### • Módulo *estrut*

Este módulo compõe-se das rotinas que manipulam as estruturas de dados que armazenam as informações associadas as sessões de cooperação. Abaixo descrevemos cada estrutura:

- Lista de sessões: cada servidor pode gerenciar várias sessões. Para cada sessão há uma estrutura composta dos seguintes campos:

- \* *gerente* - estrutura que contém o identificador do membro gerente e de sua estação.
- \* *firstmemb* - lista de membros efetivos
- \* *firstprog* - lista de programas do espaço compartilhado
- \* *firstwait* - lista de espera de membros
- \* *session\_id* - identificador da sessão
- \* *title* - título da sessão
- \* *mode* - modo de operação da sessão
- \* *dat* - data e horário do início da sessão

As sessões gerenciadas por um servidor constituem a lista de sessões.

- Lista de membros efetivos - guarda informações sobre cada membro da sessão. O nó desta lista compõe-se dos identificadores do membro e de sua estação. Sempre que um membro entra ou sai da sessão a lista é atualizada.

- Lista de programas do espaço compartilhado - cada nó desta lista contém os seguintes campos:
  - \* *sock* - identificador do *socket* de comunicação com o servidor gerente;
  - \* *pid* - identificador do processo que deu origem aos processos que compõem o programa;
  - \* *hostname* - estação na qual executa o programa;
  - \* *command* - linha de comando que originou o programa;
  - \* *type* - tipo do programa;
- Lista de espera de usuários - guarda informações sobre usuários não convidados que solicitaram permissão para entrar na sessão de cooperação. A qualquer momento o gerente pode consultar a lista e convidar os usuários de seu interesse.

• Módulo *sistrnc\_proc*

Este módulo contém todos os procedimentos correspondentes aos serviços oferecidos pelo sistema gerenciador de sessão. A seguir descrevemos cada serviço:

- Criação de Sessão - este serviço é implementado pelo procedimento *create\_1*. O procedimento recebe como parâmetros o título e o modo da sessão, o identificador do gerente e de sua estação. Atualiza a estrutura que armazena as informações relativas à sessão e o arquivo de sessões.
- Convite de Usuário - o procedimento *invitation\_1* implementa este serviço. Os parâmetros deste procedimento são: lista de convidados e identificador da sessão. O procedimento utiliza as rotinas do módulo *convite* para a localização dos usuários e exibição de convite em seus terminais.
- Execução de programas - este serviço é implementado pelo procedimento *execprog\_1*. Os parâmetros do procedimento são uma estrutura contendo informações sobre o programa a ser executado (linha de comando, estação e tipo do programa), geometria da janela de entrada e saída no caso de programas monousuário compartilhados, ambiente e identificador do usuário solicitante. O serviço consiste da verificação do modo da sessão, da execução do programa e da atualização da lista de programas.

O modo democrático permite que qualquer membro da sessão execute programas, enquanto o modo autocrático autoriza apenas o gerente.

Na execução propriamente dita do programa, caso o programa seja monousuário compartilhado, é executado um processo controlador passando-lhe como parâmetros a geometria da janela de entrada e saída, a lista de membros efetivos, a linha de comando do programa a ser compartilhado e o ambiente do usuário solicitante. No caso de outros programas multiusuário, um processo é executado utilizando-se a linha de comando passada como parâmetro. Em ambos os casos, é estabelecido o canal de comunicação entre o programa e servidor gerente de sessão.

- Entrada de Membro - o procedimento *entermember\_1* implementa este serviço. Os parâmetros deste procedimento são os identificadores da sessão, do usuário e de sua estação, além do *status* do solicitante. A nível de processo cliente verifica-se se o usuário é convidado ou não e essa informação estabelece o seu *status*. O serviço consiste da verificação do modo da sessão e da integração ou não do usuário solicitante. Além disso, uma mensagem é enviada para todos os programas do espaço compartilhado comunicando o evento. No caso de uma sessão de modo público duas atitudes distintas são tomadas de acordo com o *status* do usuário. Se o usuário é convidado ele é integrado a sessão e todos os outros participantes são informados da entrada do novo membro. Se o usuário não é convidado, o gerente é informado de seu desejo de entrar na sessão. O usuário é colocado numa lista de espera. Neste caso, a inclusão do novo membro dependerá da vontade do gerente.
- Obtenção de Informações - este serviço é implementado pelo procedimento *getinfo\_1*. O parâmetro do procedimento é o identificador da sessão e o identificador do usuário solicitante. Se a sessão tiver modo secreto apenas membros têm acesso as informações. O procedimento devolve como resultado uma estrutura contendo informações completas sobre a sessão. A exibição destas informações fica a cargo do procedimento que invocou o serviço.
- Alteração do Modo de Operação - este serviço é implementado pelo procedimento *change mode\_1*. O processo cliente encarrega-

se de consultar o arquivo de sessões para verificar se o usuário solicitante é o gerente. O serviço consiste de simples alteração da estrutura referente a sessão.

- Eliminação de Membro - *dismissmember\_1* é o procedimento que implementa este serviço. Somente o gerente pode eliminar um membro de uma sessão. O processo cliente encarrega-se de consultar o arquivo de sessões para verificar se o usuário solicitante é o gerente. O procedimento do servidor recebe como parâmetros os identificadores da sessão e do membro a ser eliminado. O serviço consiste da verificação se usuário a ser eliminado é realmente membro da sessão e em caso afirmativo na alteração da lista de membros. O evento da eliminação de um usuário é comunicado a todos os programas do espaço compartilhado.
- Saída de Membro - este serviço é implementado pelo procedimento *leaveession\_1*. O procedimento tem como parâmetros os identificadores da sessão e do usuário que deseja sair da sessão. O processo cliente encarrega-se de verificar se o usuário solicitante é o gerente. Em caso afirmativo, dois serviços são solicitados: *alteração de gerente e saída de membro*. O serviço de saída de membro consiste da verificação se o usuário é o membro da sessão e, em caso afirmativo, da eliminação de usuário da lista de membros efetivos. Se o número de membros reduzir-se a zero, o serviço de encerramento da sessão é invocado. Além disso, o evento é comunicado a todos os programas do espaço compartilhado.
- Alteração do Gerente - este serviço é implementado pelo procedimento *changemanager\_1*. Os parâmetros do procedimento compõe-se dos identificadores da sessão e do membro substituto. O processo cliente encarrega-se de verificar se o usuário solicitante é o gerente da sessão. O serviço consiste da alteração da informação referente ao usuário gerente contido na estrutura que armazena informações sobre sessão. O servidor gerente da sessão permanece o mesmo.
- Encerramento da Sessão - o procedimento *endsession\_1* implementa este serviço. O parâmetro do procedimento é o identificador da sessão. O processo cliente encarrega-se de verificar se o usuário solicitante é o gerente, consultando o arquivo de sessão. O serviço consiste da eliminação do nó da lista de sessão referente a mesma e o envio da mensagem para todos os programas

do espaço compartilhado.

