

Este exemplar corresponde à redação final da
Tese/Dissertação devidamente corrigida e defendida
por: Paulo Roberto Ferreira Jr.

e aprovada pela Banca Examinadora.

Campinas, 23 de agosto de 00

M. S. Mendes
COORDENADOR DE PÓS-GRADUAÇÃO
CPG-IC

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Agendamento de Reuniões através de
Sistemas Multiagentes

Paulo Roberto Ferreira Jr.

Dissertação de Mestrado

Agendamento de Reuniões através de Sistemas Multiagentes

Paulo Roberto Ferreira Jr.

Julho de 2000

Banca Examinadora:

- Jacques Wainer (Orientador)
- Jaime Simão Sichman
Departamento de Engenharia de Computação
Universidade de São Paulo
- Ariadne Maria Brito Rizzoni Carvalho
- Eliane Martins (Suplente)

UNIDADE	80
N.º CHAMADA	70NICAMP
	F413a
V.	Ex.
TOMBO BC	45595
PROX	16-392101
C	☐
D	☒
PREÇO	R\$11,00
DATA	01/08/01
N.º CPD	

CM00158273-7

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Ferreira Jr., Paulo Roberto

F413a Agendamento de reuniões através de sistemas multiagentes / Paulo Roberto Ferreira Jr. -- Campinas, [S.P. :s.n.], 2000.

Orientador : Jacques Wainer

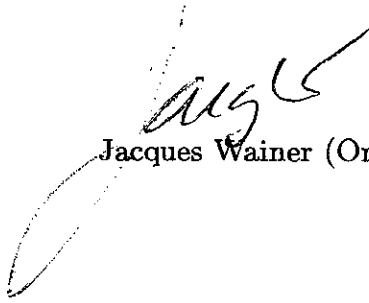
Dissertação (mestrado) - Universidade Estadual de Campinas,
Instituto de Computação.

1. Inteligência artificial distribuída. I. Wainer, Jacques. II.
Universidade Estadual de Campinas. Instituto de Computação. III.
Título.

Agendamento de Reuniões através de Sistemas Multiagentes

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Paulo Roberto Ferreira Jr. e aprovada pela
Banca Examinadora.

Campinas, 30 de junho de 2000.

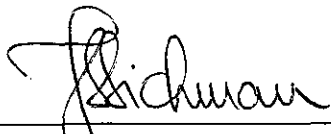


Jacques Wainer (Orientador)

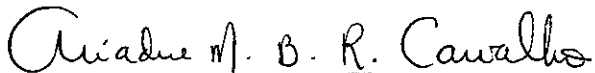
Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

TERMO DE APROVAÇÃO

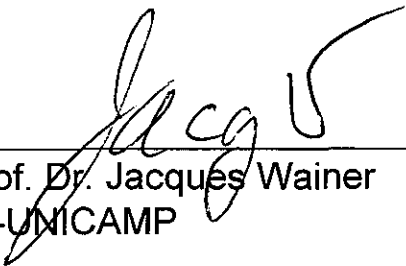
Tese defendida e aprovada em 30 de junho de 2000, pela Banca Examinadora composta pelos Professores Doutores:



Prof. Dr. Jaime Simão Sichman
POLI -USP



Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho
IC-UNICAMP



Prof. Dr. Jacques Wainer
IC-UNICAMP

Prefácio

Esta dissertação apresenta uma abordagem para a solução do problema do agendamento automático de reuniões, baseada em um conjunto de protocolos de negociação para um sistema multiagentes. Este conjunto de protocolos têm o objetivo de garantir a máxima satisfação do grupo de participantes da reunião, considerando as preferências individuais e as restrições de privacidade destes participantes. É também discutida a análise destes protocolos realizada segundo uma métrica de avaliação proposta e um simulador que também é apresentado. Por fim, é demonstrado a aplicação prática desta abordagem no desenvolvimento de um protótipo de um sistema automático de agendamento de reuniões.

Sumário

Prefácio	v
1 Introdução	1
1.1 <i>CSCW</i>	3
1.2 Agendamento de Reuniões	4
1.3 Agendamento Automático de Reuniões	7
1.4 Sitemas Multiagentes	8
2 Agendamento de Reuniões Através de Sistemas Multiagentes	10
2.1 Trabalhos Relacionados	11
2.2 Definição Formal	18
2.2.1 Métricas de Avaliação	19
2.3 Conjunto de Protocolos de Negociação para o Agendamento de Reuniões .	23
2.3.1 Protocolo de Negociação do Nível de Privacidade	24
2.3.2 Modo 1: Protocolo Direto	26
2.3.3 Modo 2: Protocolo de Votação	26
2.3.4 Modo 3: Protocolo de Aprovação	29
2.3.5 Modo 4: Protocolo de Negociação	31
2.4 Cancelamento e Reagendamento de Reuniões	33
2.4.1 Protocolo de Cancelamento	35
2.4.2 Protocolo de Reagendamento	35
2.5 Coalizões	36
2.6 Análise dos Protocolos	37
2.6.1 Simulador	37
2.6.2 Resultados Obtidos	38
3 Protótipo MENES (Multiagent meEtiNg schEduling System)	50
3.1 Localização e Comunicação	52
3.1.1 ANS	53

3.2	Linguagem de comunicação dos agentes	53
3.3	Arquitetura do <i>host</i>	54
3.4	Arquitetura do <i>representative</i>	55
3.5	Interface com o Usuário	57
4	Conclusões	62
A	<i>Gramática da linguagem de comunicação dos agentes</i>	64
B	<i>vProfile</i>	66
B.1	Propriedades de <i>vProfile</i> : Categorias	66
B.2	Propriedades de <i>vProfile</i> : Dias da Semana	67
B.3	Definição Formal	68
B.3.1	Definições Preliminares	68
B.3.2	Definição da Entidade <i>vProfile</i>	69
	Bibliografia	70

Lista de Tabelas

2.1	$\alpha(t)$ para cada participante.	20
2.2	$\alpha^*(t)$ para cada participante.	21
2.3	$\beta(t)$, N e R para cada intervalo possível.	21
2.4	$\beta(t)$ para cada intervalo possível segundo os métodos alternativos.	22
2.5	Novo $\alpha(t)$ para cada participante.	22
2.6	Comparação entre o cálculo alternativo $\beta^4(t)$ e o original $\beta(t)$	22
2.7	Exemplo de votação.	27
2.8	Padrões de preferências utilizados no gerador de agendas.	37

Lista de Figuras

2.1	Interação dos agentes com as informações do participante.	11
2.2	Diagrama da interação da sociedade para o agendamento utilizando <i>host</i> interno.	11
2.3	Diagrama da interação da sociedade para o agendamento utilizando <i>host</i> externo.	12
2.4	Reticulado para a determinação do escopo da negociação.	24
2.5	Reticulado para determinação do modo de privacidade.	26
2.6	Representação gráfica do Protocolo de Negociação do Nível de Privacidade.	27
2.7	Representação gráfica do Protocolo Direto.	28
2.8	Representação gráfica do Protocolo de Votação.	30
2.9	Representação gráfica do Protocolo de Aprovação.	30
2.10	Representação gráfica do Protocolo de Negociação.	32
2.11	Probabilidade de agendar uma reunião.	39
2.12	Análise do Protocolo de Votação.	40
2.13	Comparação entre os métodos Borda e Copeland.	41
2.14	Comparação entre estratégias.	42
2.15	Comparação entre estratégias utilizando Regressão Linear.	43
2.16	Comparação entre as estratégias lacônicas, egoísta e mista (1 egoísta e 4 lacônicos).	44
2.17	Negociação entre lacônicos.	45
2.18	Negociação entre egoístas.	45
2.19	Negociação entre 4 lacônicos e 1 egoísta.	46
2.20	Comparação entre as estratégias altruísta, egoísta e mista (1 egoísta e 4 altruístas).	46
2.21	Negociação entre altruístas.	47
2.22	Negociação entre 4 altruístas e 1 egoísta.	47
2.23	Negociação sem coalizão.	48
2.24	Negociação com coalizão de 2 participantes.	48
2.25	Negociação com coalizão de 3 participantes.	49

2.26	Negociação com coalizão de 4 participantes.	49
3.1	Estados internos dos agentes.	51
3.2	Arquitetura da sociedade com a camada de comunicação e localização. . .	54
3.3	Arquitetura do <i>host</i>	55
3.4	Arquitetura do <i>representative</i>	56
3.5	Janela para a definição de restrições.	59
3.6	Janela para a chamada de reuniões.	60
3.7	Janela para avisar que o <i>host</i> está tentando agendar a reunião.	61
3.8	Janela para comunicar que uma nova reunião foi agendada.	61
3.9	Janela para comunicar que não foi possível agendar a reunião.	61

Capítulo 1

Introdução

Para que duas ou mais pessoas se reunam, física ou virtualmente, é necessário um processo de interação para que cada uma possa consultar sua agenda e verificar suas possibilidades. O processo de marcar, ou agendar, uma reunião pode ser formalmente descrito como o processo de selecionar um intervalo comum nas agendas dos participantes de um grupo de forma consensual.

A grande maioria das pessoas que já precisou marcar uma reunião se deparou com a dificuldade que é fazê-lo. Entrar em contato com os participantes e negociar intervalos tentando agradar a todos é uma tarefa que despende bastante esforço. Muitas vezes perdemos um considerável tempo no processo de agendamento e por fim não conseguimos estabelecer um horário comum ou acabamos desagradando algum participante da reunião. Além disso, toda esta interação e negociação demorada com os participantes é razoavelmente tediosa.

Tornar transparente o processo de agendar uma reunião, livrando os participantes do complicado processo de negociação e automatizando uma tarefa que é deveras corriqueira, é uma necessidade atual.

Quando teve início a automação dos escritórios, no início da década passada, surgiram programas de computador para a automatização de inúmeras tarefas diárias destes ambientes de trabalho, dentre elas a edição de documentos, a planilha de movimentação financeira e as agendas. Chapanis [4] apresentou nesta época um interessante estudo sobre as possibilidades de tornar eletrônicas as agendas de papel, onde podemos ver a grande diversidade de formas como as agendas são utilizadas.

A integração dos sistemas de agendas para facilitar o agendamento de reuniões entre os grupos foi um dos primeiros problemas a serem considerados na recente área da computação denominada Trabalho Cooperativo Apoiado por Computador - CSCW (sigla para o inglês *Computer-Supported Cooperative Work*), ou simplesmente *groupware*.

Os sistemas de *groupware* têm o objetivo de ajudar as pessoas, através do computador,

a realizarem atividades que necessitam da interação entre elas. Existem hoje muitos sistemas de *groupware* sendo utilizados com sucesso, dentre eles podemos citar o correio eletrônico (*e-mail*), os sistemas de teleconferência e os sistemas de bate-papo (*chat*).

O desenvolvimento e a adoção dos sistemas de *groupware* são, de certa forma, complicados. Estes sistemas devem atender as necessidades dos usuários individualmente considerando também os objetivos do grupo. Além disso, para que este tipo de sistema seja adotado, é preciso que uma massa crítica passe a usá-lo para que este seja efetivamente útil para todos. Grudin [9] comenta sobre oito desafios no desenvolvimento de *groupwares*, dentre eles o problema da massa crítica.

Como mencionávamos anteriormente, começaram a surgir sistemas que auxiliavam os participantes de um grupo a agendar uma reunião. A idéia era simples: o sistema verifica as agendas dos participantes e marca a reunião em um dos horários livres em comum. Contudo, estes sistemas enfrentaram questões importantes como a privacidade das informações e a preferência dos usuários por determinados horários, além do problema das agendas eletrônicas ainda não serem muito populares.

Os sistemas desenvolvidos e que passaram a ser utilizados segundo Grudin e Palen [10], apenas ajudam no processo de agendamento, apresentando as chamadas das reuniões em interfaces amigáveis e simplificando a interação utilizando o correio eletrônico, dentre outras facilidades. Porém, é importante destacarmos que toda a decisão depende do usuário e um bom trabalho adicional é necessário para responder as consultas do sistema.

Assim, pesquisadores começaram a investir no desenvolvimento de sistemas inteligentes que livram o usuário de tarefas extras, tentando garantir a privacidade e o respeito às preferências do usuário e realizando agendamentos de forma automática.

O problema de agendar uma reunião de forma automática tornou-se um problema clássico de *groupware* uma vez que nenhuma solução até hoje implementada é considerada eficiente e foi adotada fora do próprio ambiente de pesquisa.

Neste trabalho apresentamos uma abordagem para o problema do agendamento automático de reuniões que acreditamos ser a solução para este problema. No capítulo 2 apresentamos esta abordagem e os resultados obtidos em um ambiente de simulação.

Para tentar mostrar a viabilidade da implementação prática e utilizável de nossa abordagem apresentamos um protótipo de um sistema chamado MENES (*Multi-agent meEtINg schEduling System*) que utiliza esta abordagem. No capítulo 3 descrevemos o sistema e detalhes de sua implementação.

Por fim, no capítulo 4 apresentaremos as conclusões obtidas com este trabalho e suas possíveis contribuições para a ciência.

Para situar o leitor no contexto de nosso trabalho vamos apresentar ainda neste capítulo o conhecimento fundamental para a leitura desta dissertação.

1.1 CSCW

CSCW (*Computer Supported Cooperative Work*) é uma área da Computação que busca solucionar problemas de interação entre pessoas utilizando computadores. Os sistemas que visam facilitar o contato entre as pessoas na realização das mais diversas tarefas são conhecidos como sistemas de *groupware*.

O objetivo dos sistemas de *groupware*, segundo Ellis, Gibs e Rein [3], é dar suporte computacional as atividades de grupos como comunicação, coordenação e colaboração. Estes autores definem os sistemas *groupware* como sendo aqueles que dão suporte a grupos de pessoas engajados em uma tarefa comum e que oferecem uma interface para o compartilhamento do ambiente.

Ellis e Wainer [21] classificam os sistemas de *groupware* baseado na função do sistema no grupo em que foi adotado. De acordo com eles os sistemas de *groupware* podem ser distribuídos em quatro classes, chamadas de “aspectos”, que são:

- *keeper*: sua função é relacionada com o armazenamento e manipulação de informações compartilhadas. Um bom exemplo de um *keeper* não computacional é a lousa utilizada por um grupo de pessoas em uma reunião. Nos sistemas de *groupware* temos alguns *keepers* típicos como os editores de texto concorrentes, ferramentas CASE e CAD, etc.
- *coordinator*: sua função é relacionada com a ordenação e sincronização das atividades que formam o processo colaborativo inteiro. Neste caso, cada participante desenvolve uma tarefa e uma ordem previamente definida, e a atividade do *coordinator* é coordenar a evolução temporal do processo, abilitando que uma atividade seja realizadas assim que todas as atividades que a precedem sejam terminadas. Uma linha de produção fabril é um bom exemplo de um *coordinator* não computacional. Quanto aos sistemas de *groupware*, são *coordinators* típicos os sistemas de *workflow*, sistemas de suporte a decisão, etc.
- *communicator*: sua função é relacionada com a comunicação explícita entre os participantes do grupo. Um exemplo de um *communicator* não computacional são as cartas e o telefone. Já em relação a sistemas de *groupware* temos alguns *communicators* típicos como o e-mail, os sistemas de bate-papo, etc.
- *team-agent*: são sistemas inteligentes ou semi-inteligentes que desenvolvem alguma atividade específica com a função de ajudar na dinâmica do grupo. Podemos ver os *team-agents* como participantes artificiais que realizam alguma tarefa específica. Um bom exemplo não computacional para os *team-agents* são os especialistas em desempenho de um grupo de engenheiros de software. Nos sistemas de *groupware* os agendadores de reuniões são exemplos típicos de *team-agents*.

1.2 Agendamento de Reuniões

Os sistemas de agendamento não automáticos foram uma das primeiras tecnologias de *groupware* a surgir. As agendas convencionais de papel ajudam as pessoas de muitas maneiras diferentes e podem ser até indispensáveis para a organização do trabalho.

Formalmente, estas agendas são artefatos cognitivos de memorização e planejamento que estendem as funções cognitivas pela externalização da informação em uma forma nova e útil [14].

Agendas são usadas para uma grande variedade de tarefas de apoio pessoal, incluindo planejamento e agendamento, repositório de lista de afazeres, lembretes, etc. As pessoas que utilizam agendas costumam personalizá-las com anotações, cores, cliques de papel e lembretes.

Além disso, agendas são tipicamente artefatos privados, onde seu dono é o principal manipulador e o acesso de demais pessoas é controlado por convenções sociais, permissões explícitas e a proximidade ao artefato físico.

Os sistemas de *groupware* modificaram o conceito de agenda mudando as agendas pessoais para o espaço social. Desta forma, em uma rede de computadores as agendas estão próximas de todos que estão conectados, e níveis de privacidade estabelecidos por seu dono determinam o controle de acesso, substituindo as restrições naturais de localização física e formato das agendas de papel, e as convenções sociais.

Devido a importância do agendamento de reuniões nas operações de negócios, sistemas com esta finalidade são alvo das convenções institucionais e influenciam a prática do trabalho em larga escala.

O projeto, desenvolvimento, adoção e suporte dos sistemas de *groupware* não são tarefas simples de serem realizadas. Os aplicativos de *groupware* devem ser muito bem projetados para usuários individuais e ainda serem socialmente viáveis, dando apoio a coordenação do grupo na realização de seus objetivos.

Grudin [9] apresentou oito desafios para os desenvolvedores de *groupware* que resumem estas dificuldades. Estes desafios são:

- A disparidade entre quem faz o trabalho e quem recebe o benefício: em uma aplicação de *groupware* não há uma diferença clara entre quem tem mais trabalho com sua utilização e quem se beneficia com esta utilização;
- O problema da massa crítica e o dilema do prisioneiro: A maioria dos sistemas de *groupware* somente são úteis se uma alta percentagem de membros do grupo o utilizarem. Além disso, se os membros do grupo atuarem em seu próprio benefício o resultado é pior, não só para o grupo, mas para cada membro individualmente;

- Fatores motivacionais, políticos e sociais: Os sistemas de *groupware* são muito influenciados pelas questões sociais, econômicas, motivacionais e políticas das complexas relações humanas nos grupos;
- Tratamento de exceções: Os sistemas de *groupware* são bastante prejudicados pela diferença entre como se pensa que as coisas funcionam e como elas realmente funcionam;
- Projeto de recursos infreqüentemente utilizados: Os desenvolvedores de sistemas de *groupware* exageram na importância da aplicação de coordenação e comunicação nos sistemas que desenvolvem;
- A dificuldade subestimada do desenvolvimento de *groupware*: A análise, projeto e desenvolvimento de sistemas multiusuários são bem mais complexas que aplicações para um único usuário;
- A quebra da tomada de decisão intuitiva: Tomar decisões intuitivas durante o processo de desenvolvimento dos sistemas de *groupware* faz com que muitas aplicações deste tipo não sejam utilizadas. Sistemas de *groupware* exigem projetos bem analisados e elaborados;
- Gerenciando a aceitação: Sistemas de *groupware* são muito suscetíveis a eventuais falhas no processo de introdução do sistema em um novo ambiente, fazendo com que o sistema seja rejeitado por seus usuários.

Dentre estes desafios os que mais se destacam em relação aos sistemas de agendamento de reuniões são o problema da disparidade e o problema da massa crítica.

Consideremos um sistema de agendamento de reuniões utilizado em uma empresa que realiza o agendamento buscando na agenda dos participantes um intervalo disponível em comum. O maior beneficiário da utilização deste sistema é o participante que está convocando a reunião, tipicamente o chefe ou sua secretária. Mas para que este sistema funcione eficientemente todos os participantes do grupo devem manter sua agenda atualizada, caso contrário por exemplo, uma reunião pode ser agendada num horário que o participante não tem mais livre.

Neste caso, os funcionários da empresa têm mais uma tarefa diária que é manter sua agenda atualizada, enquanto que o grande beneficiário da utilização do sistema é o chefe. Com isso podemos ver claramente a disparidade que pode existir entre quem tem mais trabalho a fazer e quem recebe o maior benefício.

Agora consideremos que este trabalho a mais que os funcionários tem de realizar desmotivou-os a utilizar o sistema e apenas uma pequena porcentagem deles o faz. Quando o chefe convocar uma reunião com um grupo deste funcionários, onde apenas um ou dois

não utilizam o sistema, não será possível agendar a reunião. Desta forma o sistema passa a ser cada vez menos utilizado e pode cair em desuso.

Assim, para que um sistema de agendamento de reunião possa ser aplicado com eficiência nos diversos grupos em que um determinado usuário está inserido é preciso que se tenha uma grande quantidade de pessoas utilizando-o.

Grudin e Palen [10] apresentaram um estudo sobre a utilização de sistemas de agendamento de uma perspectiva social. Neste trabalho os autores discutem a utilização bem sucedida de sistemas de agendamento nas empresas Sun e Microsoft, que apresentaram um conjunto de fatores tecnológicos e sociais que tornaram a adoção de um sistema deste tipo possível e que não seguiu a tradicional trajetória, de cima para baixo na hierarquia da empresa, que normalmente acontece com sistemas de *groupware*.

O sistema adotado na empresa Microsoft foi o Microsoft Schedule+, onde as agendas dos participantes são mantidas em seus próprios computadores e são atualizadas frequentemente por um servidor central. Este sistema permite que o usuário especifique quem pode ou não visualizar sua agenda através do estabelecimento de restrições de acesso, que podem apenas visualizar se o horário está livre ou não, sem ver o conteúdo dos agendamentos marcados, visualizar tudo ou não visualizar nada.

