

Um modelo de ambiente interativo de aprendizagem baseado numa arquitetura multi-agentes¹

Evandro de Barros Costa

Resumo. A maioria da pesquisa relacionada à concepção de Sistemas Tutores Inteligentes envolve interações tutoriais entre duas entidades principais, a saber: um computador desempenhando um papel de tutor e um aprendiz humano. Nesse contexto, um dos problemas fundamentais em boa parte da pesquisa em Sistemas Tutores Inteligentes diz respeito à possibilidade de um tal sistema prover ações individualizadas a estudantes particulares, adaptando-se ao seu perfil cognitivo. Essa capacidade de adaptação tem sido assumida como central na busca por um processo de ensino e aprendizagem mais efetivo. A pesquisa, ora veiculada no presente texto, desenvolve-se nessa direção, situando-se na concepção de tais sistemas, visando propor um modelo de ambiente interativo de ensino e aprendizagem baseado numa arquitetura multi-agentes. Em particular, esse modelo de ambiente busca primeiramente dar suporte às interações pedagógicas entre um Aprendiz humano e uma sociedade de agentes tutores artificiais. Essa sociedade, por sua vez, é mantida por uma sociedade de especialistas humanos. Com o referido modelo podem ser desenvolvidos sistemas específicos, considerando principalmente os seus princípios de concepção, arquitetura e comportamento. No trajeto da concepção do referido ambiente, investiu-se na estrutura do domínio de conhecimento, definindo-lhe uma visão multidimensional. Daí, adotou-se uma abordagem multi-agentes baseada nessa visão e assim definiu-se uma sociedade de agentes tutores. Enfim, foram desenvolvidos os principais elementos responsáveis por viabilizar as interações envolvidas. Entre esses elementos, incluem-se: modelo de agente, arquitetura de agente, linguagem e protocolos de interação. Finalmente, convém destacar que o ambiente de aprendizagem concebido juntamente com sua arquitetura multi-agentes representam uma contribuição horizontal para a tese. A contribuição vertical da presente investigação, por sua vez, está relacionada basicamente com a proposição de um modelo multidimensional para o domínio de aplicação, de um sistema tutor multi-agentes e de um processo de modelagem distribuído de apoio ao diagnóstico das ações do aprendiz.

Sumário

1. Introdução

- 1.1 Motivação e História
- 1.2 Contexto da Pesquisa
- 1.3 Problemática e Objetivos da Pesquisa de Tese
- 1.4 Organização da Tese

2. Noções Preliminares

- 2.1 Da IA Clássica para a IA Distribuída
- 2.2 Aspectos Educacionais
- 2.3 Inteligência Artificial em Educação

3. Tutores Inteligentes via Agentes

- 3.1 Considerações Preliminares
- 3.2 Principais Abordagens

4. Concepção do Ambiente MATHEMA

- 4.1 Prolegômenos
- 4.2 Do Modelo Mínimo de Aprendizagem ao MATHEMA
- 4.3 Arquitetura do MATHEMA
- 4.4 Funcionalidade Geral
- 4.5 Considerações sobre as Interações no MATHEMA

5. Sociedade de Agentes Tutores Artificiais

- 5.1 Características Gerais
- 5.2 Arquitetura de um Agente Tutor
 - 5.2.1 Nível Macro
 - 5.2.2 Nível Micro
- 5.3 Formalização do Modelo de Agente Tutor

6. Visão Funcional das Atividades Cooperativas

- 6.1 Mecanismo de Raciocínio Social
- 6.2 Coordenação e Cooperação
- 6.3 Tratamento dos Protocolos
- 6.4 Tratamento das Mensagens

7. Interações entre Agentes Tutores

- 7.1 Linguagem Social
- 7.2 Linguagem de Interação
- 7.3 Protocolos de Interação

8. Interações entre um Aprendiz e um Agente Tutor

- 8.1 Modelo de Ensino-Aprendizagem
- 8.2 Modelo de Interação
- 8.3 Arquitetura de um Sistema Tutor
- 8.4 Sobre o Funcionamento Básico do Sistema Tutor
- 8.5 O Sistema Tutor Multi-agentes

9. Aspectos de Implementação e Aplicações

- 9.1 Aspectos de Implementação
 - 9.1.1 A Classe Agente
 - 9.1.2 A Classe Tutor
 - 9.1.3 A Classe Social
- 9.2 Aplicações

10. Conclusões e Perspectivas Futuras

- 10.1 Situação do Trabalho
- 10.2 Principais Contribuições da Tese
- 10.3 Algumas Reflexões sobre a Tese
- 10.4 Perspectivas Futuras

Referências Bibliográficas

¹ Tese submetida à Coordenação de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba, como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica e defendida em dezembro de 1997. *Orientadores:* Edilson Férneda e Angelo Perkusich

1. Introdução

O trabalho aqui relatado situa-se na área que envolve a concepção e o desenvolvimento de ambientes de aprendizagem assistidos por computador. Neste capítulo, apresenta-se um panorama geral da presente tese. Inicia-se por uma breve motivação para se investir na referida área. Em seguida há um relato histórico e taxonômico relativo às idéias que permeiam os trabalhos desenvolvidos na área onde se localiza a pesquisa ora veiculada neste documento. Prossegue-se com uma discussão sobre o contexto, a problemática e os objetivos do trabalho da tese em questão. Por fim, apresenta-se a organização desta tese.

1.1 Motivação e história

A idéia de empregar o computador no suporte a atividades em educação e treinamento não é nova. Muito pelo contrário, ela surgiu pouco tempo depois da construção e uso do computador digital. Daí então, surgiram os ambientes computacionais destinados a tais atividades, sendo estes apoiados em propostas as mais variadas, refletindo, direta ou indiretamente, alguma concepção pedagógica. Assume-se neste documento que todas as iniciativas relacionadas à concepção dos referidos ambientes fazem parte do trabalho na vasta área denominada, entre a comunidade acadêmica brasileira, de Informática na Educação. Tais iniciativas podem, com efeito, ser vistas como configurações diferentes referentes aos papéis atribuídos, de algum modo, a três grandes entidades: o aprendiz humano, o professor e a máquina (através do *software*).

Sem dúvidas, há motivações e justificativas demais para se investir nessa área. Por um lado, existe uma quantidade significativa de problemas desafiadores de pesquisa, onde as soluções para os mesmos abrangem diferentes áreas do conhecimento, a exemplo da Psicologia, Educação, Linguística e Ciência da Computação. Portanto, trata-se de uma área altamente interdisciplinar. Por outro lado, há uma demanda social cada vez maior por resultados que possam ser originados de tais pesquisas. Observe-se a velocidade com a qual os computadores estão entrando nas instituições educacionais, e, por conseguinte, a necessidade por um suporte de *software* educativo adequado que atenda à demanda por tal tecnologia.

É difícil conseguir uma completeza num levantamento dos ambientes em questão, pois está em jogo toda uma perspectiva histórica, que pode ser localizada acompanhando-se uma trajetória vasta que se iniciou na década de 50, e que continua nos dias atuais. Neste sentido, historiar e classificar as iniciativas em concepção e consolidação dos ambientes de aprendizagem assistidos por computador não é uma tarefa fácil. Entretanto, sem pretensões de esgotar o assunto, procurou-se neste documento investir numa tal tarefa, oferecendo ao leitor uma visão da questão de maneira particular, abrangente horizontalmente e, ao mesmo tempo, sucinta. Neste sentido, buscou-se uma abordagem histórico-taxonômica, vista numa perspectiva que contempla três momentos cronologicamente distintos, a saber: o primeiro abrange os enfoques surgidos até meados da década de 70, o segundo é o que se inicia por volta da segunda metade da referida década até o final da década de 80, e finalmente o terceiro se inicia na década de 90, desenvolvendo-se no presente momento. Nesse ponto, convém a ressalva de que a entrada em um momento mais atual, não implica necessariamente o cancelamento das atividades do

anterior, ao contrário, em muitos casos, o que se observa é apenas a ocorrência de extensão, ou mesmo abordagens híbridas.

1.1.1 O primeiro momento

O primeiro momento, envolve as três categorias seguintes, que coincidem com as apontadas em [BF 82]:

- (i) Sistemas de Instrução Assistida por Computador (da sigla inglesa *CAI: Computer Aided Instruction*),
- (ii) Micromundos e
- (iii) Simuladores e Jogos Educacionais.

As categorias (i) e (ii) são de enfoques pedagógicos extremamente opostos, no que diz respeito, por exemplo, aos papéis desempenhados pelo aprendiz e pela máquina. Já a categoria (iii) pode ser vista como situada numa posição mais intermediária às duas primeiras, estando em algumas ocasiões mais próxima de (i) ou de (ii).

Os sistemas *CAIs* foram os primeiros a surgir como uma categoria de *software* educacional. Eles enfocam um modelo computacional de ensino e aprendizagem inspirado na idéia de instrução programada, apoiando-se numa abordagem comportamentalista (*behaviorista*) [Ski 58]. Uma de suas características básicas é a tentativa de redução do processo de aprendizagem, a um modelo causal do tipo estímulo-resposta. Nesse sentido, um sistema *CAI* opera com base no conceito de “programa linear” [VO 92]. Isto se traduz numa situação na qual o sistema apresenta uma determinada unidade de ensino ao estudante, visando conduzi-lo a um dado comportamento. A realização disso é obtida mediante a proposição de questões, sobre uma unidade de ensino, ao estudante. O estudante, por sua vez, responde às questões e o sistema devolve imediatamente as realimentações correspondentes. Com isso, diz-se que o aluno pode “aprender” em seu próprio ritmo.

Uma das mais significativas limitações atribuídas aos sistemas *CAIs* é que eles não consideram as características cognitivas individuais dos estudantes, como por exemplo: conhecimento prévio do domínio, estilo e capacidade de aprendizagem. Isso significa que todos os estudantes recebem uma mesma unidade de ensino, na mesma forma e seqüência. Dessa maneira, impõem uma postura de atuação passiva ao estudante.

Os Micromundos [Pap 85, Pap 87] surgiram na década de 60, constituindo uma proposta pedagógica extremamente contrária aos *CAIs*. Sua proposta central recai sobre a aprendizagem pela ação, sob uma perspectiva de construção do conhecimento. Os Micromundos foram propostos, inicialmente, pelo pesquisador Seymour Papert e sua equipe, no *Massachusetts Institute of Technology* (MIT). Uma das primeiras concretizações dessa categoria se deu com o surgimento do projeto do ambiente LOGO. Nesse ambiente, encontra-se disponível, dentre outros recursos, um micromundo gráfico. Nesse micromundo há um objeto representado por uma tartaruga, que é controlado pelo usuário através da manipulação de primitivas de uma linguagem de programação subjacente ao ambiente em apreço. É com esta tartaruga que o estudante interage de forma ativa para resolver problemas.

Além do LOGO, existem muitos outros Micromundos, tanto com propostas amplas de aplicação, como aqueles que são orientados a domínios de aplicações específicas.

Um dos Micromundos já bem consolidado e adotado, inclusive em algumas instituições educacionais no Brasil é o Cabri-

géomètre (“**Cahier de b**ruillon interactif en géométrie” - Caderno de rascunho interativo em geometria) [Bau 89,90]. Trata-se de um micromundo desenvolvido pelo LSDD-IMAG (Grenoble, França). Seu objetivo central é servir de ferramenta de aprendizagem em Geometria Euclidiana Plana, fornecendo um conjunto de objetos desse domínio e um conjunto de ferramentas de manipulação direta desses objetos. Com isso, permite ao aluno realizar seus experimentos no escopo do referido domínio.

Com os Micromundos, a ênfase no processo de aprendizado está na construção do conhecimento por parte do aprendiz. Isso se inspira em aportes construtivistas ou mesmo sócio-construtivistas. Para tanto, levam-se em consideração, principalmente, as idéias difundidas por Piaget [Pia 77], Bruner [Bru 69] e outros, com respeito ao construtivismo, e Vygotsky [Vyg 91] e mais alguns outros pesquisadores no tema, no caso do sócio-construtivismo. Dessa maneira, os micromundos contrapõem-se à abordagem adotada pelos sistemas CAIs, onde a essência no processo de aprendizagem está na transmissão do conhecimento. Uma das limitações atribuídas aos Micromundos (pelo menos aos tradicionais) é concernente à falta de realimentação adequada por parte do computador, ou seja, o computador desempenha um papel de suporte quase passivo junto às ações do aprendiz.

Os Simuladores e os Jogos Educacionais apresentam, circunstancialmente, aspectos dos CAIs e/ou dos Micromundos. Os Simuladores envolvem a criação de modelos do mundo real ou imaginário. Nos Simuladores o computador recebe entradas e então responde como se fosse o sistema que está sendo simulado, permitindo aos estudantes experimentarem os resultados de decisões boas ou más, sem riscos ou consequências caras. Só para tentar esclarecer, considere, por exemplo, os recursos oferecidos pelos simuladores de voo. A simulação oferece a possibilidade do aluno desenvolver hipóteses, testá-las, analisar resultados e refinar os conceitos. Já os jogos oferecem um aspecto lúdico incorporado num conteúdo instrucional a ser trabalhado, visando assim proporcionar mais motivação através de algo como “aprender brincando” (sendo comum, por exemplo, a disponibilidade para criação de cenários, personagens, etc.). Nesse sentido, boa parte dos jogos propõem-se a colocar os seus usuários diante de situações de desafios, em atividades de resolução de problemas. Uma das críticas dirigidas a certos jogos é que suas interfaces muitas vezes desviam a atenção dos aprendizes na percepção dos conceitos envolvidos nas atividades interativas.

Dependendo do nível de intervenção oferecido por um *software* para simulação ou jogo, estes ficarão mais distantes (tendendo para Micromundo) ou mais próximos de um CAI.

