Inteligência artificial distribuída

Guilherme Bittencourt
Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
88040-900 - Florianópolis - SC - Brazil
E-mail: gb@lcmi.ufsc.br

Conteúdo

1 Introdução

A *inteligência artificial distribuída (IAD)* é uma das áreas da *inteligência artificial (IA)* que mais se desenvolveram nos últimos anos e apresenta um enorme potencial de aplicações [DLC89]. A IAD estuda o conhecimento e os métodos de raciocínio que podem ser necessários ou úteis para que agentes computacionais participem de sociedades de agentes. Em termos de técnicas computacionais, a IAD reúne as da área de *sistemas distribuídos* às de IA.

A história da IA é a história da lenta conscientização dos pesquisadores da área a respeito das limitações dos métodos computacionais e da incomensurável ignorância a respeito dos mecanismo que tornam a inteligência possível. A gradativa mudança de metas da IA, desde o sonho de construir uma inteligência artificial de caráter geral comparável à do ser humano até os bem mais modestos objetivos atuais de tornar os

computadores mais úteis através de ferramentas que auxiliam as atividades intelectuais de seres humanos, coloca a IA na perspectiva de uma atividade que praticamente caracteriza a espécie humana: a capacidade de utilizar representações externas, seja na forma de linguagem, seja através de outros meios [Hil89].

Atualmente, as principais áreas de pesquisa em IA simbólica são: sistemas especialistas, aprendizagem, representação de conhecimento, aquisição de conhecimento, tratamento de informação imperfeita, visão computacional, robótica, controle inteligente, modelagem cognitiva, arquiteturas para sistemas inteligentes, linguagem natural e interfaces inteligentes. Em relação a estas áreas de pesquisa, a IAD aparece como um novo ponto de vista, onde as técnicas tradicionais da IA, combinadas com as funcionalidades dos sistemas distribuídos, dão origem à noção de *agente*, e conseqüentemente à de *sociedade de agentes*.

No início dos anos setenta, a IA já oferecia técnicas robustas para o desenvolvimento de aplicações práticas envolvendo sistemas inteligentes. Técnicas de representação de conhecimento, como quadros e redes semânticas, aliadas à tecnologia dos sistemas de produção formavam a base para a construção de ferramentas para desenvolvimento de sistemas especialistas. Estas ferramentas são baseadas, analogamente à grande parte da pesquisa em IA da época, na visão de um sistema inteligente composto por apenas um centro de controle, um foco de atenção e uma única base de conhecimento. Esta visão, herança do modelo computacional de von Neumann e das concepções da psicologia, tem como características principais a concepção centralizada, a "não-reusabilidade" de seus componentes e a não-abertura em direção ao exterior. No fim dos anos setenta, esta metáfora de *comportamento individual* é contrabalançada, por um lado, pelos trabalhos em redes neuronais, que incluem distribuição e paralelismo, e, por outro lado, pelos sistemas simbólicos baseados no chamado *modelo de quadro negro* (do inglês, "blackboard model") [HR85], que são caracterizados por um controle distribuído.

Estas primeiras idéias de distribuição em IA, aliadas à maturidade alcançada pelas tecnologias de redes e de sistemas distribuídos,[1] deram origem à área de IAD onde se desenvolvem métodos de IA para a solução distribuída de problemas.

2 Distribuição

Existem diversos motivos para distribuir sistemas inteligentes. O principal deles é que alguns domínios de aplicação – por exemplo, controle de tráfego aéreo, distribuição de energia elétrica, controle ambiental, etc. – são inerentemente distribuídos no espaço. Outras razões incluem:

- Melhorar a adaptabilidade, a confiabilidade e a autonomia do sistema.
- Reduzir os custos de desenvolvimento e manutenção.
- Aumentar a eficiência e a velocidade.
- Permitir a integração de sistemas inteligentes existentes de maneira a aumentar a capacidade de processamento e, principalmente, a eficiência na solução de problemas.
- Permitir a integração dos computadores nas redes de atividades humanas.

[1] Na década de noventa, foram realizados grandes esforços no sentido de padronizar a *interoperabilidade* de sistemas computacionais. Neste sentido, pode-se citar o modelo ODP (do inglês, "open distributed systems") [Tay92] proposto pela ISO ("International Standards Organization") e CCITT ("Consultative Committee on International Telephony and Telegraphy") e o modelo CORBA (do inglês, "common request broker architecture") [Rym91] proposto pelo OMG ("Object Management Group").

Além dessas razões, deve-se ainda salientar que, para problemas realmente grandes, a única possibilidade de solução é a solução distribuída, o que coloca a IAD como a única técnica indicada quando o problema ultrapassa um grau razoável de complexidade. Finalmente, o fato de a maioria das atividades inteligentes dos seres humanos envolverem mais de uma pessoa deveria ser motivo suficiente para que os aspectos sociais da inteligência fossem colocados entre os fundamentos de qualquer teoria da inteligência [Gas91].

As técnicas desenvolvidas em IAD permitem aplicações em diversos níveis: (i) o desenvolvimento de novas aplicações baseadas em metodologias tradicionais de desenvolvimento de "software", mas que se beneficiem, em seu projeto conceitual, das idéias da IAD. (ii) A expansão das funcionalidades de sistemas existentes através do "encapsulamento" destas aplicações em plataformas de IAD e, finalmente, (iii) o desenvolvimento de "verdadeiros sistemas de IAD" que incorporem as técnicas de IAD desde sua concepção até a implementação. Sistemas deste último tipo, para aplicações reais, ainda não foram construídos.