- Impressão de Mensagens - este serviço é implementado pelo procedimento *print\_I*, que tem como parâmetro a mensagem a ser impressa. É responsável pelo *Sistrac Console*. Quando uma mensagem deve ser impressa, a janela do *Sistrac Console* é criada (caso não exista) e a mensagem é exibida. Uma *shell* executa nesta janela, possibilitando, além da exibição de mensagens, a execução dos comandos do *Sistrac*.

- Módulo *conmit*

Neste módulo estão todas as rotinas relacionadas com a localização e exibição de convite para usuários.

## 5.2 Sistema de Compartilhamento de Programas Monousuário - SCPM

A implementação do módulo de compartilhamento de programas monousuário envolve o conceito de *pseudo-terminal* do Unix ([Cou 85], [Ref 90]). Um pseudo-terminal é realmente um par de dispositivos, mestre e escravo, que possibilitam a um processo atuar como agente de comunicação entre processos e usuários. Dados escritos no lado escravo do pseudo-terminal, são enviados como entrada para o processo que está lendo no lado mestre. Assim, o processo manipulando o lado mestre do pseudo-terminal tem controle sobre a informação lida ou escrita para o lado escravo, como se ele estivesse manipulando o teclado e lendo a tela de um terminal real.

Para cada programa compartilhado há dois processos controladores, um processo referente ao programa compartilhado, processos responsáveis pela leitura de teclado e exibição da saída do programa e processos que tratam os comandos de controle (veja figura 5.2).

Todos os processos executam na mesma estação e utilizam o *X Protocol* para transmissão de dados pela rede. A figura 5.2 ilustra a arquitetura do sistema de compartilhamento de programas. O SCPM foi construído de forma a poder executar tanto dentro do ambiente gerenciado pelo SGS como isoladamente. Quando é executado no ambiente de cooperação do *Sistrac*, a interface proporcionada pela janela de controle é incorporada ao *Sistractool*. Isoladamente, o SCPM exibe duas janelas por usuário:

- Janela de entrada e saída: através desta janela, é mostrada a saída do



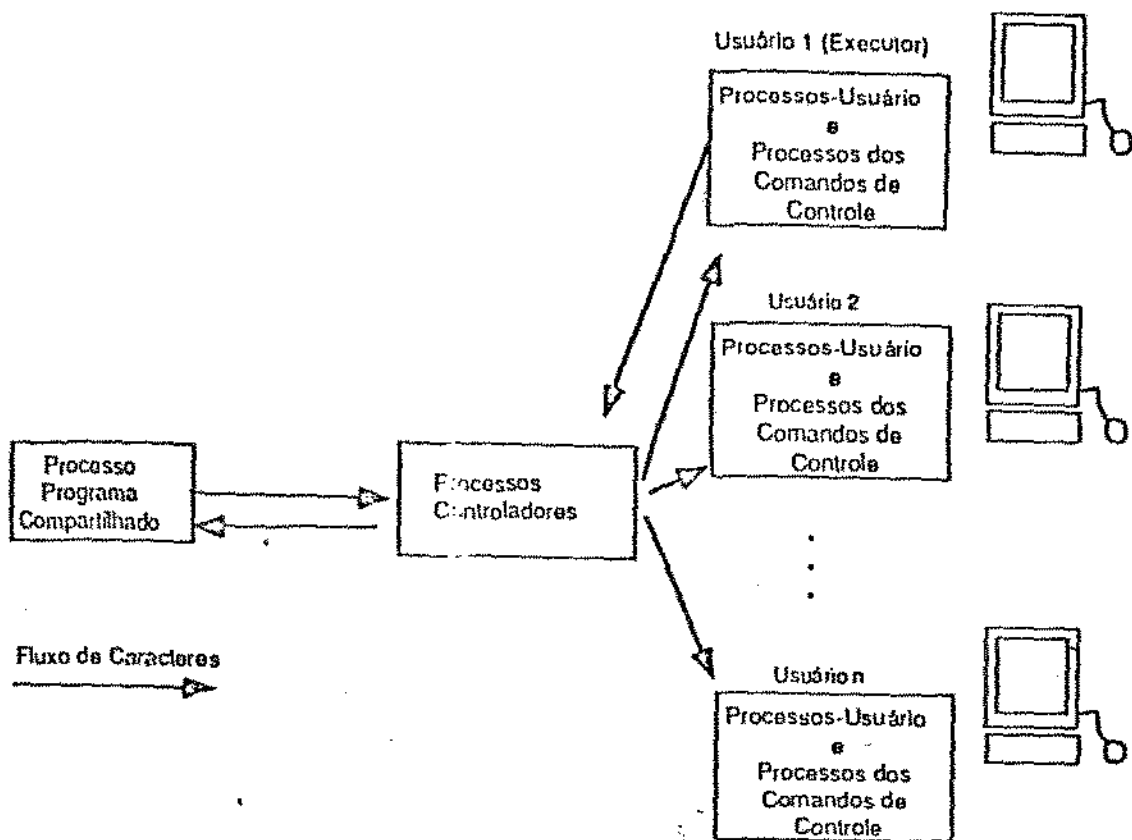


Figura 5.2: Sistema Compartilhador de Programas Monousuário

programa compartilhado. Cada usuário vê a mesma janela, O executor tem direito de entrar com comandos ou dados para o programa.

- Janela de controle: através desta janela o usuário pode entrar com comandos referentes ao gerenciamento do direito de escrita e troca de mensagens. Os comandos implementados são: liberação e solicitação do direito de escrita e troca de mensagens;

A execução de um programa compartilhado implica inicialmente na execução de um processo controlador. Este processo recebe os seguintes parâmetros:

- executor inicial;
- geometria da janela de entrada e saída;
- lista de usuários e respectivas estações;
- identificadores da porta e estação que compõem o endereço *Internet* do *socket* de comunicação com o servidor gerente da sessão da qual faz parte o programa. Estes parâmetros estão presentes somente quando o programa compartilhado é executado dentro do ambiente gerenciado pelo SGS;
- ambiente do usuário solicitante;

O processo controlador aloca um pseudo-terminal e executa o processo programa compartilhado, além dos outros programas que compõem o SCPM. A seguir são descritos em detalhes cada um dos processos:

- **Processos Controladores**

São dois os processos controladores por programa compartilhado (veja figura 5.3). O primeiro processo é responsável, pela alocação do pseudo-terminal e execução dos outros processos. Além disso, o primeiro processo controlador lê do canal de comunicação com um dos processos-usuário os caracteres digitados pelo executor, e os envia para o processo programa. O processo controlador 1 ainda trata as mensagens provenientes do servidor gerente da sessão.

O segundo processo controlador lê a saída do programa compartilhado e replica esta saída para todos os processos-usuário 1.