Para agendar uma reunião o usuário deste sistema pode utilizar-se de dois procedimentos diferentes: o usuário entra com os dados do agendamento no sistema, incluindo o nome ou *e-mail* dos participantes da reunião, e em seguida o sistema mostra uma sobreposição das agendas dos participantes da reunião e salienta os intervalos livres em comum; ou de forma reversa, o usuário verifica o intervalo livre primeiro e depois entra com os dados do agendamento.

Seja qual for o procedimento adotado para convocar uma reunião, este sistema envia para os participantes um *e-mail* com os detalhes da reunião que será agendada e solicita uma resposta. Para responder se aceita ou não o agendamento o participante escolhe uma das opções de resposta que são mostradas na mensagem e seleciona o botão correspondente. Em qualquer etapa do processo o participante que convocou a reunião e os demais podem visualizar no sistema qual foi a resposta dos demais participantes. Uma reunião é marcada na agenda de todos quando todos concordam com o agendamento.

A empresa Sun adotou o sistema Sun's Calendar Manager, onde as agendas dos participantes ficam armazenadas nos servidores dos participantes e para acessar determinado participante é preciso informar seu nome e servidor. Este sistema permite os mesmos níveis de privacidade do Microsoft Schedule+, adicionando a facilidade de permitir que o participante habilite outros participantes a manipular sua agenda inserindo e removendo compromissos.

Assim como o Microsoft Scheduler+, o Sun's Calendar Manager oferece dois procedimentos diferentes para se convocar uma reunião: o participante que quer convocar

a reunião seleciona um dos intervalos livre nas agendas de todos os participantes, utilizando uma ferramenta do sistema que realiza a interseção das agendas, entra com as informações da reunião e envia a convocação via *e-mail* a partir do próprio sistema; ou o participante seleciona dois ou três intervalos livres comuns e os envia aos participantes via *e-mail*, para que os participantes indiquem quais preferem, o intervalo de maior preferência é determinado pelo sistema e o participante que está convocando a reunião entra com as informações da reunião e envia a convocação via *e-mail*.

Independente do procedimento adotado o sistema anexa um botão ao *e-mail* enviado com a convocação final da reunião para que os participantes possam arrastá-lo do seu leitor de e-mail para o Sun's Calendar Manager. Neste sistema o participante não tem a opção de não aceitar a convocação.

Como podemos ver, estes sistemas que acabamos de descrever facilitam bastante o processo de agendamento de reuniões, oferecendo interfaces eficientes e práticas, além de uma série de facilidades para manipular as agendas e realizar agendamentos. Estas facilidades que trazem benefícios individuais contribuíram para gerar a massa crítica necessária para sua bem sucedida adoção.

Outro fator que influenciou na aceitação dos sistemas em questão foi social e teve um papel crucial. Algumas pessoas nessas empresas começaram a utilizar os sistemas de agendamento satisfatoriamente e passaram a pressionar informalmente seus colegas a também utilizá-lo. Assim, com frases do tipo: “você ainda não usa o sistema de agenda?”, algumas pessoas influenciaram outras, e assim por diante, até que grandes grupos que utilizam os agendadores se formaram.

1.3 Agendamento Automático de Reuniões

Podemos distinguir dois paradigmas no agendamento automático de reuniões: sistemas abertos e sistemas fechados.

Nos sistemas fechados temos uma relação hierárquica dos usuários imposta pela organização do grupo a que estes pertencem. Neste tipo de sistema, onde a utilização de ferramentas para agendamento é imposta aos participantes do grupo e existe uma obrigação em realizar o agendamento, há uma suscetibilidade à sabotagem. Um participante do grupo, por não gostar de manter sua agenda atualizada e ainda assim correr o risco do sistema agendar um compromisso em um horário não muito bom em relação a seus critérios pessoais, pode marcar todos os intervalos de sua agenda como ocupados. Esta atitude, considerada sabotagem, faz com que o sistema não funcione uma vez que não é possível encontrar um intervalo livre comum a todos os participantes. Ephrati, Zlotkin e Rosenchein [8] discutem mais detalhadamente esta questão e apresentam uma abordagem interessante para os sistemas fechados que discutiremos na seção 2.1.

Já nos sistemas abertos não há uma relação hierárquica como mencionamos e todos tem interesse em que a reunião se realize. Contudo, não há a obrigatoriedade de se estabelecer um agendamento em detrimento das preferências e da ocupação dos participantes. Esta característica torna os sistemas abertos, além de mais interessantes e aplicáveis, mais complexos. Neste trabalho estamos abordando os sistemas abertos de agendamento de reuniões.

Para os sistemas abertos de agendamento de reuniões existem duas questões principais que determinam sua utilização sob o aspecto de grupo: a possibilidade do participante expressar suas preferências e a garantia que esta será fortemente considerada no processo; e a confiança que sua privacidade será assegurada segundo seus critérios de confiança nos demais participantes [4].

Um participante de uma reunião pode ter preferência a determinados horários de sua agenda. Desta forma, é preciso permitir ao usuário de um sistema de agendamento automático expressar seu gosto pessoal para os horários em que a reunião pode ser agendada. O sistema então deve considerar a preferência de cada usuário para determinar o intervalo em que a reunião será agendada.

A privacidade das informações sobre as preferências e da própria agenda dos usuários do sistema deve ser garantida. Desta forma, um sistema automático de agendamento deve considerar as restrições à divulgação de informações particulares para outros usuários.

Suponhamos que, por exemplo, o chefe de um setor convoca seu grupo de funcionários para marcar uma reunião. Imaginemos que um dos participantes do grupo não se sente a vontade de mostrar para seu chefe seus horários livres, que são muitos. Além disso, outro funcionário do grupo não deseja que seu grupo saiba que ele prefere agendar seus compromissos a tarde, pois pela manhã sua produção é baixa.

Para evitar que estes usuários sabotem o sistema ou deixem de utilizá-lo, o agendador automático que o grupo está utilizando tem que permitir aos funcionários em questão restringir as informações que serão trocadas durante a negociação do melhor intervalo para o agendamento e garantir que suas preferências serão consideradas nesta negociação.

O intervalo que o sistema determinou para que se faça o agendamento da reunião de nosso exemplo deve satisfazer a todos os participantes do grupo e ao grupo como um todo.

1.4 **Sistemas Multiagentes**

O agendamento de reuniões é uma tarefa intrinsecamente distribuída e para automatizá-la as abordagens que vêm sendo apresentadas baseiam-se nos sistemas multiagentes.

Os sistemas multiagentes são sistemas de raciocínio distribuídos, subárea da Inteligência Artificial paralela e distribuída, que se vale dos recursos do paralelismo para obter vantagens sobre os sistemas inteligentes monolíticos.

Podemos definir sistemas multiagentes como sistemas de raciocínio distribuído compostos por um grupo de agentes, conhecido por sociedade de agentes, e por um conjunto de canais de comunicação entre eles.

Além de trocarem informações entre si, os agentes também interagem com seus usuários com a finalidade de receber especificações de objetivos a serem atingidos e tarefas a serem realizadas.

Existem muitas formas de definir agentes nos amplos domínios de suas aplicações. Uma das definições encontradas na bibliografia sobre o assunto é um tanto descompromissada, mas traz uma clara noção do motivo à que vieram os agentes [11]:

”Um agente é um elemento de software que sabe como fazer coisas que provavelmente você mesmo faria se tivesse tempo.”

Contudo, os agentes que compõem os sistemas multiagentes nada mais são que programas de computador dotados de algumas propriedades específicas, as quais são [22]:

- Autonomia: agentes operam sem a intervenção direta de humanos e tem controle sobre suas ações e estados internos;
- Habilidade social: agentes interagem com outros agentes através de caminhos de comunicação;
- Reatividade: agentes percebem o ambiente através de sensores de dados e reagem às mudanças que ocorrem neste;
- Iniciativa própria: agentes não agem simplesmente em resposta ao ambiente, eles estão habilitados a agir seguindo seu comportamento orientado a um objetivo e a tomar a iniciativa neste sentido.

A arquitetura de um agente é uma metodologia particular para sua construção, especificando como o agente pode ser decomposto em um conjunto de módulos e como estes módulos interagem. O conjunto dos módulos e sua interação mostram como os sensores de dados e os estados internos do agente determinam suas ações.

Existem muitas aplicações para os agentes, apesar de ainda serem em grande maioria em nível experimental, que visam realmente habilitá-los a agir no lugar de humanos em várias atividades que consomem esforços e tempo.

Assim, sociedades de agentes devem interagir com agentes participantes buscando realizar seus objetivos, ou seja, o objetivo do usuário do sistema multiagentes especificamente.

Capítulo 2

Agendamento de Reuniões Através de Sistemas Multiagentes

Os mais importantes trabalhos em agendamento de reuniões que utilizam os sistemas multiagentes não resultaram em sistemas que chegaram a ser utilizados realmente. Contudo, não foram os sistemas multiagentes os responsáveis por esta falta de eficiência, e sim a forma como os agentes interagem e lidavam com as informações de seus usuários.

Em nossa abordagem cada usuário possui um par de agentes denominados *host* e *representative*.

Os agentes representam seus usuários na negociação de um intervalo para o agendamento da reunião. Para orientar a negociação elaboramos um conjunto de protocolos de negociação para os agentes. Com este conjunto de protocolos os agentes interagem e realizam o agendamento.

Os *hosts* centralizam o processo de negociação, servindo como mediadores para este processo. Estes agentes também podem aparecer sozinhos na sociedade, não estando necessariamente vinculados a um usuário. Quando uma reunião vai ser convocada o usuário que deseja fazê-lo ativa seu agente *host*.

Já os *representatives* são responsáveis por representar seu usuário na negociação, agindo considerando as restrições de privacidade e as preferências deste usuário. Estes agentes tem acesso total as preferências e a agenda de seu usuário, utilizando estas informações na negociação.

Possibilitar a utilização de um *host* que não pertença a um participante da reunião é estratégico, uma vez que os participantes do grupo podem desejar que o mediador da negociação seja um agente totalmente imparcial. Com isso, podemos ter reuniões onde o *host* pertence a um usuário do grupo de participantes da reunião, que denominamos *host* interno, figura 2.2, e o caso em que o *host* não pertence a nenhum participante, o qual denominamos *host* externo, figura 2.3.

A figura 2.1 mostra o diagrama da interação dos agentes com as informações do participante. Estes agentes interagem entre si através de um conjunto de protocolos que apresentaremos na seção 2.3, com o objetivo de atingir a melhor reunião possível para o grupo, tentando melhor satisfazer cada participante da reunião.

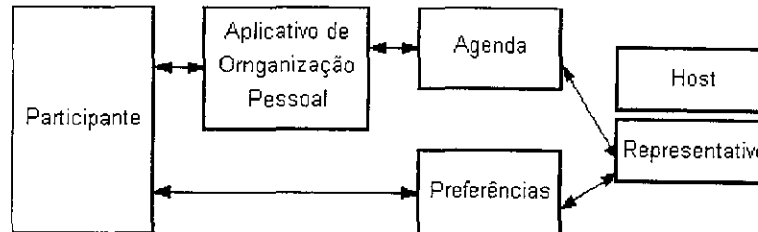


Figura 2.1: Interação dos agentes com as informações do participante.

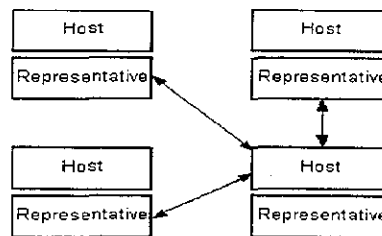


Figura 2.2: Diagrama da interação da sociedade para o agendamento utilizando *host* interno.

2.1 Trabalhos Relacionados

Por se tratar de um problema clássico, muitos pesquisadores já tentaram elaborar sistemas e aplicar suas teorias na busca de uma solução para o problema do agendamento automático de reuniões.

Ephrati, Zlotkin e Rosenchein [8] desenvolveram uma abordagem para o agendamento de reuniões em sistemas fechados. Estes autores criaram três mecanismos para agendamento de reuniões que eles consideram a prova de manipulação.

Esta abordagem utiliza um sistema de pontuação, onde o usuário pontua diferentes informações referentes a reunião em função de qual mecanismo esta usando. No mecanismo orientado ao calendário os usuários pontuam convenientemente os intervalos de tempo

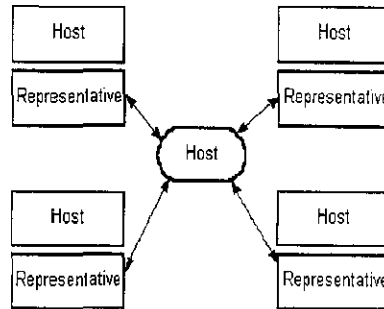


Figura 2.3: Diagrama da interação da sociedade para o agendamento utilizando *host* externo.

disponíveis, ou seja, o usuário recebe um conjunto de pontos para distribuir entre os intervalos de tempo livres simplesmente. No orientado à reunião o usuário distribui os pontos que recebe entre os intervalos de tempo livres de acordo com o objetivo ou assunto da reunião. No mecanismo orientado ao agendamento os usuários pontuam o agendamento como um todo, ou seja, são determinados todos os agendamentos possíveis e o usuário distribui os pontos que recebe entre estes agendamentos.

A pontuação é avaliada de acordo com métodos da teoria do voto e um agendamento é determinado. Estes mecanismos são manipuláveis, podendo um participante por não desejar determinado intervalo ou reunião pontuá-la com o mínimo possível. O diferencial do trabalho em questão é a utilização de um método da teoria do voto para evitar este tipo de manipulação, denominado “Clarke Tax Mechanism”.

Este mecanismo faz com que os participantes tenham menor poder de influência sobre o agendamento à medida que estes dificultam o processo, ou seja, o usuário perde pontos para utilizar em um próximo agendamento conforme a pontuação que deu para a reunião agendada. Desta maneira, se o participante usar de má fé na hora de atribuir os pontos vai acabar sem poder de decisão nos agendamentos futuros, tendo que aceitar o que for determinado pelos demais.

Neste sistema as informações de agenda e preferências não são privadas e não há uma descentralização no sistema; todos os usuários têm suas informações guardadas no mesmo repositório e o sistema tem acesso a elas sem restrições.

Nossa abordagem difere muito da abordagem de Ephrati, Zlotkin e Rosenschein principalmente por tratar os sistemas abertos, onde são considerados fatores bem diferentes dos sistemas fechados, como discutimos na seção 1.3.

Já em relação aos sistemas abertos, Sycara e Garrido [19] desenvolveram um sistema multiagentes para agendamento de reuniões com controle distribuído através da utilização de um protocolo dinâmico.

Cada agente neste sistema tem um conjunto de preferências estabelecidas por seu usuário para cada reunião em particular, e cada agente tem como objetivo agendar a reunião que maximize sua preferência individual sobre a reunião. Estes agentes podem trocar informações sobre suas preferências e intervalos livres de acordo com alguma política de privacidade que é um dos parâmetros dos testes realizados pelos autores como discutiremos em seguida. Adicionalmente as preferências, os agentes também possuem pesos que os orientam a relaxar suas preferências conforme o número de interações aumenta.

O protocolo a seguir orienta a negociação dos agentes no sistema em questão:

1. Todos os agentes do grupo são aleatoriamente enumerados;
2. Um agente é selecionado aleatoriamente e passa a coordenar a negociação encaminhando a primeira proposta;
3. Cada agente recebe a proposta e aceita ou rejeita, respondendo ao coordenador. Caso o agente aceite deve enviar ao coordenador sua prioridade com relação a reunião que está sendo agendada. A prioridade considera as preferências e os fatores de relaxamento;
4. Quando o coordenador recebe todas as respostas e o agendamento foi aceito por todos, este envia uma mensagem de confirmação do agendamento e calcula a qualidade do agendamento, que é a média aritmética das prioridades;
5. Caso a proposta seja rejeitada por um agente que seja, o processo é reiniciado no item 2. Neste caso o novo coordenador já aplica seu fator de relaxamento e reduz suas preferências.

Os autores desta abordagem realizaram uma análise de seu sistema considerando alguns fatores de *groupware*, principalmente considerando a privacidade das informações. Para isso, duas formas de trocar informações sobre as agendas foram experimentadas: na primeira todos os intervalos livres do agente são divulgados; na segunda os agentes apenas divulgam parte de seus horários livres. Além da privacidade sobre o calendário, foi considerada a privacidade sobre as preferências, onde também foram experimentadas duas formas diferentes: na primeira os agentes divulgam suas preferências para todos os intervalos que sugerem; e na segunda esta preferência só é divulgada depois do intervalo ter sido aceito por todos.

No caso em que as preferências são divulgadas previamente os agentes podem usar a qualidade do agendamento, ao invés de suas preferências, para melhor satisfazer o grupo. Além disso, no caso das preferências públicas e de todos os intervalos disponíveis serem divulgados, a possibilidade de se estabelecer o agendamento na primeira interação é alta.

Uma série de testes foram realizados considerando os fatores de privacidade acima descritos e um fator orientado ao objetivo. Este fator determina se todos os agentes tem as mesmas preferências ou preferências distintas. Além disso, foi considerada a densidade das agendas, ou seja, a porcentagem de horários da agenda que estão ocupados.

Através de seus experimentos os autores determinaram que a qualidade de um agendamento é maior quando todos os agentes têm as mesmas preferências, e diminui conforme as agendas ficam mais densas. Além disso concluíram que utilizando a divulgação de todos os intervalos livres também se tem um agendamento de melhor qualidade. Com relação ao número de interações, fator também considerado pelos autores, foi concluído que utilizando a divulgação prévia das preferências consegue-se agendar uma reunião com muito menos interações do que com a outra forma.

Os resultados desta experimentação também mostraram que quanto mais privadas são as informações mais estáveis são os resultados obtidos pelo agendamentos em relação a qualidade e ao número de interações.

Com o objetivo de modelar cenários mais reais Sycara, Brena e Garrido [20] passaram a estudar o comportamento de aprendizado e inferência dos agentes sobre as atitudes mentais e comportamentos dos demais agentes. Com este estudo, os autores buscaram verificar como a adaptação mútua pode diminuir o número de interações necessárias para que o sistema chegue a um agendamento.

Desta forma, esta nova abordagem visa otimizar o número de interações entre os agentes, fazendo com que este fique menor a cada agendamento. Isto é conseguido uma vez que os agentes buscam aprender as preferências e conhecer melhor a “personalidade” dos demais agentes à medida que os agendamentos são realizados, fazendo com que o agente possa utilizar, além de suas preferências pessoais, a qualidade dos possíveis agendamentos. Não conhecemos resultados de experimentos com esta nova abordagem.

Vários pontos diferem nosso trabalho deste que estamos discutindo:

- Em nosso trabalho a negociação é intermediada inteiramente por um agente de um dos participantes ou por um agente externo, não existindo rotatividade na coordenação do processo. Adicionalmente, este intermediário é determinado pelo grupo em um processo de negociação, garantindo que todos confiam na imparcialidade do coordenador da negociação;
- Em nosso trabalho determinamos vários níveis de privacidade que podem ser escolhidos pelos participantes segundo suas restrições aos membros do grupo de forma dinâmica, ou seja, durante a negociação. A abordagem de Sycara, Brena e Garrido considera diferentes níveis de privacidade apenas para a análise do protocolo, recomendando a utilização de um ou outro segundo os resultados desejados;

- Sycara, Brena e Garrido não dão prioridade as preferências dos participantes e estabeleceram fatores de relaxamento que diminuem as preferências dos participantes com o passar da negociação. Em nosso trabalho não flexibilizamos as preferências mesmo quando um nível de privacidade muito restrito é adotado, fazendo com que estas preferências sejam prioridade no processo de agendamento;
- Em nosso trabalho não estamos preocupados com o número de interações necessárias para o agendamento e sim utilizamos uma métrica cujo objetivo é a satisfação do grupo e dos participantes com relação ao agendamento;
- Em nosso trabalho consideramos e medimos a influência do aprendizado das preferências dos demais agentes no agendamento, apesar de não tratarmos as questões de como o agente aprende estas preferências.

Outro trabalho que também trata os sistemas abertos utilizando multiagentes é de Sen e Durfee [6] [7]. Estes autores desenvolveram um sistema adaptativo que utiliza multiagentes negociando via um protocolo de negociação e um conjunto de heurísticas para estabelecer um agendamento. Nessa abordagem existe a figura de um agente, denominado *host*, que tem a função de centralizar as negociações, não havendo comunicação entre os demais agentes.

No protocolo utilizado pelos autores cada agendamento tem um *host* diferente que é encarregado de intermediar a negociação. Quando uma reunião precisa ser agendada o *host* contata os participantes do agendamento e recebe as informações que os participantes querem compartilhar para o agendamento. Este processo é repetido várias vezes até que um intervalo de tempo livre comum a todos os participantes seja encontrado. Vários agendamentos podem ser realizados simultaneamente, podendo o agente atuar como *host* ou como participante em várias negociações ao mesmo tempo.

Este protocolo depende fundamentalmente da informação que os agentes desejam compartilhar, ou seja, as estratégias de privacidade e de divulgação das informações que os agentes utilizam.

O conjunto de heurísticas, que são as estratégias que os agentes adotam, são selecionadas conforme fatores da organização do grupo. Tais estratégias são: "Search Biases", que determina a forma como os intervalos são procurados na agenda do participante, podendo ser linear ou hierárquica; "Announcement Strategies", que determina como o *host* propõe o agendamento, podendo ser o melhor possível ou considerando algumas das melhores soluções; "Bidding Strategies", que determina a resposta dos agentes, podendo ser apenas sim ou não, ou um conjunto de intervalos alternativos; "Commitment Strategies", que determina se mais de um agendamento pode ser realizado simultaneamente.