Os principais problemas tratados pela IAD são [Gas91]:

- Formular, descrever, decompor e alocar problemas e sintetizar resultados em um grupo de agentes inteligentes.
- Permitir a comunicação e interação entre agentes.
- Assegurar a coerência dos agentes a respeito de suas decisões e ações e evitar interações indesejadas.
- Permitir que agentes individuais raciocinem a respeito das ações, planos e conhecimentos dos outros agentes visando à coordenação e à cooperação entre os agentes.
- Reconhecer e conciliar diferentes pontos de vista e interesses dos diversos agentes.
- Criar metodologias e projetar ambientes para a implementação de sistemas inteligentes distribuídos.

Estes problemas são inerentes ao projeto e implementação de qualquer sistema que seja composto por solucionadores de problemas distribuídos e coordenados.

3 Dois enfoques

Por motivos históricos, a IAD se dividiu em dois enfoques: (i) a *solução distribuída de problemas (SDP)* (do inglês, "distributed problem solving") e (ii) *sistemas multiagentes (SMA)* (do inglês, "multi-agent systems") [DR94].

A SDP tem como foco principal o *problema*, conforme a tradição na IA simbólica, da qual este enfoque é diretamente derivado. Seus objetivos são utilizar a capacidade de processamento e a robustez oferecidas pela tecnologia de redes para atacar problemas naturalmente distribuídos ou excessivamente complexos, como, por exemplo, a decomposição e alocação de tarefas em uma rede de computadores ou o controle de tráfego aéreo. Para a SDP, os agentes são pré-programados para cooperar, e seus métodos visam garantir que esta cooperação ocorra de maneira coerente, robusta e eficiente. A qualidade de um sistema de SDP é medida pelo desempenho global do sistema na solução de um problema específico.

Os agentes envolvidos em SDP são programados para cooperar, dividir tarefas, comunicar-se de maneira confiável, etc. No entanto, experiências em ciências sociais mostram que não é simples estabelecer tais propriedades em uma coleção de indivíduos. O estudo das pressuposições básicas sobre agentes que garantam a possibilidade de ação cooperativa em sociedade é o foco de estudo dos SMA, isto é, neste

Tabela 1: Temas de pesquisa em IAD

	Propriedades		
	Agentes	Ambiente	Sistema
SMA	variável	fixa	fixa (interna)
SDP	fixa	variável	fixa (externa)
?	fixa	fixa	variável

caso o foco das pesquisas é o *agente*. O estudo de SMA é naturalmente multidisciplinar e envolve conceitos provenientes de diversas disciplinas, por exemplo, economia, teoria de jogos, ciências sociais, etologia, etc. Dentre estes conceitos, o de *agente racional*, isto é, aquele que age no sentido de maximizar seus benefícios e minimizar suas perdas, apresenta especial interesse para a área de SMA.

A relação entre as áreas de SDP e SMA, além de modificar-se ao longo tempo, não é clara. Três visões alternativas, mas não mutuamente exclusivas, são propostas em [DR94].

- SDP é um subconjunto de SMA: segundo esta visão, sistemas para SDP são SMA em que se verificam certas hipóteses sobre o comportamento dos agentes, por exemplo, benevolência e compartilhamento de objetivos globais. Uma terceira hipótese é de que sistemas para SDP são caracterizados por um projetista centralizador. Esta hipótese engloba as duas anteriores, pois neste caso o projetista pode introduzir benevolência ou objetivos comuns nos agentes sob sua responsabilidade. Esta visão, no entanto, não permite uma clara separação entre SMA e sistemas para a SDP, pois, para atingir objetivos comuns, nem sempre ações puramente cooperativas são as mais adequadas (por exemplo, um agente pode negar um recurso a outro por saber que a ação a ser realizada com o recurso não contribui para o objetivo global, o que visto por um observador externo pareceria um comportamento não cooperativo).

- SMA (e sistemas distribuídos) fornecem a base para SDP: esta visão se baseia nos fatos de que em SDP assume-se que os agentes devem comportar-se cooperativamente e que em SMA constata-se que isto não é tão simples e buscam-se meios para que isto possa realmente ser realizado. Assim a tarefa da SDP, que é solucionar o *problema* que deu origem ao sistema, pode ser subdividida. Em SMA estuda-se como uma comunidade de agentes pode ser levada a interagir cooperativamente e, em SDP, utiliza-se esta capacidade para levar a comunidade de agentes a solucionar um problema externo. Novamente, a linha de separação entre SDP e SMA não fica clara.

- SMA e SDP são linhas de pesquisa complementares: a dificuldade de separar SDP de SMA deve-se, defende esta visão, ao fato de que a diferença entre os dois enfoques se deve menos ao tipo de sistema desenvolvido e mais aos diferentes interesses teóricos subjacentes a esses sistemas. Assim, enquanto pesquisadores em SDP se concentram em questões a respeito da eficiência e robustez da comunicação em face de falhas como atraso e perda de mensagens, pesquisadores em SMA se preocupam com a possibilidade de cooperação mesmo que a comunidade contenha agentes insinceros e manipulativos.

Para caracterizar estes interesses teóricos que dividem os pesquisadores em IAD, Durfee e Rosenschein propõem uma classificação baseada nas propriedades dos elementos cuja variação cada enfoque está disposto a permitir durante seus experimentos. De acordo com esta classificação, mostrada no quadro 1, em SMA

estudam-se as propriedades de sistemas de agentes, em um ambiente dado, quando as propriedades internas destes agentes variam e, em SDP, estuda-se como garantir certas propriedades globais de um sistema de agentes, com propriedades preestabelecidas, quando o ambiente em que estes se encontram varia. Finalmente, os autores deixam em aberto a classificação de sistemas de agentes com propriedades fixas em ambientes fixos, mas cujas propriedades globais podem variar. No entanto, os autores não deixam de notar que redes neuronais e computação evolutiva são possíveis candidados para esta posição.