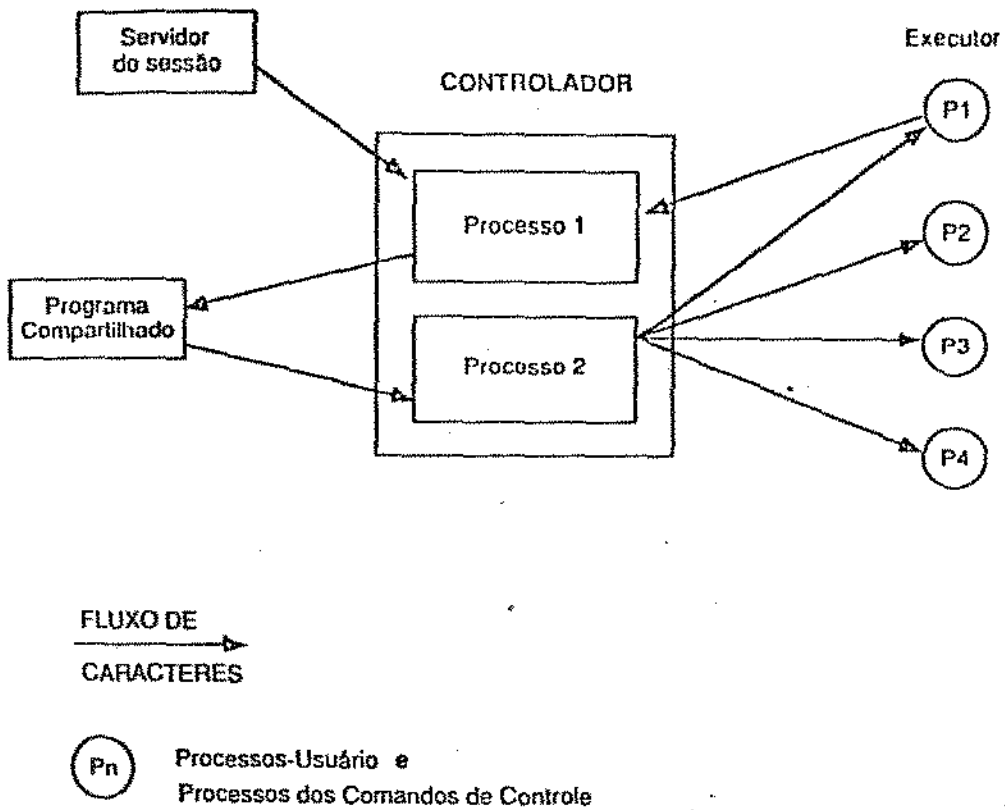


Figura 5.3: Processos Controladores: o primeiro lê caracteres do canal de comunicação com processos-usuário do executor e os envia ao processo programa compartilhado. Além disso trata as mensagens do servidor gerente da sessão. O segundo processo lê a saída do programa compartilhado e a replica para todos os processos-usuário

- **Processos-usuário**

Estes processos são responsáveis pela leitura do teclado do usuário e exibição da saída do programa compartilhado (veja figura 5.4). Cada janela de entrada e saída corresponde a um terminal *xterm*<sup>4</sup>. O modo do terminal é alterado, de forma que não haja pré-processamento de entrada ou saída. Assim, o processo-usuário 2 do executor, lê os caracteres digitados pelo usuário e os envia para o processo controlador 1. O processo-usuário 1 lê a saída do programa compartilhado, proveniente do processo controlador 2, e a exibe na janela de entrada e saída. Somente o processo-usuário 2 do executor está ativo, isto é, está lendo o teclado do usuário. A alteração do executor implica no envio de mensagem do processo controlador para processos-usuário do antigo e novo executor e processos dos comandos de controle de todos os usuários.

- **Processos dos Comandos de Controle**

Estes processos são responsáveis pelos comandos de liberação e requisição do direito de escrita e troca de mensagens. Eles lêem e tratam os comandos da janela de controle e exibem as mensagens de controle referentes ao programa compartilhado (veja figura 5.5).

O processo controlador 1 mantém uma lista de usuários que solicitaram o direito de escrita. Quando um usuário requisita ou libera o direito de escrita, uma mensagem é enviada ao processo controlador 1. Este processo é interrompido para tratar o evento. O usuário solicitante é colocado na lista de espera ou passa a ser  $\delta$ -executor no caso de não haver executor naquele momento. Se o usuário está liberando o direito de escrita, o primeiro da lista de espera ou sucessor especificado na linha de comando passa a ser o executor.

Sempre que há alteração de executor os processos-usuário e processos dos comandos de controle recebem mensagem do processo controlador 1. O protocolo de comunicação entre estes processos consiste de uma estrutura com os seguintes campos:

- controle: identificador de evento;
- usuário: identificador do usuário solicitante;
- mensagem: texto que deve ser exibido no caso de troca de mensagens;

---

<sup>4</sup>emulador de terminal X Window

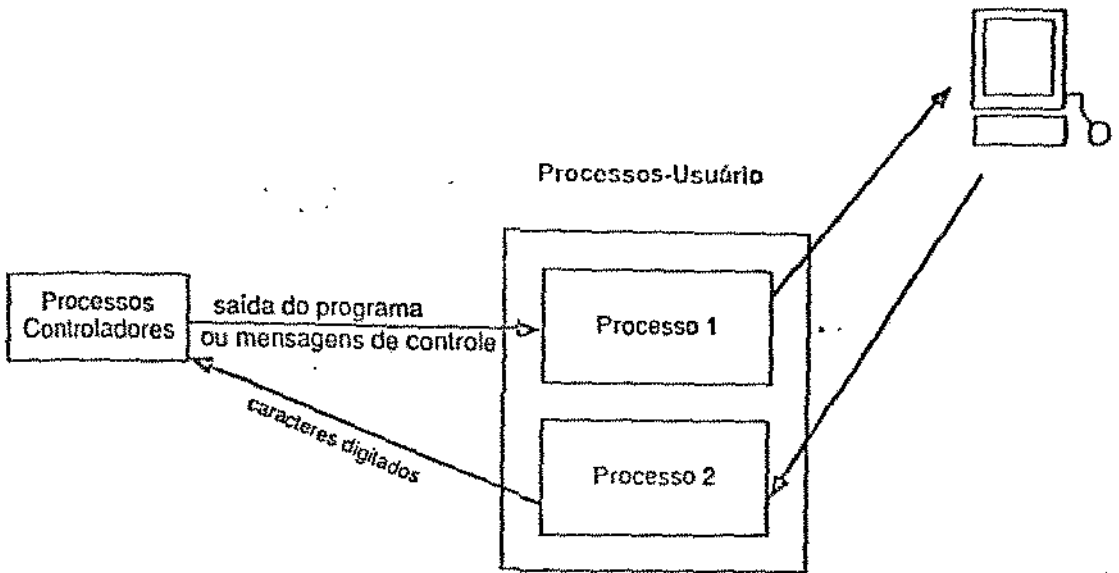
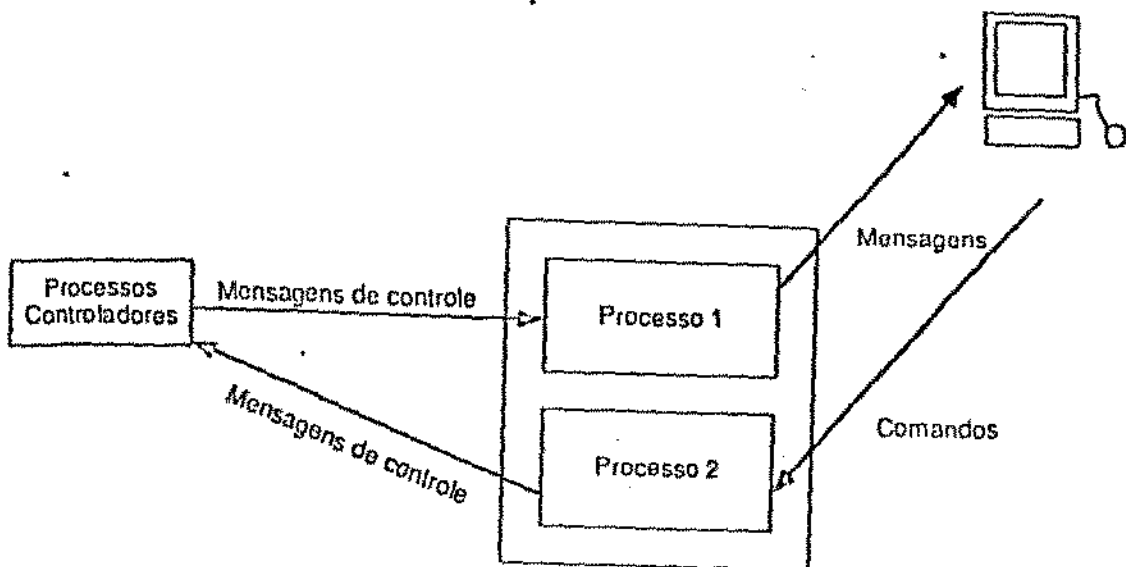