Variando a combinação de estratégias em um ambiente de simulação os autores realizaram um conjunto de experimentos com o objetivo de avaliar como as diferentes com-

binações de estratégias influenciam na probabilidade de se conseguir uma agendamento em função da ocupação das agendas e o número de interações necessários para o agendamento. Estes resultados não são relevantes para o escopo de nosso trabalho e em nenhum momento os autores consideraram as preferências dos participantes para estabelecer o agendamento.

Em um outro trabalho Sen, Haynes e Arora [16] passaram a considerar as preferências dos participantes. Para isso, os autores desenvolveram um outro protocolo de agendamento e deixaram de considerar as diferentes estratégias que mencionamos a pouco.

O protocolo utilizado pelos agentes para agendar a reunião nesta nova abordagem é o seguinte:

1. Quando um agendamento precisa ser realizado, o *host* procura em sua agenda intervalos onde a reunião pode ser agendada. O *host* então anuncia o chamado contrato contendo informações sobre o agendamento e um ou mais dos melhores intervalos segundo suas preferências.
2. Cada participante recebe o contrato e também procura em sua agenda os melhores intervalos para a reunião proposta e envia ao *host* estes intervalos. Este conjunto de intervalos pode ser um subconjunto dos intervalos sugeridos pelo *host* ou um novo conjunto de intervalos.
3. O *host* recebe os intervalos e verifica se existe algum intervalo comum nas respostas de todos os participantes que também está livre para ele. Caso exista este intervalo a reunião é agendada e o *host* comunica aos participantes; caso contrário o *host* determina entre as propostas dos participantes e sua agenda um novo conjunto de intervalos que logo em seguida são enviados para os participantes.
4. O participantes verificam sua disponibilidade em relação ao novo contrato enviado pelo *host* e respondem a ele dizendo se podem ou não realizar o agendamento nos horários indicados no contrato.
5. Caso o *host* não encontre nenhum horário comum nas novas respostas dos participantes o protocolo é reiniciado no passo 1 até que um agendamento seja possível ou que o sistema verifique que não existe intervalo comum para o agendamento.

Para determinar os melhores intervalos para sugerir durante a negociação os agentes utilizam suas preferências em relação a um grande número de fatores relacionados com o agendamento. Desta forma, os autores estabeleceram várias dimensões de preferências relacionadas, por exemplo, com a hora, o dia, o tempo de duração ou *host* do agendamento.

Quando um conjunto de horários é solicitado ao agente este distribui os intervalos livres em listas ordenadas de acordo com cada dimensão de preferência. Com os intervalos

organizados em um ranking o agente seleciona os melhores utilizando o método Borda de seleção do vencedor da teoria do voto, já discutido neste trabalho.

A contrário deste trabalho, nossa abordagem:

- Considera diferentes níveis de privacidade que buscam atender aos diversos tipos de grupos, aumentando a satisfação de todos em relação ao agendamento;
- Não considera a possibilidade de múltiplos agendamentos pois não acreditamos que grupos de pessoas tenham grandes benefícios com este recurso;
- Considera as preferências dos usuários de forma mais simples e prática, atendendo a todas as necessidades do participante sem complicar sua interação com os agentes. Com isso, não temos conflitos de preferências dentro do agente, dispensando um processo mais complicado de decisão;
- Utiliza as preferências dos participantes para determinar a satisfação do grupo em relação ao agendamento, não considerando o número de interações como fator de medida de eficiência;
- Foi implementada e experimentada em um ambiente de simulação.

Apesar de ter obtido bons resultados segundo seus autores, os últimos dois trabalhos discutidos consideram de forma muito modesta as questões de *groupware* que influenciam diretamente a aplicação deste tipo de sistema. Isto ocorre principalmente pelo fato destes autores considerarem o agendamento de reuniões apenas como um cenário para o desenvolvimento de tecnologias relacionadas aos sistemas multiagentes.

Além das diferenças já mencionadas, nosso trabalho ainda aborda:

- O processo de reagendamento e cancelamento de reuniões, estudando suas implicações e apresentando um protocolo para tratar estes eventos;
- A formação de coalizões, estudando suas implicações para o agendamento e estabelecendo protocolos livres de coalizão;
- Os diferentes níveis de privacidade que diferentes tipos de grupos podem desejar adotar, aumentando ainda mais a eficiência dos agendamentos;
- A arquitetura do sistema distribuído pela Internet e a implementação de um protótipo de um sistema de agendamento de reuniões baseado nos sistemas multiagentes.

Assim, podemos verificar que nossa abordagem é essencialmente baseada nas questões de *groupware*, considerando as necessidades dos participantes e objetivando a maior satisfação do grupo.

2.2 Definição Formal

Apresentaremos a seguir uma definição formal para o problema de agendar uma reunião utilizando sistemas multiagentes.

Consideramos uma reunião a ser agendada m como uma 8-tupla $\langle A, a, h, q, l, wb, we, R \rangle$, onde:

- A é o conjunto $\{a_1, a_2, \dots, a_n\}$ dos agentes convidados a participar da reunião;
- a é o agente que chama a reunião;
- h é o agente que intermedia a reunião, denominado *host*. $h \in A$ se um dos participantes da reunião assume a tarefa de agendá-la, ou $h \notin A$ caso este agente não seja um dos participantes, por exemplo, por razões de privacidade.
- q é a unidade de tempo para o estabelecimento de uma reunião m , a qual denominamos *quantum*. q é uma fração de hora, ou seja, $0 < q \leq 1$. Desta forma, se determinada reunião tem no mínimo meia hora de duração então $q = 0.5$ e esta reunião pode ser agendada, por exemplo, as 10h e 30min, as 11h, as 11h e 30 min, e assim por diante;
- l é o tempo de duração da reunião. Uma reunião pode durar l vezes o *quantum* q utilizado;
- wb e we são as datas de início e fim do período do calendário onde uma reunião deve ser agendada. Desta forma, se uma reunião deve ser agendada na próxima semana, wb será a primeira hora útil da próxima segunda-feira, e we será a última hora útil da próxima sexta-feira;
- R é o conjunto de informações reais sobre a reunião, como pauta, material necessário, etc.

Cada agente a_i tem um conjunto de horários livres $F(a_i)$ representado por triplas $\langle d, h, w \rangle$, onde:

- d é o dia da semana;
- h é a hora do dia, fracionada de acordo com o *quantum*;
- w é o peso atribuído pelo usuário ao intervalo de tempo segundo suas preferências, sendo que $0 \leq w \leq 10$ e a ordem de preferência é crescente em relação ao peso.

Para uma reunião dada $m = \langle A, a, h, l, wb, we, R \rangle$: $P(m) = \{t_1, t_2, \dots, t_k\}$ é o conjunto de intervalos de tempo possíveis para se agendar a reunião, de forma que:

- $t_i \in F(a_j)$ para todo intervalo t_i e para todo agente $a_j \in A$;
- a duração de cada t_j é igual a $l * q$;
- cada t_j começa depois ou juntamente com wb e termina antes ou juntamente com wc ;

2.2.1 Métricas de Avaliação

Com o objetivo de quantificar o quão bom são os protocolos de agendamento elaboramos uma métrica a qual denominamos Grau de Otimização, ou simplesmente N .

É importante destacar que não estamos preocupados em medir tempo de execução, número de interações, e outros fatores de processamento, pois cremos que tais fatores não são relevantes para o escopo de nosso trabalho.

Primeiramente, estabelecemos como Satisfação do Agente a_i em relação ao agendamento em um intervalo possível $t = \{\langle d_1, h_1 \rangle, \langle d_2, h_2 \rangle, \dots, \langle d_n, h_n \rangle\} \in P(m)$, como sendo:

$$\alpha_a(t) = \frac{\sum_{j=h_0}^{h_0+l} w_j}{l} \quad (2.1)$$

A Satisfação Padronizada $\alpha_a^*(t)$, calculada pela equação 2.2, tem média igual a 0 e variância igual a 1, fazendo com que os graus de satisfação de cada agente $\alpha_a(t)$, de todos os agendamentos possíveis em $P(m)$, tenham desvio padrão 1.

Este recurso estatístico, também conhecido por normalização [17], garante a todos os agentes o mesmo poder de influência sobre a satisfação do grupo uma vez que as preferências não são absolutas, e sim relativas aos intervalos $P(m)$.

$$\alpha_a^*(t) = \frac{\alpha_a(t) - \frac{\sum_{t_i \in P(m)} \alpha_a(t_i)}{|P(m)|}}{\sqrt{\frac{|P(m)| \sum_{t_i \in P(m)} (\alpha_a(t_i))^2 - (\sum_{t_i \in P(m)} \alpha_a(t_i))^2}{|P(m)|(|P(m)|-1)}}} \quad (2.2)$$

A Satisfação do Grupo é dado pela média das Satisfações Padronizadas $\alpha_a^*(t)$ de cada agente, como mostrado na equação 2.3.

$$\beta(t) = \frac{\sum_{a_i \in A} \alpha_{a_i}^*(t)}{|A|} \quad (2.3)$$

Finalmente, podemos definir o melhor horário possível para o agendamento de uma reunião m , denotado por m^+ , como o intervalo de tempo que maximiza β . Da mesma

forma, o pior horário possível para este agendamento, denotado como m^- , é o intervalo que minimiza β .

O Grau de Otimização do agendamento de uma reunião em um intervalo t é:

$$N(t) = \frac{\beta(\bar{t}) - \beta(\bar{m}^-)}{\beta(\bar{m}^+) - \beta(\bar{m}^-)} \quad (2.4)$$

O Grau de Otimização, apesar de ser eficiente, não é muito intuitivo para ser analisado. Devido a isso, utilizamos o Grau de Otimização para dispor as reuniões possíveis em uma lista ordenada de melhor a pior, e a posição que determinado intervalo ocupa na lista denominamos Ranking.

Desta forma, o intervalo ótimo tem Ranking igual a 1, o segundo melhor tem Ranking igual a 2 e assim sucessivamente. Utilizando o Ranking, ou simplesmente R , podemos dizer se a reunião agendada é a melhor possível, ou a segunda melhor possível, ou a terceira melhor possível, etc.

Assim, dado um conjunto de intervalos possíveis e seus Graus de Otimização $\{N(t_1) > N(t_2) > \dots > N(t_i)\}$, teremos seus Rankings $R(t_1) = 1, R(t_2) = 2, \dots, R(t_i) = i$.

Como realizamos alguns experimentos visando medir quão bom para os agentes é o intervalo escolhido para o agendamento e considerando que a Satisfação do Agente, assim como o Grau de Otimização, não é uma medida muito intuitiva, criamos também um Ranking para os agentes que denominamos Ranking Pessoal ou Rp .

O Rp nada mais é do que a ordenação dos intervalos possíveis baseada na Satisfação do Agente. Assim, tendo $\{\alpha_a(t_1) > \alpha_a(t_2) > \dots > \alpha_a(t_i)\}$ temos $Rp_a(t_1) = 1, Rp_a(t_2) = 2, \dots, Rp_a(t_i) = i$.

Para ilustrar o cálculo de nossa métrica, vamos supor uma reunião de uma hora a ser agendada com três participantes X, Y e Z e três possíveis intervalos t_1, t_2 e t_3 . Consideremos também que as preferências dos participantes em relação a estes intervalos são como demonstradas na tabela 2.1.

	$\alpha_X(t)$	$\alpha_Y(t)$	$\alpha_Z(t)$
t_1	6	5	5
t_2	6	2	3
t_3	5	4	10

Tabela 2.1: $\alpha(t)$ para cada participante.

Podemos encontrar a Satisfação Padronizada para cada participante calculada na tabela 2.2.

De acordo com a Satisfação do Grupo e o Ranking, que apresentamos na tabela 2.3, podemos verificar que o intervalo de tempo que é melhor para o grupo é o t_1 . Se não

	$\alpha_X^*(t)$	$\alpha_Y^*(t)$	$\alpha_Z^*(t)$
t_1	0,577350269	0,872871561	-0,277350098
t_2	0,577350269	-1,091089451	-0,832050294
t_3	-1,154700538	0,21821789	1,109400392

Tabela 2.2: $\alpha^*(t)$ para cada participante.

usarmos nossa métrica e procurarmos verificar intuitivamente qual dos intervalos é mais consensual, de certo escolheremos t_1 também.

	$\beta(t)$	N	R
t_1	0,390957244	1	1
t_2	-0,448596492	0	3
t_3	0,057639248	0,60298194	2

Tabela 2.3: $\beta(t)$, N e R para cada intervalo possível.

Em relação ao Ranking Pessoal dos *representatives* para a reunião agendada, temos que para X a reunião agendada é a melhor possível, $Rp = 1$, para Y a reunião também é a melhor possível, $Rp = 1$; e para Z é a segunda melhor possível, $Rp = 2$.

Como mencionamos anteriormente, utilizamos a normalização pois nos parece ser o método que melhor garante que o interesse do grupo seja preservado. Para ilustrar isto apresentaremos alguns exemplos onde a Satisfação do Grupo é calculada sem considerar a Satisfação Padronizada.

Vamos supor então a Satisfação de Grupo calculada das seguintes formas:

1. Através da média aritmética entre as Satisfações dos Agentes para cada intervalo:

$$\beta(t) = \frac{\sum_{a_i \in A} \alpha_{a_i}(t)}{|A|}.$$

2. Através da média geométrica entre as Satisfações dos Agentes para cada intervalo:

$$\beta(t) = \sqrt[|A|]{\prod_{a_i \in A} \alpha_{a_i}(t)}.$$

3. Através da média harmônica entre as Satisfações dos Agentes para cada intervalo:

$$\frac{1}{\beta(t)} = \frac{1}{|A|} \sum_{a_i \in A} \frac{1}{\alpha_{a_i}(t)}$$

4. Através da média aritmética entre as Satisfações dos Agentes para cada intervalo

$$\text{subtraído seu desvio padrão: } \beta(t) = \frac{\sum_{a_i \in A} \alpha_{a_i}(t)}{|A|} - \sqrt{\frac{|P(m)| \sum_{t_i \in P(m)} (\alpha_{a_i}(t_i))^2 - (\sum_{t_i \in P(m)} \alpha_{a_i}(t_i))^2}{|P(m)|(|P(m)|-1)}}$$

A tabela 2.4 traz o calculo do $\beta(t)$ para cada intervalo considerando os quatro métodos recém apresentados e as Satisfações dos Agentes do primeiro exemplo, tabela 2.1.

	$\beta^1(t)$	$\beta^2(t)$	$\beta^3(t)$	$\beta^4(t)$
t_1	5,333333333	5,313292846	5,294117647	4,755983064
t_2	3,666666667	3,301927249	3	1,585000667
t_3	6,333333333	5,848035476	5,454545455	3,11878308

Tabela 2.4: $\beta(t)$ para cada intervalo possível segundo os métodos alternativos.

Como podemos ver, a metodologia de cálculo do β utilizando as médias aritmética, geométrica e harmônica não apresenta o resultado que esperávamos selecionando o intervalo t_3 , ou seja, foi selecionado um intervalo não tão consensual como nossa métrica original que determinou o t_1 como melhor.

Contudo, o método que usa a média subtraída do desvio padrão parece se comportar da mesma forma como o que utiliza a padronização. Vamos supor um novo conjunto de preferências para os agentes como mostra a tabela 2.5.

	$\alpha_X(t)$	$\alpha_Y(t)$	$\alpha_Z(t)$
t_1	6	5	5
t_2	6	2	3
t_3	7	5	5

Tabela 2.5: Novo $\alpha(t)$ para cada participante.

Na tabela 2.6 temos o cálculo da satisfação do grupo utilizando a média subtraída do desvio padrão e o método original.

	$\beta(t)$	$\beta^4(t)$
t_1	0,19245009	4,755983064
t_2	-0,962250449	1,585000667
t_3	0,769800359	4,511966128

Tabela 2.6: Comparação entre o cálculo alternativo $\beta^4(t)$ e o original $\beta(t)$.

Como vemos, se calcularmos a Satisfação do Grupo como propusemos originalmente teremos o t_3 como o melhor intervalo. Porém, se utilizarmos o método em questão teremos t_1 como o escolhido, o qual assumimos intuitivamente, que não é melhor que t_3 . Assim, nenhum dos demais métodos estudados mostrou ser tão eficiente para os nossos propósitos quanto o que adotamos.

2.3 Conjunto de Protocolos de Negociação para o Agendamento de Reuniões

Mencionamos anteriormente que nosso conjunto de protocolos de negociação garante a privacidade e a consideração às preferências dos usuários. Como veremos na seção 2.2.1, o usuário terá suas preferências respeitadas e consideradas, pois elas são a medida de eficiência de nosso conjunto de protocolos, ou seja, determinado protocolo só é considerado eficiente se garante um alto grau de satisfação do grupo. Quanto a privacidade, vários fatores determinarão a forma como a negociação será conduzida.

Inicialmente, o participante pode ter restrições a quais de suas informações pessoais serão divulgadas, não querendo que os demais participantes conheçam seus horários livres ou suas preferências, ou até ambos em determinados casos.

Além disso, em outra dimensão, devemos considerar se as informações trocadas com o *host* serão divulgadas para todos os demais participantes. Isto irá determinar se o *representative* considerará as restrições de divulgação de informações pessoais do usuário em relação ao grupo todo ou ao *host* apenas.

Da mesma forma, em uma terceira dimensão, o usuário pode não confiar em determinado *host*. Isto pode ocorrer pelo fato do participante acreditar que existem truques na programação do *host*, a fim de favorecer um ou outro participante. Assim, o usuário pode impor a troca de *host* para outro *host* dos participantes, ou até para um *host* imparcial externo ao grupo, por exemplo, em algum lugar da Internet.

A primeira dimensão de privacidade, que chamamos modo de privacidade, determina que informação será divulgada durante a negociação e define quatro modos de negociação:

1. o participante não tem problemas em tornar disponível seus horários livres e suas preferências;
2. o participante pode disponibilizar seus horários livres, mas não suas preferências;
3. o participante pode disponibilizar suas preferências, mas não seus horários livres;
4. o participante não aceita tornar disponíveis seus horários livres e nem suas preferências.

A segunda dimensão, que chamamos de escopo, define dois modos de iteração:

- público - todas as informações trocadas entre os *representatives* e o *host* são divulgadas para os demais *representatives*;
- privado - a informação trocada entre o *host* e os *representatives* não pode estar disponível aos demais *representatives*.

Finalmente, a terceira dimensão define o *host* que será utilizado no processo de agendamento.

Para realizar um agendamento o usuário ativa seu *host* passando as informações sobre a reunião que deseja convocar. Este *host* será chamado de *pre-host* porque ainda não sabemos se será ele que irá intermediar toda a negociação. Então, o *pre-host* envia aos *representatives* uma chamada de agendamento. Cada *representative* responde ao *pre-host* com o *quantum* que utiliza. Logo após, o *pre-host* determina o *quantum* do agendamento, que será o maior entre os *quantums* enviados pelos *representatives*.

Assim, o *pre-host* compõe a 8-tupla com as informações sobre a reunião que será agendada. Na sequência o *pre-host* envia a todos a 8-tupla referente a reunião e, em seguida, o Protocolo de Negociação do Nível de Privacidade, que iremos detalhar na próxima seção, determina o modo de privacidade que será utilizado.

Com isso, um dos quatro protocolos que implementam os modos de privacidade, que também serão apresentados nas próximas seções, é utilizado para determinar o agendamento da reunião, se possível.

Com a finalidade de facilitar a compreensão dos protocolos mais rebuscados apresentados neste trabalho, elaboramos uma representação gráfica simples, que pode ser facilmente entendida. Esta representação acompanha a descrição em linguagem natural dos protocolos nas seções 2.3.1, 2.3.2, 2.3.3, 2.3.4 e 2.3.5.

2.3.1 Protocolo de Negociação do Nível de Privacidade

O primeiro estágio da negociação é chegar a um consenso sobre qual nível de privacidade utilizar.

Para isso, cada *representative* envia ao *pre-host*, depois de receber uma solicitação de agendamento, o escopo em que este pretende que se realize o processo de agendamento. O *pre-host* então determina no reticulado da figura 2.4 qual será o escopo da negociação e envia esta informação aos *representatives*.



Figura 2.4: Reticulado para a determinação do escopo da negociação.

Conhecendo o escopo, os *representatives* passam a utilizar o Protocolo de Negociação do Modo e *Host*, apresentado a seguir, para determinar o modo de privacidade. Caso algum dos *representatives* não concorde que o *pre-host* a intermedie a negociação, também será determinado o *host*.

Protocolo de Negociação do Modo e *Host*

Inicialmente cada *representative* verifica se seu usuário concorda que o *pre-host* seja o intermediário da negociação. Caso concorde, o *representative* verifica os modos privacidade nos quais está disposto a negociar e responde para o *pre-host* concordando com sua intermediação e enviando os modos aceitos. Caso não concorde, o agente deve enviar n , onde $n \geq 0$, sugestões de novos *hosts* para intermediarem a reunião, juntamente com os modos aceitos em relação a cada *host* sugerido.

O *pre-host* armazena as sugestões dos agentes de forma intercalada, com o objetivo de oferecer ao grupo uma sugestão de cada participante. Interações se sucederão até que todos concordem de maneira consensual que determinado *host* intermedie a negociação, onde, em cada interação, o *pre-host* apresenta para todos os *representatives* as sugestões de cada um, uma a uma.

É importante esclarecer que o *pre-host* indicado pelo participante que chamou a reunião pode intermediar a negociação do *host* sem a restrição dos participantes, uma vez que este participante é o maior interessado que a reunião que está sendo convocada se realize.