4 Solução distribuída de problemas

A SDP pode ser vista como a intersecção entre a IA e os *sistemas distribuídos* [DLC89]. Seu objetivo global é desenvolver técnicas de raciocínio e representação de conhecimento necessárias para que nodos, contendo solucionadores de problemas interligados em uma rede fracamente acoplada, cooperem efetivamente para solucionar um problema distribuído complexo. Entre seus objetivos genéricos pode-se citar:

- Aumentar a eficiência através do paralelismo.
- Aumentar o número de tarefas realizáveis através do compartilhamento de recursos (informação, conhecimento, recursos físicos, etc.).
- Aumentar a confiabilidade (tarefas duplicadas por diferentes métodos).
- Diminuir a interferência entre tarefas evitando interações inúteis.

Existem diversos domínios de aplicação para SDP: Interpretação Distribuída – por exemplo, redes de sensores, diagnóstico de falhas em redes de comunicação –, Planejamento e Controle Distribuídos – por exemplo, controle de tráfego aéreo, robôs cooperantes, veículos com controle remoto, controle distribuído de processos em manufatura, alocação de recursos em redes de comunicação –, Sistemas Especialistas Cooperantes – por exemplo, controle de veículos autônomos, negociação entre especialistas sobre preços e prazos – e Cooperação Humana via Computador – por exemplo, coordenação de projetos (do inglês, “groupware”), editores distribuídos.

Os principais temas de pesquisa em SDP são os seguintes: negociação, protocolo de licitação, negociação em diversos estágios, centralização de tarefas, cooperação funcionalmente correta, conciliação entre conhecimentos e pontos de vista globalmente inconsistentes e cooperação efetiva, estrutura organizacional e planejamento multiagentes.

5 Sistemas multiagentes

Não existe uma definição para *agente* que seja aceita por toda a comunidade de IAD. Uma possível definição é proposta por [FG91]:

> Chama-se agente uma entidade real ou abstrata que é capaz de agir sobre ela mesma e sobre seu ambiente, que dispõe de uma representação parcial deste ambiente, que, em um universo multiagente, pode comunicar-se com outros agentes, e cujo comportamento é conseqüência de suas observações, de seu conhecimento e das interações com outros agentes.

Esta definição se preocupa principalmente com os mecanismos internos de tratamento de cada agente, sem estabelecer, por exemplo, o tipo de comunicação possível entre agentes. Nada é dito também sobre a granularidade dos agentes. Uma definição complementar à primeira, que ressalta o aspecto *identidade* do agente, foi proposta por [Gas92]:

> Um agente é uma entidade à qual se pode associar uma identidade única, e que é capaz de realizar cálculos formais. Um agente pode ser considerado como um meio que produz um certo número de ações a partir dos conhecimentos e mecanismos internos que lhe são próprios.

A métafora de inteligência utilizada em SMA é a *comunidade inteligente*, isto é, o comportamento social que é a base para a inteligência do sistema. Os membros de uma comunidade inteligente podem ser desde extremamente simples até extremamente complexos. Os primeiros são chamados agentes *reativos* e os segundos agentes *cognitivos*.

Os agentes reativos são baseados em modelos de organização biológica ou etológica, como, por exemplo, as sociedades de formigas ou cupins. Embora uma formiga sozinha não possa ser considerada uma entidade inteligente, o formigueiro como um todo apresenta um comportamento visivelmente inteligente no sentido em que existe uma busca de alimentos e posterior estocagem, uma organização da reprodução com berçários e enfermeiras, etc. O modelo de funcionamento de um agente reativo é o de *estímulo-resposta*. Em geral, estes agentes não apresentam memória, não planejam sua ação futura e não se comunicam com outros agentes, cada agente tomando conhecimento das ações e comportamentos dos outros agentes apenas através de modificações no ambiente. Normalmente as sociedades de agentes reativos são numerosas, com populações da ordem de milhares de membros. Existem diversos trabalhos na literatura sobre SMA baseados em agentes reativos, por exemplo, a arquitetura de subsunção de Brooks [Bro86], o modelo PACO [Dem93], as ações situadas [RK86], entre outros.

Os agentes cognitivos são baseados em organizações sociais humanas como grupos, hierarquias e mercados. Os agentes possuem uma representação explícita do ambiente e dos outros agentes, dispõem de memória e, graças a esta memória, são capazes de planejar suas ações futuras. Estes agentes podem ainda comunicar-se entre si diretamente, isto é, seus sistemas de percepção (que permite examinar o ambiente) e de comunicação (que permite a troca de mensagens entre agentes) são distintos, o que não acontece com os agentes reativos. O número de membros em uma sociedade de agentes cognitivos, em geral, é da ordem de algumas dezenas no máximo. Diversos modelos de agentes cognitivos foram propostos na literatura. Alguns têm como objetivo a caracterização formal das atitudes mentais de um grupo de agentes, como objetivos, planos, intenções, etc., por exemplo, [CL87], [Pol90] e [Jen93], outros visam à construção de arquiteturas que permitam a implementação dos diversos mecanismos internos de um agente, por exemplo, [Lev90], [BD94] e [OC91], finalmente, alguns modelos apresentam características de ambos os enfoques anteriores, por exemplo, [Gas94] e [Sic95].

A diferença entre um agente reativo e um cognitivo pode ser também explicada como um compromisso entre eficiência e complexidade. Como o comportamento dos agentes reativos é definido funcionalmente, este pode ser compilado e executado com um atraso temporal pré-estabelecido, tornando os sistemas baseados neste tipo de agente adequados para aplicações em tempo real ou robótica. No entanto, a ausência de deliberação neste tipo de sociedade pode limitar a complexidade dos comportamentos possíveis.