Figura 5.4: Processos-usuário: cada usuário tem dois processos-usuário associados. O primeiro exibe a saída do programa compartilhado e trata as mensagens de controle provenientes dos processos controladores. O segundo, caso o usuário seja o executor, lê o teclado e envia os caracteres lidos para processo controlador.



Processos dos Comandos de Controles

Figura 5.5: Processos dos Comandos de Controle

### 5.2.1 Problemas de Implementação

Processos Unix convencionalmente têm acesso a três arquivos: eles lêem de seu arquivo de entrada padrão, escrevem sua saída no seu arquivo de saída padrão e escrevem mensagens de erro no seu arquivo de erro padrão. Estes arquivos estão normalmente associados ao terminal de controle do processo (entrada — teclado, saída — tela). A idéia inicial de implementação era redirecionar os descritores padrões de entrada e saída (descritores 0, 1, 2) do processo programa antes de executá-lo. Estes descritores passavam a ser pipes de comunicação com um dos processos controladores. Essa abordagem trazia sérias restrições. O processo controlador tinha que assumir o papel dos módulos de disciplina de linha [Bac 86] interpretando a entrada e saída de dados. Além disso, era necessário capturar as alterações feitas nos parâmetros do terminal. Essas alterações são normalmente efetuadas através da chamada ao sistema *ioctl*. Um dos parâmetros do *ioctl* é o descritor do arquivo sobre o qual pretende-se efetuar a operação. Quase todos os programas alteram os parâmetros do terminal, através dos descritores padrões [Bac 86]. Assim, o simples redirecionamento dos descritores de entrada e saída não possibilitava o compartilhamento de programas que alterassem os

parâmetros do terminal ou redirecionassem a entrada e/ou saída padrões. A solução encontrada foi a utilização de pseudo-terminal. O SCPM aloca um pseudo-terminal por programa. Para o programa é absolutamente transparente se ele está interagindo com o usuário via tela e teclado ou com um processo. Os processos controladores escrevem e lêem no lado mestre e o programa compartilhado escreve e lê do lado escravo do pseudo-terminal. Desta forma, todas as alterações dos parâmetros do terminal são efetuadas no dispositivo escravo. Os processos controladores, ao escreverem no dispositivo mestre, exercem o papel de um usuário ao pressionar o teclado, ao mesmo tempo que atua como dispositivo de saída. Não há necessidade de se fazer nenhum pré-processamento de entrada e/ou saída, pois isso já é feito pelo módulo de disciplina de linha associado ao dispositivo escravo.

### 5.3 Interação entre o SCPM e o SGS

O SGS comunica a cada um dos programas do espaço compartilhado os eventos relacionados com a sessão de cooperação da qual fazem parte:

- Entrada de membro;
- Saída ou eliminação de membro;
- Encerramento de sessão;

As mensagens entre o SCPM e SGS consistem numa estrutura com os seguintes campos:

- controle: indica o tipo de evento;
- usuário: identifica o usuário que entrou ou saiu da sessão;
- estação: identifica a estação do referido usuário;

O tratamento dos eventos pelo SCPM é feito por interrupção e é descrito abaixo:

**Entrada de membro:** o SCPM mantém um arquivo por programa compartilhado que armazena a saída de cada um destes programas. Quando um novo usuário ingressa na sessão, os processos e janelas referentes ao usuário são criados. O arquivo associado ao programa é exibido de forma que o novo membro tenha visão idêntica da janela de entrada e saída do programa compartilhado;

**Saída ou eliminação de membro:** todas as janelas e processos relativos ao usuário são encerrados;

**Encerramento da sessão:** todos os processos relacionados com o programa compartilhado são encerrados



## Capítulo 6

# Conclusão

Foi projetado e implementado um sistema para suporte de trabalho de grupo. Ele consiste de um compartilhador de programas monousuário e um módulo para a administração da atividade coletiva. A partir deste segundo módulo, programas multiusuário podem ser incorporados.

O protótipo foi desenvolvido para redes locais de estações de trabalho Unix com sistema X Window.

O SCPM é capaz de compartilhar qualquer programa projetado para uso em terminais alfanuméricos. Nesta categoria encontram-se a maioria dos programas tradicionais do Unix como *vi*, *emacs*, *cshell* ou *mail*. O suporte a aplicações que utilizam recursos gráficos do sistema X Window é altamente desejável, constituindo-se uma das extensões do corrente trabalho. O SCPM foi projetado de forma que o compartilhamento fique totalmente transparente aos programas compartilhados, isto é, o código do programa não precisa sofrer nenhuma alteração. Além das restrições dos recursos gráficos mencionadas acima, um programa que utiliza recursos de áudio também perderá parte de sua funcionalidade ao ser compartilhado, pois produzirá saída de áudio apenas em uma das estações envolvidas no compartilhamento.

O gerenciador de sessão foi implementado utilizando-se tecnologias novas de sistemas distribuídos. É o núcleo do ambiente de cooperação. Deve ser ampliado a partir do desenvolvimento de novas aplicações multiusuário. É possível que novas operações sejam definidas, que se estabeleçam novos modos de operação para as sessões e a estrutura social seja enriquecida com o surgimento de novos elementos como, por exemplo, sub-gerente (usuários com alguns dos privilégios e atribuições dos gerentes).

A interface do compartilhador de programas do Sistrac compõe-se de

duas janelas *xterm*. A interface do sistema como um todo (Sistractool) foi especificada e deve ser implementada utilizando-se os padrões OpenLook ou Motif.