A negociação do *host* pode terminar sem que seja determinado um *host* consensual. Isto ocorrerá se algum dos agentes responder ao *pre-host* discordando sem enviar uma sugestão de um novo *host*. Assumimos, então, que se um agente não sugerir um novo *host* é porque seu usuário não concorda com mais nenhum *host* além dos que já sugeriu e que não foram aceitos pelos demais. Caso isto ocorra, o *host* envia uma mensagem de insucesso para indicar que não foi possível agendar a reunião em questão.

O primeiro *host* aceito por todos os participantes será o *host* do agendamento. Este novo *host* irá receber a reunião a ser agendada e os modos de privacidade e do escopo determinados, dando continuidade no processo de agendamento.

Juntamente com a negociação do *host* os agentes estabelecem o modo de privacidade. Para isso, os agentes utilizam as informações que recebem sobre as restrições de privacidade durante a negociação do *host* e calculam o mínimo do reticulado da figura 2.5.

O *representative* estabelece o modo de privacidade considerando que: se o escopo for público o *representative* verifica as restrições do seu usuário aos modos da primeira dimensão em relação aos demais participantes e ao *host*; ou se o escopo for privado, o agente verifica suas restrições apenas ao *host*.

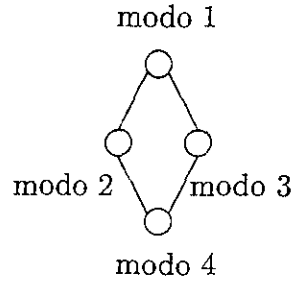


Figura 2.5: Reticulado para determinação do modo de privacidade.

2.3.2 Modo 1: Protocolo Direto

Este protocolo é apropriado quando nenhum dos participantes têm restrição a divulgar suas preferências e horários livres aos demais.

Cada *representative* envia ao *host* o conjunto de seus intervalos disponíveis $F(a_i)$, correspondente a reunião chamada, juntamente com suas preferências para estes intervalos. De posse destas informações, o *host* determina o conjunto $P(m)$ de intervalos livres comuns a todos os participantes e, usando as preferências, determina o melhor agendamento possível m^+ . Se o conjunto $P(m)$ é vazio o *host* retorna um sinal de insucesso a cada participante, caso contrário, retorna m^+ .

Garantimos que a melhor reunião será agendada pois utilizamos a informação de preferência do participante aplicadas nas equações de satisfação do agente e de satisfação de grupo que apresentaremos na seção 2.2.1, que acreditamos melhor atender as necessidades do sistema e que também utilizamos como métrica de avaliação para o conjunto de protocolos.

A figura 2.7 apresenta a representação gráfica do protocolo direto.

2.3.3 Modo 2: Protocolo de Votação

Este protocolo, apropriado ao modo 2 de privacidade, utiliza um método de votação para determinar o intervalo mais consensual aos participantes. Isto é necessário pois o *host* conhece apenas os horários livres dos participantes, não tendo acesso aos pesos atribuídos aos intervalos possíveis para o agendamento.

Garantir a máxima satisfação do grupo em relação ao resultado de uma eleição com múltiplos candidatos, que é o objetivo deste protocolo, não é uma tarefa simples.

Vamos supor que uma eleição com quatro alternativas A, B, C, D e cinco eleitores U, V, X, Y e Z, acontece no método mais simples e tradicional, conhecido como votação por Pluralidade, os eleitores U e V escolhem a alternativa A, o eleitor X prefere a B, o eleitor

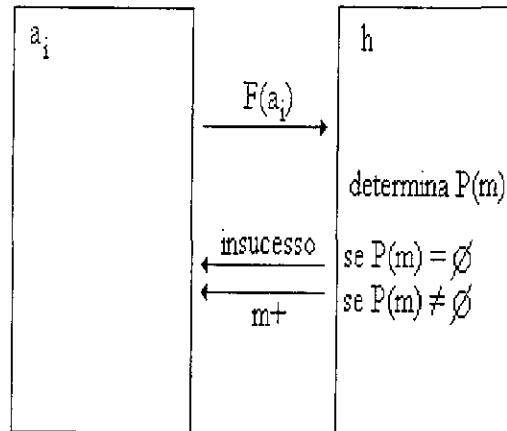


Figura 2.6: Representação gráfica do Protocolo Direto.

Y vota na C e o eleitor Z na D, como na tabela 2.7.

	A	B	C	D
U	✓			
V	✓			
X		✓		
Y			✓	
Z				✓

Tabela 2.7: Exemplo de votação.

Na contagem de votos, a alternativa A venceria com 2 votos e as demais empatariam em segundo com 1 voto. Suponhamos agora que a alternativa A seja considerada a pior por todos os outros eleitores que não o U e V. O resultado da eleição desagradaria 3 dos 5 eleitores, ou seja, a maioria deles.

Existem métodos de votação mais complexos e elaborados que resolvem este tipo de problema [12], sendo que o mais apropriado para este protocolo é o método Ranking, pois permite ao participante indicar sua preferência à todas as reuniões possíveis sem divulgar os valores de w para cada horário livre. Isto ocorre por que neste método de votação o eleitor manifesta seu voto ordenando os candidatos segundo sua preferência, como por exemplo $A > B > C > D$.

Contudo, o método Ranking oferece várias formas pelas quais o vencedor pode ser escolhido, permitindo ao *host* utilizar estas diversas formas para determinar a reunião a ser agendada.

Antes de prosseguirmos precisamos mencionar um conceito muito conhecido na teoria do voto, o candidato Condorcet [5]. O candidato Condorcet é o candidato que não perde de nenhum outro candidato se a eleição fosse realizada em pequenas eleições par-a-par decididas por Pluraridade. A divisão em muitas eleições pode ser eliminada utilizando a votação do tipo Ranking, que mencionamos anteriormente. É importante destacar que o candidato Condorcet pode não existir caso ocorra o chamado Paradoxo da Votação [15], o que indica que para todo o candidato há pelo menos um que o derrota.

Um método de seleção do vencedor para o Ranking que determina o Condorcet é denominado Copeland [13]. Caso o Condorcet não exista ainda assim o método Copeland determina um vencedor através do Escore de Copeland. Tal escore é a contagem de quantos candidatos cada um venceria ou empataria em uma eleição par-a-par. Um candidato que tenha seu escore de Copeland maior ou igual ao escore individual de todos os outros é denominado Copeland Winner. Se o escore desse candidato for igual a $(n - 1)$ onde n é o número total de candidatos, então ele também é um candidato Condorcet.

Outro método de seleção do vencedor chama-se Borda [2]. Este método recomenda que se pontue cada candidato com a posição no ranking de cada eleitor; aquele que obtiver menos pontos será o vencedor.

Desta forma, selecionamos os dois métodos de escolha do vencedor anteriormente mencionados, que consideramos bons representantes da maioria dos métodos, com o objetivo de determinar, através de experimentações, qual destes torna o protocolo de votação mais otimizado.

Durante a análise dos protocolos, que discutiremos mais adiante, verificamos que não há uma diferença significativa no Grau de Otimização resultante dos métodos de seleção do vencedor que optamos por experimentar. Desta maneira, resolvemos adotar para este protocolo aquele que possui a menor complexidade computacional.

A seguir podemos ver uma análise da complexidade dos métodos de seleção do vencedor discutidos nesta seção, onde $|A|$ é o número de participantes e $|P(m)|$ é o número de intervalos possíveis para o agendamento.

- Borda $O((|A| + 1).(|P(m)|))$;
- Copeland $O((|A| + 1).(|P(m)|)^2)$.

O protocolo de votação utiliza a forma de votação Ranking, selecionando o vencedor pelo método Borda.

Cada *representative* envia ao *host* seus intervalos disponíveis sem a informação de preferência w . O *host* retorna a cada *representative* o conjunto $P(m)$; caso $P(m) = \emptyset$ o *host* retorna uma mensagem que indica que um agendamento não é possível. Depois disso, cada *representative* ordena o conjunto de intervalos recebido do *host* em ordem

decrecente em relação a suas preferências. O *host*, então, usa o método Borda [2, 12] para determinar o intervalo preferido por todos e o retorna aos participantes.

Na figura 2.8 temos a representação gráfica deste protocolo.

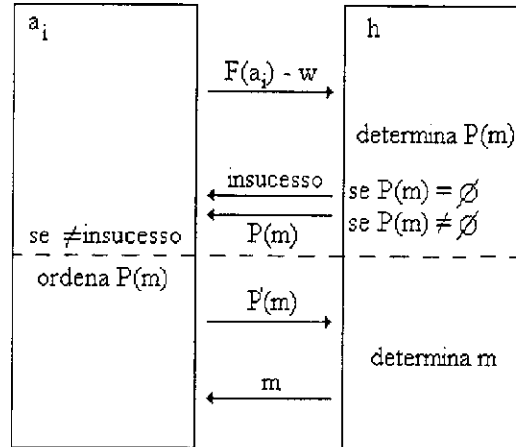


Figura 2.7: Representação gráfica do Protocolo de Votação.

2.3.4 Modo 3: Protocolo de Aprovação

Neste protocolo, onde apenas as preferências são divulgadas, cada agente envia para o host seu perfil de preferências, o qual consiste do peso para cada um dos intervalos existentes entre wb e we , não importando se está livre ou não. Assumimos que o usuário pode ter padrões de preferências diferentes para diferentes dias.

O *host* considera que todos os intervalos entre wb e we estão livres e calcula a satisfação do grupo para cada intervalo. O *host* então envia um intervalo de tempo de cada vez para que os *representatives* aceitem se este estiver livre e rejeitem caso contrário. Uma vez que todos os *representatives* concordem com um intervalo este será enviado a todos e a reunião agendada.

Como o cálculo da Satisfação do Grupo é feito a cada interação, o intervalo que todos os participantes possuem em comum primeiro será o melhor intervalo possível, que é o mesmo que o protocolo direto determina. Assim, o protocolo de votação também determina a reunião ótima m^+ , caso haja uma.

Podemos ver na figura 2.9 a representação gráfica deste protocolo.

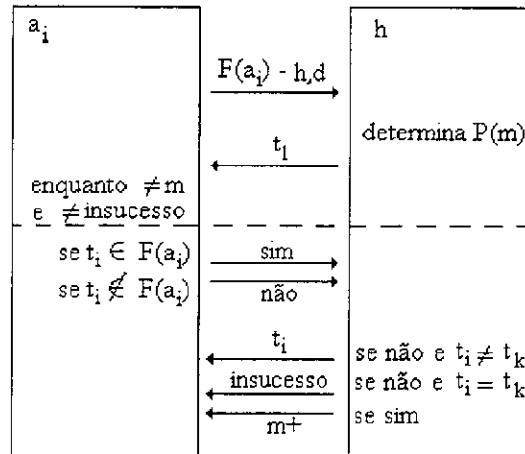


Figura 2.8: Representação gráfica do Protocolo de Aprovação.

2.3.5 Modo 4: Protocolo de Negociação

Apropriado para quando não há informação disponibilizada, o protocolo de negociação é baseado na idéia intuitiva de sugestões.

Em cada interação da negociação cada *representative* a_i deve mandar uma nova sugestão para o agendamento $\in F(a_i)$. O *host* recebe as sugestões de todos os *representatives*, as agrupa e envia de volta aos *representatives* para que estes as considerem para realizarem suas próximas sugestões.

Desta forma, os *representatives* apresentam suas sugestões, diferentes a cada ciclo de interação, ou um sinal de que não tem mais nada para sugerir $\perp$, até que o *host* determine um horário em comum para realizar o agendamento. O *host* armazena todos os intervalos de tempo que foram sugeridos por cada *representative* separadamente e o agendamento é determinado quando um destes intervalos aparece na lista de intervalos de todos os agentes simultaneamente. O *host* verifica esta condição toda vez que recebe um novo conjunto de sugestões dos *representatives*.

Devemos destacar que mais de um intervalo em comum pode ser encontrado pelo *host* e isto gera a necessidade de determinarmos qual dentre eles é o melhor para o grupo. Para isso realizamos uma votação em Ranking utilizando o método Borda de seleção do vencedor que discutimos na seção 2.3.3. O *host* envia para os *representatives* a lista de intervalos possíveis para que estes a ordenem de acordo com suas preferências e a recebe de volta, determinando o intervalo no qual a reunião será agendada.

Se todos os *representatives* enviarem o sinal $\perp$ para o *host* o agendamento não será realizado pois não existem horários livres em comum.

Assim, mais formalmente, cada *representative* a_i envia um intervalo possível para o agendamento $t_{ij} \in F(a_i)$. Na interação j o *host* envia o conjunto de intervalos S^j sugerido por todos os *representatives* nesta interação, sendo que na primeira delas $S^1 = \{t_{10}\}$.

Cada agente a_i então responde com o conjunto R_i^j tal que $R_i^j \subseteq F(a_i)$, $R_i^{j-1} \subseteq R_i^j$ e $R_i^{j-1} = R_i^j$ se e somente se $R_i^j = F(a_i)$. Caso exista um intervalo $\hat{x} \in \cap_i R_i^j$ a reunião será agendada neste horário. Se para todos *representatives* i $R_i^{j-1} = R_i^j$ e $\cap_i R_i^j = \emptyset$ o *host* responde aos *representatives* dizendo que a reunião não pode ser agendada. Na interação $j + 1$, o *host* envia $S^{j+1} = \cup R_i^j$.

Este protocolo sempre converge para uma solução caso ela exista, não sendo necessariamente a ótima. Isto ocorre pois os *representatives* sempre devem responder ao *host* com alguma sugestão em cada interação; então se houver algum horário em comum depois de $\max |F(a_i)|$ interações todos os *representatives* terão sugerido este intervalo e a reunião será agendada.

Na figura 2.10 podemos ver a representação gráfica do protocolo de negociação.

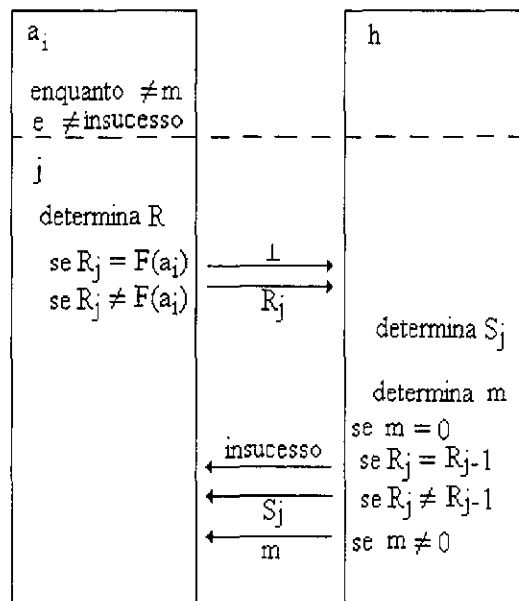


Figura 2.9: Representação gráfica do Protocolo de Negociação.

Estratégias Alternativas

O protocolo de negociação é o único de nosso conjunto de protocolos que permite variações de estratégias dos *representatives* durante a negociação.

Um *representative* pode ser egotísta, tentando agendar a reunião nos seus melhores intervalos mesmo que isso aumente as informações que este estará divulgando principalmente em negociações públicas.

Nesta estratégia o *representative* ordena seu conjunto de intervalos livres na ordem decrescente de sua preferências $\langle x_1, x_2 \dots x_k \rangle$, enviando como sugestão em cada interação j o intervalo x_j .

Por outro lado, um *representative* pode ser lacônico, preferindo não divulgar muitas informações em troca de agendar um reunião em um intervalo que não seja muito bom de acordo com suas preferências.

Esta estratégia baseia-se na idéia de sugerir intervalos que o *representative* possui disponível e que já foram sugeridos pelos demais *representatives* em interações anteriores. Assim, na interação j , se S^j é o conjunto de sugestões feitas na interação anterior, se $X = \langle x_1, x_2 \dots x_k \rangle$ é o conjunto ordenado dos intervalos livres para o *representative* e se R^{j-1} o conjunto de sugestões que o *representative* fez em interações anteriores, então o *representative* irá sugerir:

- se $X \cap (S^j - R^{j-1}) \neq \emptyset$, o intervalo $x_i \in X \cap (S^j - R^{j-1})$ com mais alta preferência em X ;
- se $X \cap (S^j - R^{j-1}) = \emptyset$, o intervalo $x_i \in X - R^{j-1}$ com mais alta preferência em X .

Desta forma o *representative* lacônico prefere não divulgar informações a seu respeito durante a negociação, mas se isto se fizer necessário este sugere seu melhor intervalo possível.

Ainda podemos ter um *representative* enganador que se comporta de forma parecida com o lacônico, porém este não oferece seu melhor intervalo caso não possa repetir uma sugestão. Nesta estratégia é preferível divulgar menos informação possível. Utilizando a mesma notação da estratégia lacônica temos:

- se $X \cap (S^j - R^{j-1}) \neq \emptyset$, o agente irá sugerir um intervalo randomicamente $x_i \in X \cap (S^j - R^{j-1})$;
- se $X \cap (S^j - R^{j-1}) = \emptyset$, o agente irá sugerir um intervalo randomicamente $x_i \in X - R^{j-1}$.

Por fim, podemos ter também um *representative* que apenas se preocupa com o grupo, o qual chamamos altruísta.

Consideramos que este *representative* consegue, ao receber as informações de um agendamento com informações públicas, aprender sobre as preferências dos demais *representatives*. Com isso, este *representative* determina o melhor horário para o grupo, como o *host* no Protocolo de Aprovação o faz, e a cada interação sugere este intervalo.

Caso o intervalo não seja aceito isto significa, como no Protocolo de Aprovação, que algum dos participantes tem este horário ocupado e na próxima interação o *representative* irá recalcular o melhor intervalo, sem contar com o rejeitado, e fará nova sugestão, até que o protocolo convirja em uma solução.

Os quatro tipos de *representatives* apresentados resumem as diferentes estratégias que podem ser adotadas pelo participante quando o Protocolo de Negociação estiver sendo usado.

2.4 Cancelamento e Reagendamento de Reuniões

Além de agendar reuniões, através do conjunto de protocolos apresentado nas seções anteriores, acreditamos que o cancelamento e o reagendamento de reuniões são fundamentais para a eficiência de um sistema de agendamento automático de reuniões.

Quanto ao cancelamento de reuniões, devemos considerar que os participantes de uma reunião normalmente não tem o poder de determinar que uma reunião não mais aconteça, ou seja, um participante apenas pode cancelar sua participação na reunião. Apenas o participante que convocou a reunião pode cancelá-la. Para permitir o cancelamento de reuniões desenvolvemos o protocolo de cancelamento.

Já em relação ao reagendamento de reuniões, a principal questão que devemos considerar é que o interesse em reagendar uma reunião surge para um participante quando se torna vago um intervalo em sua agenda, que não estava disponível quando a reunião em questão foi agendada, e que irá melhor atender as suas preferências. Com isto, é do interesse deste participante estabelecer um processo de reagendamento.

Devemos ressaltar que o reagendamento garante que a reunião seja reagendada em um intervalo melhor ou tão bom para o grupo quanto o agendamento primeiramente realizado. Consideremos o fato de que os protocolos agendam sempre o melhor horário para o grupo, que não necessariamente é o ótimo mas que é o melhor possível segundo as restrições de privacidade adotadas na negociação, e o intervalo em que a reunião está agendada é considerado livre para o reagendamento.

Assumamos que um intervalo torna-se vago na agenda de algum participante, e este inicia o reagendamento. No reagendamento será agendada a melhor reunião possível para o grupo, que novamente não necessariamente é a ótima mas que é o melhor possível segundo as restrições de privacidade adotadas na negociação. Assim ou a reunião será reagendada no mesmo intervalo ou em um intervalo melhor que o anterior considerando a configuração da agenda neste momento.

Porém, existem fatores independentes das preferências e disponibilidade dos participantes que são importantes no reagendamento.

Imaginemos que na reunião que determinado participante pretende reagendar, outro

participante irá realizar uma apresentação de um relatório. Caso a reunião seja adiantada pelo reagendamento pode ficar inviável para o apresentador preparar seu material, uma vez que já estava em seus planos a apresentação no dia em que a reunião tinha sido agendada primeiramente. Nesta situação o participante deve interferir no reagendamento para que a reunião só possa ser adiada, para não prejudicá-lo.

Outro exemplo para estes fatores externos é a dependência de equipamentos locados para o evento. Consideremos que um sistema de projeção visual foi alugado para o dia da reunião. Desta forma, apenas o horário da reunião pode ser alterado, já que modificar seu dia irá implicar em novo aluguel do equipamento.

Com isto, verificamos que deve ser facultado a todos participantes o direito de determinar se a reunião pode ser reagendada e qual o novo período possível para este reagendamento. Para permitir o reagendamento de reuniões, considerando as questões apresentadas, desenvolvemos o protocolo de reagendamento.

2.4.1 Protocolo de Cancelamento

O *host* do participante, que precisa cancelar sua presença na reunião, envia uma mensagem a todos *representatives* anunciando que este não mais tomará parte na reunião. Para isto o *host* do desistente envia a mensagem de cancelamento, a descrição de seu participante, pertencente a A , e o intervalo de agendamento da reunião m que este não mais participará.

Caso o *host* do participante que convocou a reunião envie a mensagem de desistência, que descrevemos acima, a reunião então é cancelada. Desta forma, todos os *representatives* devem remover a reunião das agendas de seus usuários e disponibilizar o horário para novos agendamentos.

Cabe a cada *representative* comunicar a seu usuário a desistência de outros participantes. Sabendo da desistência, os outros participantes podem também desistir, ou o participante que convocou a reunião pode cancelar a reunião, dependendo da importância do desistente para o grupo ou para a reunião. Mas é importante lembrar que esta é uma decisão do usuário e o seu agente não tomará nenhuma atitude automaticamente nestes casos.