1.1.2 O segundo momento

No segundo momento, o que se observa é, de certo modo, uma expansão e eventualmente transformação das categorias citadas no primeiro momento, só que agora contando com duas fortes alianças. Por um lado, a aliança das áreas de Inteligência Artificial (IA) e Psicologia Cognitiva, e por outro, a disponibilidade de um aparato tecnológico mais apropriado, como, por exemplo, *hardware* e *software* mais possantes, tecnologia de comunicação mais sofisticada, implicando certas melhorias.

Com respeito às influências da tecnologia, os CAIs, por exemplo, ressurgem atrelados à tecnologia de Multimídia, utilizando-se, em muitos casos, dos sistemas hipermídia (hipertexto + multimídia). Do mesmo modo, as outras categorias também passaram a tirar

proveito de tal tecnologia. Ainda nesta ótica, outros avanços fazem surgir categorias que incluem Telemática e Robótica Educacional. Telemática diz respeito ao uso do computador em educação através das redes de computadores. O termo telemática vem da união de (Tele)comunicações mais Infor(mática). Uma concretização dessa categoria pode ser verificada através de trabalho cooperativo interpessoal e interinstitucional através da rede *Internet*, a exemplo do projeto *Kidlink* descrito, por exemplo, em [GC 93]. A Robótica Educacional é uma categoria que tem objetivos relacionados à construção e intermediação entre o computador e certos tipos de dispositivos [Abr 91] (por exemplo, os de natureza eletromecânica). Um exemplo bem difundido dessa categoria é a ferramenta LEGO-LOGO.

Com a incorporação de recursos da IA e resultados da Psicologia Cognitiva e Educação [Kea 87], os sistemas CAI dão origem aos sistemas *Intelligent CAI (ICAI)* ou Sistemas Tutores Inteligentes (STI) (do inglês: *Intelligent Tutoring System (ITS)*). Estes sistemas têm como característica básica a representação de conhecimentos relacionados às questões: o que ensinar, a quem ensinar e como ensinar. Os STIs têm com proposta básica oferecer instrução individualizada aos aprendizes. Mais informações sobre os STIs são apresentadas no Capítulo 2.

1.1.3 O terceiro momento

Inicialmente, constata-se um movimento em favor de modelos computacionais de apoio à aprendizagem calcados na noção de cooperação. Nesse sentido, propõem-se que os STIs tradicionais evoluam para os Ambientes Interativos/Inteligentes de Aprendizagem (do inglês: *Interactive/Intelligent Learning Environment - ILE*) ou ainda Sistemas Tutores Cooperativos. Um tal sistema pode, com efeito, ser entendido como uma combinação de aspectos das categorias STI e Micromundo. De uma maneira genérica, de agora em diante nesta tese, os sistemas computacionais dedicados à aprendizagem que utilizam métodos e técnicas de Inteligência Artificial em Educação, a exemplo dos STIs e ILEs poderão eventualmente ser designados por Sistemas IA-ED. Isso devido à existência de uma área com tal abrangência, já está estabelecida mundialmente, sendo na literatura em língua inglesa denominada *Artificial Intelligence in Education (AI-ED)*.

Mais recentemente, os ambientes educacionais apoiados por computador começam a dar ênfase à tecnologia de computação distribuída, tendo como uma das consequências o emprego de modelos de trabalho cooperativo. Assim, tais ambientes começam a ser dotados de suporte para a tecnologia de *groupware* e de *CSCW (Computer Supported Cooperative Work)* [Bae 93], reunindo pessoas (estudantes e professores) em atividades de aprendizagem cooperativa numa espécie de sala de aula virtual [Her 96].

Neste mesmo período, os STIs e ILEs passaram a incorporar resultados da Inteligência Artificial Distribuída, através de uma abordagem de Sistemas Multi-agentes, aumentando ainda mais as suas capacidades de interação/cooperação².

Finalmente, há também um investimento bastante recente na construção de ambientes que suportem a tecnologia de realidade virtual. Uma iniciativa nesta direção pode ser encontrada em [CMF 97]. Um ambiente dotado de recursos de realidade virtual

² Essa iniciativa em conceber ambientes com a abordagem de agentes é discutida no Capítulo 3 desta tese.

permite ao usuário uma interação bastante intuitiva com a simulação de um sistema sintético em um espaço tridimensional [GV 94].

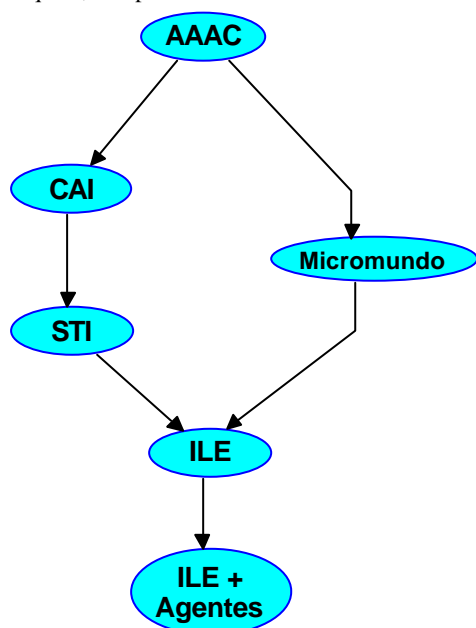
1.1.4 Comentários finais

Esboçou-se sucintamente, nas seções anteriores, uma perspectiva histórico-taxonômica dos ambientes de aprendizagem assistidos por computador. Outras iniciativas similares podem ser encontradas. Por exemplo, em trabalhos, tais como os encontrados em [Tay 80], [HP 88], [Sta 90], há uma tentativa de fazer um tal apanhado como objetivo central do documento veiculado. Já no presente trabalho de tese, bem como em [VG 96] e muitos artigos e dissertações, verifica-se que essa questão é abordada com objetivo mais periférico, ainda que de forma interessante.

A expectativa com a presente discussão é primeiramente localizar a pesquisa desta tese. No mais, como ainda há pouca literatura, notadamente em língua portuguesa, com propósito de disponibilizar um material amplo e atual sobre a área de Informática na Educação, o apanhado aqui descrito pode, de certo modo, contribuir nesse sentido. Evidentemente, trata-se mais de um roteiro, tentando referir-se à área com algo atual e vasto, ao mesmo tempo apresentando uma visão particular da questão.

1.2 Contexto da pesquisa

No geral, o presente trabalho se situa no âmbito da concepção dos ambientes computacionais cujo foco principal está sobre uma relação de ensino e aprendizagem envolvendo por um lado, um humano desempenhando o papel de aprendiz e por outro lado, a máquina, comportando-se como um tutor.



AAAC = Ambiente de Aprendizagem Assistido por Computador

CAI = Computer Assisted Instruction

STI = Sistemas Tutores Inteligentes

ILE = Intelligent / Interactive Learning Environment

Figura 1.1: Taxonomia simplificada para os AAACs.

Nesta perspectiva, o domínio de estudo abordado nesta tese é, conforme foi indicado na seção anterior, o de Sistemas IA-ED, mais especificamente a concepção de ambientes interativos de aprendizagem sob um enfoque multi-agentes. Trata-se de um domínio essencialmente interdisciplinar, envolvendo, no mínimo,

disciplinas, tais como: Inteligência Artificial Distribuída, Educação e Psicologia.

Na Figura 1.1 sintetiza-se a taxonomia ampla, apresentada na seção anterior, localizando mais especificamente o contexto da presente pesquisa. Nesse sentido, esta pesquisa se encontra na categoria que envolve ILE + Agentes. Essa síntese retrata mais diretamente a temática abordada na problemática deste trabalho, refletindo, de algum modo, os esforços empreendidos para se chegar aos ILEs, e daí, à incorporação de um enfoque multi-agentes.

O objetivo global desta investigação é o de elaborar um modelo de ambiente de aprendizagem, servindo para concepção e desenvolvimento de Sistemas IA-ED particulares. Busca-se com isso, a possibilidade do sistema oferecer a um dado estudante, condições favoráveis ao seu aprendizado. Uma vez concluído, um tal modelo de ambiente oferecerá condições para o desenvolvimento de aplicações particulares, levando em conta principalmente a sua arquitetura e os seus princípios. A busca por um tal modelo, entretanto, envolve várias questões não triviais, relacionadas aos elementos que lhe constituem. Estas questões são apresentadas e abordadas na presente tese.

1.3 Problemática e objetivos da pesquisa de tese

O objetivo fundamental para os ambientes de aprendizagem assistidos por computador é, evidentemente, facilitar a aprendizagem dos estudantes em relação a um dado domínio de conhecimento. Subjacente a essa pretensão, encontra-se a definição de alguns elementos básicos, tais como: concepção de aprendizagem e modelo de interação entre as entidades envolvidas no processo. Nesses elementos estão embutidas as visões relacionadas a questões, tais como: o que é aprender e como se aprende.

Perseguindo com o objetivo supracitado, verifica-se na maior parte da pesquisa em IA-ED, como estratégia adotada para promover a aprendizagem dum estudante, a existência de uma questão admitida como central. Essa questão, diz respeito à possibilidade de adaptação de um tutor artificial a um dado estudante humano, em uma situação de aprendizagem. Essa adaptabilidade tem sido assumida como um recurso fundamental para se alcançar um processo de ensino e aprendizagem mais efetivo. Conseguir realizar tal possibilidade através do computador tem sido uma tarefa complexa. Esta tarefa está relacionada ao que será chamado doravante neste texto de **problema da adaptação**. Este problema, entretanto, é amplo demais e pode ser decomposto em vários subproblemas, muitos dos quais, por si só, justificam um segmento de pesquisa na área em discussão. Alguns desses subproblemas são destacados a seguir. Entretanto, a problemática geral desse trabalho é a modelagem das interações sistema tutor versus aprendiz, com vistas ao problema da adaptação.

A pretensão de se conceber um bom ambiente³ de aprendizagem envolve uma preparação especial para o sistema tutor. De uma forma mais geral, nesta pesquisa, o interesse está focalizado na seguinte questão, Q0: O que buscar na concepção de um ambiente para assegurar um suporte apropriado para melhorar a intera-

³ Entenda-se por bom ambiente, neste caso, um local no qual pretende-se que se disponibilizem condições favoráveis para que ocorra a aprendizagem do estudante quando de sua interação com um sistema tutor.

ção entre o sistema tutor e o estudante? Essa melhora concerne ao provimento de condições favoráveis à aprendizagem do estudante. Esse preparo no sistema inclui, por exemplo, um investimento na qualidade do conhecimento deste sistema, assim como nos seus mecanismos para manipular esse conhecimento.

Nesta perspectiva, a elaboração dum Sistema IA-ED rumo à obtenção de uma solução ao problema da adaptação, passa, normalmente e inicialmente, pelo investimento em três categorias de questões gerais assumidas aqui como fundamentais:

Q1: Como obter um modelo adequado para o domínio de conhecimento?

Q2: Como elaborar um modelo do estudante efetivo, isto é, como dispor de recursos adequados para que, em cada momento duma interação Tutor × Estudante, o sistema tenha um bom modelo do estado cognitivo deste estudante?

Q3: Como definir um modelo pedagógico abrangente e de qualidade?

As respostas a essas questões constituem requisitos importantes na definição de um ambiente interativo de aprendizagem. Assume-se aqui, no tratamento adequado das questões acima, a hipótese de que um bom desempenho do sistema tutor depende fortemente da sua capacidade de atender de forma adaptada à demanda do estudante. Nesse sentido, assume-se mais especificamente, a hipótese de que a definição do sistema tutor como uma sociedade de agentes relativamente a um domínio de conhecimento, pode contribuir significativamente para a melhora nas condições de interação e aprendizagem do aprendiz. O emprego de uma tal sociedade, tal como será visto melhor nos Capítulos 4 e 5, está comprometido principalmente com questões de engenharia do conhecimento.

Neste sentido, numa tentativa de responder adequadamente a tais questões, particularmente, Q1 e Q2, foram estabelecidos primeiramente os objetivos seguintes:

Ob1: Propor um modelo para o domínio de conhecimento levando em conta um compromisso entre aspectos de riqueza e estruturação relativo a um tal domínio, como elementos de requisito para assegurar sua qualidade.

Ob2: Com a realização da proposição feita para o modelo do domínio, espera-se contribuir para tornar o sistema tutor mais perceptivo. Nesse sentido, pretende-se investir na modelagem do aprendiz, propondo um módulo de diagnóstico de qualidade, tirando um bom proveito do modelo do domínio nos termos propostos.

Esses são os objetivos basilares e iniciais desta tese. Paralelamente⁴ a esses dois objetivos, há também um outro que é naturalmente mais amplo, mas ao mesmo tempo, todos podem ser vistos, de algum modo, como interdependentes. Trata-se assim de mais um objetivo, já levemente mencionado acima, a saber:

Ob3: Definir um modelo de ambiente interativo de aprendizagem, considerando um enfoque multi-agentes e a noção de aprendizagem cooperativa.

O investimento nos três objetivos acima declarados, gerou desdobramentos que conduziram à inclusão de novos objetivos a

serem perseguidos nesta tese. Tal investimento ensejou, após um certo estudo⁵, a adoção de uma abordagem multi-agentes⁶ na definição do sistema tutor. Tal adoção ocorreu motivada principalmente pelos resultados obtidos para alcançar o objetivo Ob1 e seus reflexos, importância e relacionamento com respeito a Ob2. Em decorrência da adoção de uma tal abordagem, tornou-se contingente o surgimento de uma série de problemas que lhe estão relacionados. As tentativas de soluções para esses problemas passaram a compor os novos objetivos desta tese. Entre esses novos objetivos pautados na definição do sistema tutor multi-agentes, incluem-se os seguintes:

Ob4: definição de modelos de cooperação e comunicação entre os agentes tutores,

Ob5: definição da arquitetura do agente tutor,

Ob6: definição de linguagem e protocolos necessários para viabilizar a interação entre os agentes tutores envolvidos.