O compartilhador de programas do Sistrac está funcional desde de abril de 1991 para estações Sun. O gerenciador de sessão tem quase todas as suas operações implementadas. O código do Sistrac foi escrito na linguagem C e tem aproximadamente 7000 linhas de código. Foi implementado em SunOs utilizando essencialmente recursos do BSD Unix. O seu porte para o SYS V.4 deve ser bastante simples uma vez que o SYS V.4 incorpora a funcionalidade do BSD Unix utilizada no Sistrac.

## 6.1 Expectativas e Extensões

O desenvolvimento de programas multiusuário deve completar o ambiente de cooperação Sistrac. Todos os programas multiusuário devem utilizar o módulo gerenciador de sessão como administrador das sessões de cooperação. A primeira versão do gerenciador deve passar por uma bateria de testes quanto a interface e protocolo de comunicação a fim de que sejam avaliados aspectos relacionados com o comportamento humano no trabalho de grupo.

O programa de grupo mais importante para o ambiente de cooperação será o sistema de conferência por computador [Yam 91]. Este sistema proporcionará aos membros de uma sessão facilidades de comunicação por texto e voz, essenciais em atividades cooperativas. O sistema de conferência funcionará em estações de trabalho ou terminais alfanuméricos. Além disso, permitirá a gravação da conferência para edição posterior.

Outro produto de grupo que deve ser incorporado ao Sistrac é uma agenda de equipe. Esta agenda deve entender o conceito hierárquico de projeto com uma estrutura social composta por coordenadores, sub-coordenadores e membros. A cada um desses elementos devem ser atribuídos direitos diferenciados. Os compromissos agendados podem ser individuais ou de projeto. Dependendo do *status* do usuário, compromissos de projeto podem se sobrepor a compromissos individuais.

Quanto ao compartilhador de programas, uma extensão importante seria o compartilhamento de programas que utilizassem recursos gráficos. Em programas orientados a texto a interação com o terminal do usuário ocorre através de fluxo de caracteres. Aplicações gráficas, especificamente aplicações X Window, interagem com a estação gráfica através de fluxo de requisições [Jon 89]. Cada estação executa um programa X servidor da rede.

O servidor aceita solicitações de conexão de programas de aplicação (clientes). Através desta conexão o servidor recebe e responde a requisições (por exemplo, requisições para criar uma janela ou desenhar uma reta) dos programas. O servidor também usa a conexão para enviar eventos (como o acionamento do botão do *mouse*) para as aplicações que se interessem por eles. Deve ser estudado a viabilidade de se construir um sistema que permita o compartilhamento de programas X.

Enfim, os módulos do Sistrac hoje implementados constituem o ponto de partida para o desenvolvimento de um ambiente de colaboração. A partir da experiência adquirida com os protótipos, versões futuras poderão incorporar características novas para adaptá-los a atividade coletiva.

# Apêndice A

## Comandos do Sistrac

Este apêndice apresenta a sintaxe dos comandos do Sistrac. A seguinte notação foi utilizada:

[ ] opcional

< > símbolo não terminal

— alternativas

... repetição

{ } opções, uma das quais é obrigatória

### COMANDO

```
sistrac -c [<título>] [-l <lista_de_convitados>] [-m <modo_de_operacao>]
```

### DESCRIÇÃO

Esse comando cria uma sessão de cooperação. O argumento da opção *-c* e as opções *-l* e *-m*, quando omitidas, fazem com que o sistema solicite do usuário o título, a lista de usuários convidados e o modo de operação da sessão de cooperação. A criação de uma sessão de cooperação consiste da inicialização das estruturas de dados referentes à sessão e do envio de convites.

## OPÇÕES

-c [<título>]

Especifica o título da sessão de coperação.

-l [<lista\_de\_convitados>]

Especifica a lista de convidados no formato: usuário1@usuário2@...

-m [<modo\_de\_operação>]

Especifica o modo de operação da sessão. O modo de operação é dado pelos seguintes valores numéricos:

1. Sessão secreta e democrática
2. Sessão secreta e autocrática
3. Sessão pública e democrática
4. Sessão pública e autocrática

## ARQUIVOS

/etc/sessionfile - arquivo que contém informações sobre sessões correntes. A estrutura esta definida em <file.e> e compõe-se dos seguintes campos:

identificador da sessão

título da sessão

estação do servidor gerente da sessão

nome do gerente

lista de convidados

/etc/hostfile - arquivo que contém os identificadores das estações componentes da rede.

## COMANDO

sistrac -i [<identificador\_da\_sessao> <lista\_de\_convitados>]

## DESCRIÇÃO

Esse comando convida uma lista de usuários para participar de uma sessão de cooperação. Cada usuário da lista de convidados é localizado e um convite em forma de ícone é mostrado na sua tela. Caso o usuário não esteja conectado a rede, o convite é registrado no arquivo de sessões, o que possibilita listagem de convites recebidos.

## OPÇÕES

-i [<identificador\_da\_sessao> <lista\_de\_convitados>]

Essa opção tem dois argumentos: o primeiro -identificador da sessão- é um inteiro que especifica a sessão para qual os usuários estão sendo convidados; o segundo -lista\_de\_convitados- especifica os usuários a serem convidados. A lista de convidados deve apresentar os nomes dos usuários separados por '@'.

## ARQUIVOS

/etc/sessionfile - este arquivo é atualizado.

## COMANDO

sistrac -L

## DESCRIÇÃO

Esse comando permite ao usuário listar as sessões correntes. As informações apresentadas consistem dos seguintes itens:

- identificador da sessão
- título da sessão
- nome do gerente
- estação do servidor gerente
- lista de convidados

## ARQUIVOS

/etc/sessionfile - este arquivo é consultado.

## COMANDO

sistrac -I

## DESCRIÇÃO

Esse comando permite ao usuário listar as sessões para as quais foi convidado. As informações apresentadas consistem dos seguintes itens:

- identificador da sessão
- título da sessão
- nome do gerente
- estação do servidor gerente

## ARQUIVOS

/etc/sessionfile - este arquivo é consultado.

## COMANDO

sistrac -o {ALL—identificador\_da\_sessão}

## DESCRIÇÃO

Mostra informações completas sobre sessões correntes. Essas informações consistem dos seguintes itens:

- Identificador da sessão
- Título da sessão
- Nome do gerente
- Estação do servidor gerente
- Lista de membros efetivos
- Data e horário do início da sessão
- Modo da sessão
- Programas que fazem parte do espaço compartilhado

## OPÇÕES

-o {ALL—<identificador\_da\_sessão>}

Se o argumento dessa opção for ALL, então informações sobre todas as sessões são apresentadas. Caso o argumento seja o identificador de uma sessão, informações específicas daquela sessão são exibidas.

## COMANDO

sistrac -e [<identificador\_da\_sessão>]

## DESCRIÇÃO

Esse comando encerra uma sessão de cooperação. Todos os membros da sessão são avisados através do *Sistrac Console* e os processos e janelas do espaço compartilhado são encerrados.

## OPÇÕES

-e [<identificador\_da\_sessão>]

O argumento desta opção especifica a sessão que deve ser encerrada. Caso a linha de comando não contenha o identificador da sessão, este é solicitado ao usuário.