Existem diversas aplicações desenvolvidas a partir do enfoque de SMA. Alguns exemplos que utilizam agentes reativos são: cartografia [BDA95], simulação de partículas [BJV95], a simulação de veículos autônomos [HDL92], robótica móvel [Ste90] e manipuladores em robótica [OPP95]. Outros exemplos de aplicações, agora construídas a partir de agentes cognitivos são: reconhecimento de linguagem natural

[Ste93], tutores inteligentes [CGR92], visão por computador [BD94], robótica [OC91], sistemas de automação industrial [Jen92] e telecomunicações [KDEQ95].

6 Exemplo: o sistema Expert-Coop

O exemplo apresentado a seguir faz parte dos resultados da dissertação de mestrado de Augusto Cesar Pinto Loureiro da Costa [Cos97]. O problema que originalmente motivou esse trabalho foi o processo de recomposição da rede de transmissão de energia elétrica da região sul do Brasil, após esta rede, ou parte dela, ter sido afetada por um "blackout".

No problema em questão, cada uma das unidades da rede de transmissão de energia elétrica – usina hidroelétrica, termoelétrica ou subestação – possui seu próprio domínio de conhecimento sobre os equipamentos e procedimentos operacionais das respectivas unidades (ex. grupo-gerador, linha de transmissão, disjuntores, barramentos, etc). Entretanto, informações globais tais como a completa topologia da rede não pertencem a esse domínio de conhecimento. O processo de recomposição é realizado atualmente pelos operadores das respectivas unidades, que seguem um conjunto de procedimentos pré-determinados de acordo com o estado específico da unidade na rede. Tais procedimentos podem ser codificados em um sistema especialista. Esses procedimentos locais são coordenados pelo Centro de Operação do Sistema (COS), localizado em Curitiba-PR, onde todas as informações referentes à rede estão disponiveis em um sistema de aquisição de dados. Por razões práticas esses procedimentos visam minimizar a comunicação entre os operadores das unidades em detrimento de uma operacionalidade ótima do sistema. Esse problema da recomposição caracteriza uma das situações do mundo real onde o comportamento cooperativo pode ser fundamental para a otimização da solução do problema.

Para solucionar este problema foi proposto um ambiente para desenvolvimento de SMA, chamado Expert-Coop [GBA97]. O ambiente suporta SMA *cognitivos*, *heterogêneos* e *abertos* projetados para cooperar, mesmo que os agentes pertencentes à comunidade não possuam uma representação do conhecimento global desta. O ambiente utiliza processos licitatórios como estratégia de cooperação. Na comunidade de agentes em questão, as metas globais a serem atingidas, através da cooperação de seus membros, surgem a partir das necessidades particulares de cada um dos agentes.

O ambiente foi implementado, utilizando-se o CMU Common Lisp/CLOS, segundo uma abordagem orientada a objetos, permitindo que novas funcionalidades possam ser facilmente adicionadas ao ambiente. Um outro aspecto que incrementa a portabilidade do ambiente é o fato dos mecanismos de comunicação serem baseados nos "sockets" do UNIX. Este mecanismo, suportado pelo CMU Common Lisp, assegura que as conexões realizadas na rede de computadores ocorrem de acordo com o modelo OSI. Dois mecanismos adicionais, provenientes da tecnologia de *Sistemas Distribuídos*, foram implementados no ambiente: *Ordenamento Total de Eventos* [Lam78] e *Tolerância a Falhas de Sistema* [Jal94]. O Expert-Coop é de domínio publico e compatível com os seguintes sistemas operacionais: SunOS, Solaris, HP-UX, Irix, FreeBSD e Linux.

Atualmente uma nova versão do Expert-Coop, está sendo implementada utilizando C++ como linguagem de programação. Também uma nova estrategia de cooperação chamada *DSK, Dynamic Social Knowledge* [CB98] será adicionada a esta nova versão. Esta nova versão do Expert-Coop permitira a implementação de sistemas multiagentes cognitivos com restrições de tempo real.

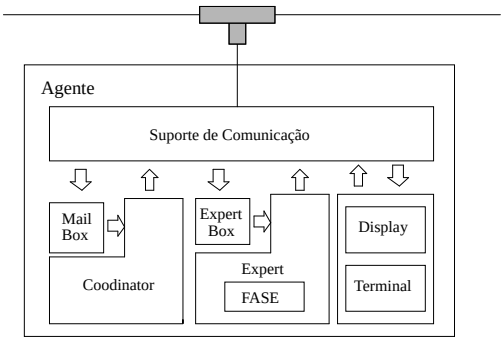

Figura 1: Arquitetura do Agente

6.1 Arquitetura do Agente

A arquitetura do agente suportado pelo Expert-Coop, mostrada na figura 1, é composta por seis processos concorrentes: Mail-box, Expert-box, Coordenador, Expert, Terminal e Display. Estes interagem através de troca de mensagens e de um suporte de comunicação de alto nível.

- **Mail-box:** O Mail-box é dedicado a receber todas as mensagens, provenientes dos demais agentes da comunidades e dos processos Expert e Terminal, e armazena tais mensagens em uma "caixa postal".

- **Expert-box:** O processo Expert-box, similar ao Mail-box, tem como objetivo receber todas as mensagens destinadas ao processo Expert. Essas mensagens são enviadas exclusivamente pelo Coordinator. O Expert-box foi adotado visando aumentar a concorrência entre os processos Coordinator e Expert, aumentando assim a eficiência do agente.