Esses objetivos podem, a partir de um certo momento, ser integrados e vistos sob um outro ângulo, sendo encarados como parte da busca pela concepção e realização do modelo de ambiente interativo de aprendizagem, relativo ao Ob3. Esse ambiente emergiu a partir dos investimentos nas questões supracitadas e passou a se chamar **MATHEMA**. Seus primeiros resultados podem ser encontrados em [CLF 95].

Relativamente ao **MATHEMA**, fazem parte também desta tese mais três objetivos, que são:

Ob7: Elaboração de um modelo que descreva as interações didáticas entre o aprendiz humano e a sociedade de agentes tutores.

Ob8: Definição da arquitetura interna da componente tutorial no agente: o sistema tutor, modelando as interações envolvendo os seus elementos. É aqui onde se vai localizar, por exemplo, o modelo do domínio, o modelo do aprendiz e o modelo pedagógico pertinentes a cada agente tutor.

Ob9: Desenvolvimento de um protótipo em um certo domínio, como uma forma de validação dos resultados obtidos no âmbito do ambiente MATHEMA. Além disso, um tal protótipo servirá também para subsidiar uma revisão nos conceitos envolvidos na concepção do ambiente.

1.4 Organização da tese

Esta tese está estruturada em capítulos, sendo esse o primeiro capítulo e os demais estando organizados como segue abaixo. Esses capítulos, a partir do segundo, podem ser vistos como agregados em quatro partes. Na Parte 1 estão contidos os Capítulos 2 e 3, servindo, respectivamente, como os conhecimentos de apoio à leitura da tese e uma orientação ao estado da arte no tema em apreço. A Parte 2 é formada pelos Capítulos 4, 5, 6, 7 e 8, representando a essência do presente trabalho, prestando-se assim, para mostrar como foram alcançados os objetivos pretendidos para esta tese. Já a Parte 3 é constituída pelos Capítulos 8 e 9, servindo para apresentar os aspectos relativos à especificação, implementação e aplicações.

⁵ Este estudo considerou questões sobre ganhos ao adotar-se uma abordagem de agentes, principalmente quanto à possibilidade de melhoria na interação Sistema Tutor - Aprendiz. Veja detalhes no Capítulo 4.

⁶ Este trajeto saindo de Q1 e chegando a um Sistema Tutor baseado em múltiplos agentes está descrito no Capítulo 4 desta tese.

⁴ A sequencialidade no estabelecimento dos objetivos e de suas execuções nem sempre está presente. O que ocorre algumas vezes, é que algumas atividades ocorrem paralelamente e de forma interdependente, sendo muitas vezes colaborativas.

ção, implementação e aplicações. Finalmente, a Parte 4 é constituída pelo Capítulo 10, onde são apresentadas as conclusões, contribuições principais da tese e as propostas para trabalhos futuros. Visando ser mais específico, descrevem-se, a seguir, os propósitos de cada um desses capítulos.

No Capítulo 2, procura-se apresentar sucintamente um conhecimento de suporte aos principais assuntos da tese. Assim, faz-se uma revisão das noções fundamentais em Inteligência Artificial Distribuída comparando-a à Inteligência Artificial clássica, enfatizando os Sistemas Multi-agentes e em Sistemas Tutores Inteligentes, considerando basicamente o necessário para o entendimento do conhecimento envolvido no presente trabalho de pesquisa. Posteriormente, abordam-se alguns aspectos educacionais e, conclui-se com uma descrição sobre os já mencionados Sistemas IA-ED.

No Capítulo 3, apresenta-se uma discussão sobre a combinação entre Tutores Inteligentes e a noção de Agentes. O objetivo é discutir algumas das idéias e soluções que estão sendo trabalhadas nas pesquisas atuais na concepção de alguns Sistemas IA-ED com enfoque baseado em sistemas multi-agentes. Além disso, mencionam-se algumas das principais abordagens que assumem tal combinação; e por fim, introduz-se o ambiente MATHEMA.

No Capítulo 4, descreve-se a concepção do ambiente MATHEMA. A discussão inicia-se pelo esboço de um modelo mínimo de aprendizagem, passando por um ambiente dotado de sistema tutor multi-agentes e, finalmente chegando ao MATHEMA. Daí então, enfocam-se os seus princípios e sua arquitetura. Ademais, definem-se os componentes da arquitetura e caracterizam-se os relacionamentos entre esses componentes. Finalmente, é fornecida uma visão sobre o funcionamento geral do ambiente e são indicadas as interações entre os componentes principais a serem considerados nesta tese.

No Capítulo 5, define-se a Sociedade de Agentes Tutores Artificiais que compõe o ambiente MATHEMA. Inicialmente, são apresentadas as características gerais dessa sociedade e, em seguida, enfocam-se os aspectos de cooperação e comunicação entre os agentes. Finalmente, define-se a arquitetura desses agentes tutores, discutindo a funcionalidade de cada um de seus componentes.

No Capítulo 6, discute-se o comportamento dinâmico de um agente durante a sua participação em atividades cooperativas. Com isso, pretende-se apresentar a funcionalidade interior de um agente no momento em que ele necessita se envolver numa interação.

No Capítulo 7, descreve-se o suporte lingüístico necessário para compor a definição da linguagem de interação entre agentes tutores. Conclui-se com uma descrição e comentários sobre os protocolos definidos e adotados.

No Capítulo 8 concentra-se atenção sobre um elemento particular da arquitetura de um agente, que é o sistema tutor. Desse modo, são discutidas principalmente as características do modelo de interação entre esse sistema e um aprendiz. Daí, são enfocados aspectos sobre os elementos de base do sistema, quais sejam: o modelo do domínio, o modelo do aprendiz e o modelo pedagógico.

No Capítulo 9 apresentam-se os aspectos de implementação e aplicações relacionadas ao ambiente proposto.

No Capítulo 10, apresentam-se as conclusões finais, refletindo

sobre os aspectos ligados às contribuições desta tese. Prosseguem-se com as indicações sobre algumas direções sugeridas para serem seguidas na continuação deste trabalho.

2. Noções preliminares

Neste capítulo, pretende-se fornecer ao leitor uma visão panorâmica relativa ao conhecimento de suporte no entendimento dos capítulos seguintes. Assim, inicia-se descrevendo-se algumas noções em Inteligência Artificial Distribuída (IAD), enfatizando mais uma abordagem de Sistemas Multi-Agentes (SMA). Prosseguindo, comentam-se alguns aspectos em Educação presentes na área que envolve os mencionados Sistemas IA-ED e, em seguida, discutem-se suas características.

A idéia aqui pretendida é que as noções selecionadas e discutidas neste capítulo, auxiliem, de algum modo, a leitura dos capítulos seguintes. Entretanto, não se tem a intenção de que tais preliminares sejam um texto para leitura independente dos objetivos expostos nesta tese. Portanto, aos leitores interessados em maiores detalhes nos temas envolvidos, são recomendadas sugestões de leitura no decorrer de cada uma das seções seguintes.

2.1 Da IA clássica para a IA distribuída

A IAD é uma disciplina que objetiva desenvolver métodos e técnicas relacionadas à solução de problemas complexos, trabalhados sob uma metáfora de inteligência coletiva. A complexidade destes problemas está vinculada à possibilidade de suas soluções envolverem conhecimento de mais de uma especialidade. Para lidar com tal complexidade, considera-se na tarefa de resolução de problemas, o exercício de um comportamento inteligente e cooperativo por parte de suas entidades, dispostas em um sistema distribuído. A inteligência considerada pode residir em unidades separadas chamadas de Agentes ou emergir de suas interconexões.

No artigo de autoria de Sichman, Demazeau & Boissier [SDB 92], foram destacadas as metáforas associadas às visões sobre a noção de inteligência, tanto na IA clássica quanto na IA Distribuída. Por um lado, a IA clássica se baseia no *comportamento humano individual*, cuja ênfase é colocada em representação do conhecimento e de métodos de inferência. Por outro lado, a IAD é baseada no *comportamento social*, e a ênfase é colocada nas ações e interações entre agentes.

Algumas das motivações para investimentos em técnicas de IAD estão vinculadas às expectativas em ganhos relacionados a, pelo menos, dois aspectos: por um lado, do ponto de vista da Engenharia de *Software*, incluem-se fatores, tais como: modularidade, manutenção, desenvolvimento, paralelismo, robustez [SDB 92], e por outro lado, pelo potencial que pode oferecer às implementações da idéia de adaptação nos sistemas adaptativos em geral.

Os enfoques de pesquisa em sistemas adotados na área de IAD podem ser vistos, de um modo geral, segundo um dos dois seguintes eixos: Resolução Distribuída de Problemas (do inglês: *Distributed Problem Solving - DPS*) e SMA [BG 88, SDB 92, Oli 95]. Em um enfoque de *DPS*, já se supõe *a priori* a existência de um dado problema, para então conceber os agentes dotados de capacidades específicas e comprometidas com a resolução deste problema. Com respeito a um enfoque de SMA, não há necessariamente a existência de um problema definido *a priori*; sendo assim os agentes podem já existir numa dada sociedade,

independentemente de algum problema específico. Com efeito, um *DPS* pode ser visto como um caso particular de SMA [AS 97].

É num enfoque de SMA que se baseia a presente pesquisa. Assim sendo, a discussão seguinte recai apenas sobre esse enfoque.

Até agora, foi feita referência apenas à área de IAD, sem, no entanto, dizer quase nada à respeito do termo agente. Trata-se de um termo largamente empregado, ao mesmo tempo em que o acesso à literatura no tema revela a ausência de uma definição consensual e única a seu respeito. Ao invés disso, observa-se na mesma, uma ampla variedade de definições, formuladas segundo um comprometimento com algum contexto específico. Além disso, o termo agente também tem sido empregado em outras áreas, como por exemplo: Sistemas Distribuídos e Orientação a Objetos.

Franklin & Graesser [FG 96] organizaram uma coletânea de definições sobre agentes, mostrando como cada autor, entre os relacionados em tal referência, se pronuncia com respeito à questão. Com base nessa coletânea, pode-se perceber a presença de uma série de propriedades que servem para mostrar os pontos comuns entre as visões dos autores, ao mesmo tempo em se observa os pontos que servem para distinguir tais visões.

Algumas destas definições são destacadas a seguir:

Um agente é qualquer coisa que pode ser vista como percebendo seu ambiente através de sensores e agindo sobre este ambiente através de efetadores.

Agentes Inteligentes realizam constantemente três funções: percepção das condições dinâmicas no ambiente; ações para afetar as condições no ambiente; e raciocínio para interpretar percepções, resolver problemas, efetivar inferências, e determinar ações.

Agentes Inteligentes são entidades de software que executam algum conjunto de operações em favor de um usuário ou um outro programa com algum grau de independência ou autonomia, e em assim fazendo, empregar alguma conhecimento ou representação dos objetivos ou desejos dos usuários.

Apesar das definições acima, uma das definições muito bem aceita pela comunidade de SMA e que se mostra abrangente e adequada aos propósitos da presente tese, é a proposta por [FG 91]:

Um agente é definido como sendo uma entidade (real ou abstrata) capaz de agir sobre ela mesma e sobre seu ambiente, dispondo de uma representação parcial deste ambiente, podendo comunicar-se com outros agentes, e cujo comportamento é consequência de suas observações, de seu conhecimento e das interações com outros agentes.

Existem essencialmente duas abordagens diferentes na concepção de um SMA, consideradas de acordo com o critério da granularidade dos agentes, quais sejam: agentes reativos (*granularidade fina*, muito simples, não inteligentes) e agentes cognitivos/intencionais (*granularidade grossa*, inteligentes) [SDB 92]. Nesse ponto, entretanto, convém ressaltar uma tendência bastante atual em investir-se sobre abordagens híbridas, em relação às duas anteriores, tal como se pode constatar em, dentre outros trabalhos, [SF 97].

Numa organização de agentes reativos [Bro 91], o desempenho de um comportamento inteligente não é devido a um agente isoladamente, mas resulta da emergência de um comportamento global inteligente a partir dos comportamentos individuais reativos. Nesse sentido, trabalha-se um dinamismo vinculado à idéia de um comportamento emergente, onde a estrutura é criada dinamicamente em resposta a estímulos e perturbações vindas do ambiente. Trata-se de um modelo de organização baseado em metáforas biológicas, tal como ocorre, por exemplo, numa colônia de formigas. Neste modelo, a ativação do sistema para desempenhar um comportamento inteligente resulta da emergência dos comportamentos individuais. Nesse modelo, não há num agente a representação explícita do ambiente, nem representação dos outros membros que compõem a organização, e também, não há comunicação via envio e recebimento de mensagens entre os agentes.

Já na organização de agentes cognitivos (intencionais), os agentes possuem representações de si mesmo, dos demais agentes e do ambiente. Aqui, existe um conjunto de agentes intencionais, onde a ativação do sistema resulta da comunicação direta através de envio e recebimento de mensagens. Estes modelos de agentes cognitivos são baseados em algum modelo de organização social, no sentido de sociedades humanas (grupos, departamentos, etc.) [FG 91]. Alguns comportamentos sociais que se encontram nesses modelos, incluem: cooperação, colaboração, competição, co-habitação, distribuição, etc.

Interessa, por enquanto, ao presente trabalho, apenas a abordagem de agentes cognitivos. Por esse motivo, apenas ela será considerada nas discussões seguintes. Nesse sentido são apresentados os elementos básicos para caracterizar um Sistema Multi-agentes cognitivo.

A idéia de agentes numa sociedade articularem-se para resolver um dado problema, conduz a duas noções importantes em SMA, que são: a de organização e a de controle. Em [IS 97], define-se organização como sendo o modo pelo qual um grupo de agentes se organiza para resolver um problema alvo. A idéia é que esse problema seja decomposto em subproblemas e, assim esses subproblemas sejam repartidos entre os agentes envolvidos. Segundo esses autores, [Sic 95] categoriza a forma de concepção de uma organização de agentes em estática e dinâmica.