## ARQUIVOS

/etc/sessionfile - este arquivo é atualizado



## COMANDO

sistrac -s [<identificador\_da\_sessão>]

## DESCRIÇÃO

Esse comando permite ao usuário sair da sessão, caso não seja o gerente. Os outros membros da sessão são informados através de mensagem no *Sistrac Console*.

## OPÇÕES

-s [<identificador\_da\_sessão>]

O argumento desta opção especifica a sessão da qual usuário pretende sair. Caso a linha de comando não contenha o identificador da sessão, este é solicitado ao usuário.

## COMANDO

sistrac -n [<identificador\_da\_sessão>]

## DESCRIÇÃO

Esse comando permite que um usuário convidado ou não solicite seu ingresso numa sessão de cooperação. Caso o usuário seja convidado, ele automaticamente é incorporado a sessão e os processos e janelas referentes a todos os programas do espaço compartilhado são criados para o novo membro. Caso o usuário não esteja na lista de convidados, o gerente é informado do seu desejo, cabendo a ele enviar ou não um convite para o mesmo.

## OPÇÕES

-n [<identificador\_da\_sessão>] O argumento dessa opção especifica a sessão da qual o usuário solicitante deseja fazer parte. O argumento quando omitido, é solicitado.

## COMANDO

sistrac -X <identificador\_da\_sessão> [-t <tipo\_de\_programa>] [-g <geometria>] [-x <executer>] -p <command>

## DESCRIÇÃO

Esse comando permite ao usuário executar um programa dentro do espaço compartilhado. Esse programa pode ser um programa monousuário compartilhado ou multiusuário. Para cada membro efetivo da sessão são criados os processos e janelas referentes ao programa. Se o modo da sessão é democrático, todos têm direito de executar programas. Caso o modo da sessão seja autocrático, tal direito pertence apenas ao gerente.

## OPÇÕES

-X <identificador\_da\_sessão>

O argumento desta opção especifica a sessão na qual pretende-se executar um programa.

-t <tipo\_de\_programa>

Esta opção especifica se o programa que vai ser executado é monousuário compartilhado ou um outro programa multiusuário. O argumento <tipo\_de\_programa> é um valor inteiro que pode assumir os seguintes valores:

1. Monousuário compartilhado
2. Multiusuário

Se opção -t for omitida, assume-se que o programa é multiusuário.

-g <geometria>

Essa opção refere-se à geometria da janela de entrada e saída de programas compartilhados. O formato do argumento segue o padrão Xwindows. Se essa opção for omitida assume-se a seguinte geometria: "30x80+0+0".

**-x <executer>**

Essa opção refere-se a programas compartilhados e determina o executor inicial do programa que está sendo executado. Caso esta opção seja omitida, assume-se que o executor é o usuário solicitante.

**-p <command>**

Essa opção determina a linha de comando do programa a ser executado.

## COMANDO

sistrac -a <identificador\_da\_sessão> <mode>

## DESCRIÇÃO

Esse comando possibilita ao gerente alterar o modo de operação da sessão.

## OPÇÕES

-a <identificador\_da\_sessão> <mode>

O argumento <identificador\_da\_sessão> especifica a sessão da qual deseja-se alterar o modo. O argumento <mode> é um valor numérico que determina o novo modo da sessão e pode assumir os seguintes valores:

1. Sessão secreta e democrática
2. Sessão secreta e autocrática
3. Sessão pública e democrática
4. Sessão pública e autocrática

## COMANDO

sistrac -d [<identificador\_da\_sessão> <identificador\_do\_usuario>]

## DESCRIÇÃO

Esse comando possibilita ao gerente eliminar um participante de uma sessão de cooperação.

## OPÇÕES

-d [<identificador\_da\_sessão> <identificador\_do\_usuario>]

O argumento <identificador\_da\_sessão> especifica a sessão da qual deseja-se eliminar o participante. O argumento <identificador\_do\_usuario> identifica o participante a ser eliminado.

## ARQUIVOS

/etc/sessionfile - este arquivo é atualizado

## COMANDO

sistrac -M [<identificador\_da\_sessão> <identificador\_do\_usuario>]

## DESCRIÇÃO

Através desse comando o gerente solicita a sua substituição.

## OPÇÕES

-M [<identificador\_da\_sessão> <identificador\_do\_usuario>]

O argumento <identificador\_da\_sessão> especifica a sessão da qual deseja-se substituir o gerente. O argumento <identificador\_do\_usuario> identifica o novo gerente.

## ARQUIVOS

/etc/sessionfile - este arquivo é atualizado

## Apêndice B

# Funções Exportáveis

Este apêndice apresenta as funções exportáveis do Sistrac que encontram-se na biblioteca `usr/local/lib/sistraclib`.

### NOME

`CreateSession` - criação de uma sessão de cooperação.

### SINOPSE

```
#include <sistrac.g>
int CreateSession(title,mode,invitedlist,logname,hostname)
char *title
unsigned short mode
char *invitedlist
NAMETYPE logname
char hostname[MAXHOSTLEN]
```

### DESCRIÇÃO

Esta função solicita do servidor de sessão que executa na estação especificada por *hostname* a criação de uma sessão de cooperação com título, modo de operação, nome do gerente e lista de convidados dados por *title*, *mode*, *logname* e *invitedlist* respectivamente.

### VALORES DE RETORNO

**SUCESSO** - sucesso  
**ERRHAND** - erro na criação do *handler* do cliente  
**ERRCALISVC** - erro na chamada do servidor  
**ERRALLOC** - erro na alocação de memória

ERROPENFILE - erro na abertura do arquivo de sessões

## NOME

LeaveSession - saída de membro da sessão

## SINOPSE

```
#include <sistrac.g>
int LeaveSession(sesid, logname)
int sesid;
NAMETYPE logname;
```

## DESCRIÇÃO

Esta função requisita do servidor gerente da sessão identificada por *sesid* a saída do usuário solicitante identificado por *logname*.

## VALORES DE RETORNO

SUCESSO - sucesso  
ERRHAND - erro na criação do *handler* do cliente  
ERRCALLSVC - erro na chamada do servidor  
ERRALLOC - erro na alocação de memória  
ERROPENFILE - erro na abertura do arquivo de sessões

## NOME

EnterMember - entrada de usuário na sessão

## SINOPSE

```
#include <sistrac.g>
int EnterMember(newmember, sesid)
MEMBER newmember
int sesid;
```

## DESCRIÇÃO

Esta função permite que um usuário identificado por *newmember* requisite sua entrada na sessão identificada por *sesid*.

## VALORES DE RETORNO

OK - usuário incluído na lista de espera  
SUCESSO - usuário incluído como membro efetivo  
ERRMEMBER - usuário já é membro efetivo  
ERRSECRETSESSION - sessão de modo secreto  
ERRHAND - erro na criação do *handler* do cliente  
ERRCALLSVC - erro na chamada do servidor  
ERRALLOC - erro na alocação de memória  
ERRLOCKFILE - erro no bloqueio de registro no arquivo de sessão  
ERRINVRECORD - registro do arquivo de sessões inválido  
ERROPENFILE - erro na abertura do arquivo de sessões



## NOME

InviteMember - convite de usuário

## SINOPSE

```
#include <sisrac.g>
int InviteMember(invitedlist,logname,sesid)
char *invitedlist
NAMETYPE logname
char *hostname
int sesid
```

## DESCRIÇÃO

Esta função permite ao gerente de uma sessão em andamento convidar uma lista de usuários a participarem da mesma. O parâmetro *invitedlist* deve ter o seguinte formato: usuário1@usuário2@... O parâmetro *logname* identifica o usuário solicitante. Este usuário deve ser o gerente. Caso contrário, a função retornará um código de erro. O parâmetro *sesid* identifica a sessão de cooperação.