- **Coordinator:** O processo Coordinator é responsável pela realização das atividades de comunicação e cooperação entre os agentes. A comunicação consiste em coletar, ordenar e tratar as mensagens recebidas pelo agente, armazenadas na "caixa postal" do agente pelo processo Mail-box. A atividade cooperativa consiste em: decodificar mensagens, abrir e fechar licitações e requisitar ao processo Expert propostas a serem submetidas a licitações abertas por outros agentes. Além dessas atividades, o processo Coordinator também é responsável pela subscrição e desligamento de agentes da comunidade e pela execução do algoritmo de tolerância a falhas de sistema para assegurar que uma licitação aberta será finalizada com qualquer número de agentes ativos na comunidade.

- **Expert:** O processo Expert é responsável pelas habilidades cognitivas do agente. Este processo contém um sistema especialista encapsulado, mais todo o suporte de comunicação necessário para integrá-lo na arquitetura do agente. O sistema especialista associado ao processo Expert é baseado no sistema FASE [BM93]. Um manipulador de mensagens, chamado Fase-Comm, e uma nova estrutura para representação de conhecimento foram adicionados ao FASE. O Fase-Comm permite ao sistema especialista receber mensagens provenientes do processo Coordinator e respondê-las como se estivesse manipulando conhecimento em sua base local. Tais características permitem incluir comandos para envio e recebimento de mensagens nas regras do sistema especialista.

- **Terminal:** O processo Terminal possibilita ao usuário enviar mensagens ao processo Coordinator.

- **Display:** O processo Display permite a visualização de todas as mensagens válidas recebidas pelo agente. Este processo em conjunto com o processo Terminal formam a interface do agente com o usuário.

6.2 Comunicação

No Expert-Coop, todos os agentes possuem um processo chamado Mailbox que recebe as mensagens destinadas ao agente. Estas mensagens são periodicamente coletadas pelo processo Coordinator. Apenas o processo Coordinator se comunica com os demais agentes da comunidade. A comunicação entre dois agentes, ou dois processos de um mesmo agente, é realizada utilizando-se um mesmo padrão de mensagens. Esta mensagem é uma estrutura de dados contendo os seguintes atributos: From, To, Time-Stamp, Body, Round, Grade, Alfa. Por exemplo:

```
((From agent-1)
 (To agent-2)
 (Time-Stamp 13)
 (Body (ANNOUCE ((logic (sincronizar-grupo-gerador 2)))))
 (Round 13))
```

O atributo From, contém o nome de quem envia a mensagem: um agente, no caso de mensagens recebidas de outros agentes, ou um processo, no caso de mensagens recebidas de processos do próprio agente. O atributo To armazena o nome do agente ou processo a que se destina a mensagem. O atributo Body contém a mensagem propriamente dita, codificada de acordo com a linguagem para comunicação de agentes adotada. O atributo Time-Stamp contém o rótulo temporal da mensagem. O atributo Round é utilizado para associar a mensagem a um dado processo licitatório em andamento. O atributo Grade permite atribuir um valor numérico a uma proposta. Finalmente, o atributo Alfa permite estabelecer um corte α para determinar o vencedor de uma licitação. Dentre este atributos, From, To, Time-Stamp e Body são obrigatórios; Round, Grade e Alfa são facultativos.

Uma linguagem para comunicação de agentes, chamada, Parla [CB97], foi consebida para ser utilizada pelo Expert-Coop. As sentenças desta linguagem são utilizadas nos atributos Body das mensagens. A linguagem consiste em um conjunto de primitivas seguidas de um argumento. Este argumento pode ser qualquer um dos formalismos de representação de conhecimento suportados pelo FASE: lógica, quadros ou redes semânticas. As primitivas de comunicação que compõem a linguagem informam ao agente que ação deve ser tomada com a respectiva mensagem. As primitivas suportadas pela linguagem são as seguintes:

- **ANNOUNCE:** Utilizada por um agente, para informar aos demais agentes da comunidade a abertura de uma licitação. Neste caso o objeto da licitação será representado no formalismo, passado como parâmetro, o atributo Alfa poderá ser utilizado para estabelecer um corte α para a licitação e o atributo Round obrigatoriamente conterá o identificador da licitação.

- **ACCEPT:** Utilizada para envio de uma proposta de realização de uma tarefa licitada. Essa proposta terá o mesmo Round da mensagem referente ao edital da licitação (primitiva ANNOUNCE). Um valor numérico associado à proposta será armazenado no atributos Grade.

- **INFORM:** Utilizada para enviar uma informação a um processo ou agente.

- **CONFIRM:** Utilizada pelo agente responsável pela licitação para informar ao agente vencedor da licitação que sua proposta foi a vencedora. Essa proposta terá o mesmo Round da mensagem referente ao edital da licitação (primitiva ANNOUNCE).
- **RECALL:** Utilizada pelo agente responsável pela licitação para informar aos demais agentes que suas respectivas propostas não obtiveram êxito.
- **REFUSE:** Utilizada por um ou mais agentes da comunidade para enviar uma proposta, rejeitando realizar a tarefa licitada pelo agente que abriu a licitação. Essa proposta terá o mesmo Round da mensagem referente edital da licitação (primitiva ANNOUNCE).
- **REQUEST:** Utilizada para solicitar uma informação a um agente ou processo.
- **REPLY:** Utilizada para responder uma solicitação (primitiva REQUEST).
- **SUBSCRIBE:** Utilizada por um agente que não pertence à comunidade para solicitar sua inclusão. Esta solicitação pode ser feita a qualquer membro da comunidade e no formalismo de representação de conhecimento passado como parâmetro, deve aparecer o nome do agente e seu "host". Uma vez recebida tal solicitação o agente inclui o novo agente na lista de membros e difunde a nova lista de participantes da comunidade.
- **UNSUBSCRIBE:** Utilizado para solicitar a saída da comunidade.