2.4.2 Protocolo de Reagendamento

O reagendamento é um recurso que os participantes devem usar apenas quando uma reunião é cancelada, como discutimos anteriormente. Para determinar que deve começar um processo de reagendamento os *hosts* dos usuários verificam se o horário que se tornou disponível é preferido em relação ao que se encontra qualquer uma de suas reuniões.

Uma vez que se determinou a situação mencionada acima, o *host* do participante que verificou tal situação envia a todos os *representatives* uma solicitação de reunião, como

descrito na seção 2.3, diferente do agendamento apenas por um sinal indicando que será feito um reagendamento.

Os *representatives* consultam as restrições que cada usuário tem ao reagendamento de determinada reunião. Estas restrições apenas determinam o intervalo de agendamento que o usuário concorda para este reagendamento.

Cada *representative* então envia uma mensagem ao *host* composta por um par $\langle we, wb \rangle$, como em m , que delimitam o novo intervalo de agendamento segundo suas restrições. Em seguida, o *host* determina a interseção dos intervalos enviados pelos agentes e estabelece um novo período de agendamento para a reunião. Desta forma, o *host* envia aos *representatives* o novo período para o agendamento, que substituirá o período antigo, e o sistema passará a utilizar o conjunto de protocolos de agendamento.

Devemos destacar que durante todo o processo de agendamento que acontecerá então, por se tratar de um reagendamento, o horário onde a reunião está agendada entrará na lista de horários livres para as negociações.

2.5 Coalizões

Vamos considerar um grupo de participantes de uma reunião qualquer, onde alguns destes participantes têm uma relação pessoal mais forte. É perfeitamente aceitável, e ocorre muitas vezes, que estes participantes combinem entre si intervalos para o agendamento que melhor os satisfaçam e, uma vez no grupo a que pertencem, todos defendem a mesma posição e têm maiores chances de que seu intervalo preferido seja agendado.

Chamamos este processo de coalizão. Em nosso conjunto de protocolos estudamos a possibilidade da existência de coalizões e quais suas implicações para o grupo.

As coalizões podem ser formadas para que a negociação dentro da coalizão seja feita em um nível menos restrito de privacidade, garantindo uma maior satisfação para os sub-grupos, que negociam num nível mais restrito de privacidade com os demais participantes.

Sabendo que o Protocolo Direto e o de Aprovação resultam em um agendamento ótimo, como comentamos nas seções 2.3.2 e 2.3.4, não há nenhum motivo para que os agentes realizem coalizões quando o grupo negocia com este protocolos.

De acordo com esta análise concluímos que só pode ser interessante para os participantes estabelecer coalizões quando os Protocolo de Negociação e de Votação estiverem sendo utilizado no grupo. Contudo, resolvemos estudar apenas as coalizões com o Protocolo de Negociação pois, como apresentaremos na seção 2.6.2, o protocolo de votação apresenta bons resultados para o grupo e individualmente, não gerando a necessidade de uma subnegociação.

Assim, um determinado número, menor que $|A|$, de participantes de uma reunião negocia utilizando os protocolos Direto, de Negociação ou de Aprovação e determina as

informações que todos irão utilizar no protocolo de Negociação que está sendo usado pelo grupo todo.

Podemos encarar as coalizões como prejudiciais para o grupo, uma vez que pequenos subgrupos podem se formar com o objetivo de melhor se satisfazerem e de alguma forma prejudicarem a satisfação do grupo como um todo. Uma coalizão pode, por exemplo, se formar entre todos os participantes do grupo a exceção de um. Este participante que não pertence a coalizão pode ser muito prejudicado na negociação se todos investirem em um intervalo que seja o pior para ele.

Tendo feito esta análise verificamos que a formação de coalizões não aumenta o poder de influência na decisão que o grupo tomará. Isto é possível pois neste protocolo mesmo que um intervalo seja sugerido mais de uma vez, este só aparecerá uma vez na lista de intervalos sugeridos que os agentes usam para tomarem suas decisões.

Assim, nosso conjunto de protocolos pode ser dito livre de coalizões, não oferecendo vantagens para a formação de subgrupos.

2.6 Análise dos Protocolos

O conjunto de protocolos que estamos apresentando foi implementado em um ambiente de simulação, com o objetivo de testar e medir sua eficiência de acordo com nossas métricas de avaliação.

2.6.1 Simulador

Antes de analisarmos o conjunto de protocolos, devemos descrever o ambiente de simulação que foi utilizado nas experimentações.

Todos os algoritmos de simulação foram implementados na linguagem C++ em LINUX e executados em um micro computador 586(266Mhz) com 32MB de RAM.

Desenvolvemos um gerador de agendas aleatórias que gera agendas preenchidas aleatoriamente segundo uma porcentagem de ocupação determinada pelo simulador e quatro padrões de distribuição de pesos.

Consideramos que todos os agentes utilizam *quantum* de uma hora. Cada agenda possui um mês de tamanho (31 dias), com cinco dias por semana, e cada dia é subdividido em onze intervalos de uma hora ininterruptos, resultando em 341 intervalos.

Os quatro padrões de distribuição dos pesos nestas agendas, que consideramos mais propícios para os testes, estão na tabela 2.8.

O simulador propriamente dito gera reuniões aleatórias para as quais o gerador de agendas cria e preenche as agendas.

horário	padrão1	padrão2	padrão3	padrão4
8h	0	10	2	10
9h	1	9	4	8
10h	2	8	6	6
11h	3	7	8	4
12h	4	6	10	2
13h	5	5	10	2
14h	6	4	8	4
15h	7	3	6	6
16h	8	2	4	8
17h	9	1	2	10
18h	10	0	2	10

Tabela 2.8: Padrões de preferências utilizados no gerador de agendas.

Cada reunião m deve ser agendada no intervalo $wb = 01/05/1999$ $we = 31/05/1999$ e seu tamanho l é gerado aleatoriamente segundo a seguinte distribuição de probabilidade $p(l)$:

$$p(1) = 60\%, p(2) = 30\%, p(3) = 10\%$$

Em uma experimentação são agendadas 20 reuniões para cada porcentagem de ocupação de agenda, que pode variar entre 0% e 100% crescendo de cinco em cinco por cento. O simulador só considera reuniões com mais de uma possibilidade de agendamento, fazendo com que agendas sejam geradas até que se consiga no mínimo dois intervalos comuns nas agendas de todos os participantes.

O simulador retorna o Grau de Eficiência N e o Ranking R de cada protocolo para cada porcentagem de ocupação.

2.6.2 Resultados Obtidos

Probabilidade de Agendar um Reunião

Um algoritmo para realizar a negociação de acordo com o primeiro modo de privacidade é computacionalmente simples e leva à escolha do melhor horário possível para o agendamento. Contudo, determinar quando um agendamento será possível ou não, depende do número de participantes da reunião e do nível de ocupação de suas agendas.

Realizamos um conjunto de simulações para determinar para quantos participantes e para qual nível de ocupação das agendas ainda conseguimos um intervalo livre comum para realizar o agendamento. A figura 2.11 mostra a porcentagem de sucessos em agendamentos tentados, de acordo com a porcentagem de ocupação das agendas, para 5, 10 e 15 participantes.

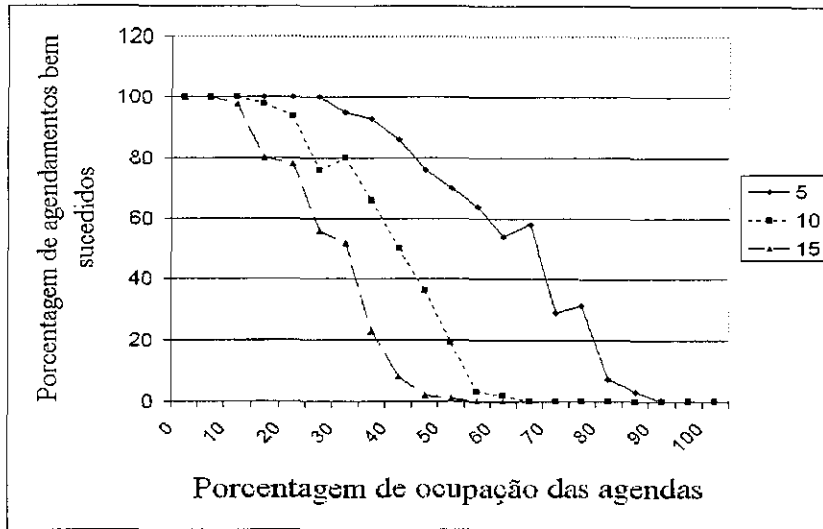


Figura 2.10: Probabilidade de agendar uma reunião.

De posse desta informação passamos a realizar nossas simulações apenas até 70% da ocupação pois, como já mencionamos, o simulador trabalha com agendamentos onde existem no mínimo dois intervalos possíveis. Além disso, não realizamos outras simulações com 10 e 15 participantes pois acreditamos que cinco participantes são suficientes para a análise dos protocolos uma vez que o aumento no número de participantes apenas diminui a probabilidade de se agendar a reunião, não interferindo no comportamento geral do sistema.

Eficiência do Protocolo de Votação

O protocolo de votação é muito eficiente, garantindo em média o agendamento da segunda melhor reunião e N médio de 0,92. Apresentamos na figura 2.12 o gráfico com o N e o Ranking para cada porcentagem de ocupação das agendas.

Como comentamos na seção 2.3.3 analisamos este protocolo utilizando dois métodos diferentes de seleção do vencedor. Como também dissemos anteriormente, estes métodos não apresentaram variação significativa em termos de eficiência.

Na figura 2.13 temos o Ranking da mesma simulação utilizando o método Borda e o método Copeland.

Para comprovar que não é significativa a pequena variação entre os resultados obtidos pelos dois métodos aplicamos um teste estatístico denominado Análise de Variância [17]. Este teste serve essencialmente para se verificar se existe variação entre médias, ou equivalentemente, testar a hipótese de nulidade de que as médias são todas iguais. Esta

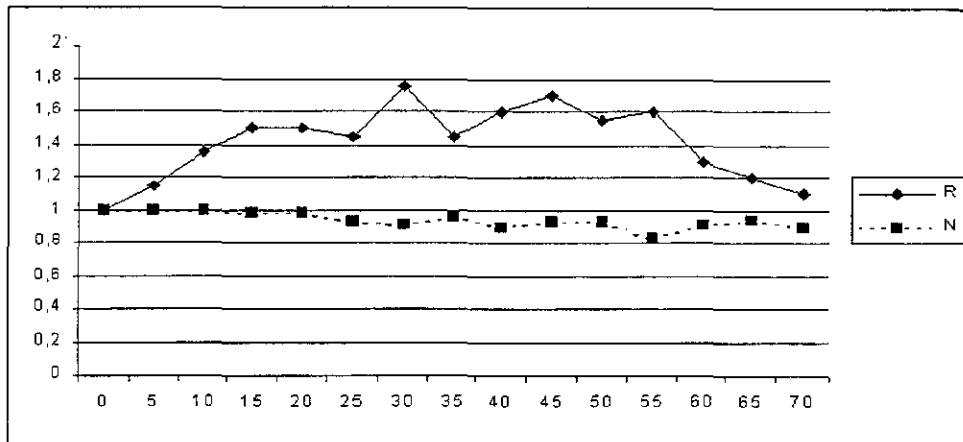


Figura 2.11: Análise do Protocolo de Votação.

técnica utiliza a distribuição F .

Assim, calculamos a média dos Rankings entre todas as porcentagens de ocupação para o método Borda e Copeland, $\frac{409}{300}$ e $\frac{403}{300}$ respectivamente, e calculamos o F , cujo resultado foi $F = 0,07955$. Com isso, segundo a teoria da Análise de Variância, como $F_{0,95} = 3,84$ e $F < 3,84$ não podemos rejeitar a hipótese nula de que as médias são iguais. Esta análise feita com $F_{0,95}$ significa que desejamos verificar a probabilidade de igualdade das médias com uma probabilidade de 95% de acerto, ou seja, que a chance de existir diferença entre as médias é de 5%.

Com este teste e o resultado obtido comprovamos estatisticamente que não existe diferença significativa na eficiência entre os diferentes métodos de seleção do vencedor, o que nos permitiu adotar o menos complexo computacionalmente.

Estratégias para o Protocolo de Negociação

O Protocolo de Negociação, como já explicamos, permite que sejam adotadas diferentes estratégias na negociação. Por este motivo, realizamos simulações para comparar a eficiência das diferentes estratégias em relação ao grupo e em relação a cada agente individualmente.

Inicialmente estabelecemos que todos os *representatives* de uma reunião utilizam a mesma estratégia, que chamamos de negociação simples. Com isso, realizamos uma simulação com o objetivo de comparar as estratégias quanto ao Ranking obtido. Na figura 2.14 apresentamos o Ranking para cada porcentagem de ocupação para as quatro estratégias que detalhamos na seção 2.3.5.

Devido a variação natural dos Rankings, optamos por realizar uma Regressão Linear

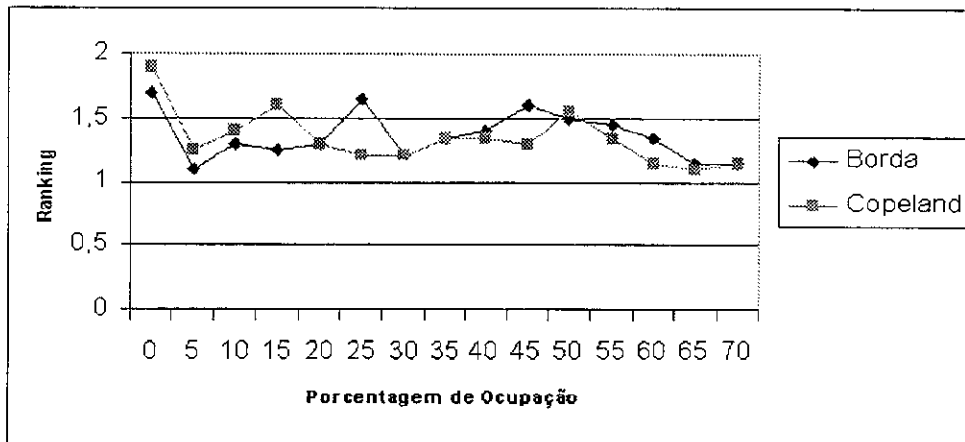


Figura 2.12: Comparação entre os métodos Borda e Copeland.

[17] para facilitar a comparação visual entre as estratégias. A figura 2.15 traz a Regressão Linear das curvas da figura 2.14.

Como podemos observar, o grupo de *representatives* altruístas conseguem agendamentos com média muito próxima do ótimo. Isto ocorre pois, conhecendo a preferência de todos os participantes, estes atuam na negociação com o objetivo de melhor satisfazer o grupo.

Este tipo de estratégia é interessante, porém só pode ser usada se os agentes já aprenderam as preferências dos demais participantes e para isto ocorrer as informações tem de ser públicas. Para a simulação fizemos os *representatives* altruístas acessarem diretamente as informações de preferência dos outros participantes.

Contrariamente, os *representatives* enganadores, por tentarem preservar as informações de seus usuários, acabam agindo na negociação de forma a aceitar qualquer intervalo e sugerir intervalos que não são seus melhores, fazendo com que as reuniões agendadas não sejam boas para o grupo. No gráfico observamos que o grupo de enganadores foi quem obteve o pior resultado.

Já os *representatives* lacônicos e egoístas apresentaram resultados que, além de muito semelhantes, atingiram o melhor Ranking, desconsiderando o caso especial dos altruístas que têm algum conhecimento sobre os outros. Assim, apesar dos Rankings parecerem ruins a primeira vista, conseguimos em média a quarta melhor reunião, o que consideramos satisfatório dadas as altas restrições de privacidade.

Utilizamos a Análise de Variância primeiramente discutida na seção 2.6.2 para verificar se existe alguma diferença significativa entre as médias dos Rankings obtidas com os lacônicos e com os egoístas. As médias entre os Rankings são $\frac{939}{300}$ e $\frac{919}{300}$ para os egoístas

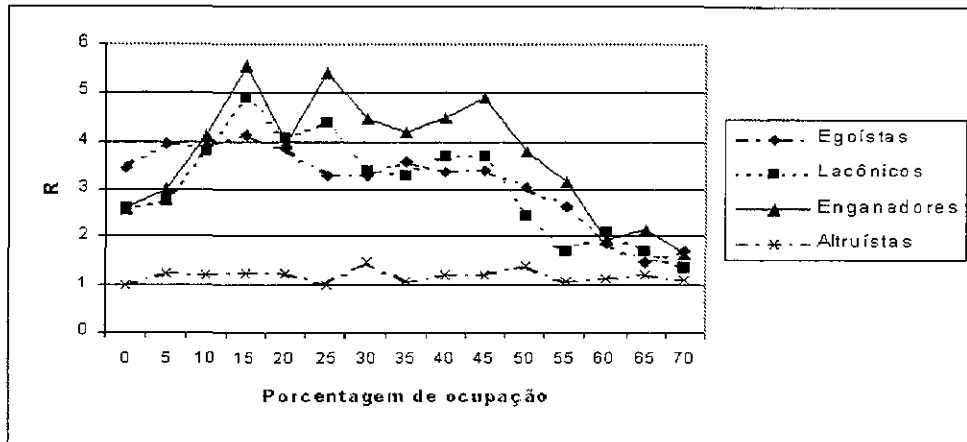


Figura 2.13: Comparação entre estratégias.

e lacônicos, respectivamente, e encontramos $F = 0,10239$. Assim, como $F_{0,95} = 3,84$ e $F < 3,84$ não podemos rejeitar a hipótese nula de que as médias são iguais. Desta forma, não existe diferença na eficiência das estratégias egoísta e lacônica.

Fizemos o mesmo teste para verificar se a diferença que visualizamos entre os lacônicos e os enganadores é significativa. A média dos Rankings dos enganadores é $\frac{1109}{300}$ e $F = 6,3109$. Desta maneira, como $F_{0,95} = 3,84$ e $F > 3,84$ podemos rejeitar a hipótese nula de que as média são iguais. Podemos afirmar então que as médias analisadas possuem um diferença que não pode ser desprezada.

Como os lacônicos e os egoístas não apresentaram diferença quanto a sua eficiência medida em relação ao Ranking e estes apresentaram um bom resultado, misturamos um egoísta a um grupo de lacônicos para estudar suas implicações. Chamamos a negociação com diferentes estratégias sendo utilizadas simultaneamente de negociação mista. Na figura 2.16 apresentamos a Regressão Linear das curvas dos Rankings de uma simulação utilizando um grupo com todos lacônicos, um grupo com todos egoístas e um grupo com quatro lacônicos e um egoísta.

Visualmente verificamos que é pequena a diferença entre as curvas e que a reta do grupo misto está encerrada pelas outras retas. As médias dos Rankings são para os lacônicos $\frac{850}{300}$, para o grupo misto $\frac{927}{300}$ e para os egoístas $\frac{901}{300}$, e $F = 0,7022$. Como $F_{0,95} = 3$ e $F < 3$ não podemos rejeitar a hipótese nula de que as médias são iguais. Assim, utilizando a Análise de Variância comprovamos que em relação ao grupo não há diferença se misturarmos um egoísta aos lacônicos.

A figura 2.17 mostra os Rankings Pessoais para os cinco *representatives* em uma simulação com um grupo de lacônicos. A diferença que observamos entre o *representative*

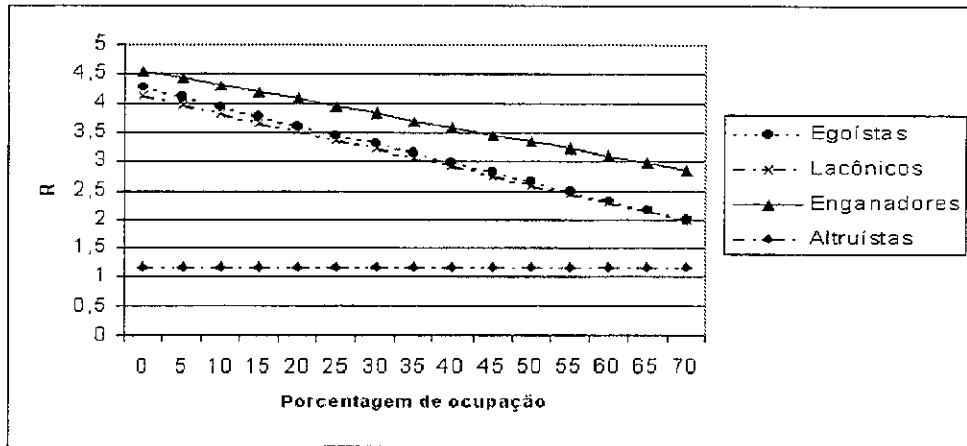


Figura 2.14: Comparação entre estratégias utilizando Regressão Linear.

1 em relação aos demais se justifica pois é ele que primeiro sugere um intervalo e como os lacônicos aceitam qualquer intervalo desde que este esteja disponível, os demais acabam aceitando a sugestão do primeiro. A diferença diminui com o aumento da ocupação das agendas pois os demais participantes passam a concordar menos à medida que tem menos horários em comum com o *representative* 1. Por este motivo o egoísta foi inserido no grupo de lacônicos substituindo o altruísta 2 para evitar a influência deste detalhe do protocolo em nossa avaliação.

Os agentes no grupo de egoístas tem seus Rankings Pessoais praticamente iguais, como podemos ver na figura 2.18.

Porém, na figura 2.19 podemos verificar que o *representative* que utiliza a estratégia egoísta entre os demais lacônicos leva uma vantagem significativa uma vez que seu Ranking Pessoal é muito melhor se comparado aos demais.

Podemos concluir, baseado nestes fatores, que não existem vantagens em ser lacônico nem para o grupo e nem individualmente, pois para o grupo não há diferença entre as duas estratégias e individualmente os lacônicos levam desvantagem se um egoísta estiver presente na negociação.