Ainda de acordo com esses autores, numa organização estática, os agentes já possuem um problema global a ser solucionado. Isso significa que a partir de um tal problema, organiza-se a sociedade de agentes de tal forma que estes encontrem a solução desejada. Numa organização dinâmica, os agentes não possuem *a priori* um único problema a ser solucionado: tais agentes pré-existem a eventuais problemas que possam ser solucionados pelo sistema. Com isso, surge a noção de formação de coalizões dinâmicas entre agentes, com vistas a solucionar um dado problema.

São os modelos de organização dinâmica que interessam ao presente trabalho. Sobre esses modelos, podem-se distinguir duas categorias [IS 97]: a Rede Contractual e as Coalizões Baseadas em Dependência. No modelo de Rede Contractual, um agente não possui informações à respeito das capacidades dos outros agentes. Desse modo, quando não puder solucionar um problema por si só, realiza uma busca não informada por agentes que lhe possam ajudar. No caso do modelo de Coalizões Baseadas em Dependência, um agente quando necessita de cooperação realiza uma busca informada de agentes que potencialmente possam lhe

ajudar.

Quanto ao controle, normalmente há duas soluções básicas, a saber: o controle centralizado e o controle distribuído. No primeiro caso, há normalmente um agente controlador encarregado de coordenar todas as atividades cooperativas envolvendo um grupo de agentes. No caso do controle distribuído, cada agente possui um controle local.

Finalmente, há mais duas noções importantes na concepção de um Sistema Multi-agentes. A primeira, aqui denominada de cooperação, tem a ver com os mecanismos que viabilizam as atividades cooperativas entre agentes. A segunda, denominada de comunicação, diz respeito aos mecanismos responsáveis pela comunicação entre os agentes, sendo normalmente baseados num modelo de troca de mensagens ou num modelo de compartilhamento de informações (a exemplo de uma arquitetura *blackboard*). Ademais, há também a questão da arquitetura de agentes, assim como linguagem e protocolos que viabilizem as interações entre os agentes, a exemplo da bem estabelecida na comunidade - o padrão **KQML**: *Knowledge Query and Manipulation Language* - que é uma linguagem que faz uso de performativos para expressar suas crenças, necessidades, e modalidades de comunicação [AS 97], tendo sido definida independente de algum domínio particular.

Quanto a arquitetura, há várias propostas com enfoques bem diferenciados, podendo o leitor acessá-las, por exemplo, num abrangente apanhado realizado em [WJ 95]. Entretanto, para os propósitos do presente trabalho é importante destacar que a arquitetura de agentes definida teve sua inspiração em duas famosas arquiteturas: uma que é a definida no âmbito do projeto ARCHON [JW 92] e ASIC [Sic 95]. Tratam-se de duas arquiteturas que consideram em seu modelo de agente, a noção de conhecimento social relativamente a cada agente.

2.2 Aspectos educacionais

Como foi mencionado no capítulo anterior, o domínio de estudo da tese, ora apresentada, é o da concepção em sistemas IA-ED. Descreveu-se acima, um pouco sobre Inteligência Artificial, sendo mais específico na denominada IA Distribuída. Agora, entretanto, é a vez de apresentarem-se algumas colocações relacionadas ao termo Educação.

No contexto da pesquisa em IA-ED, considera-se apenas a questão da aprendizagem como uma atividade fundamental em Educação. Acredita-se que o propósito geral da educação seja promover aprendizagem. De fato, essa tem sido a tônica dos ambientes de aprendizagem assistidos por computador, tal como mencionados no capítulo anterior. A concepção de um tal ambiente envolve, de algum modo, uma visão sobre ensino-aprendizagem. Essa visão tem um compromisso com o enfoque dado à relação estudante-computador no ato de aprender.

Há, por um lado, os ambientes que colocam o ensino como elemento fundamental para alcançar a aprendizagem. Por exemplo, os sistemas *CAIs* são representantes fiéis desse ponto de vista. Por outro lado, os Micromundos, pelo menos nos termos definidos por Papert, descartam a atividade de ensino. As demais propostas de ambientes educativos podem ser enquadradas numa proposta intermediária ou coincidente com esses dois extremos.

Os modelos com propostas para lidar com a questão a aprendizagem são muitos. Eles normalmente fazem parte de abordagens

com diferentes denominações, tais como: *behavioristas*, *cognitivistas* e *situacionistas (contextualistas)*. A primeira diz respeito às idéias inspiradas nos trabalhos de Skinner [Ski 57], onde sua visão sobre conhecimento e aprendizagem baseia-se num modelo de caixa-preta: estímulo-resposta. Ela tem sido cada vez mais deixada de lado à medida que as abordagens cognitivistas ganham mais espaço, principalmente entre as pesquisas em Sistemas IA-ED, ou mais geralmente em Inteligência Artificial. Situa-se aí a linha construtivista [Pia 77], que considera o conhecimento como resultado de um processo de construção pessoal que envolve conflito epistêmico, auto-reflexão e auto-regulação. Finalmente, há a linha dos partidários da cognição situada [LW 91], [Cla 93] partilhando do princípio que a aprendizagem é sempre localizada num contexto e é permanentemente evolutiva. Eles consideram a aprendizagem como algo que ocorre em comunidades de prática. A aprendizagem é, assim, conceitualizada como um processo de iniciação, de integração e de colaboração no interior de diferentes comunidades de prática.

A polêmica associada a qual abordagem é melhor do que outra não faz parte do escopo desse tese. A preocupação aqui é apenas com a natureza e o processo de aprendizagem.

2.3 Inteligência Artificial em Educação

2.3.1 Preliminares

O marco inicial dessa área em discussão nesta tese data da década de 70, no momento em que se resolveu aliar métodos e técnicas da Inteligência Artificial na concepção de ambientes de ensino e aprendizagem assistidos por computador. Com isso, surgiu a área denominada de *ICAI* (Intelligent CAI), posteriormente cunhada de Sistemas Tutores Inteligentes. Essa área e seus desdobramentos é aqui chamada, conforme foi mencionado no Capítulo 1, de Inteligência Artificial em Educação, com a sigla IA-ED.

2.3.2 Sistemas IA-ED: dos STIs clássicos para os ILEs

A pesquisa em sistemas IA-ED iniciou-se na década de 70, podendo os trabalhos veiculados em [Car 70] e [Sel 74], serem considerados como estando entre os pioneiros. Um STI, no sentido mais clássico, é um sistema computacional que, essencialmente, visa reproduzir o comportamento atribuído a um professor humano, supostamente competente em um domínio de ensino particular. Essa competência, normalmente, está vinculada, por exemplo, a sua capacidade de prover instrução individualizada a um certo estudante, através da adaptação dinâmica da forma e conteúdo de apresentação do material instrucional ao mesmo. Para Sleeman e Brown [SB 82], um STI é um programa de computador que utiliza técnicas de IA para representar o conhecimento e para conduzir uma interação adaptativa com o estudante.

Para fazer isso, um STI inclui em sua estrutura básica, conhecimentos especializados para tratar questões, tais como: o que ensinar? a quem ensinar? como ensinar? Essas questões têm sido freqüentemente mapeadas em três módulos, tal como é mostrado mais adiante nesta seção.

Para lidar com as questões envolvidas em tais módulos, os STIs clássicos se utilizam de domínios de estudos, que incluem a Inteligência Artificial (IA), a Psicologia Cognitiva e a Pedagogia. A Figura 2.1, adaptada de [Kea 87], ilustra e posiciona os STIs em relação a esses domínios. Eles incorporam métodos e técnicas de IA permitindo a representação e manipulação dos conhecimentos

mencionados acima. A Psicologia Cognitiva serve como suporte para buscar entendimento sobre indivíduos. A Pedagogia oferece os métodos de ensino.

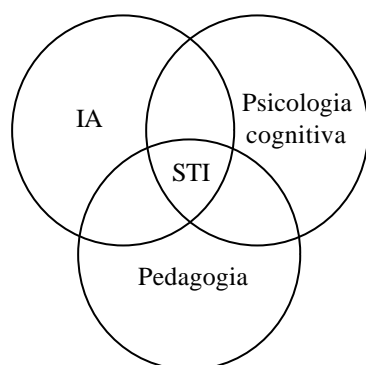


Figura 2.1: Posicionamento dos STIs

Um dos STIs pioneiros foi o sistema SCHOLAR [Car 70] proposto por Carbonell. O seu objetivo geral é o ensino de geografia. Para isso, ele conta com uma Rede Semântica como sua linguagem de representação de conhecimento do domínio em apreço. A comunicação entre esse sistema e o aprendiz se faz sob a forma de intervenções em meio a um diálogo em linguagem natural.

No mais, convém destacar que existem muitos outros sistemas, entretanto, ao invés de continuar citando e comentando sistemas já desenvolvidos, optou-se por indicar ao leitor referências clássicas, tais como: [SB 82], [Wen 87] e [GS 88], as quais cumprem muito bem esse papel de, entre outras coisas, divulgar vários trabalhos realizados no assunto. Notadamente, Wenger [Wen 87], apresenta um apanhado bastante representativo do que se pode atribuir ao estado da arte em STI, até pelo menos a década de 80, dando conta de uma visão panorâmica de qualidade relativa às idéias, princípios e aplicações na referida área.

2.3.2.1 Arquitetura geral de um STI

Apesar de não haver uma concordância total entre as arquiteturas em STIs clássicos (baseada em modelos), observa-se na literatura sobre a área, a existência de um consenso significativo com respeito à adoção dos três módulos citados anteriormente: módulo do modelo do domínio, módulo do modelo do aprendiz e módulo do modelo pedagógico. Além desses, há mais um módulo que se encarrega da *interface* de comunicação entre o sistema tutor e o aprendiz [Wen 87]. Esses módulos são descritos sucintamente a seguir.

O módulo do modelo do domínio

Este módulo, como já mencionado, é responsável pela questão *o que ensinar*, contendo, por exemplo, o conhecimento especializado sobre o domínio de aplicação. Nele, representa-se o conhecimento sobre um domínio particular de ensino, a ser trabalhado pelo sistema tutor. Neste módulo, os conhecimentos são geralmente representados por algum dos métodos clássicos, escolhido de acordo com a natureza do domínio: declarativo ou procedural. Pode-se observar na literatura em STI a existência de outras denominações equivalentes para esse módulo, por exemplo: módulo do modelo do especialista ou perito, devido ao conhecimento especializado nele contido.

Este módulo serve de base para a operacionalização das funções pedagógicas, a exemplo de resolução de problemas, diagnóstico cognitivo e instrução.

O Módulo do modelo do aprendiz

O módulo do modelo do aprendiz, encarregado da questão *a quem ensinar*, é uma representação abstrata que o sistema constrói sobre o conhecimento e desempenho do estudante, durante sua interação dinâmica com o mesmo. Pretende-se determinar a cada momento, o estado cognitivo do estudante (*o que ele sabe, o que não sabe, o que entendeu mal*). Esse conhecimento servirá para apoiar as decisões do sistema na escolha de estratégias de ensino adequadas ao estudante numa dada situação. Além disso, serve também para ajudar no diagnóstico e conseqüentes remediações em relação ao desempenho do estudante. O modelo do estudante é de importância central para um STI. Dependendo da sofisticação que se queira para esse modelo, problemas complicados podem surgir. Tanto que a construção de tais modelos tem justificado pesquisas e mais pesquisas.

Este módulo é fundamental na tarefa de adaptação do sistema ao aprendiz. Os modelos de representação do aprendiz são, na maior parte, baseados no modelo do domínio. Nesse sentido, existem dois modelos bastante conhecidos: o modelo de Sobreposição (*overlay*) e o modelo de Perturbação. No primeiro, considera-se o conhecimento do aprendiz como um subconjunto do conhecimento do especialista. No segundo, trabalha-se sobre essa perspectiva de ser baseada no domínio, mas considera um conhecimento relacionado com a noção de erros e mal-entendidos por parte do aprendiz. Maiores detalhes sobre esses modelos podem ser encontrados em, por exemplo, [Wen 87], [VO 92], [VG 96].

Módulo do modelo pedagógico

O módulo do modelo pedagógico, também chamado de *módulo do modelo de tutor ou instrutor*, está relacionado com as estratégias de ensino-aprendizagem a serem adotadas sobre o domínio de aplicação, sendo o responsável pela questão *como ensinar*. Uma estratégia de ensino-aprendizagem envolve seleção e planejamento das atividades a serem apresentadas ao aprendiz. Um plano corresponde a uma seqüência de ações que visam atingir determinados objetivos. As estratégias de ensino-aprendizagem lidam com esquemas de planos, definindo formas de aprender o material instrucional [VO 92].

Em suma, é neste módulo é onde se situa a funcionalidade básica relativa à decisão sobre qual atividade pedagógica apresentar e como apresentá-la, a partir de informações provenientes do modelo do aprendiz e do modelo do domínio.

2.3.3 A evolução para os ILEs

Ultimamente o foco de atenção nas pesquisas em STI tem se voltado mais para um estilo de interação mais flexível. A idéia é considerar o aprendiz como um elemento ativo e com uma certa autonomia no processo de aprendizagem. Essa visão tem conduzido a novas abordagens na concepção de ambientes computacionais de ensino e aprendizagem. Nesse sentido, os STIs tradicionais estão se transformando para atender a essas novas percepções. Uma dessas transformações resultaram nos AIAs (ILEs) ou STI Cooperativos.

Um ILE pode ser caracterizado como uma categoria de *software* educacional na qual o aprendiz é colocado em uma situação de descoberta, envolvendo normalmente atividades de resolução de problemas. Durante esse processo, o tutor artificial assiste o aprendiz em sua tarefa e monitora o seu aprendizado. O foco de atenção na resolução de problema por parte do aprendiz passa a considerar mais o processo de solução, verificando passo-a-

passo o seu conteúdo.