O ambiente permite ainda adicionar facilmente novas primitivas à linguagem.

6.3 Estratégia de Cooperação

O comportamento cooperativo da comunidade de agentes é obtido utilizando-se a estratégia de licitação. Quando um dos agentes da comunidade necessita realizar uma atividade cooperativa, ele abre uma licitação e difunde na comunidade o edital da licitação. Uma vez recebido um edital, os demais agentes da comunidade responderão com propostas, aceitando ou rejeitando a realização da tarefa licitada. Cada uma das propostas em que o agente se dispõe a realizar a tarefa licitada, possui um valor numérico associado a proposta. Esse valor numérico, chamado de Grade, expressa quão bem o agente pode realizar a tarefa licitada. Uma vez recebida uma proposta de cada um dos agentes ativos da comunidade, a melhor proposta, a que apresentar o maior Grade, é proclamada vencedora. Os processos licitatórios podem ser abertos paralelamente por qualquer um dos agentes da comunidade, todos os agentes possuem o mesmo nível hierárquico, sendo que cada um dos agentes é responsável pelas licitações por eles abertas.

6.4 Recomposição

Uma parte da chamada Área 1 da rede de transmissão de energia elétrica é apresentada na figura 2. Três usinas hidroelétricas, Salto Santiago (UHSS), Segredo (SGD) e Bento Munhoz (GBM), e duas subestações, Areia (ARE) e Ivaiporã (IVP), são apresentadas. Cada uma das usinas possui quatro grupo geradores, de diferentes potências. Segundo a *Instrução de Operação da Eletrosul* para a recomposição fluente, para recompor a linha de transmissão LT-IVP, o operador da usina UHSS precisa certificar-se de que ao menos quatro dos doze grupo geradores estejam sincronizados, de que as linhas de transmissão LT-SGD, LT-ITA, LT-ARE e LT-ARE-1 já tenham sido recompostas e finalmente e de que o anel de sincronismo entre LT-ARE

e LT-ARE-1 tenha sido fechado. Atualmente a verificação desta condições é realizada pelos operadores das unidades envolvidas e o COS através de um sistema de comunicação via microondas. Devido a esta limitações a recomposição desta linha de transmissão pode demorar até vinte minutos.

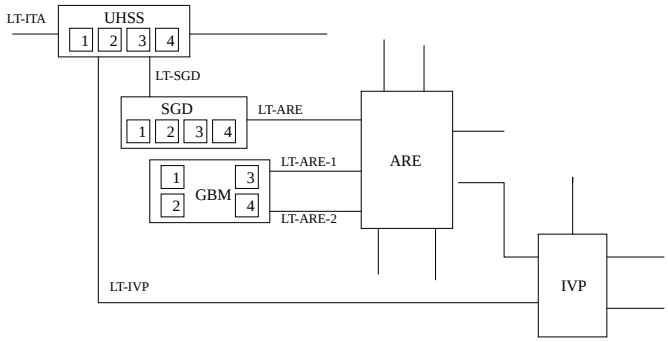

Figura 2: Parte da Rede de Transmissão de enegia elétrica da área 1 da Região Sul do Brasil

A abordagem multiagentes proposta consiste em alocar um agente cognitivo em cada subestação ou usina da rede onde será tratado o problema. Cada um destes agentes contém (na base de conhecimento do processo Expert) o conhecimento necessário sobre a respectiva unidade, subestação ou usina. A idéia central levantada pelo comportamento da comunidade de agentes proposta é que na ocorrência de uma situação anormal em alguma das unidades, caso uma determinada unidade necessite da ajuda de outras unidades da vizinhança, o respectivo agente pode iniciar um processo de cooperação entre os agentes da comunidade, que conduzirá para a solução do problema sem que haja a intervenção do COS. A identificação da situação e as respectivas trocas de mensagens necessárias à solução do problema podem ser codificadas em forma de regras no processo Expert existente em cada um dos agentes da comunidade. Por exemplo:

```
(rule-19
  ((logic (request (recompose lt uhss-ivp)))
   (frame (circuit-breaker uhss (dj-1090 off))
          (circuit-breaker uhss (dj-1092 off))))
  ((export (request (synchronized-generators x y))
   (lisp (message-to all)))).
```

Por questões práticas, as seguintes hipóteses simplificadoras foram adotadas: a requisição para recompor a linha de transmissão que leva energia elétrica de UHSS para IVP parte da interface do agentes UHSS ao invés de ser enviada por um outro agente que estaria na subestação IVP; o anel de sincronismo entre LT-ARE e LT-ARE-1 encontra-se fechado; as transformações do ambiente, ou seja a alteração dos estados dos geradores, barramentos disjuntores, etc. serão percebidos através de mensagens enviadas pelo terminal da interface.

Uma vez recebida a requisição para recompor a linha de transmissão, o agente UHSS, solicita aos agentes SGD e GBM o número de grupo geradores sincronizados em suas respectivas unidades (veja tabela 2).

De acordo com as mensagens recebidas dos demais agentes da comunidade (ver tabela 2), existem três geradores sincronizados, sendo necessário licitar apenas mais um gerador. Então o processo Expert solicita ao processo Coordinator que seja aberta uma licitação correspondente ao fornecimento de mais um gerador. O processo Coordinator manterá esta licitação aberta até que cada um dos agentes ativos tenha colocado sua proposta (ver tabela 3).