## VALORES DE RETORNO

SUCESSO - usuários convidados  
ERRNOTMANAGER - usuário não é gerente da sessão  
ERRALLOC - erro na alocação de memória  
ERROPENFILE - erro na abertura do arquivo de sessões

## OBSERVAÇÕES

Os convites são enviados para todos os membros da lista de convidados que estiverem conectados à rede. O arquivo de sessões é atualizado de forma que quando um usuário se conecta à rede, pode listar todos os convites dirigidos a ele.

## NOME

GetInformation - obtenção de informações

## SINOPSE

```
#include <sistrac.g>
session_res *GetInformation (int sesid, char *hostname)
int sesid
char *hostname
```

## DESCRIÇÃO

Esta função permite aos usuários obter informações completas sobre uma sessão de cooperação. A estrutura *session\_res* está definida em "sistrac.h".

## VALORES DE RETORNO

SUCESSO - sucesso

ERRNOTEXISTSESSION - sessão inexistente

ERROPENFILE - erro na abertura do arquivo

ERRALLOC - erro na alocação de memória

ERRHAND - erro na criação do *handler* do cliente

ERRCALLSVC - erro na chamada do servidor

## NOME

ChangeMode - alteração do modo da sessão

## SINOPSE

```
#include <sisrac.g>
int ChangeMode(int sesid, unsigned short mode)
int sesid
unsigned short mode
```

## DESCRIÇÃO

Esta função permite ao gerente da sessão de cooperação identificada por *sesid* alterar o modo da sessão. O parâmetro *mode* identifica o novo modo da sessão e pode assumir os seguintes valores:

Secreta e Democrática : 0x00

Pública e Democrática : 0x0F

Pública e Autocrática : 0xFF

Secreta e Autocrática : 0xF0

## VALORES DE RETORNO SUCESSO - Sucesso

ERRNOTEXISTSESSION - sessão inexistente

ERROPENFILE - erro na abertura do arquivo

ERRALOC - erro na alocação de memória

ERRHAND - erro na criação do *handler* do cliente

ERRCALLSVC - erro na chamada do servidor

## NOME

EndSession - Encerramento de sessão

## SINOPSE

```
#include <sistrac.g>
```

```
int EndSession (sesid,logname)
```

```
int sesid
```

```
char *logname
```

## DESCRIÇÃO

Esta função possibilita ao gerente da sessão identificada por *sesid* encerrá-la. O parâmetro *logname* identifica o usuário solicitante.

## VALORES DE RETORNO

SUCESSO - sucesso

ERRNOTEXISTSESSION - sessão inexistente

ERRNOTMANAGER - usuário não é gerente

ERRALLOC - erro na alocação de memória

ERROPENFILE - erro na abertura do arquivo

ERRHAND - erro na criação do *handler* do cliente

ERRCALLSVC - erro na chamada do servidor

## NOME

DismissMember - elimina usuário da sessão

## SINOPSE

```
#include <sisrac.g>
```

```
int DismissMember(sesid,logname, loguser)  
int sesid;  
NAMETYPE logname;
```

## DESCRIÇÃO

Esta função possibilita ao gerente da sessão eliminar um membro identificado por *loguser*. O usuário solicitante é identificado por *logname* e a sessão por *sesid*.

## VALORES DE RETORNO SUCESSO - sucesso.

ERRNOTMANAGER - usuário solicitante não é gerente

ERRNOTMEMBER - usuário a ser expulso não é membro da sessão

ERRALLOC - erro na alocação de memória

ERROPENFILE - erro na abertura do arquivo

ERRHAND - erro na criação do *handler* do cliente

ERRCALLSVC - erro na chamada do servidor

## NOME

ExecProg - executa programas em uma sessão

## SINOPSE

```
#include <sisrac.g>
```

```
int ExecProg(prog,geom,sesid,enviro)  
TOOL prog  
char *geom  
int sesid  
char **enviro
```

## DESCRIÇÃO

Através dessa função programas podem ser executados no espaço compartilhado de uma sessão de cooperação. Na estrutura TOOL são armazenadas as informações relativas aos programas. O parâmetro *geom* refere-se à geometria da janela de entrada e saída do programa, caso ele seja monousuário compartilhado (deve ser *NULL* no caso de aplicativo multiusuário). O parâmetro *sesid* identifica a sessão e o parâmetro *enviro* corresponde ao ambiente do usuário solicitante. O ambiente deve ser *NULL* se o programa não for monousuário compartilhado.

## VALORES DE RETORNO SUCESSO - sucesso.

ERREXEC - erro na execução do programa

ERRPERMDENIED - usuário não tem direito de executar programas (modo de operação autocrático)

ERRALLOC - erro na alocação de memória

ERROPENFILE - erro na abertura do arquivo

ERRHAND - erro na criação do *handler* do cliente

ERRCALLSVC - erro na chamada do servidor

## NOME

Interface - interface servidor de sessão - programa

## SINOPSE

```
#include <sisstrac.g>
```

```
Interface(smporta, smhost)  
int smporta  
char smhost[MAXHOSTLEN]
```

## DESCRIÇÃO

Esta função estabelece o canal de comunicação entre servidor de sessão e um programa que seja executado dentro de uma sessão de cooperação. Os parâmetros *smporta* e *smhost* especificam a porta do *socket* de leitura do servidor de sessão e a estação onde executa o servidor respectivamente.

## VALORES DE RETORNO

descritor do *socket* - sucesso.

ERRSOCKET - erro na abertura ou conexão de *socket*

## Apêndice C

# Mensagens do Sistrac

Este apêndice apresenta as mensagens do sistema. Essas mensagens são impressas no *Sistrac Console* pelo serviço de impressão de mensagens do sistema gerenciador de sessões.

### Mensagem:

Session <session-id> was created

Operação: criação de sessão

Destino: participante que solicitou criação da sessão

Significado: criação da sessão identificada por *session-id* foi bem sucedida

### Mensagem:

<login name> wants to be a member of <session-id>

Operação: entrada de membro

Destino: gerente da sessão

Significado: um usuário não convidado identificado por *login name* solicitou sua integração à sessão identificada por *session-id*.



Mensagem:

<login name> became an effective member of session <session-id>

Operação: entrada de membro

Destino: todos os participantes da sessão identificada por *session-id*

Significado: o usuário identificado por *login name* convidado para a sessão *session-id* se integrou a mesma.

Mensagem:

<command line> was executed in session <session-id>

Operação: execução de programas

Destino: todos os participantes da sessão identificada por *session-id*

Significado: um programa foi executado.