As estratégias que apresentaram melhor resultado foram a altruísta e a egoísta, por isso vamos mesclar *representatives* egoístas e altruístas para medir o que ocorre em uma negociação quando estas estratégias interagem. Na figura 2.20 temos o gráfico com a Regressão Linear das curvas resultantes de uma simulação com um grupo de altruístas, um grupo de egoístas e um grupo com quatro altruístas e um egoísta.

Como já esperávamos, a presença de um egoísta diminuiu o N piorando o Ranking médio da reunião agendada. Contudo, podemos observar que as reuniões agendadas pela

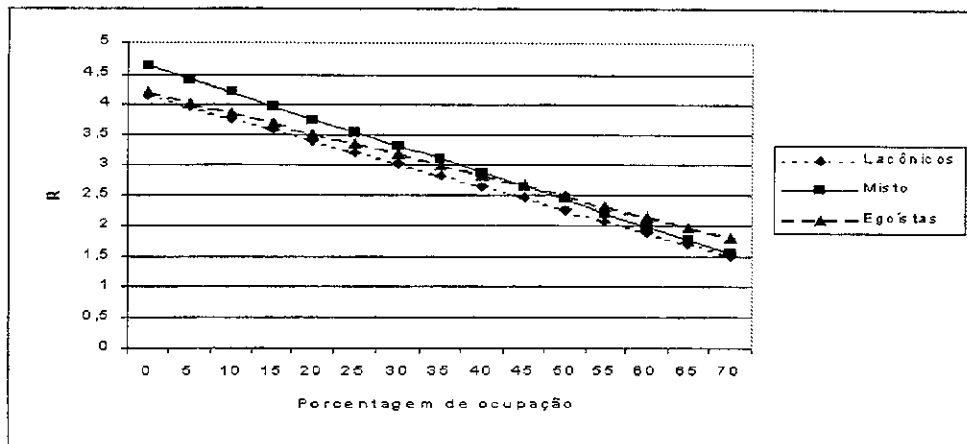


Figura 2.15: Comparação entre as estratégias lacônicas, egoísta e mista (1 egoísta e 4 lacônicos).

mistura das estratégias não foram em média piores que as reuniões agendadas pelo grupo de egoístas. Assim, combinar estas estratégias, onde existem *representatives* que não aprenderam as preferências dos demais e *representatives* mais inteligentes que têm este conhecimento, é perfeitamente aceitável e não prejudica a eficiência do protocolo.

Analisando as figuras 2.18, 2.22 e 2.21 podemos verificar que o egoísta tem vantagem pessoal sobre os altruístas na negociação mista, enquanto que nas negociações simples com altruístas e egoístas nenhum *representative* leva vantagem.

Assim, só é interessante para o *representative* ser lacônico se todos o forem. Como não é possível saber qual a estratégia que os demais *representatives* adotaram, o *representative* deve utilizar a estratégia egoísta para atingir os melhores índices de satisfação pessoal.

Ineficiência da Coalizão

Na seção 2.5 garantimos que nosso Protocolo de Negociação não oferece vantagens que justifiquem a formação de coalizões. Para ilustrar isto realizamos uma simulação onde dois ou mais agentes participam de uma coalizão negociando através do Protocolo Direto no subgrupo e no Protocolo de Negociação no grupo, cujos resultados podemos ver nos gráficos das figuras ??, ??, ?? e ??.

Como podemos observar, os *representatives* que tomam parte na coalizão sempre têm um Ranking Pessoal pior para a reunião agendada do que aqueles que estão fora da coalizão. Desta forma vemos que ao invés de aumentar o poder de influência dos participantes

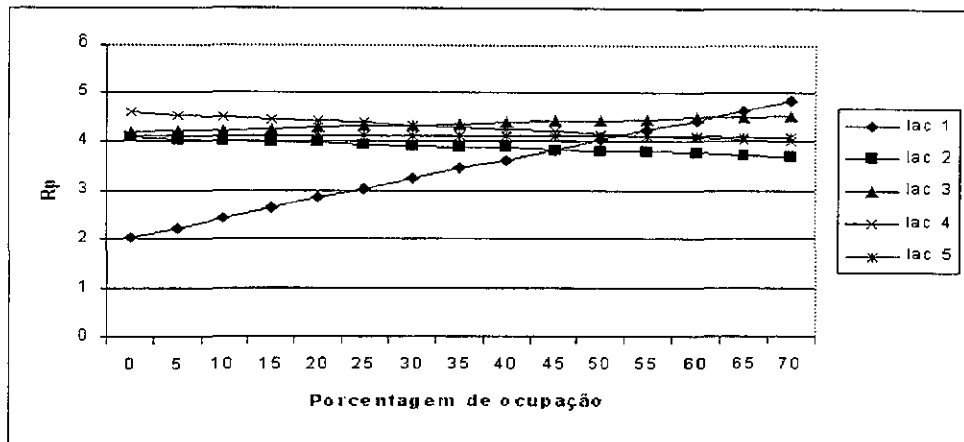


Figura 2.16: Negociação entre lacônicos.

a coalizão acaba fazendo o contrário.

Não desenvolvemos o Protocolo de Negociação com a intenção de prejudicar os participantes membros de uma coalizão. Contudo verificamos que isto ocorre através desta simulação e conseguimos identificar o motivo o porquê.

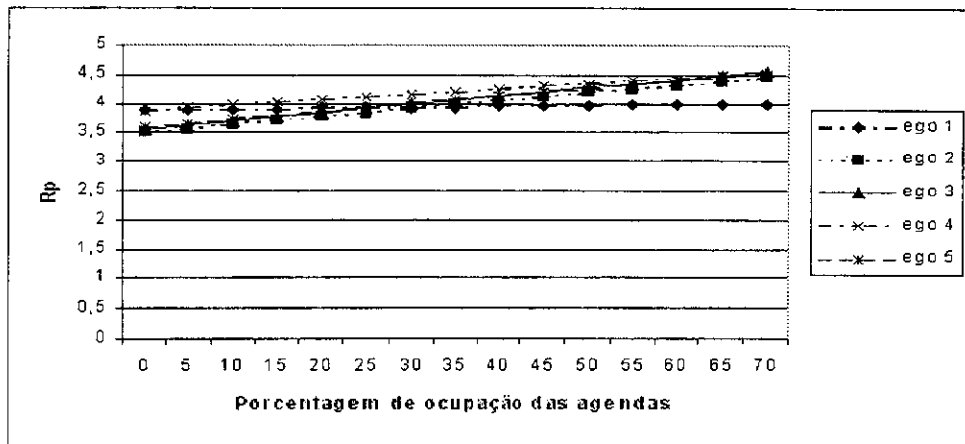


Figura 2.17: Negociação entre egoístas.

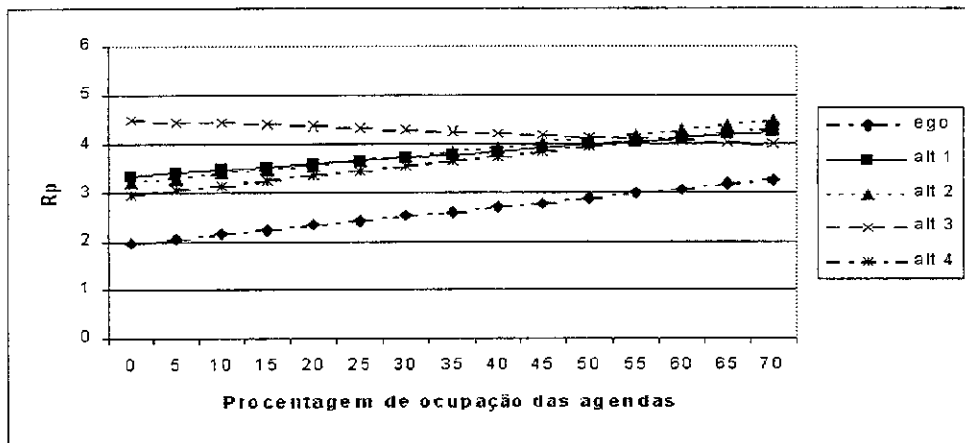


Figura 2.18: Negociação entre 4 lacônicos e 1 egoísta.

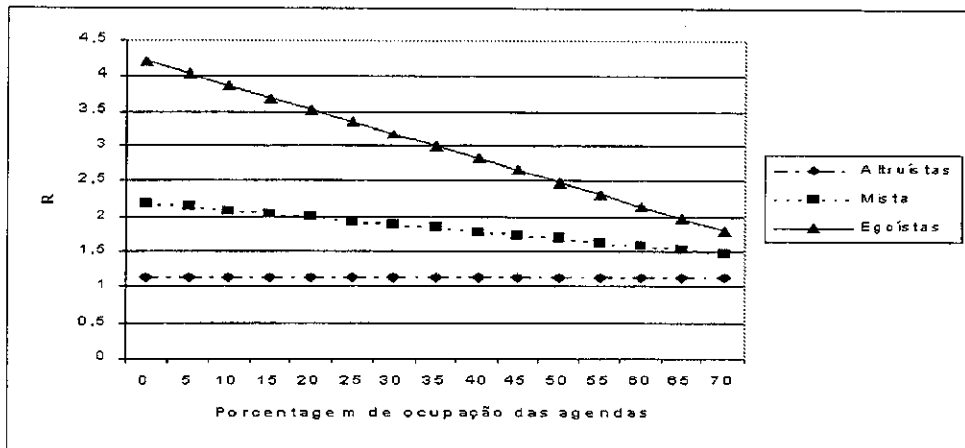


Figura 2.19: Comparação entre as estratégias altruísta, egoísta e mista (1 egoísta e 4 altruístas).

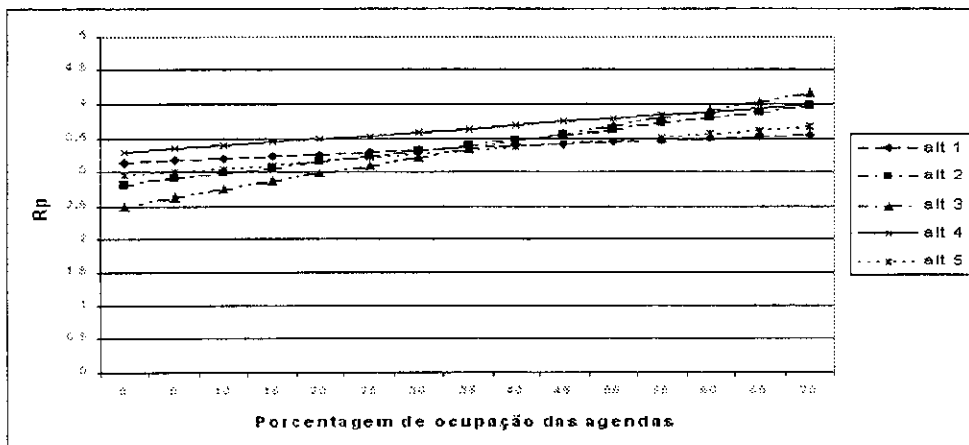


Figura 2.20: Negociação entre altruístas.

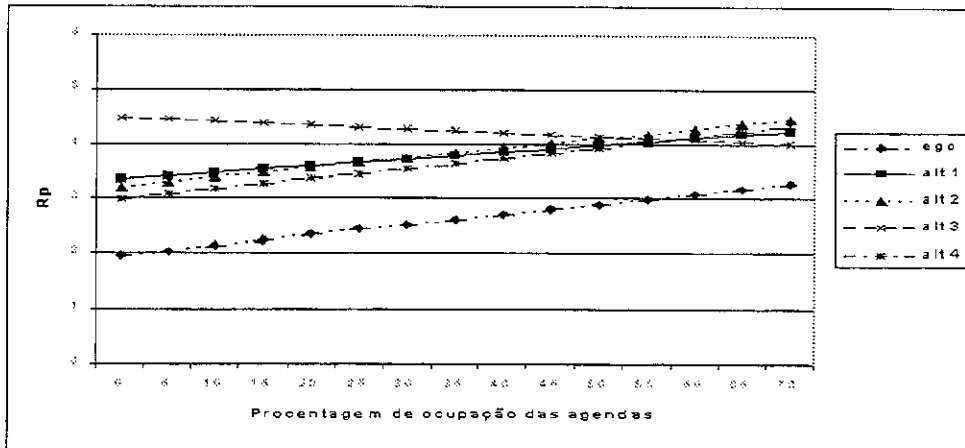


Figura 2.21: Negociação entre 4 altruístas e 1 egoísta.

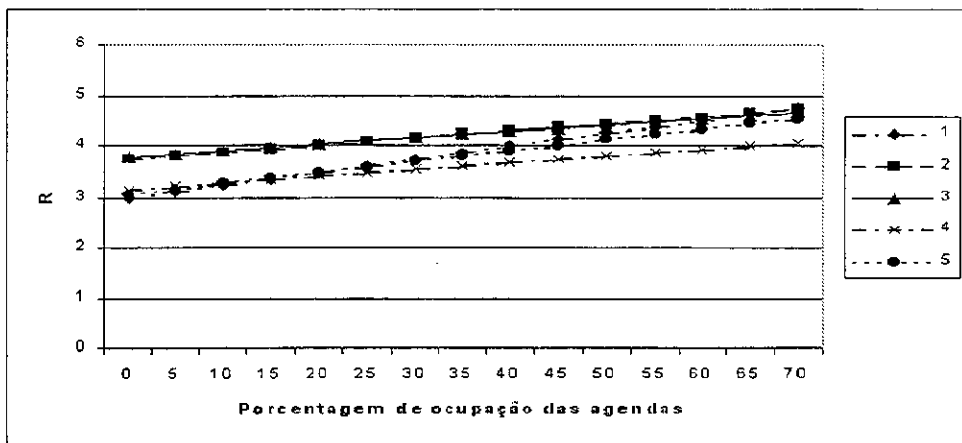


Figura 2.22: Negociação sem coalizão.

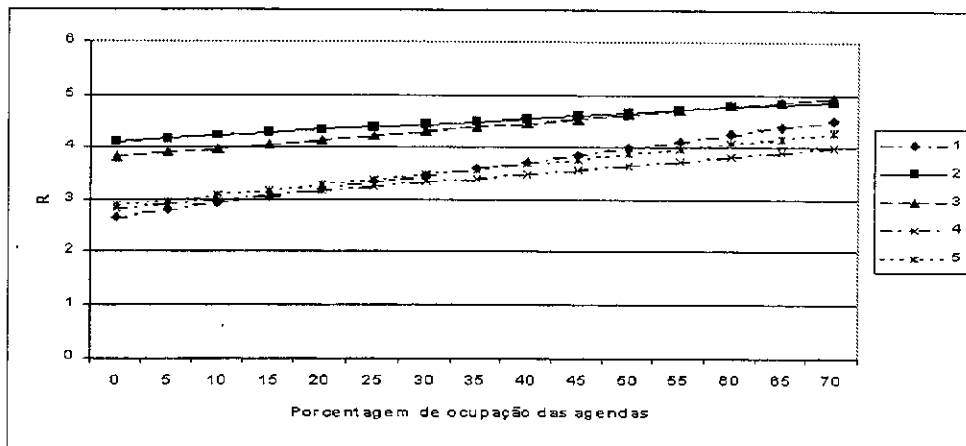


Figura 2.23: Negociação com coalizão de 2 participantes.

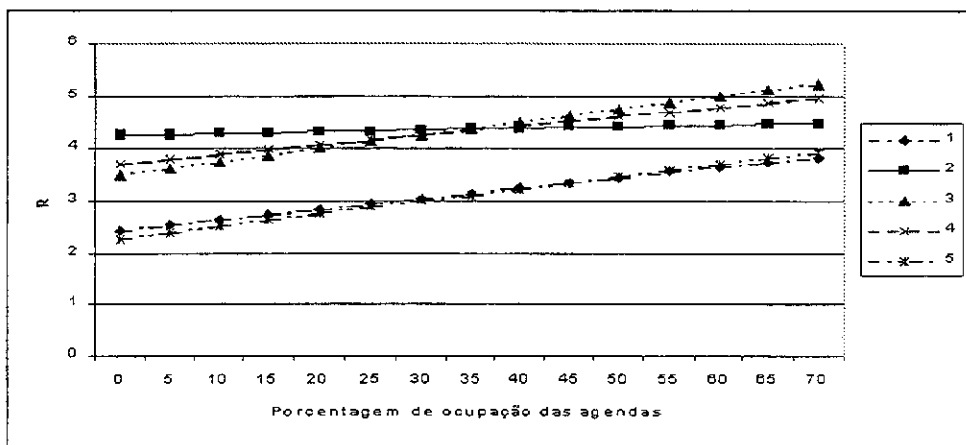


Figura 2.24: Negociação com coalizão de 3 participantes.

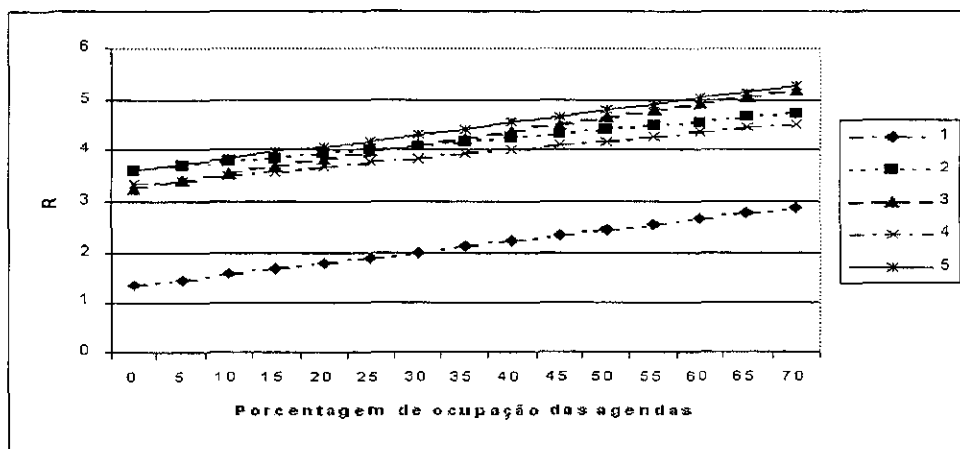


Figura 2.25: Negociação com coalizão de 4 participantes.

Capítulo 3

Protótipo MENES (Multiagent meEtiNg schEduling System)

Desenvolvemos um protótipo de um sistema de agendamento automático de reuniões baseado em um sistema multiagentes que implementa parte da teoria apresentada neste trabalho. Nosso protótipo foi implementado em C++ para UNIX/Solaris, utilizando o compilador G++ e sua interface foi concebida em Tcl/Tk.

Nosso objetivo foi, além de verificar a viabilidade da implementação prática de nosso conjunto de protocolos, a possibilidade de colocar nossas idéias a prova no futuro através da implantação deste sistema para um teste prático. Somente com um teste em um ambiente próximo ao real, com a utilização do sistema por pessoas envolvidas em seus grupos, podemos validar completamente nossa teoria sob os aspectos de *groupware*.

Antes de prosseguirmos devemos destacar que no protótipo não implementamos o Protocolo de Negociação do *host*. Portanto os participantes não podem ter restrições a utilização de qualquer *host* e não existe a figura do *host* externo. Além disso, também não implementamos os Protocolos de Cancelamento e Reagendamento.

Outra observação é que como não programamos os *representatives* capazes de aprender através do comportamento dos demais, no Protocolo de Negociação aplicado no protótipo implementamos todos os agentes utilizando a estratégia egoísta pois, como já explicamos, caso os *representatives* não estejam habilitados para aprender as preferências dos demais, a estratégia egoísta é a melhor para o grupo e para o indivíduo.

Vamos nos concentrar inicialmente em como implementamos a sociedade de agentes e como estes agentes interagem entre si e com o ambiente. Somente na seção 3.5 discutiremos os aspectos do sistema vistos por um usuário, como sua interface e usabilidade.

Nossos agentes são implementados internamente como uma máquina de estados, onde a linguagem de comunicação dos agentes que veremos na seção 3.2 orienta sua execução. Os estados internos dos agentes não mudam para *representatives* e *hosts*, apenas as ações

dos agentes para cada estado é que são obviamente diferentes. Na figura 3.1 apresentamos a máquina de estados que implementa nossos agentes.

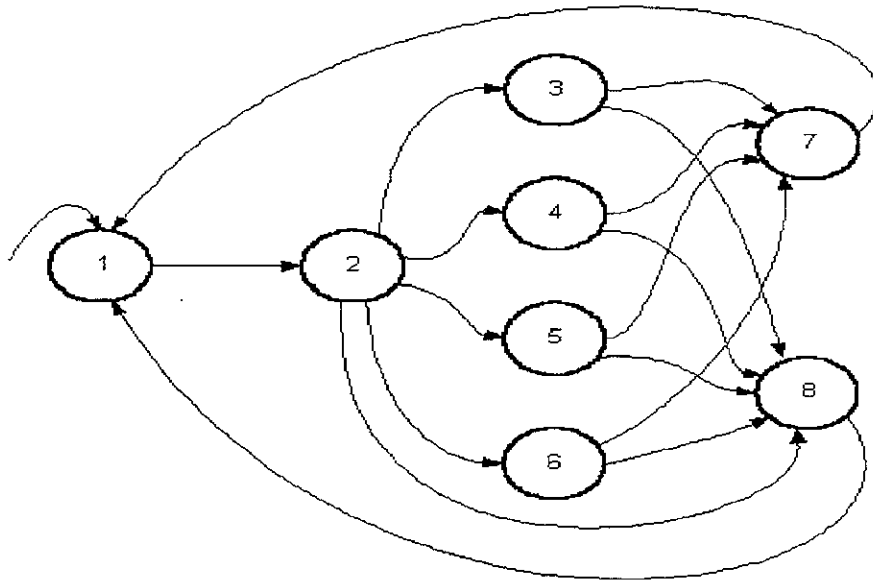


Figura 3.1: Estados internos dos agentes.