Portanto, a diferença básica entre um STI tradicional e um *ILE* está no estilo de interação entre o sistema e o aprendiz [OVC 94a, CP 96]. Um STI tradicional é baseado num estilo rígido de interação, significando que o sistema detém sempre o controle da interação. Num *ILE*, entretanto, considera-se um estilo flexível, onde a iniciativa da interação é normalmente compartilhada entre o sistema e o aprendiz.

3. Tutores inteligentes via agentes

Neste capítulo, apresenta-se uma discussão geral sobre determinados trabalhos cuja iniciativa recai na combinação entre Tutores Inteligentes e a noção de Agentes. O seu objetivo é, após as considerações preliminares, discutir idéias e soluções que estão sendo empregadas nas pesquisas atuais na concepção de alguns Sistemas IA-ED com enfoque baseado em agentes. Apresentam-se propostas com um tal enfoque, sendo algumas delas com objetivos específicos bem diferente do estabelecido para esta tese e outras com abordagens e objetivos específicos mais próximos dela. Por fim, introduz-se brevemente o ambiente MATHEMA, analisando-o numa perspectiva de compará-lo com as abordagens apresentadas.

3.1 Considerações preliminares

Boa parte da pesquisa em Sistemas IA-ED⁷, desde os Sistemas Tutores Inteligentes tradicionais aos atuais Ambientes Interativos de Aprendizagem, está relacionada com a busca por soluções cada vez mais sofisticadas e efetivas ao **problema da adaptação**. No atual estado da arte nas investigações na área, esse problema tem sido atacado à luz de novos entendimentos, principalmente com respeito à postura e forma de encarar as três questões clássicas, já mencionadas, que permeiam um tal sistema: *o que ensinar?*, *a quem ensinar?* e *como ensinar?*. A primeira questão é concernente à modelagem do conhecimento do domínio de aplicação, a segunda é atinente à modelagem do estado cognitivo do estudante e a terceira diz respeito à modelagem de estratégias pedagógicas. Essas questões (ou pelo menos algumas delas) continuam, de um modo ou de outro, sendo objeto de pesquisa na concepção de boa parte dos sistemas em discussão. Nesse sentido, muitos dos trabalhos atuais têm dado atenção especial a elas, só que agora colocando-as diante de um novo enfoque, mediante a adoção da noção de agentes.

O estado da arte concernente ao tratamento atual das três questões citadas acima, numa perspectiva de concepção de sistemas IA-ED, aponta principalmente para as direções comentadas a seguir.

Com respeito à modelagem do domínio, as propostas seguem mais na direção de tratar múltiplas representações de um mesmo domínio de conhecimento e de permitir uma evolução dinâmica no conhecimento do domínio.

Em relação ao modelo do estudante, começa-se a investir fortemente em mecanismos para diagnóstico distribuído, notadamente em situações de resolução de problemas. Uma outra tendência considera a possibilidade do estudante poder inspecionar o modelo que o sistema elaborou a seu respeito e, eventualmente,

interferir em etapas diferentes do processo de modelagem.

Quanto ao modelo pedagógico, há uma tendência em adotarem-se diferentes enfoques pedagógicos (construtivista, comportamentalista, etc.) em um mesmo ambiente educativo. Uma outra, menos ambiciosa, visa a integração de múltiplas funções pedagógicas.

A seguir são apresentados alguns dos principais trabalhos em Tutores Inteligentes, envolvendo as considerações sobre o estado da arte, além de algumas iniciativas diferentes no emprego de uma abordagem de agentes. A expectativa é que a amostra dos trabalhos escolhidos seja significativa, até mesmo porque ainda é pouco expressivo o investimento feito em uma tal direção. O critério de escolha dos trabalhos discutidos abaixo está relacionado à pertinência ao enfoque citado e a proximidade da abordagem utilizada no presente trabalho de tese, que começa a ser melhor delineado a partir do próximo capítulo. Esta combinação entre STI/*ILE* e a noção de agentes representa uma linha de investigação bastante promissora e ainda muito incipiente. Os trabalhos nessa linha, portanto, ainda se encontram nos seus primórdios, observando-se o que já foi feito e o muito que ainda resta a fazer.

3.2 Principais abordagens

Apresentam-se a seguir alguns dos principais trabalhos no tema em discussão, sem, no entanto, ter a pretensão de ser exaustivo, mas destacar aqueles com mais afinidade com as questões discutidas e, assim, com o presente trabalho.

Em [Kea 93], Kearsley apresenta uma reflexão acerca da adoção de agentes inteligentes na concepção de sistemas instrucionais. Ele procura examinar o relacionamento entre agentes inteligentes e sistemas tutores inteligentes, observando inicialmente que a noção de agente já determina um novo paradigma em Inteligência Artificial. Daí, prossegue chamando atenção para que se comece a pensar nos agentes inteligentes, procurando saber o que eles representam em termos de sua adoção na concepção dos STIs: se apenas uma extensão ou um novo paradigma. Esse autor finaliza o seu artigo opinando favoravelmente à abordagem de tutores inteligentes através de agentes, acreditando tratar-se de um novo paradigma para instrução, tomando por base o conceito de habilidades compartilhadas e aprendizagem cooperativa envolvendo humanos e computadores.

Identificam-se na literatura diferentes abordagens em relação ao emprego da noção de agentes em Sistemas Tutores Inteligentes. A fim de agrupá-las, buscando facilitar a discussão sobre elas, tais abordagens foram distribuídas em três categorias⁸, a saber: Canônicas, Mentalista e Especializadas.

3.2.1 Abordagens canônicas

Boa parte das abordagens que utilizam noção de agentes consiste em organizar a arquitetura dum STI em agentes, de tal maneira que cada agente possa ficar encarregado de dar suporte a uma das especialidades presentes em boa parte das arquiteturas clássicas de tais sistemas. Por exemplo, um agente especializado no domínio de aplicação, um agente encarregado de capturar o estado cognitivo do estudante, um agente pedagógico e, assim por diante. Portanto, as abordagens aqui denominadas de canônicas, refletem, de algum modo, este enfoque de agentes.

O trabalho veiculado em [Fut 89] figura como um dos precursor-

⁷ Neste texto, são usadas indiferentemente, sem prejuízo de entendimento, as siglas STI ou *ILE*, significando Sistemas IA-ED.

⁸ Essa categorização não tem a pretensão de ser precisa, mas apenas de facilitar a discussão.

res em termos de utilização de abordagem multi-agente na concepção de STIs. Nele, o autor apresentou a definição do que ele denominou Tutor Inteligente Distribuído. Neste, a arquitetura está relacionada a um sistema com múltiplos agentes, sendo um deles um agente controlador, denominado planejador pedagógico, e os demais são agentes encarregados dos outros módulos comumente encontrados num STI clássico, tais como: resolução de problemas, explicação, diagnóstico, etc. O agente planejador pedagógico é o responsável pelas sessões de aprendizagem e pela supervisão dos trabalhos de cada um dos outros agentes.

Em [GGL 92], define-se um STI baseado numa arquitetura multi-agente. Os agentes que compõem o sistema são divididos em duas categorias: humano e programa. Na primeira categoria estão o aluno e o professor. Na segunda, estão três agentes, quais sejam: um planejador, um tutor e um micromundo. Cada um desses agentes programa possui conhecimento que diz respeito ao domínio de ensino, ao aluno em questão e aos aspectos pedagógicos. Nesse artigo, apresenta-se a arquitetura multi-agente, discutem-se as interações entre os agentes e exibe-se um esquema de comunicação entre os agentes programa.

Num enfoque similar, em [Ted 97] propõe-se uma arquitetura baseada em agentes para um sistema tutor inteligente. Trata-se do SEI - Sistema de Ensino Inteligente. Os agentes em SEI constituem uma sociedade composta por cinco agentes: Agente Domínio, Agente Tutor, Agente Estudante, Agente Controlador e Agente Comunicador (*Interface*). Além dos agentes, o SEI conta com duas bases de conhecimento, a saber: o modelo do estudante e o modelo do domínio. Desenvolveu-se com esse sistema, um protótipo implementado na linguagem JAVA e voltado para o ensino de Introdução à Computação. Pode-se perceber, portanto, que esse sistema trabalha sobre as mesmas funcionalidades requeridas nas arquiteturas clássicas de sistema tutores inteligentes, só que agora sob um enfoque de agentes. Essa mudança de enfoque visa basicamente oferecer ganhos em relação a certos aspectos em Engenharia de Software, notadamente: extensibilidade, modularidade e portabilidade.

3.2.2 Abordagem mentalista

Uma direção particular na abordagem de sistemas IA-ED tem sido perseguida num projeto de pesquisa que envolve uma arquitetura de agente constando, em sua composição, a noção de estado mental. O objetivo é, dada uma situação de diálogo, simular o funcionamento das mudanças que ocorrem nos estados mentais seguintes: crença, desejo, intenção e expectativa, referentes a dois agentes cognitivos autônomos envolvidos numa interação de ensino/aprendizagem relativamente à solução de um problema. Nesse diálogo, os dois agentes podem alcançar uma das seguintes situações:

- (i) concordam a respeito do conhecimento envolvido mas discutem o método de solução;
- (ii) possuem conhecimentos diferentes sobre o problema e necessitam entrar em acordo sobre o conhecimento envolvido na solução do problema [OVC 94b], e
- (iii) diálogos em que um dos agentes pretende conversar sobre um determinado problema e o outro não está interessado, ou não tem condições de cooperar.

Para tanto, utiliza-se da arquitetura SEM - Sociedade dos Estados Mentais, proposta por Corrêa [Cor 94]. Mais especificamente, alguns resultados desta pesquisa já foram divulgados,

podendo-se acessá-los em [MV 95] e [MVC 96].

A idéia de usar estado mental está relacionada com a proposta de uma das famosas correntes de pesquisa em agentes, tendo Shoham [Sho 93], com a sua proposta de Programação Orientada a Agentes, como um dos seus principais proponentes. Nesse contexto, um agente é definido como uma entidade cujo estado é visto como consistindo de componentes mentais tais como: crenças, capacidades, escolha e compromisso. Essa visão inspirou Corrêa na proposição de sua arquitetura SEM.

3.2.3 Abordagens especializadas

Essas abordagens dizem respeito às suas preocupações com principalmente um dos aspectos supracitados relativos ao estado da arte restritos à modelagem do domínio, modelagem do estudante ou ainda modelagem pedagógica. Algumas das abordagens abaixo não fazem menção ao uso de agentes, mas foram aqui incluídas por terem relação próxima com os objetivos dessa tese.

Com respeito ao modelo do domínio, em [Sel 92, CS 91] destaca-se a importância da noção de pontos de vista, através de múltiplas representações sobre um certo objeto de conhecimento. Ao mesmo tempo, na primeira referência ressalva-se que é hora de se buscar meios para tornar essa noção computacionalmente proveitosa.

Percebe-se em [Oli 94], um investimento na melhoria da capacidade de interação cooperativa entre o sistema tutor e o aprendiz. Neste trabalho, procurou-se auferir ganhos relacionados às capacidades de adaptação e auto-avaliação do sistema, exigindo-se, para tanto, que o mesmo seja dotado de capacidade de aprendizagem e de avaliação de seus próprios processos de aprendizagem. Isso conduziu à incorporação de recursos na estrutura do sistema, permitindo criar condições para evolução da base de conhecimento sobre o domínio de ensino, através da interação entre o sistema e o aprendiz.

Em [GB 95], constata-se que quando se considera de forma explícita a noção de *curriculum* associada ao domínio num STI, normalmente esta tal noção é tratada como uma componente não ativa, tal como o fez [GGL 92] na proposta de sua arquitetura multi-agentes acima mencionada. Contrapondo-se a essa posição, propõem-se a dar um papel ativo a um tal componente no modelo do domínio. Nesse sentido, eles sugerem uma organização particular para o domínio, olhando-o em dois níveis: os objetivos de aprendizagem e os pontos de vista associados a esses objetivos. Com base nisso, eles propõem um STI multi-agentes, onde os agentes estariam envolvidos tanto com os objetivos quanto com os pontos de vista. A pretensão é, portanto, chegar a uma definição duma representação centrada em agentes do domínio de ensino e daí, num estudo das interações dos agentes envolvidos. Concluindo, o autor evidencia sua pretensão em chegar na definição de um modelo de agente e de uma arquitetura onde possa integrar os agentes.

Com relação à modelagem do estudante, as pesquisas atuais estão apontando uma alternativa na qual o estudante participa na elaboração de um tal modelo. Trata-se de uma idéia destacada inicialmente por Self [Sel 88a], onde ele se refere à possibilidade do conteúdo deste modelo ser aberto ao estudante, de maneira que este possa inspecioná-lo. Essa idéia continua enfatizada em [CS 91], ressaltando-se a necessidade de uma representação compreensível do modelo, permitindo ao aprendiz não apenas observar o que o sistema “sabe” sobre ele, mas poder, em alguns casos,

modificar o modelo. Mais recentemente, em [BPB 93, BP 95], essas idéias passam, de algum modo, a ser realizadas, de acordo com um protocolo de negociação entre as crenças do aprendiz e as do sistema em relação ao modelo do estudante. Isso avança um passo adiante nesta evolução, permitindo ao aluno retificar os eventuais erros de diagnóstico (de seu ponto de vista). Uma das grandes virtudes deste modelo inspecionável e modificável, é fornecer ao estudante uma realimentação (*feedback*) sobre seus conhecimentos e ao sistema um *feedback* sobre a qualidade de seu diagnóstico [Dil 93]. Também em [PSH 95], considera-se ainda essa questão sobre um modelo do estudante aberto, só que numa perspectiva um pouco diferente. Assim, adota a idéia de disponibilizar o modelo do aprendiz a agentes externos, através de *interfaces* diferentes, provendo informações apropriadas, para agentes diferentes.