Tabela 2: Mensagens recebidas pela usina UHSS

Message Body	From	To	Rd	Gd	T.S.
(REQUEST ((logic (recompose lt uhss-ivp))))	Inter.	Coord.			1
(REQUEST ((logic (recompose lt uhss-ivp))))	Coord.	Expert			3
(REQUEST ((logic (synchronized-generators x y))))	Expert	Coord.			5
(REQUEST ((logic (synchronized-generators x y))))	UHSS	All			9
(REPLY ((logic (synchronized-generators sgd 1))))	SGD	UHSS			11
(REPLY ((logic (synchronized-generators gbm 1))))	GBM	UHSS			13
(REPLY ((logic (synchronized-generators sgd 1))))	Coord.	Expert			15
(REPLY ((logic (synchronized-generators gbm 1))))	Coord.	Expert			17

Tabela 3: Mensagens recebidas pela usina UHSS

Message Body	From	To	Rd	Gd	T.S.
(REQUEST ((logic (synchronize-generator x 1))))	Expert	Coord.	0		21
(ANNOUCE ((logic (synchronize-generator x 1))))	UHSS	All	23		26
(ANNOUCE ((logic (synchronize-generator x 1))))	Coord.	Expert	23		28
(ACCEPT ((logic (synchronize-generator gbm 1))))	GBM	UHSS	23	0.8	30
(ACCEPT ((logic (synchronize-generator sgd 1))))	SGD	UHSS	23	0.6	32
(ACCEPT ((logic (synchronize-generator uhss 1))))	Expert	Coord.	23	0.6	34

Uma vez recebidas todas as propostas, a melhor delas (ver tabela 3), isto é, a que apresentar o maior Grade, será declarada vencedora e o agente que a enviou receberá uma mensagem com a primitiva CONFIRM. Os demais agentes receberão mensagens com a primitiva RECALL, informando-os de que não obtiveram êxito na licitação (ver tabela 4).

Tabela 4: Mensagens recebidas pela usina UHSS

Message Body	From	To	Rd	Gd	T.S.
(CONFIRM ((logic (synchronize-generator uhss 1))))	UHSS	GBM	23	0.8	36
(RECALL ((logic (synchronize-generator x 1))))	UHSS	LOOSERS	23	0.6	39
(RECALL ((logic (synchronize-generator x 1))))	Coord.	Expert	23	0.6	41
(REPLY ((logic (synchronize-generator gbm 1))))	Coord	Expert	23	0.8	43
(REPLY ((logic (recomposed lt uhss-ivp))))	Expert	Coord.			45

7 Conclusão

As perspectivas de desenvolvimento da IAD podem ser medidas pela magnitude do evento *Agents' World* realizado em Paris, França, de 04 a 07 de julho de 1998. Este evento reuniu o ICMAS'98 (3rd International Conference on Multi-Agent Systems), a mais importante conferência internacional na área de SMA, e diversos workshops sobre assuntos relacionados: 2nd International Workshop on Cooperative Information

Agents (CIA'98), 2nd International Workshop on Intelligent Agents for Telecommunications Applications (IATA'98), 1st International Workshop on Collective Robotics (CRW'98), 1st International Workshop on Agents in CommunityWare (ACW'98), 1st International Workshop on Multi-Agent Systems and Agent-Based Simulation (MABS'98), além de duas copas do mundo de robots a RoboCup'98 e a MIROSOT'98.

Os títulos dos Workshops mostram as principais áreas onde a tecnologia de sistemas multiagentes tem encontrado aplicações: sistemas de informação, telecomunicações, robótica, trabalho cooperativo e simulação. Outra área de aplicação interessante são as competições de futebol de robôs. A tecnologia necessária para colocar um time de robôs em campo e fazê-los jogar cooperativamente evolve desde problemas de controle e visão computacional até problemas de representação de conhecimento e raciocínio simbólico, passando por enfoques neurais e/ou evolutivos. O interessante é que o resultado pode ser testado através dos jogos, criando um campo de experimentação padronizado para comparar técnicas de IAD.

Pode-se concluir que a IAD tem muito a contribuir para o futuro da IA em particular e da computação em geral, principalmente no que diz respeito à ambientes como a Internet e a aplicações de robótica.

Referências

[BD94] O. Boissier and Y. Demazeau. Mavi: A multi-agent system for visual integration. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, USA*, pages 731–738, 1994.

[BDA95] C. Baeijs, Y. Demazeau, and L.O. Alvares. Application des systèmes multi-agent à la généralization cartographique. In *Actes des 3èmes Journées Francophones d'Intelligence Distribuée & Systèmes Multi-Agents, Université de Savoie, Chambéry, France*, March 1995.

[BJV95] S. Bijnens, W. Joosen, and P. Verbaeten. Language constructs for coordination in an agent space. In Yves Demazeau, Jean-Pierre Müller, and John W. Perram, editors, *Pre-Proceedings of the 6^{th} European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Odense, Denmark*, pages 65–77, August 1995.

[BM93] G. Bittencourt and M. Marengoni. A customizable tool for the generation of production-based systems. In G. Rzevski, J. Pastor, and R.A. Adey, editors, *Eighth International Conference on Applications of Artificial Intelligence in Engineering (AIENG'93)*, pages 337–352. Elsevier Applied Science, 1993.

[Bro86] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):435–453, March 1986.

[CB97] A.C.P.L. da Costa and G. Bittencourt. Parla: A cooperation language for cognitive multi-agent systems. *EPIA'97, 8th Portuguese Conference of Artificial Inteligence*, 1323:207–215, October 1997. Spring-Verlag, Lecture Notes in Artificial Inteligence.

[CB98] A.C.P.L. da Costa and G. Bittencourt. Dynamic social knowledge: A cooperation strategie for cognitive multi-agent systems. *Third International Conference on Multi-Agent Systems (ICMAS'98)*, pages 415–416, Paris, France, July 2-7 1998. IEEE Computer Society.