Os estados apresentados na figura 3.1 são:

1. Descanso: é o estado inicial do agente. Até que o *representative* receba uma solicitação de agendamento do *host* ou o *host* seja chamado para iniciar um agendamento por seu usuário os agentes permanecem neste estado;
2. Estabelecimento: neste estado o *quantum* do agendamento é estabelecido, como mostrado na seção 2.3, e o protocolo de negociação do nível de privacidade descrito na seção 2.3.1, é executado;
3. Negociação 1: caso o modo de privacidade determinado seja o modo direto este estado é ativado e o protocolo direto da figura 2.7 é executado;
4. Negociação 2: caso o modo de privacidade determinado seja o modo de votação este estado é ativado e o protocolo de votação da figura 2.8 é executado;
5. Negociação 3: caso o modo de privacidade determinado seja o modo de aprovação este estado é ativado e o protocolo de aprovação da figura 2.9 é executado;

6. Negociação 4: caso o modo de privacidade determinado seja o modo de negociação este estado é ativado e o protocolo de negociação da figura 2.6 é executado;
7. Sucesso: se o agendamento for conseguido com sucesso o agente passa para este estado. O agente retorna para o estado de Descanso logo em seguida.
8. Insucesso: caso ocorra algum problema no processo de agendamento em qualquer um dos estados, exceto o de Descanso, este estado é ativado. O agente retorna em seguida ao estado de Descanso.

Em nosso protótipo a sociedade de agentes interage via Internet, permitindo aos agentes estarem localizados em qualquer máquina que esteja conectada a Rede.

Para que os agentes sejam localizados na Internet eles carregam rótulos que os relacionam a seus usuários. Convencionamos utilizar o endereço do *email* do usuário para facilitar a identificação do *representative*, e para o *host* adicionamos ao *user name* do endereço a cadeia de caracteres “host.”.

Assim, o nome do *representative* de um determinado usuário pode ser “determinado_usuario@onde.esta.br” e do seu *host* “host.determinado_usuario@onde.esta.br”.

Cada agente da sociedade precisa conhecer seu endereço, rótulo e outras informações, para isso criamos um diretório no diretório local do agente, denominado *config*, onde convencionamos colocar arquivos com estas informações necessárias para o agente. Desta forma, o agente conhece seu rótulo através do arquivo “config/my_name.cnf”.

Veremos a seguir em detalhes como se dá o processo de localização e comunicação dos agentes.

3.1 Localização e Comunicação

Do ponto de vista computacional, nossos agentes são processos cliente/servidor que utilizam TCP/IP (do inglês *Transmission Control Protocol / Internet Protocol*) [18] para se comunicarem pela Internet. Para utilizar o TCP/IP os agentes criam canais de comunicação entre processos denominados *sockets*. Para permitir que vários processos, em uma mesma máquina, possam criar *sockets* são utilizadas as Portas.

Para que um agente possa receber mensagens através da Rede este cria uma *socket* na máquina na qual está rodando e se vincula a uma Porta e a um endereço IP. Para que o agente saiba em qual máquina está e que Porta deve usar utilizamos o arquivo de configuração “config/my_adress.cnf”.

Quando um agente deseja enviar uma mensagem para outro este deve conhecer o endereço de IP e a Porta deste agente, assim como as pessoas precisam saber o número telefônico das outras para fazer uma chamada. Porém, dissemos anteriormente que os

agentes são conhecidos por rótulos; então deve existir uma forma pela qual possuindo o rótulo de um agente se possa saber seu endereço IP e Porta.

Para que os agentes possam saber onde os outros agentes se encontram apenas conhecendo seu rótulo criamos um sistema de servidor de nomes para os agentes, o qual denominamos ANS, sigla para o inglês *Agent Name Server*. Na seção seguinte apresentaremos as características e o modo de funcionamento do ANS.

3.1.1 ANS

O ANS é um servidor que atende a requisições dos agentes, comunicando-se também através de *sockets* e TCP/IP.

Quando o usuário ativa seu agente este envia para o ANS seu rótulo, endereço IP e Porta. Para saber o endereço IP e Porta associado a determinado rótulo um agente faz uma consulta ao ANS. Toda vez que o usuário desativa seu agente este envia para o ANS um requisição para que o ANS passe a considerá-lo desativado.

Como podemos perceber o ANS atende a todos os participantes do grupo e a várias requisições de cada um. Devido a esse fato, implementamos o ANS como um servidor multiprocessado, ou seja, a cada requisição recebida de um agente o ANS cria um processo filho para atendê-la e fica pronto para receber uma nova requisição.

O ANS armazena suas informações em uma base de dados implementada através de um arquivo ASCII formatado especificamente para guardar a informação de rótulo e localização. Este arquivo se encontra em seu diretório local e é denominado “able.ans”.

Para que os agentes conheçam o endereço IP e a Porta do ANS é preciso que seu usuário as indique através do arquivo de configuração “config/ans.cnf”.

Para a comunicação entre os agentes e o ANS criamos um protocolo de mensagens baseado em cabeçalhos. Existem apenas três tipos de cabeçalhos que associam a informação que os acompanha à atitude que o ANS deve tomar. O cabeçalho “T” indica que o agente está ativo, o cabeçalho “Q” indica que o agente está fazendo uma consulta ao ANS e o cabeçalho “X” indica que o agente está sendo desativado.

Assim, toda a comunicação entre agentes é precedida por uma consulta ao ANS. A figura 3.2 ilustra a arquitetura da sociedade de agentes com a camada de localização e comunicação.

3.2 Linguagem de comunicação dos agentes

Na seção 3.1 vimos que os canais de comunicação permitem a troca de cadeias de caracteres entre os agentes. Nesta seção discutiremos a linguagem que os agentes utilizam

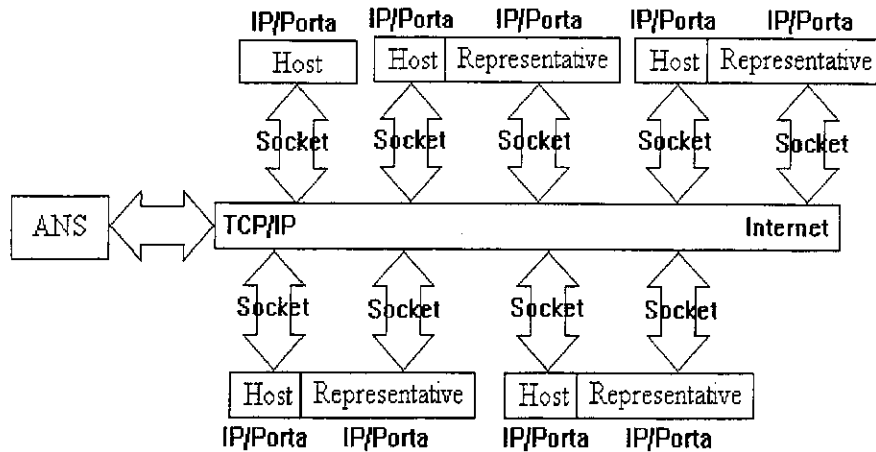


Figura 3.2: Arquitetura da sociedade com a camada de comunicação e localização.

para se comunicar, ou seja, a forma como estas cadeias de caracteres são compostas para que os agentes se entendam.

Como dissemos anteriormente, a linguagem de comunicação dos agentes determina a mudança entre os estados internos do agente. Conforme as mensagens vão sendo interpretadas e respondidas se estabelece a negociação e o agendamento.

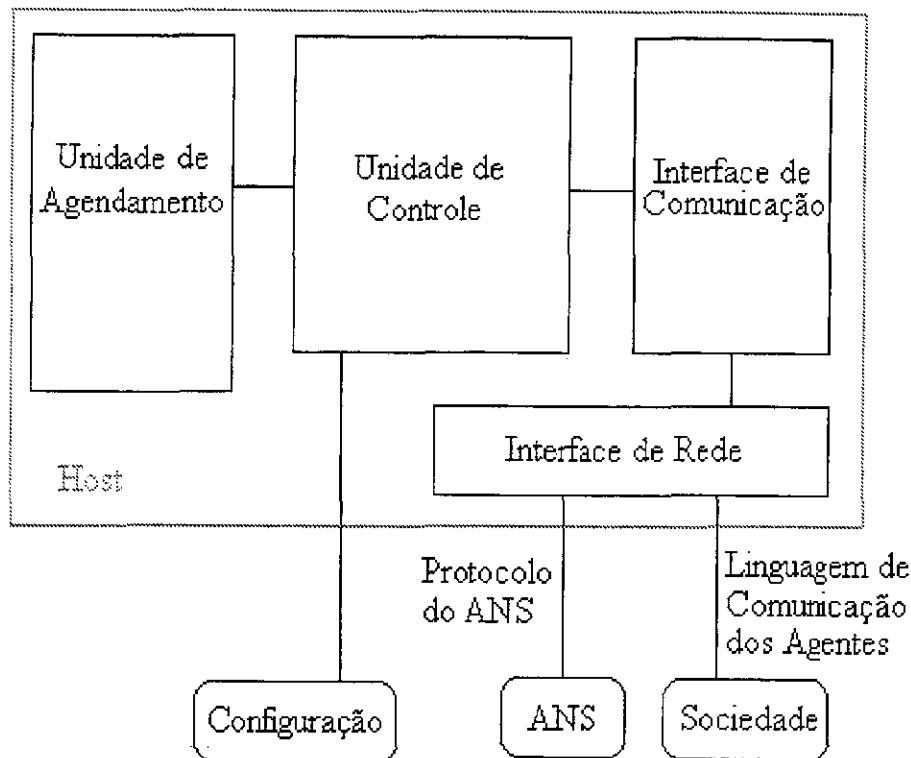
Implementamos vários pequenos interpretadores que recebem as mensagens e traduzem-nas para a forma de representação interna do agente. Nossa linguagem é baseada em cabeçalhos, assim os primeiros caracteres indicam o significado da mensagem.

No apêndice A apresentamos a gramática da linguagem de comunicação dos agentes.

3.3 Arquitetura do *host*

A figura 3.3 traz a arquitetura do *host* onde cada módulo tem a seguinte função:

- Interface de Rede: é responsável pela conexão do *host* a Internet. Para isto a Interface de Rede cria o *socket* no endereço IP e Porta recomendados pela Unidade de Controle para que o *host* fique aguardando uma solicitação de agendamento de seu usuário. Cabe a este módulo também toda a comunicação com o ANS e a indicação de possíveis erros de conexão para a Unidade de Controle.
- Interface de Comunicação: é responsável pela tradução das mensagens recebidas em forma de seqüências de caracteres para as estruturas internas do agente. Cabe a este módulo indicar possíveis erros na linguagem de comunicação para a Unidade de Controle.

Figura 3.3: Arquitetura do *host*.

- Unidade de Controle: esta unidade é o cérebro do agente; é nela que está a máquina de estados do agente e é ela quem contém os planos de ação e quem executa todas as ações do agente. Além disso, é responsável pela configuração do agente através da interpretação dos arquivos de configuração, por controlar os estados internos do agente conforme a comunicação, pela ativação da Unidade de Agendamento quando for necessário e pelo controle dos erros de comunicação.
- Unidade de Agendamento: é responsável por realizar operações de agendamento como a interseção de agendas entre si, a interseção de agendas e perfis de preferência, o cálculo de intervalos existentes entre duas datas, etc. Esta unidade recebe solicitações de operações da Unidade de Controle e responde com seu resultado.

3.4 Arquitetura do *representative*

Apresentamos na figura 3.4 a arquitetura do *representative* onde cada módulo tem a seguinte função:

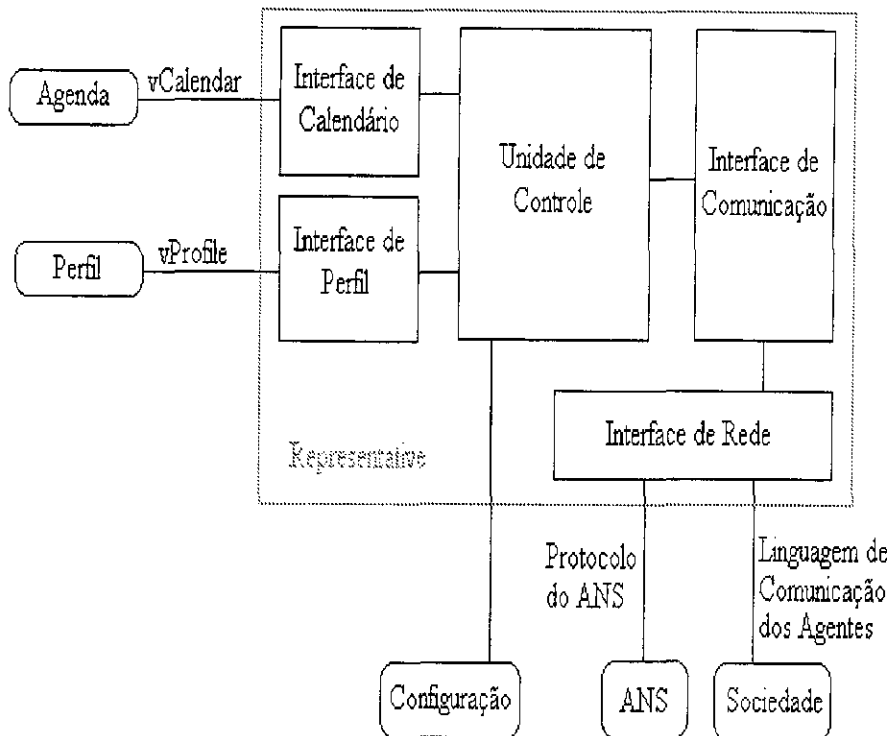


Figura 3.4: Arquitetura do *representative*.

- Interface de Rede: é responsável pela conexão do *representative* com a Internet. Para isto a Interface de Rede cria o *socket* no endereço IP e Porta recomendados pela Unidade de Controle para que o *representative* fique aguardando uma solicitação de agendamento de algum *host*. Cabe a este módulo também toda a comunicação com o ANS e a indicação de possíveis erros de conexão para a Unidade de Controle.
- Interface de Comunicação: é responsável pela tradução das mensagens recebidas em forma de seqüências de caracteres para as estruturas internas do agente. Cabe a este módulo indicar possíveis erros na linguagem de comunicação para a Unidade de Controle.
- Unidade de Controle: esta unidade é o cérebro do agente; é nela que está a máquina de estados do agente e é ela quem contém os planos de ação e quem executa todas as ações do agente. Além disso, é responsável pela configuração do agente através da interpretação dos arquivos de configuração, por controlar os estados internos do agente conforme a comunicação, pelo controle dos erros de comunicação e por tratar as informações de agenda e perfil de preferências recebidas pelas Interfaces

de Calendário e de Perfil.

- Interface de Calendário: é responsável por manipular a agenda do usuário. Esta unidade atende a requisições da Unidade de Controle quando o *representative* precisa enviar informações para o *host* sobre sua agenda ou quando uma reunião foi agendada e esta precisa ser inserida na agenda. A agenda do usuário deve estar em um formato chamado *vCalendar* para que a Interface de Calendário possa interpretá-la corretamente. *vCalendar* define um formato para calendários eletrônicos que permite capturar e manipular informações normalmente armazenadas em aplicações de agendas [1].
- Interface de Perfil: é responsável por interpretar o perfil de preferência do usuário. Esta unidade atende a requisições da Unidade de Controle quando o *representative* precisa enviar informações para o *host* sobre seu perfil de preferências. Este perfil do usuário deve estar em um formato chamado *vProfile* para que a Interface de Perfil possa interpretá-lo corretamente. *vProfile* é uma nova entidade para *vCalendar* que desenvolvemos para permitir o armazenamento de informações sobre as preferências dos usuários nos moldes do *vCalendar* que já existia. Apresentamos no apêndice B a definição de *vProfile*.

3.5 Interface com o Usuário

Nesta seção, como mencionamos anteriormente, vamos tratar dos aspectos da interação entre os usuários e seus agentes.

Inicialmente, definimos o arquivo “config/restric.cnf”, baseado em regras, que contém as restrições de privacidade do usuário com explicado na seção 2.3.1. Este arquivo permite que o usuário indique suas restrições em relação a primeira dimensão de privacidade mencionada na seção 2.3, pois as demais dimensões de privacidade não são tratadas neste protótipo.

Utilizamos a linguagem descrita abaixo em BNF para formatar o arquivo:

INICIO	→ ROTULO RESTRICAO
ROTULO	→ NAME '@' NAME
RESTRICAO	→ RES
RESTRICAO	→ / RES ',' RESTRICAO
RES	→ '1' / '2' / '3' / '4'
NAME	→ '*'
NAME	→ '*' '.' NAME
NAME	→ cadeia_de_caracteres
NAME	→ cadeia_de_caracteres '.' NAME

Com este formato podemos não só indicar um rótulo específico, como generalizar endereços ou nomes. Por exemplo, podemos ter um arquivo de restrições com o seguinte conteúdo:

```
@dcc.unicamp.br 4
ferreira, wainer@dcc.unicamp.br 0
outros@*.*.* 3,4
@*.edu 1,3
@* 1,2,3
```

De forma resumida, os endereços dos agentes são atrelados às restrições aos modos de privacidade que os participantes têm em relação a estes agentes. Quando a restrição tem valor 0 significa que não há nenhuma restrição ao usuário referido. Os números significam que o usuário tem restrição ao primeiro modo de privacidade se for o 1, ao segundo se for o 2, e assim por diante.

Como vemos, nossa linguagem admite regras ambíguas. Se mais de uma regra for aceita durante a verificação do arquivo a regra mais específica será considerada. Desta forma, para o endereço “ferreira@dcc.unicamp.br” a regra considerada será “ferreira, wainer@dcc.unicamp.br”.

Quando máscara “*” é utilizada o sistema considera que em sua posição possa existir qualquer nome válido para endereços e nomes. No caso da descrição “*@*.edu”, por exemplo, todos os participantes da reunião que têm seu rótulo composto por qualquer nome e por um endereço terminado por “.edu” serão considerados por esta regra. Assim, “joseh@msc.edu”, “claudio@mwd.ci.edu”, etc. terão restrições aos modos 1 e 3.

Caso a Unidade de Controle do *representative* não encontre uma regra para um determinado rótulo de agente, ela abre uma janela para que o usuário informe suas restrições, como pode ser visto na figura 3.5.

A agenda e o perfil do usuário são mantidos no formato *vCalendar* [1] e *vProfile*

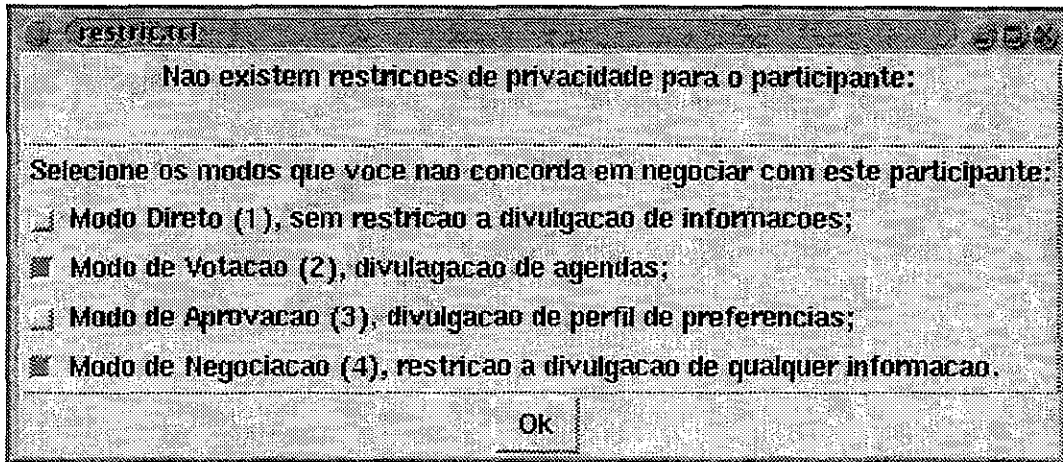


Figura 3.5: Janela para a definição de restrições.