Ainda com respeito à modelagem do estudante, há também uma vertente de pesquisa investindo na questão do diagnóstico distribuído. Em [LGM 95], os autores evidenciam um problema particular em modelagem do estudante, relativo ao reconhecimento de falsos entendimentos (*misconceptions*), notadamente, quando se trata de erros sobre o raciocínio. Nesse sentido, eles apresentam uma proposta para uma modelagem distribuída do estudante, considerando uma abordagem multi-agentes. Com isso, referem-se à existência de múltiplos pontos de vista para um dado problema a ser resolvido. Desse modo, de posse de uma *a priori* pretendida solução global, permite-se o raciocínio em paralelo seguindo os múltiplos pontos de vista. No artigo, eles discutem como o modelo distribuído do estudante é construído pelos agentes. Procuram ilustrar as idéias através de um problema particular em geometria, tratando da construção de um quadrado, considerando três pontos de vista em uma tal construção. Essa mesma equipe prossegue com mais publicações [GML 96] visando prover um maior detalhamento dos algoritmos envolvidos na atividade de diagnóstico distribuído, e em [LGM 96] apresenta um modelo de agente para trabalhar em atividades de resolução de problema pelo aprendiz.

Outras iniciativas similares incluem os trabalhos que constam em [ME 92, Loi 94], tal como descrito a seguir.

Em [ME 92], adotou-se também uma abordagem onde se constam vários especialistas que resolvem os mesmos problemas com diferentes pontos de vista.

Com respeito a [Loi 94], define-se um modelo instrucional de diagnóstico concorrente. Este modelo se utiliza de uma estrutura baseada em atores [Agh 86], fazendo uso de uma linguagem de programação com suporte à concorrência. Nessa referência, menciona-se o domínio de resolução de equações do 1º grau como o escolhido na aplicação do modelo.

3.2.4 Considerações finais

De fato, analisando-se as iniciativas observadas nos trabalhos discutidos anteriormente, no tocante ao uso de abordagens multi-agentes, percebem-se pretensões e rumos bem diferentes. Na primeira abordagem, os trabalhos não propõem mudanças nas bases de uma arquitetura clássica em Sistemas Tutores, ao invés disso, investem num enfoque de agentes visando tirar proveitos em relação a aspectos em Engenharia de Software. Na abordagem Mentalista, tal como já mencionado, há uma proposta de rumo bem diferente, com respeito à arquitetura básica e canônica de um Sistema Tutor. O trabalho de pesquisa nessa linha, que foi

discutido anteriormente, já indica alguns sinais parciais em direção a uma eventual consolidação. Por fim, as abordagens Especializadas seguem também um rumo promissor em relação aos avanços nas pesquisas na área em pauta, mesmo sem se afastarem muito das bases da abordagem clássica nos sistemas em questão. A idéia é mais no sentido de revisão e ampliação dos modelos de um tal sistema. Pode-se constatar entre as pesquisas indicadas uma certa incipiência e/ou mesmo um foco muito pontual. Agora, será introduzido rapidamente o ambiente MATHEMA, já referido no Capítulo 1, e que constitui um grande alvo na presente discussão. Este ambiente pode, com efeito, ser visto como situado na categoria das abordagens Especializadas.

O MATHEMA, como um ambiente integrado, propõe contribuição em vários dos aspectos destacados acima, conforme o que se poderá perceber a partir do Capítulo 4. Primeiramente, contribui-se com uma solução para modelo de domínio que abrange o que se recomenda no estado da arte, sendo definido com preocupações ligadas a sua qualidade e computabilidade. Prosseguindo-se, propõe-se também uma solução multi-agente com suas particularidades para os sistemas IA-ED, tendo principalmente o modelo do domínio como uma de suas bases e justificativas. A partir da solução multi-agentes, contribui-se ainda mais para o processo de modelagem do estudante, apresentando uma alternativa ao diagnóstico cognitivo distribuído. Enfim, o próprio ambiente é sem dúvida uma contribuição significativa principalmente devido ao seu papel de ser a unidade acolhedora das soluções específicas já citadas. Ademais, há outras contribuições vinculadas diretamente às referidas para, por exemplo: a modelagem das interações entre o Aprendiz e um Agente Tutor, apresentadas no Capítulo 8 e na conclusão dessa tese.

Aqui, primeiro estudou-se as questões relacionadas à melhoria no processo de interação entre o Aprendiz e o Sistema Tutor. Daí, investiu-se primeiramente na modelagem do domínio e, a partir de então considerando alguns aspectos da modelagem do estudante com foco na qualidade do processo de interação Sistema-Aprendiz, chegando-se à definição de um SMA e, em seguida, ao desenho e desenvolvimento do MATHEMA.

4. Concepção do ambiente MATHEMA

Neste capítulo apresentam-se, inicialmente, algumas considerações gerais sobre a concepção de sistemas IA-ED em geral. Em seguida, focalizam-se os aspectos de concepção do ambiente de aprendizagem MATHEMA, visando mostrar o trajeto de idéias que apoiaram a sua concepção. Nesse sentido, são discutidas questões ligadas ao conhecimento envolvido num sistema tutor, chegando-se à definição de um modelo do conhecimento do domínio. Posteriormente, apresenta-se uma proposta para concepção de um sistema tutor através de uma abordagem multi-agentes. Finalmente, chega-se ao MATHEMA, descrevendo seu princípio geral, sua arquitetura e sua funcionalidade global.

4.1 Prolegômenos

Boa parte da pesquisa na concepção de sistemas IA-ED (a exemplo de STI, AIA) envolve, essencialmente, interações tutoriais entre duas entidades principais, a saber: uma máquina (desempenhando o papel de tutor) e um humano (comportando-se como um aprendiz). Neste contexto, há um processo de interação, onde o objetivo maior é o sistema tutor facilitar a aquisição

de conhecimentos por parte do aprendiz, durante as atividades pedagógicas. Para isso, o sistema tutor conta com bases de conhecimento e com mecanismos de raciocínio sobre as ações do aprendiz e sobre essas bases, para adaptar-se, segundo a sua percepção, às necessidades individuais do aprendiz, no processo de interação dinâmica. Isto diz respeito à possibilidade do tutor produzir ações apropriadas e individualizadas ao aprendiz, segundo algum modelo de interação definido. É nesse momento, que a IA aparece como uma aliada, provendo, por exemplo, técnicas para representação e manipulação do conhecimento envolvido.

A missão para o sistema tutor, tal como pretendida acima, tem sido um empreendimento central da pesquisa na área em discussão. Trata-se de uma tarefa amplamente assumida como importante, mas ao mesmo tempo desafiadora em termos de se conseguir soluções consideradas efetivas do ponto de vista educacional. Um dos fortes desafios está em como lidar com a complexidade do conhecimento que o sistema deve se utilizar para promover uma interação apropriada com o aprendiz humano. Este conhecimento, entre outras coisas, diz respeito ao conhecimento do domínio de aplicação, ao conhecimento pedagógico e ao conhecimento sobre o aprendiz. Todo esse conhecimento deve ser organizado adequadamente no sistema para dar suporte na interação com o Aprendiz. Isso é feito, normalmente, tendo-se já fixado *a priori* uma concepção de aprendizagem, enfatizando-se uma visão de aprendiz e um conteúdo de aprendizagem, como elementos de concepção do modelo de interação. Levando-se em conta essas considerações, têm-se as condições básicas para se definir adequadamente um ambiente de aprendizagem.

Toda essa pretensão, rumo à concepção de tais sistemas, conduz a problemas complicados, notadamente os que dizem respeito à engenharia do conhecimento, alguns dos quais serão discutidos posteriormente nesta tese. Alguns aspectos dessa problemática fazem parte do escopo da presente pesquisa, devido ao seu foco na concepção de um modelo de ambiente interativo de aprendizagem.

A idéia aqui sobre uma tal concepção, iniciou-se por concentrar esforços na elaboração de um método para dotar o sistema tutor de um modelo adequado sobre o conhecimento de um dado domínio. Assim, procurou-se definir um mecanismo para perceber este conhecimento, levando-se em conta um compromisso entre a sua riqueza e sua estruturação. Isso, posteriormente, conduziu, por exemplo, à definição de um sistema tutor, enfocando, na sua concepção, aspectos da Inteligência Artificial Distribuída, segundo uma abordagem baseada em Sistemas Multi-Agentes. Os argumentos mais específicos para um tal investimento são apresentados no decorrer deste e dos próximos capítulos.

4.2 Do modelo mínimo de aprendizagem ao MATHEMA

Antes da definição do modelo de ambiente interativo de aprendizagem baseado em múltiplos agentes, são apresentadas algumas considerações e proposições nas quais está assentada a concepção do ambiente pretendido.

4.2.1 Modelo de ambiente mínimo de aprendizagem

A idéia de um modelo para um ambiente mínimo de aprendizagem é aqui colocada tanto como um esquema metodológico de apoio à concepção do Sistema Tutor, quanto para servir de um

modelo abstrato básico para referir-se às interações tutoriais envolvendo as duas entidades mencionadas anteriormente: Tutor e Aprendiz.

Neste sentido, buscam-se neste modelo os elementos mínimos para se atacar a questão da adaptação do Sistema Tutor ao Aprendiz, tal como colocada e motivada na seção anterior. Aqui, a pretensão é a de habilitar o Aprendiz a adquirir conhecimento como resultado de sua interação com o tutor. Para isso, trabalhou-se numa perspectiva de dispor de um modelo cooperativo de ensino/aprendizagem⁹, apoiado no modelo ilustrado na Figura 4.1.

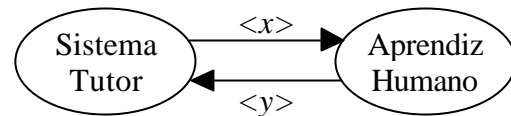


Figura 4.1: Modelo de ambiente mínimo de aprendizagem

De uma maneira mais abstrata, o processo de interação envolvido no modelo da Figura 4.1, é visto como ocorrendo dentro de um esquema de troca de mensagens ($\langle x \rangle$, $\langle y \rangle$) entre as duas entidades envolvidas, obedecendo a um certo protocolo. Neste esquema, o Sistema Tutor formula e envia uma mensagem com conteúdo $\langle x \rangle$ para o Aprendiz, e este reage produzindo e devolvendo-lhe uma mensagem com conteúdo $\langle y \rangle$. Assim, para alcançar a referida e almejada adaptação, tal como discutida no Capítulo 1, é fundamental o investimento na qualidade relacionada à natureza e conteúdo dessas mensagens, bem como em relação aos seus processamentos para, por exemplo, escolher um conteúdo adequado no momento das intervenções. Para tanto, precisa-se definir um ambiente suficientemente rico para realizar tal pretensão.

Considerando-se a necessidade por melhoramento e efetividade no processo de interação entre o Sistema e o Aprendiz, levando-se em conta a questão da adaptação, algumas exigências de qualidades desejáveis em direção à concepção de um bom ambiente para ensino/aprendizagem têm sido adotadas no desenrolar da presente pesquisa. Essas exigências de qualidade podem ser, de algum modo, vistas no atual estado da arte na pesquisa em sistemas IA-ED (incluindo STI, ILE), conforme foi destacado no Capítulo 3. Essas exigências visam o provimento de boas condições no suporte a uma aprendizagem efetiva por parte do Aprendiz. Elas incluem:

- (i) *conhecimento sobre o domínio*: envolvendo um compromisso entre “riqueza” e estruturação apropriada do mesmo, provendo o aprendiz com condições apropriadas ao seu aprendizado. Um dos fatores importantes aqui, é a disponibilidade de múltiplas visões sobre um mesmo conhecimento.
- (ii) *conhecimento pedagógico*: em teorias de ensino e aprendizagem contemporâneas, um aluno deixa de ser visto como um mero receptor passivo de informações, e passa a ser considerado como um agente ativo no processo de aprendizagem [JJ 95]. Essa visão conduz à exigência de um compromisso entre “qualidade” e “quantidade” desse conhecimento. Ressalva-se aqui, uma hipótese apriorística de que não existe uma maneira única de ensinar um dado conhecimento a qualquer estudante. Portanto, a importância de

⁹ Este modelo é discutido no Capítulo 8.

múltiplos estilos de ensino.

- (iii) *conhecimento sobre o estudante*: considerando principalmente o diagnóstico distribuído e a inspeção do modelo pelo aprendiz¹⁰.
- (iv) *capacidade de interação*: é altamente desejável contar com um “bom” modelo de interação cooperativa, aliado e influenciado pelas qualidades expostas em (i), (ii) e (iii).

As qualidades sugeridas acima são aqui assumidas como hipóteses de trabalho importantes num enfoque mais significativo ao problema da adaptação. Elas são fundamentais para serem consideradas como condições que contribuem para a emergência de interações adaptativas promovidas pelos sistemas IA-ED.

Para fazer face a um vínculo mais forte e ao mesmo tempo sendo mais específico com as qualidades discutidas acima, foi trabalhada na idealização do nosso modelo uma proposta que abrange e viabiliza, de certo modo, cada uma delas. Tendo estabelecido o modelo de interação mencionado, foi enfatizada, inicialmente, a exigência de conhecimento do domínio (qualidade (i)), culminando com uma definição de um modelo multidimensional desse conhecimento, tal como apresentado a seguir.

4.2.2 Modelagem do conhecimento sobre um domínio

Considerando as qualidades requeridas acima mencionadas como basilares, passou-se nesta pesquisa, inicialmente, a fazer considerações sobre como realizá-las tomando por base a elaboração de um modelo estrutural para o domínio de conhecimento. Assim, elaboramos uma forma interessante e particular de olhar um domínio de conhecimento dado, e daí particioná-lo e organizá-lo apropriadamente para contar com ele no apoio ao processo de interação que ocorre entre as entidades que compõem o ambiente.