[CGR92] H. Coelho, G. Gaspar, and I. Ramos. Experiments on achieving communication in communities of autonomous agents. In *Proceedings of the IFIP WG 8.3 Working Conference on Decision Support Systems: Experiments and Expectations, Fontainbleau, France*, 1992.

[CL87] P.R. Cohen and H.J. Levesque. Intention = choice + commitment. In *Proceedings of AAAI-87, Seattle*, pages 410–415, 1987.

[Cos97] A.C.P.L. Costa. *Expert-Coop: Um Ambiente para Desenvolvimento de Sistemas Multi-Agentes Cognitivos*. Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1997. Dissertação de Mestrado.

[Dem93] Y. Demazeau. La plate-forme paco et ses applications. In Yves Demazeau and Anne Collinot, editors, *Actes de la 2ème Journée Nationale du PRC-GDR Intelligence Artificielle, Montpellier, France*, December 1993.

[DLC89] E.H. Durfee, V.R. Lesser, and D.D. Corkill. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):63–83, March 1989.

[DR94] E.H. Durfee and J.S. Rosenschein. Distributed problem solving and multi-agent systems: Comparisons and examples. In *Proceedings of the International Workshop on Distributed Artificial Intelligence*, July 1994.

[FG91] J. Ferber and L. Gasser. Intelligence artificielle distribuée, 1991. Tutorial Notes of the 11^th Conference on Expert Systems and their Applications, Avignon'91, France.

[Gas91] L. Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence (Special Volume Foundations of Artificial Intelligence)*, 47(1-3):107–138, January 1991.

[Gas92] L. Gasser. Boundaries, identity and aggregation: Plurality issues in multiagent systems. In Eric Werner and Yves Demazeau, editors, *Decentralized Artificial Intelligence*, pages 199–212. Elsevier Science Publishers, Amsterdam, NL, 1992.

[Gas94] G. Gaspar. *Modelação de Agentes Autónomos Inteligentes Integrados em Sociedades de Agentes*. PhD thesis, Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa, July 1994.

[GBA97] *Expert-Coop: An Environment for Cognitive Multi-Agent Systems*, volume 2, October 1997.

[HDL92] M. Hassoun, Y. Demazeau, and C. Laugier. Motion control for a car-like robot: Potential field and multi-agent approaches. In *Proceedings of the 1992 International Conference on Intelligent Robots and Systems (IROS'92), Raleigh, USA*, August 1992.

[Hil89] W.C. Hill. The mind at ai: Horseless carriage to clock. *The AI Magazine*, pages 29–41, Summer 1989.

[HR85] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, July 1985.

[Jal94] P. Jalote. *Fault Tolerance in Distributed System*. PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[Jen92] N.R. Jennings. Towards a cooperation knowledge level for collaborative problem solving. In Bernd Neumann, editor, *Proceedings of the 10^{th} European Conference on Artificial Intelligence, Viena, Austria*, pages 224–228, August 1992.

[Jen93] N.R. Jennings. Commitments and conventions: the foudations of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[KDEQ95] J.-L. Koning, Y. Demazeau, B. Esfandiari, and J. Quinqueton. Quelques perspectives dútilization des langages et protocoles d'intéraction dans le contexte des telecommunications. In *Actes des 3èmes Journées Francophones d'Intelligence Distribuée & Systèmes Multi-Agents, Université de Savoie, Chambéry, France*, March 1995.

[Lam78] L. Lamport. Time, clocks, and ordering of events. *Communications of ACM*, 21(7):558–565, July 1978.

[Lev90] P. Levi. Architectures of individual and distributed autonomous agents. In T. Kanade, F.C.A Groen, and L.O. Hertzberger, editors, *Intelligent Autonomous Agents 2*, pages 315–324. IAS, Amsterdam, NL, 1990.

[OC91] E. Oliveira and R. Camacho. A shell for cooperating expert systems. *Expert Systems*, 8(2):75–85, May 1991.

[OPP95] L. Overgaard, H.G. Petersen, and J.W. Perram. Motion planning for an articulated robot: A multi-agent approach. In Yves Demazeau, Jean-Pierre Müller, and John W. Perram, editors, *Pre-Proceedings of the 6^{th} European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Odense, Denmark*, pages 171–182, August 1995.

[Pol90] M.E. Pollack. Plans as complex mental attitudes. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions on Communication*, pages 77–103. MIT Press, Cambridge, MA, 1990.

[RK86] J.S. Rosenschein and L.P. Kaelbling. The systhesis of digital machines with provable epistemic properties. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge*, pages 83–98. Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1986.

[Rym91] J.R. Rymer. Common object request broker - omg's new standard for distributed object management. *Network Monitor*, 6(9):3–27, September 1991.

[Sic95] J.S. Sichman. *Du raisonnement Social chez les agents, une approche fondée sur la thèorie de la dèpendance*. PhD thesis, Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle, Institut Polytechnique de Grenoble, France, September 1995.

[Ste90] L. Steels. Cooperation between distributed agents through self-organization. In Yves Demazeau and Jean-Pierre Müller, editors, *Decentralized Artificial Intelligence*, pages 175–196. Elsevier Science Publishers, Amsterdam, NL, 1990.

[Ste93] M.-H. Stefanini. Vers une architecture multi-agents pour le traitement automatique des langues naturelles. In *Actes des 1ères Journées Francophones d'Intelligence Distribuée & Systèmes Multi-Agents, Toulouse, France*, April 1993.

[Tay92] C.J. Taylor. A status report on open distributed systems. *First-Class, Object Management Group Newsletter*, 3(2):11–13, 1992.