Mensagem:

The mode of session <session-id> was changed: <old mode>  
- <new mode>

Operação: alteração do modo da sessão

Destino: todos os participantes da sessão identificada por *session-id*

Significado: o modo da sessão foi alterado

Mensagem:

<login name> was dismissed from session <session-id>

Operação: eliminação de usuário

Destino: todos os participantes da sessão identificada por *session-id*

Significado: participante identificado por *login name* foi eliminado da sessão identificada por *session-id*.

Mensagem:

`<login name> is the new manager of session <session-id>`

Operação: alteração de gerente

Destino: todos os participantes da sessão identificada por *session-id*

Significado: houve alteração do gerente da sessão.

Mensagem:

`Session <session-id> will be finished.`

Operação: encerramento da sessão.

Destino: todos os participantes da sessão identificada por *session-id*

Significado: sessão identificada por *session-id* será encerrada.

Mensagem:

`<login name> left from session <session-id>`

Operação: saída de membro

Destino: todos os participantes da sessão identificada por *session-id*

Significado: participante identificado por *login name* saiu da sessão identificada por *session-id*.

# Bibliografia

- [Abd 88] Abdel-Wahab, H. M. & Guan, S. & Nievergelt, J., *Shared Workspaces for Group Collaboration: An Experiment Using Internet and Unix Interprocess Communications*, IEEE Communications Magazine, pp 10-16, novembro de 1988.
- [Ahu 88] Ahuja, R. & Ens, R. & Ho, S., *The Rapport Multimedia Conferencing System*. SIGOIS bulletin 9(2 & 3):1-8, abril e julho de 1988.
- [Agl 86] Aguilar, L. & Garcia-Luana-Aceves, J.J. & Moran, D. & Craighill, E. J. & Brungardt, R. *Arquitetura for a Multimedia Teleconferencing System*. SIGCOMM '86 Symposium on Communications Architectures and Protocols, pp 126-136, Agosto de 1986.
- [Bac 86] Bach, J. B., *the Design of the Unix Operating System*. Prentice Hall, Englewood Cliffs, New Jersey, 1986.
- [Bou 87] Bourne, S. R., *The UNIX System V Environment*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.
- [Cou 85] Courington B., *The UNIX System: A Sun Technical Report*. Sun Microsystems, Inc., 1985.
- [Dru 87] Drummond, R. & Liesenberg II., *A\_HAND : Ambiente de Desenvolvimento de Software Baseado em Hierarquias de Abstração em Níveis Diferenciados*. IV Encontro de Trabalho do Projeto ETHOS, Petrópolis, RJ, Abril de 1987.
- [Dru 88] Drummond, R. *LegoShell: Linguagem de Computações* Relatório Interno, Departamento de Ciência da Computação, UNICAMP, 1988.

- [Elw 89] Elwart-Keys, M. & Halonen, D. & Marjorie, H. & Kass, Paul, S., *User Interface Requirements For Face To Face Groupware*. CHI'90 PROCEEDINGS Seattle, Abril de 1990.
- [Eng 88] Engelbart, D. & Lehtman, H., *Working Together*. Byte, 13(13):245-252, dezembro de 1988.
- [Fis 88] Fish, R. S. & Kraut R. E. & Leland M. D. P. & Cohen M., *Quilt: a collaborative tool for cooperative writing*. SIGOIS bulletin, 9(2 & 3):30-37, Abril e Julho de 1988.
- [Fur 91] Furuti, C. A. & Drummond, R., *Um Compilador para uma Linguagem de Programação Orientada a Objetos*. Dissertação de Mestrado Departamento de Ciência da Computação, Unicamp. Campinas, Julho de 1991.
- [Gib 89] Gibbs, S. J., *LIZA: An Extensible Groupware Toolkit*, CHI'89 PROCEEDINGS Austin, Maio de 1989.
- [Gru 88] Grudin, J., *Pcrls and Pitfalls*. Byte, 13(13):261-264, Dezembro de 1988.
- [Jon 89] Jones, J., *Introduction to the X Window System*. Prentice Hall, Englewood Cliffs, New Jersey, 1989.
- [Kra 88] Kraemer, L. K. & King, J. L., *Computer-Based Systems for Cooperative Work and Group Decision Making*. ACM Computing Surveys, 20(2):117-145, Junho de 1988.
- [Lia 90] Liang L. & Chanson S. T. & Neufeld G. W., *Process Groups and Group Communications: Classifications and Requirements*. Computer, 23(2):56-68, Fevereiro de 1990.
- [Man 89] Manson, C. & Thurber, K., *Remote Control*. Byte, pp 235-240, Julho de 1989.
- [Mas 87] Mason, W. A. *Distributed Processing: The State of the Art*. Byte, pp 291-297, Novembro de 1987.
- [Nam 89] Name, M. L. V. & Catchings, B., *Scrvng Up Data*. Byte, pp 259-264, Setembro de 1989.
- [Net 90] Sun Microsystems, *Network Programming Guide*. Sun Microsystems, Inc., Março de 1990.

- [Opp 88] Opper, S., *A Groupware Toolbox*. Byte 13(13):275-280, Dezembro de 1988.
- [Pet 89] Peterson L. L. & Buchholz N. C. & Schlichting R. D., *Preserving and Using Context Information in Interprocess Communication*. ACM Transaction on Computer Systems, 7(3):217-246, Agosto de 1989.
- [Pre 87] Pressman, R. S., *Software Engineering : A Practioner's Approach*, Second Edition, McGraw-Hill Book Co., 1987.
- [Ref 90] Sun Microsystems, *Reference Manual*. Sun Microsystems, Inc., Março de 1990.
- [Sar 85] Sarin, S. & Greif, I. *Computer-Based Real-Time Conferencing Systems*. Computer, 18(10):33-45, Outubro de 1985.
- [Ste 87] Stefik M. & Foster G. & Bobrow D. G. & Kahn K. & Lanning S. & Suchman L., *Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings*. Communications of the ACM, 30(1):32-47, Janeiro de 1987.
- [Sys 90] Sun Microsystems, *System Services Overview*. Sun Microsystems, Inc., Março de 1990.
- [Tan 85] Tanenbaum, A. S. & Renesse R. V., *Distributed Operating Systems*. Computing Survey, 17(4):419-466, Dezembro de 1985.
- [Wal 89] Walker, B. J. & Popek, G. J., *A Transparent Environment*. Byte, pp 225-233, Julho de 1989.
- [Wil 87] Wilbur, B. S. & Young, R. E., *The Cosmos Project : A Multi-Disciplinary Approach to Design For Computer-Supported Group Working*. Department of Computer Science, Queen Mary College, University of London, Mile end Road, London E1 4NS, England, 1987.
- [Win 88] Winograd, T., *Where the Action Is*. Byte 13(13):256A-258, Dezembro de 1988.
- [Yam 91] Yamamoto, L. & Drummond, R., *Sistema de Teleconferência*. Relatório Interno, Departamento de Ciência da Computação, UNICAMP, Abril de 1991.

[Yod 89] Yoder, E. & Aksyn R. & McCracken D., *Collaboration in KMS, A Shared Hypermedia System*. CHI'89 PROCEEDINGS Austin, Mayo de 1989.