Nesta perspectiva, um esquema de modelagem do conhecimento do domínio foi definido. Segundo este esquema, o conhecimento sobre um domínio é abordado segundo duas formas de visualização: uma *visão externa* e uma *visão interna*.

4.2.2.1 Visão externa

A idéia aqui é submeter um dado domínio de conhecimento (de agora em diante será também chamado neste documento de domínio alvo) a um particionamento em diferentes subdomínios. Este particionamento é de natureza epistemológica e está comprometido com alguma visão particular do domínio. A idéia é buscar um particionamento para este domínio e um corpo de conhecimento que lhe seja subordinado no momento de sua operacionalização. Essa busca é orientada de maneira a alcançar um conhecimento com especialidades distribuídas em três dimensões de conhecimento [CLF 95, CP 96, CP 97a]:

- (i) uma para contexto (ponto de vista), que é composta por diferentes contextos ou pontos de vista sobre um domínio de conhecimento. Isso diz respeito às várias interpretações que podem ser dadas a um tal domínio, constituindo-se em representações (abordagens) diferentes de um mesmo objeto de conhecimento,
- (ii) uma para profundidade, que é relativa a um contexto particular. Ela é concernente a alguma forma de refinamento na linguagem de percepção, ou seja, a estratificação dos vários níveis epistemológicos de percepção do objeto de conheci-

mento em apreciação, e

- (iii) uma para lateralidade, que diz respeito aos conhecimentos afins de suporte a um dado objeto de conhecimento do domínio alvo, proveniente duma visão particular de contexto e profundidade, constituindo-se, no escopo do presente trabalho, em pré-requisitos da referida unidade.

Portanto, esta visão multidimensional mostra a possibilidade de um domínio poder ser focado por uma visão contextual, sendo que esta visão pode vir acompanhada de várias alternativas de variação do ponto de vista de profundidade e lateralidade em relação a cada contexto escolhido no domínio.

DEFINIÇÃO 4.1: Seja D um domínio de conhecimento sobre o qual pretende-se disponibilizar uma visão multidimensional (de acordo com as três dimensões acima). Um enfoque para D segundo uma tal visão, obedece a uma ordem epistemológica de ocorrência de seus elementos, sendo formalmente definido assim:

- (i) A D , associa-se um conjunto de contextos distintos $\{C_1, C_2, \dots, C_n\}$, onde cada C_i , com $1 \leq i \leq n$, representa um contexto particular. Isto é:

$$D \rightarrow \{C_1, C_2, \dots, C_n\}$$

- (ii) Daí então, a cada C_i fixado, atribui-se um conjunto de níveis diferentes de profundidade $\{P_{i1}, P_{i2}, \dots, P_{im}\}$, onde P_{ij} , com $j = 1$, representa a j -ésima profundidade associada ao i -ésimo contexto. Isto é:

$$C_i \rightarrow \{P_{i1}, P_{i2}, \dots, P_{im}\}$$

- (iii) Finalmente, a cada par $\langle C_i, P_{ij} \rangle$, associa-se um conjunto de lateralidades distintas $\{L_{ij1}, L_{ij2}, \dots, L_{ijr}\}$, onde um L_{ijk} , com $k = 1$, representa a k -ésima lateralidade desse par. Isto é:

$$\langle C_i, P_{ij} \rangle \rightarrow \{L_{ij1}, L_{ij2}, \dots, L_{ijr}\}$$

Esse conjunto de lateralidades é relativo a outros domínios de conhecimento e pode, em algum caso ou até por motivo de conveniência, ser vazio.

DEFINIÇÃO 4.2: Um domínio D submetido a uma partição comprometida com uma visão de contexto e profundidade é um conjunto definido pela união de subdomínios, obtido como segue: para cada par $\langle C_i, P_{ij} \rangle$, representando uma visão pontual sobre D , define-se um subdomínio de D , que será denotado aqui por d_{ij} . Logo:

$$D = \bigcup_{j=1}^m d_{ij}, i = 1, \dots, n.$$

DEFINIÇÃO 4.3: Um domínio de conhecimento lateral DL , comprometido com uma visão de lateralidade, é um conjunto definido pela união de subdomínios obtidos como segue: para cada visão L_{ijk} , define-se um subdomínio¹¹ externo à D , aqui denotado por d_{ijk} , encarregado de um conhecimento lateral em relação à D , de tal modo que $d_{ijk} \notin D$. Assim:

$$DL = \bigcup_{k=1}^r d_{ijk}, i = 1, \dots, n; j = 1, \dots, m.$$

Assim, o conhecimento relativo a um domínio D , a ser disponibilizado numa sociedade (sendo também chamado nesta tese de

¹⁰ Os termos estudante e aprendiz estão sendo usados neste texto como sinônimos.

¹¹ Cada subdomínio de lateralidade pode, apesar de não ter sido considerado no escopo desta tese, ser olhado e obtido da partição de um dado domínio submetido a uma visão de contexto e profundidade.

conhecimento da sociedade relativo à D), que será aqui denotado por D' , é definido pela união dos subdomínios de D mais o conhecimento lateral adjacente, tal como expresso logo a seguir.

DEFINIÇÃO 4.4: O conhecimento em D' é definido assim:

$$D' = D \cup DL.$$

Na Figura 4.2, apresenta-se o domínio alvo D constituído por seus d_{ij} , que estão representados como pontos de um plano cartesiano (onde tais pontos são, na verdade, pares ordenados, definidos pela visão Contexto \times Profundidade), tal como ilustrado no plano mais próximo. Nos demais planos estão mostradas as lateralidades escolhidas para D , tendo as indicações dos subdomínios correspondentes a visões particulares de lateralidade. Nessa Figura, entenda-se a existência de uma ordenação epistemológica, tal como mencionada anteriormente, onde o eixo do contexto é visto como principal (determinador). Daí, visualiza-se o eixo da profundidade relativa ao contexto. Finalmente, vem a lateralidade que é subordinada ao contexto e profundidade já estabelecidos.

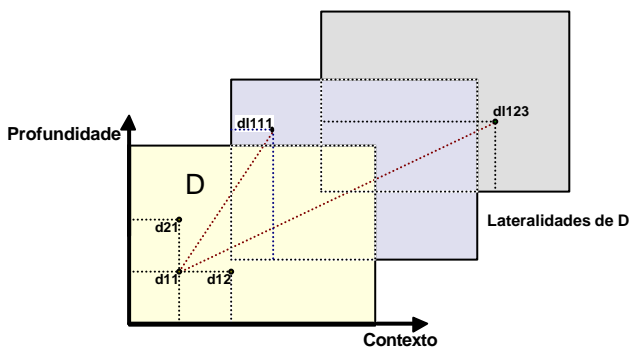


Figura 4.2: Visão Multidimensional do Conhecimento do Domínio

Visando ilustrar as noções presentes na visão multidimensional para o conhecimento do Domínio, tal como descrita acima, seguem três exemplos focalizando domínios específicos, tais como: Lógica, Álgebra e Geometria. Tratam-se de domínios que abrangem conteúdos estudados em escolas de 1º, 2º e 3º graus, escolhidos com o propósito de dar uma idéia da amplitude do potencial de uso e adequação da noção de dimensão do conhecimento nos termos definidos acima.

Exemplo 1: Domínio da Lógica Clássica

Suponha que o domínio (D) a ser modelado seja o da Lógica Clássica. Definindo este domínio de acordo com o modelo multidimensional de conhecimento apresentado anteriormente, pode-se considerar para D = Lógica Clássica, as seguintes dimensões:

Contextos

- C_1 = uma visão axiomática;
- C_2 = uma visão via Dedução Natural.
- C_3 = uma visão algébrica (Semântica).

Os três sistemas - Apresentação Axiomática, Dedução Natural e Semântica - são equivalentes no que se refere ao conjunto de fórmulas que eles geram, ou seja, o conjunto dos teoremas (gerado pelos dois primeiros) e o de tautologia ou fórmulas válidas (gerado pelo sistema algébrico ou semântico), são os mesmos.

Profundidades

Fixando-se, por exemplo, C_1 , pode-se ter as seguintes profundidades:

P_{11} = Lógica Clássica Proposicional (Ordem zero)

P_{12} = Lógica Clássica de Predicados (Ordem um)

Lateralidades

Em relação ao par $\langle C_1, P_{11} \rangle$, pode-se definir as lateralidades seguintes:

L_{111} = Teoria dos Conjuntos

L_{112} = Princípio da Indução Finita

Toda essa visão para Lógica Clássica pode, de certo modo, ser encontrada, por exemplo, em [Cos 92]. Nessa mesma referência, poderia-se ainda, considerando, por exemplo, um outro parâmetro para definir profundidade, definir o modelo multidimensional, levando-se em conta a extensão da lógica clássica para lógica modal. Portanto, trata-se de uma escolha de natureza essencialmente pedagógica.

Exemplo 2: Domínio da Álgebra (equações algébricas)

O Domínio (D) utilizado para ilustração se insere no campo da álgebra, mais especificamente: D = estudo de equação de segundo grau. Definindo este domínio de acordo com o modelo de domínio baseado nas dimensões de conhecimento propostas no MATHEMA, pode-se obter a seguinte configuração:

Contextos

- C_1 = através de métodos algébricos de resolução canônicos (fórmula de Bhaskara, e método da redução)
- C_2 = através de métodos de resolução do cálculo numérico

No escopo deste trabalho utiliza-se, para efeito de ilustração, apenas o contexto C_1 . Sobre ele foram definidas as profundidades que seguem, tomando um universo de trabalho onde se buscam as raízes num determinado universo, como parâmetro de definição dos níveis de profundidade. Convém esclarecer, entretanto, que esse é apenas um parâmetro a ser considerado, tendo muitos outros que poderiam ter sido levados em conta.

Profundidades

P_{11} = no domínio dos reais (R)

P_{12} = no domínio dos complexos (C)

Lateralidades

Em relação ao par $\langle C_1, P_{11} \rangle$, podemos definir as lateralidades seguintes:

L_{111} = expressões algébricas

L_{112} = operações com polinômios

L_{113} = produtos notáveis/fatoração

L_{114} = frações algébricas

L_{115} = equação do primeiro grau.

Diferente da facilidade de obtenção dos subdomínios tal como pode ser observada no domínio anterior, essa visão para o domínio das equações não se encontra descrita num único texto, mas pode ser observada e buscada em diferentes textos de matemática elementar e cálculo numérico.

Exemplo 3: Domínio da Geometria Euclidiana Plana

Suponha que o domínio em questão seja o da Geometria Euclidiana Plana. Assim, poderiam-se definir as dimensões do seguinte modo:

D = Geometria Euclidiana Plana - Estudo dos Triângulos

Contextos

C_1 = uma visão métrica;

C_2 = uma visão trigonométrica.

► Profundidades

Fixando-se, por exemplo, C_1 , poderiam-se ter as seguintes profundidades:

P_{11} = Triângulos Retângulos

P_{12} = Triângulos quaisquer.

► Lateralidades

Com respeito às lateralidades L para o par $\langle C_1, P_{11} \rangle$, pode-se ter:

L_{111} = Produtos Notáveis

L_{112} = Equação do 2º Grau.

Diferente também da facilidade mencionada anteriormente para o domínio da lógica, essa visão para o domínio da geometria pode ser buscada em diferentes textos de matemática elementar.

◇ ◇ ◇

Enfim, esta visão segundo as três dimensões e mais a organização interna de cada subdomínio, descrita a seguir, são o alicerce para se buscar um compromisso entre uma boa estruturação e riqueza de conhecimento de um dado domínio. Com isso e mais um mecanismo de interação mais perceptivo, incorporado num sistema multi-agentes, pretende-se dispor de um ambiente mais adequado para promover aprendizagem cooperativa e, assim, conseguir um maior potencial de adaptabilidade do sistema a um estudante em particular. A idéia é, portanto, combinar estes fatores em uma tal ordem que venham a favorecer o surgimento de um comportamento adaptativo na interação com o Aprendiz.

4.2.2.2 Visão interna

Uma vez fixados os subdomínios d_{ij} em D , onde i e j denotam, respectivamente, o i -ésimo contexto e a j -ésima profundidade associada a d_{ij} , ou fixando os domínios dl_{ijk} em DL tal como estabelecido acima, passa-se a olhá-los internamente como constituídos por um conjunto de unidades pedagógicas, definidas em função de objetivos de ensino/aprendizagem específicos que estão associados a um *curriculum*. Simbolicamente, tem-se:

$$\text{Curric} = \{up_1, up_2, \dots, up_n\},$$

onde *Curric* denota um currículo definido para um certo d_{ij} ou dl_{ijk} , sendo que cada up_i denota uma unidade pedagógica do *Curric*. Estas unidades estão relacionadas segundo uma ordem definida com base em critérios pedagógicos. A cada up_i corresponde um conjunto de problemas. A cada problema, está associado um conhecimento de suporte à sua resolução, incluindo basicamente: conceitos e resultados¹².

4.2.3 Sistema tutor via uma abordagem de agentes

De posse do modelo do domínio obtido e suas conseqüências nos modelos do estudante e pedagógico, tornou-se praticamente contingente a adoção de uma abordagem baseada em agentes na concepção do pretendido Sistema Tutor. Isso significou, portanto, buscar técnicas e funcionalidades da Inteligência Artificial Distribuída (IAD), segundo um enfoque de Sistema Multi-Agentes (SMA). O interesse na estrutura de um SMA diz respeito à sua adequação para tratar a complexidade envolvida no conhecimento e sua manipulação. Desse modo, foi adotada uma abordagem multi-agentes tanto na concepção quanto no desenvolvimento do modelo de ambiente para apoiar as interações tutoriais dotadas dos requisitos de qualidade supracitados.