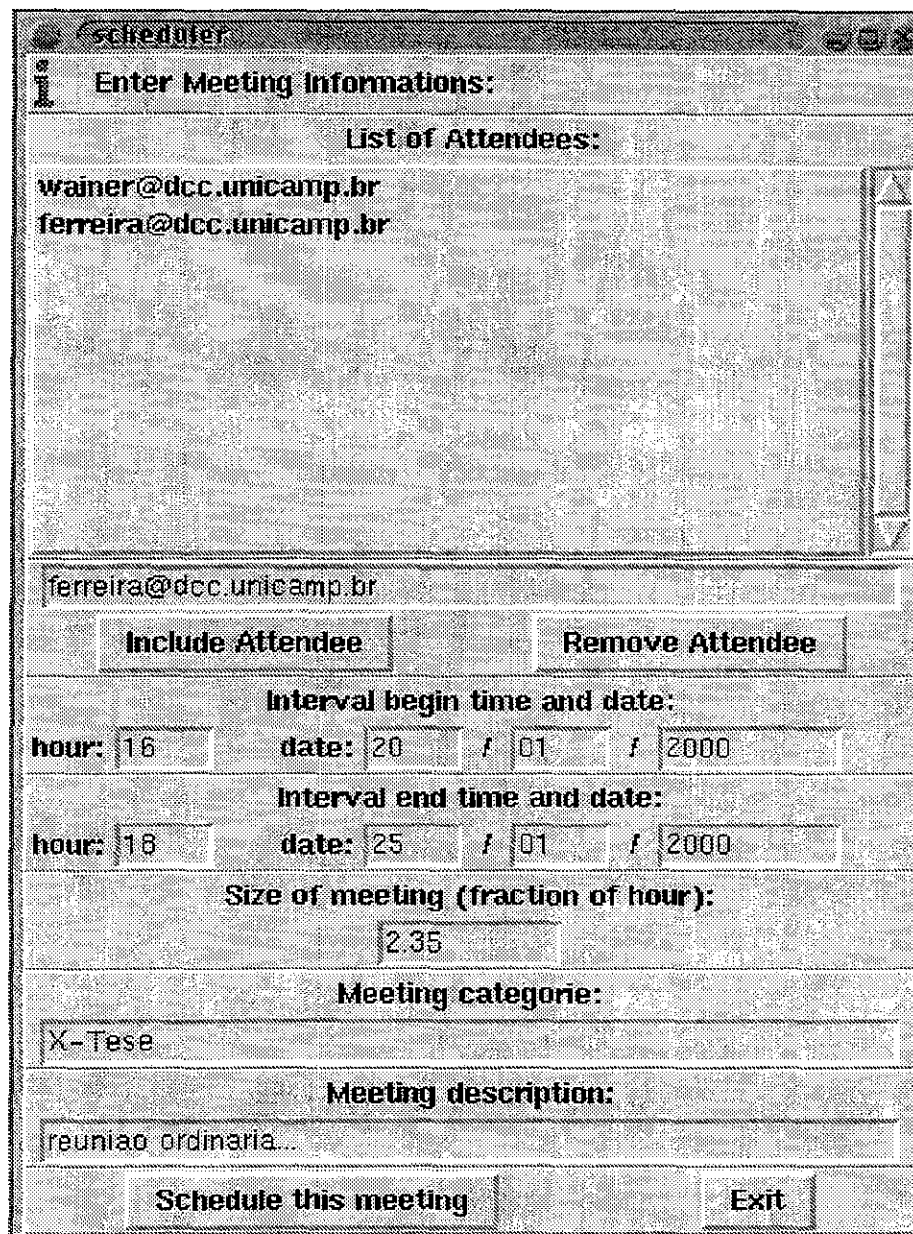
(apêndice B), e nos arquivos "config/mycalend.cnf" e "config/myprofile.cnf". Não implementamos um sistema de gerenciamento de agendas e de perfis pois achamos que isto foge um pouco de nosso objetivo principal. Para nossos testes utilizamos a edição destes arquivos manualmente.

Para chamar uma reunião desenvolvemos um programa chamado Scheduler que, utilizando um *socket*, comunica-se com o *host*. Este programa recebe através de uma janela as informações sobre a reunião que o usuário quer agendar e transmite estas informações para o *host*. Esta janela de chamada de reunião pode ser vista na figura 3.6.

Uma vez ativado, o *host* retorna uma mensagem, através da janela da figura 3.7, para avisar o usuário que este foi ativado com sucesso e que vai tentar agendar a reunião.

Quando a reunião for agendada o *representative* do usuário avisa, através da janela da figura 3.8, que existe uma nova reunião em sua agenda. Caso a reunião não possa ser marcada o *host* comunica o insucesso para seu usuário através da janela da figura 3.9.

Este conjunto de interfaces permite que o usuário interaja plenamente com o sistema e garante uma forma simples e prática de interação.



The screenshot shows a window titled "Scheduler" with a standard Mac OS X title bar. The window contains the following elements:

- Enter Meeting Information:** A section header at the top.
- List of Attendees:** A list box containing two email addresses: `wainer@dcc.unicamp.br` and `ferreira@dcc.unicamp.br`. A scroll bar is visible on the right.
- Attendee Input:** A text field containing `ferreira@dcc.unicamp.br`.
- Buttons:** Two buttons, "Include Attendee" and "Remove Attendee", are positioned below the input field.
- Interval begin time and date:** A section with two rows of input fields:
 - Row 1: `hour: 16`, `date: 20 / 01 / 2000`
 - Row 2: `hour: 18`, `date: 25 / 01 / 2000`
- Interval end time and date:** A section with two rows of input fields:
 - Row 1: `hour: 18`, `date: 25 / 01 / 2000`
 - Row 2: `hour: 18`, `date: 25 / 01 / 2000`
- Size of meeting (fraction of hour):** A text field containing the value `2.35`.
- Meeting categorie:** A text field containing the value `X-Tese`.
- Meeting description:** A text field containing the value `reuniao ordinaria...`.
- Bottom Buttons:** Two buttons, "Schedule this meeting" and "Exit", are at the bottom.

Figura 3.6: Janela para a chamada de reuniões.

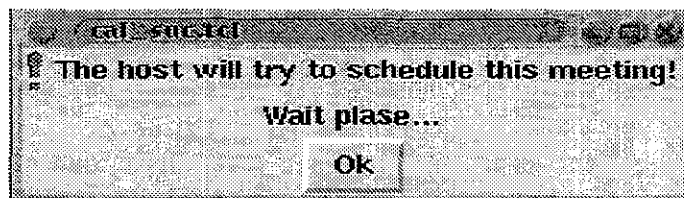


Figura 3.7: Janela para avisar que o *host* está tentando agendar a reunião.



Figura 3.8: Janela para comunicar que uma nova reunião foi agendada.

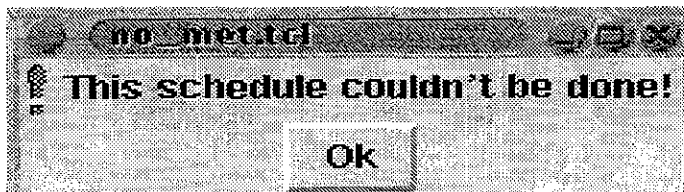


Figura 3.9: Janela para comunicar que não foi possível agendar a reunião.

Capítulo 4

Conclusões

Nesta dissertação apresentamos uma abordagem para a solução do problema do agendamento automático de reuniões, baseada em um conjunto de protocolos de negociação para um sistema multiagentes. Este conjunto de protocolos foi concebido com o objetivo de garantir a máxima satisfação do grupo de participantes da reunião, considerando as preferências individuais e as restrições de privacidade destes participantes.

Ao contrário de outros pesquisadores que já publicaram trabalhos sobre este assunto, orientamos nosso trabalho às questões de *groupware* que envolvem o problema: a possibilidade do participante expressar suas preferências e a garantia que esta será fortemente considerada no processo; e a confiança que sua privacidade será assegurada segundo seus critérios de confiança nos demais participantes.

Em nossa abordagem cada participante possui um par de agentes, autorizados a agir autonomamente pelo seu usuário, a fim de negociar com os agentes dos demais participantes para estabelecer o melhor intervalo possível para o agendamento. Esta negociação se dá orientada pelo conjunto de protocolos que elaboramos.

Com o objetivo de medir a eficiência de nosso conjunto de protocolos, criamos uma métrica de avaliação totalmente baseada na preferência dos usuários. Com isso, implementamos um simulador e realizamos várias simulações para medir a eficiência de nossa abordagem, onde obtivemos resultados muito bons segundo nossa perspectiva.

Nossos Protocolos Direto e de Aprovação utilizados na negociação são, por definição, ótimos, resultando no melhor agendamento possível. O Protocolo de Votação garante, em média, o segundo melhor agendamento.

Com relação ao Protocolo de Negociação, demonstramos que as estratégias altruísta e a egoísta apresentam os melhores resultados. Contudo, não é interessante utilizar a estratégia altruísta uma vez que não se sabe qual a estratégia dos demais e, caso existam egoístas combinados com altruístas, os altruístas acabam sendo prejudicados com o resultado do agendamento. Assim, com a utilização da estratégia egoísta por todos os

participantes da negociação temos, em média, o quarto melhor agendamento.

Demonstramos também que não existe vantagens na formação de coalizões no processo de agendamento utilizando o Protocolo de Negociação.

O Protocolo de Reagendamento é, por definição, igualmente eficiente ao protocolo do modo de privacidade que for adotado no novo agendamento.

Para testar a viabilidade da aplicação prática de nosso conjunto de protocolos implementamos o protótipo de um sistema de agendamento automático de reuniões, denominado MENES. Neste sistema implementamos a maioria das funcionalidades de nossa abordagem.

MENES é um sistema multiagentes que utiliza a Internet como meio de comunicação. Para isso, desenvolvemos um sistema de localização e comunicação entre os agentes utilizando Sockets e o protocolo TCP/IP.

Os agentes deste sistema são dirigidos por uma máquina de estados e interagem entre si através de uma linguagem própria. Para a comunicação entre o sistema e os participantes elaboramos algumas interfaces gráficas simples.

A solução do problema do agendamento automático de reuniões, que acreditamos ter atingido, pode trazer inúmeros benefícios para os grupos que têm o agendamento de reuniões como atividade comum. Utilizando um sistema automático, livramos o participante do processo trabalhoso de buscar um intervalo comum na agenda de todos os participantes de forma consensual.

Com isso, vemos a automatização desta tarefa como um passo importante no sentido de livrar pessoas de tarefas corriqueiras e trabalhosas, o que de certa forma é um dos objetivos da computação. Acreditamos que nosso trabalho conseguiu sintetizar em um conjunto de protocolos de negociação os aspectos de *groupware* envolvidos no agendamento de reuniões.

Para comprovar definitivamente a eficiência de nossa abordagem, devemos ainda complementar o protótipo para torná-lo um sistema mais completo e robusto e com isso realizar um conjunto de testes em um ambiente real de aplicação, onde a complicada ralação entre participantes de um grupo esteja presente.

Apêndice A

Gramática da linguagem de comunicação dos agentes

Definindo formalmente em BNF a gramática $G = N, T, P, S$ da linguagem de comunicação da sociedade de agentes, temos:

- S - START, símbolo inicial.
- T - Terminais
 - 'l', 'm', 'r', 'o', 'h', 'a', 'b', 'e', 't', ':', 'p', 'd', 'n', 'q', 's'.
 - email - endereço de email padrão (nickname@domain);
 - isodate - data padrão ISO 8601 (ex:19990304);
 - isotime - hora padrão ISO 8601 (ex:123000);
 - res - 4 dígitos, '0' ou '1';
 - ans - 'y' ou 'n';
 - hour - 2 dígitos, onde $'01' \leq xx \leq '24'$;
 - frac - 2 dígitos, onde $'00' \leq xx \leq '99'$;
 - weight - 2 dígitos, onde $'00' \leq xx \leq '10'$;
 - weekdays - 'mon', 'tue', 'wed', 'tur', 'fri', 'sat' e 'sun';
 - mod - 1 dígito, '1', '2', '3' e '4';
 - str - cadeia de caracteres;
 - int - 1 dígito, onde $0 \leq x \leq 9$.
- N - Não Terminais
START, BEGIN, MODE, RESTRIC, START, MEETING, M1, M2, M3, M4, ESET, TIME, TIMESET, PROFILE, END, DAYSET, WEIGHT, NEGOCI

- P - Produções

START	→ 'l' QUANTUM BEGIN MODE END
QUANTUM	→ 'q' frac 's' frac
BEGIN	→ 'm' MEETING 'r' res 'o' mod
MODE	→ M1
MODE	→ M2
MODE	→ M3
MODE	→ M4
MEETING	→ 'h' email 'a' ESET 'b' TIME 'e' TIME 'l' int 'c' str 'i' str
ESET	→ email
ESET	→ email, ESET
TIME	→ 't' isotime ':' isodate
M1	→ TIMESET PROFILE
TIMESET	→ TIME
TIMESET	→ TIME TIMESET
PROFILE	→ 'p' DAYSET WEIGHT
PROFILE	→ 'p' DAYSET WEIGHT PROFILE
END	→ 'd' TIME
END	→ 'd' 'n'
DAYSET	→ weekdays
DAYSET	→ weekdays ',' DAYSET
WEIGHT	→ hour '.' frac ':' weight
WEIGHT	→ hour '.' frac ':' weight ',' WEIGHT
M2	→ TIMESET TIMESET TIMESET
M3	→ PROFILE NEGOCI
NEGOCI	→ TIME 'a' ans
NEGOCI	→ TIME 'a' ans NEGOCI
M4	→ TIMESET TIMESET
M4	→ TIMESET TIMESET M4

Apêndice B

vProfile

Com o objetivo de permitir o armazenamento e a troca de informações sobre as preferências dos usuários, criamos a entidade *vProfile*, que é uma nova entidade para *vCalendar* [1].

Um objeto *vProfile* é um conjunto de propriedades de preferências para calendários. Este objeto representa as preferências de uma pessoa acerca de sua agenda, permitindo, por exemplo, atribuir pesos para os diferentes horários dos dias da semana.

Um entidade *vProfile* é definida em um objeto *vCalendar* pelos delimitadores:

```
BEGIN:PROFILE  
END:PROFILE
```

Para a utilização do *vProfile* é necessário definir em *vCalendar* o tamanho do intervalo de tempo utilizado nas propriedades. Para isso definimos uma nova propriedade para *vCalendar* denominada PROFILEQUANTUM. Vejamos um exemplo de utilização desta nova propriedade:

```
PROFILEQUANTUM:0.25
```

No exemplo acima definimos que o tempo de duração dos intervalos é 1/4 de hora ou 15 min. Mais adiante veremos a real aplicação de PROFILEQUANTUM.

B.1 Propriedades de *vProfile*: Categorias

Esta propriedade, denominada CATEGORIES, define as categorias para as entidades *vCalendar*.

É importante observarmos que mais de uma categoria pode ser especificada; para isso utilizamos um formato de lista onde os itens são separados pelo caracter ponto e vírgula (ASCII decimal 59).

Vejamos um exemplo da utilização desta propriedade:
CATEGORIES:X-UNICAMP;X-JACQUES;X-TESE
CATEGORIES:TRAVEL

Nas entidades *vProfile* esta propriedade tem o objetivo principal de categorizar as preferências de acordo com o tipo de evento ou afazer.

Devemos destacar que existe um conjunto de valores padrão para categorias definidos por *vCalendar*, para a utilização de outros devemos acrescentar "X-" antes do nome da categoria.

Assim, no primeiro dos exemplos anteriores, está definido que o conjunto de preferências representado nas demais propriedades da entidade, fazem referência aos eventos cujas categorias são X-UNICAMP;X-JACQUES;X-TESE.

B.2 Propriedades de *vProfile*: Dias da Semana

Dias da Semana é um conjunto de propriedades, definidas por *vProfile* que tem a função de indicar os dias da semana disponíveis para agendar eventos e o nível de preferência para as horas disponíveis em cada dia.

Cada dia da semana tem uma propriedade associada diretamente. Assim, temos as propriedades denominadas MON, TUE, WED, THU, FRI, SAT e SUN. Estas propriedades assumem o valor de uma lista de horários disponíveis e seu respectivos pesos, separados pelo caracter traço (ASCII decimal 173).

Os pesos associados a cada horário podem assumir um valor entre 0 e 10, inclusive, onde 0 é o menos preferido e 10 o mais preferido, permitindo o armazenamento do nível de preferência de cada horário em relação aos demais.

Para flexibilizar a utilização do *vProfile* os horários não são necessariamente hora cheia; podemos então especificar horas e frações de hora.

Vejamos um exemplo da utilização destas propriedades:

MON: 8.5-5 ; 9.25-10 ; 14-8 ; 16-0

SAT: 8-5 ; 9-5 ; 10.75-5 ; 11.75-5 ; 12.75-0

O exemplo indica que o usuário dispõe de segunda as 8h e 30 min, 9h e 15 min, 14h e 16h com respectivos pesos 5, 10, 8, 0 e sábado as 8h, 9h, 10h, 11h e 12h com peso 5 para todos a exceção de 12h que tem peso 0. É importante destacar que quando mencionamos como disponível as 8h, estamos indicando que o horário que começa as 8h e termina as 8h + PROFILEQUANTUM, que está disponível.

Além disso, podemos ter entidades agrupadas que atribuem os mesmos horários e pesos para diferentes dias da semana, como por exemplo:

MON; TUE; WED; SAT:8-5;9-5;10-5;11-5;12-0

Neste exemplo segunda, terça, quarta e sábado tem os mesmos horários disponíveis e respectivos pesos.

Vejamos agora um exemplo da entidade *vProfile*:

```
BEGIN:VCALENDAR
BEGIN:PROFILE
CATEGORIES:BUSINESS;X-UNICAMP
MON;TUE;WED;THU;FRI:8-4;9-4;10-5;11-5; 13-5;14-10;15-10;16-10;17-10;18-8
END:PROFILE
BEGIN:PROFILE
CATEGORIES:X-SOCCER
MON;TUE;WED;THU;FRI:18-10;19-10;20-5;21-5;22-0
SAT;SUN:14-10;15-10;16-10;17-10;18-10
END:PROFILE
END:VCALENDAR
```

Quanto as preferências, temos que destacar que quando não existe preferência definida para os horários devemos utilizar na representação o valor padrão 10.

B.3 Definição Formal

A notação utilizada nesta descrição formal é uma variação da BNF utilizada em [1] para a formalização de *emphvCalendar*.

B.3.1 Definições Preliminares

CRLF = nova linha (ASCII CR e ASCII LF, *carriage return e linefeed*)
 SPACE = espaço em branco (ASCII decimal 32)
 HTAB = tabulação (ASCII HT)
 ws = 1*(SPACE / HTAB) (um ou mais caracteres em branco ou tabulações)
 hour = "01", ..., "24" (2 dígitos numéricos em ASCII para os inteiros de 1 a 24)
 wei = "00", ..., "10" (2 dígitos numéricos em ASCII para os inteiros de 0 a 10)
 catvals = definido em [1]

B.3.2 Definição da Entidade *vProfile*

```

profileentity  = "BEGIN"[ws] ":"[ws] "PROFILE"[ws] CRLF
                entpropsf [ws] CRLF
                "END"[ws] ":"[ws] "PROFILE"[ws] CRLF
entpropsf      = categorie CRLF weekdays
categorie      = [ws] "CATEGORIES"[ws] ":"1*catvals CRLF
weekdays      = weekday
                / weekday CRLF weekdays
weekday        = days [ws] ":"[ws] weight CRLF
days           = day
                / day [ws] ";"[ws] days
day            = "MON" / "TUE" / "WED" / "THU" / "FRI" / "SAT" / "SUN"
weight         = wvalue
                / wvalue [ws] ";"[ws] weight
wvalue         = hour [ws] "-"[ws] wei

```


Referências Bibliográficas

- [1] a versit Consortium Specification. vcalendar: The eletronic calendaring and scheduling exchange format, version 1.0. <http://www.imc.org/> (updated 27/11/1999), 1996.
- [2] J. C. Borda. *Memoire sur les Elections au Scrutin*. Histoire de l'Academie Royale de Science, Paris, 1781.
- [3] S.J.Gibbs C.A.Ellis and G.L. Rein. Groupware. *Communications of the ACM*, 1991.
- [4] Alphonse Chapanis and J.F.Kelley. How professional persons keep their calendars: Implications for computadorization. *Journal of Occupational Psychology*, 1982.
- [5] Marquis de Condorcet. *Mathématique et société*. Hermann, Paris, 1974.
- [6] Edmund Durfee and Sandip Sen. On the design of an adaptative meeting scheduling. In *Tenth IEEE Conference on Artificial Intelligence Applications*, 1994.
- [7] Edmund Durfee and Sandip Sen. A formal study of distribuited meeting scheduling. *Group Decision and Negotiation*, 1998.
- [8] E. Ephrati, G. Zlotkin, and J. Rosenschein. Meet your destiny: a non-manipulable meeting scheduler. In *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work*, pages 359–371. ACM Press, 1994.
- [9] Jonathan Grudin. Groupware and social dynamics eight challenges for developers. In *Communication of ACM*, 1992.
- [10] Jonathan Grudin and Leysia Palen. Why groupware succeeds: Discretion or mandate? *Computer-Suported Cooperative Work - ACM*, 1998.
- [11] Bjorn Hermans. *Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the information society and a prediction of (near-)future developments*. PhD thesis, Tilburg University, The Netherlands, 1996.

- [12] Samuel Merrill III. *Making multicandidate elections more democratic*. Princeton University Press, Princeton, New Jersey, 1988.
- [13] H. Moulin. The strategy of social choice. *Advanced Textbooks in Economics*, 1983.
- [14] Leysia Ann Palen. *Calendars on the New Frontier: Challenges of Groupware Technology*. PhD thesis, University of California, Irvine, 1998.
- [15] Thomas Schwartz. *The logic of collective choice*. Columbia University Press, New York, 1986.
- [16] Sandip Sen, Thomas Haynes, and Neeraj Arora. Satisfying user preferences while negotiating meetings. *International Journal of Human-Computer Studies*, 1997.
- [17] Murray R. Spiegel. *Probabilidade e Estatística*. McGraw-Hill, São Paulo, 1977.
- [18] Richard W. Stevens. *Advanced Programming in the UNIX environment*. Reading Mass, Addison - Wesley, 1995.
- [19] Katia Sycara and Leonardo Garrido. Multi-agent meeting scheduling: Preliminary experimental results. In *Second International Conference on Multi Agent Systems (ICMAS-96)*, 1996.
- [20] Katia Sycara, Leonardo Garrido, and Ramón Brena. Cognitive modeling and group adaptation in intelligent multi-agent meeting scheduling. In *First Ibero American Workshop on Multi-Agents*, 1996.
- [21] Jacques Wainer and Clarence Ellis. Agents in groupware systems. In *CYTED-RITOS Workshop on Groupware (CRIWG'98)*, 1998.
- [22] Michael Wooldridge and Nicholas Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 1994.