Essa hipótese de adequação estrutural é aqui assumida como um recurso fundamental por oferecer vantagens no âmbito da Engenharia do Conhecimento envolvido. Nesse sentido, ela oferece mecanismos importantes na administração da complexidade do conhecimento via uma estratégia do tipo divida-e-conquiste, resultando em possíveis ganhos com respeito à modularidade, reusabilidade, incrementabilidade, etc. Essa hipótese oferece ainda um aumento no potencial de adaptabilidade dos sistemas em pauta, visando melhorar as interações entre as entidades envolvidas.

Assumida essa importância acima atribuída aos SMAs e com base no modelo do domínio definido, investiu-se posteriormente na construção de uma sociedade de agentes tutores, tal como descrito na seção 4.2.4. O modelo multi-agentes obtido introduz uma estruturação para o conhecimento local de cada agente e para a própria organização dos agentes da sociedade, através de um espaço multidimensional. Isto diz respeito a uma representação centrada em agentes sobre um domínio de conhecimento. Este modelo de ambiente de aprendizagem, almejado na presente pesquisa, passou a ser primeiramente definido de acordo com o modelo exibido na Figura 4.3.

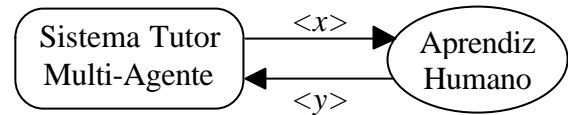


Figura 4.3: Modelo de Ambiente de Aprendizagem baseado em agentes

4.2.3.1 Construção da sociedade de agentes tutores

Para tratar cada um dos subdomínios de um domínio D , constrói-se um agente tutor considerando-se uma correspondência direta de índices obedecendo ao seguinte critério: para cada subdomínio d_{ij} de D , define-se um agente tutor AT_{ij} ($d_{ij} \rightarrow AT_{ij}$), significando que a definição de AT_{ij} é feita a partir de d_{ij} , o mesmo é feito para o subdomínio de conhecimento lateral dl_{ijk} , definindo-se um agente tutor AT_{ijk} ($dl_{ijk} \rightarrow AT_{ijk}$), responsável por alguma visão de lateralidade [CP 97b]. Nesse sentido, foram definidos dois tipos de agentes, tais como segue.

DEFINIÇÃO 4.5: O conjunto de todos os agentes tutores AT s relativo a um domínio D é definido assim:

$$AT = \bigcup_{j=1}^m AT_{ij}, i = 1, \dots, n.$$

DEFINIÇÃO 4.6: O conjunto de todos os agentes tutores ATL relativo a um domínio DL é definido assim:

$$ATL = \bigcup_{k=1}^t AT_{ijk}, i = 1, \dots, n; j = 1, \dots, m.$$

Assim, um agente AT_{ij} é dotado de conhecimento sobre o subdomínio d_{ij} e um agente AT_{ijk} é dotado de conhecimento sobre o domínio dl_{ijk} . A união desses agentes AT_{ij} e AT_{ijk} constituirá a sociedade de agentes tutores, tal como descrita a seguir.

As ligações entre os diferentes agentes são representadas por meio de relações de dependência. Essas dependências nos permitem distinguir as diferentes interações entre agentes tutores.

Baseado nas considerações acima, chega-se à definição da socie-

¹² Ver detalhes no Capítulo 8.

dade de agentes, tal como a seguir.

DEFINIÇÃO 4.7: A sociedade de agentes tutores artificiais (*SATA*) em relação a um domínio de conhecimento *D* é definida assim:

$$SATA = AT \cup ATL.$$

4.2.4 O ambiente MATHEMA

O modelo ilustrado na Figura 4.3 foi ampliado através da inclusão de novas entidades humanas e computacionais¹³. As razões para tal incremento, dizem respeito principalmente a dois aspectos: (i) uma questão de completeza conceitual e, ao mesmo tempo, por assumir que há uma influência no desempenho das entidades computacionais (no caso da inclusão das entidades humanas), especialmente as entidades que estarão monitorando o desempenho de *SATA* e, sempre que necessário, agindo sobre ela com a intenção de melhorá-la; e (ii) também por uma questão de arquitetura de *software*, no caso dos agentes de interface, procurando viabilizar apropriadamente as interações entre as entidades envolvidas. Ademais, outros argumentos em favor de uma tal iniciativa são apresentados no decorrer desta tese, a medida que as funcionalidades de cada uma dessas entidades forem esclarecidas.

Com a perspectiva supracitada, surge o MATHEMA como um modelo de ambiente interativo de aprendizagem baseado no computador, sendo concebido para apoiar atividades que venham a favorecer à realização de interações adaptativas e seus desdobramentos no processo de aprendizagem. Trata-se de um ambiente propondo uma nova abordagem no suporte à concepção de sistemas IA-ED, configurado-se como um caminho de solução adequado e efetivo ao problema da adaptação. As atividades de ensino-aprendizagem (adaptativo) são aqui vistas como consequência do processo de interações cooperativas envolvendo os seus componentes (*Aprendiz*, *Tutores*) em situações baseadas em resolução de problemas. Neste caso, o sistema envolve o aprendiz na resolução duma sucessão de problemas e passa a desempenhar um papel de tutor assistente. A noção de problema é aqui colocada de forma intuitiva, significando uma situação na qual, a partir de certas condições iniciais ou estado inicial do problema, tenta-se atingir um dado objetivo através de um conjunto de ações [Nil 82, Pea 84, Hun 87]. A aprendizagem, por sua vez, é assumida neste trabalho como sendo favorecida e decorrente de atividades provenientes do processo de resolução de problemas, significando aquisição de conhecimento. Nessa perspectiva, como um modelo conceitual, o MATHEMA propõe-se principalmente a prover princípios e uma arquitetura alternativa necessários para orientar o desenvolvimento de sistemas IA-ED particulares.

O princípio geral adotado na concepção do MATHEMA tem como proposta central o engajamento de um aprendiz humano numa relação de interação com uma sociedade de agentes tutores artificiais, visando envolvê-lo em situações de aprendizagem, a partir do seu envolvimento em atividades de resolução de problemas. Esses agentes tutores podem cooperar entre si ou com uma sociedade de especialistas humanos, em favor dum objetivo maior, que é a promoção da aquisição de conhecimento por parte do aprendiz. Nesta perspectiva, o MATHEMA representa um modelo de sistema IA-ED distribuído, onde os agentes tutores se comunicam por meio de troca de mensagens. Com isso, visa-se a

cooperação entre esses agentes com o intuito de suprir as necessidades do aprendiz.

4.3 Arquitetura do MATHEMA

A arquitetura multi-agentes do ambiente MATHEMA está baseada no princípio da integração de entidades humanas e artificiais dispostas a interagir cooperativamente. As motivações para as interações cooperativas variam entre os pares de entidades envolvidas, tal como poderá ser observado no decorrer desta tese.

DEFINIÇÃO 4.8: A arquitetura do MATHEMA, denotada por *Arq_Mat*, é definida por uma 6-upla, como segue: *Arq_Mat* = $\langle AH, SATA, SEH, AI, AM, ME \rangle$, onde:

- (i) *AH* denota um Aprendiz Humano;
- (ii) *SATA* denota uma Sociedade de Agentes Tutores Artificiais;
- (iii) *SEH* é uma Sociedade de Especialistas Humanos (funcionando como fonte de conhecimento através de operações de manutenção sobre *SATA*);
- (iv) *AI* denota o Agente de Interface que é encarregado da interação entre o aprendiz e a *SATA*;
- (v) *AM* denota o Agente de Manutenção que é responsável pela interface entre *SEH* e *SATA*;
- (vi) *ME* denota um Motivador Externo representando entidades humanas externas que podem motivar o aprendiz a trabalhar no MATHEMA.

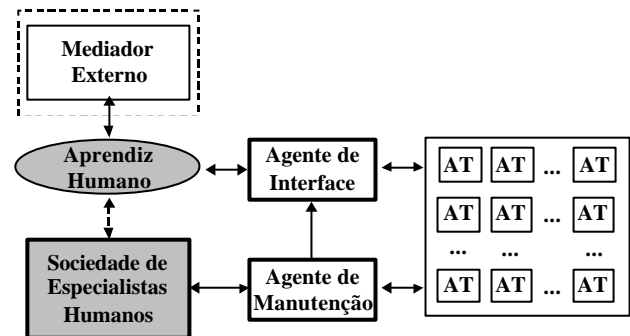


Figura 4.4: Arquitetura do MATHEMA

A arquitetura do MATHEMA, mostrada na Figura 4.4, ilustra esses seus seis componentes, sendo cada um deles descritos abaixo:

- (i) *Aprendiz Humano*: agente interessado em aprender algo sobre um dado domínio, trabalhado no ambiente MATHEMA. Ele desempenhará essencialmente o papel de um agente ativo, quando envolvido em atividades baseadas em resolução de problemas num dado domínio de aplicação, sob a assistência especializada de um agente tutor (imerso numa sociedade de agentes tutores cooperantes).
- (ii) *Sociedade de Agentes Tutores Artificiais*: coleção de agentes que podem cooperar entre si a fim de promover a aprendizagem de um dado aprendiz em atividade de resolução de problema. Cada um desses agentes é especializado em um subdomínio relacionado a um dado domínio de conhecimento (domínio alvo). Essa idéia foi inicialmente inspirada nas reflexões contidas em “Sociedade da Mente”, de Marvin Minsky [Min 85]. Segundo Minsky, a inteligência emerge da combinação de agentes mentais, cada um responsável por um pequeno processo.

¹³ Estas novas entidades são apresentadas na seção 4.2.5.

- (iii) *Sociedade de Especialistas Humanos*: fonte integrada de conhecimento externa ao sistema computacional definido para *SATA*. Ela desempenha, em certos momentos, o papel de uma entidade oracular, servindo de suporte para a *SATA*. Dessa Sociedade é requerida a criação e manutenção da *SATA* (com operações de inclusão, exclusão de agentes, bem como alterações no conhecimento dos agentes) e mais a disposição, em caso de uma falha mais crítica da *SATA*, de assistir, de um certo modo, os aprendizes. A *SEH* é, portanto, responsável pela incrementabilidade nas capacidades cognitivas da *SATA*. Ademais, é importante ressaltar que a *SEH*, mesmo sem ser requisitada por *SATA*, pode analisar um *log* com o desenvolvimento das interações entre o aprendiz e a *SATA*, avaliando o desempenho de seus agentes tutores e daí, sempre que necessário, promovendo-lhe melhorias.
- (iv) *Agente de Interface*: representa o elo de ligação entre o *Aprendiz Humano* e a *SATA*. Ele é responsável por desempenhar, primeiramente, o papel de comunicação do agente tutor com o Aprendiz e vice-versa. Além disso, é responsável particularmente por ajudar o aprendiz a escolher um agente distinguido entre os agentes *ATs* em *SATA*, para agir como seu *agente supervisor* (metáfora, por exemplo, para um orientador acadêmico num ambiente de pesquisa e pós-graduação). Desta interação, identifica-se uma lista ordenada de agentes pertinentes aos interesses do aprendiz, onde o mais pertinente é escolhido para ser o tal supervisor. É admitido que o *AI* sabe da existência dos agentes e suas capacidades em *SATA*.
- (v) *Agente de Manutenção*: representa principalmente um elo de ligação (*interface*) entre a *SEH* e a *SATA*, encarregando-se de prover uma interação entre elas. Para isso, oferece os meios necessários para *SEH* realizar operações de manutenção sobre *SATA*. No mais, há mecanismos de percepção e comunicação, bidirecionalmente, tanto no lado da *SEH* quanto de *SATA*. Mais ainda e primeiramente, fornece também operações que permitem criar *SATA*. De certo modo, pode-se entender que este agente provê meios que facilitam o processo de aquisição de conhecimento de *SATA*.
- (vi) *Motivador Externo*: entidades humanas externas que desempenham o papel de quem motiva o Aprendiz a trabalhar no MATHEMA. Alguns motivadores podem ser um professor do Aprendiz (numa situação onde, por exemplo, está bem definido o trilógismo aluno, professor e computador), seus colegas, etc. Num trabalho futuro sobre o MATHEMA, essas entidades poderão ser unidas ao Aprendiz através de um cenário que passará a dar suporte à implementação da idéia de trabalho de grupo cooperativo.

4.4 Funcionalidade geral

As setas mostradas na Figura 4.4 representam uma visão externa indicando os relacionamentos interativos entre os componentes do MATHEMA. Baseado nesses relacionamentos, será apresentada a seguir uma visão geral envolvendo um cenário de funcionamento das interações (mais de um ponto de vista abstrato e externo) que podem ocorrer no MATHEMA.

Suponha que, inicialmente, um dado *Aprendiz Humano*, supostamente incentivado, de algum modo, por um *Motivador Humano Externo*, resolve trabalhar pela primeira vez no ambiente

MATHEMA. A seguir, inicia-se uma interação dialógica entre o *Aprendiz* e o *Agente Interface (AI)*. Nesse caso, o *Aprendiz* informa o *AI* sobre seus objetivos. O *AI*, por sua vez, apresenta ao *Aprendiz* algo sobre o ambiente computacional de aprendizagem e o ajuda, mediante uma análise de seus objetivos e num contexto de diálogo, a escolher seu *Supervisor* na *SATA*. A partir daí, tendo sido escolhido o *Supervisor*, inicia-se um processo de interação cooperativa e didática entre o *Aprendiz* e o *Agente Supervisor*. O *Agente Supervisor* passa então a ser o responsável pelo *Aprendiz*, oferecendo-lhe uma orientação e suporte pedagógico necessários à promoção do seu aprendizado. Durante esta interação, situações de interação com diferentes complexidades podem ocorrer. A mais elementar é a que envolve apenas a interação entre o *Aprendiz* e o seu *Supervisor*, numa situação de resolução de problemas. A complexidade, entretanto, começa a aumentar à medida que a demanda do *Aprendiz*, no processo de solução dum problema, passa a requerer a participação de outros agentes, além do seu *Supervisor*. Nesse momento, um grupo de agentes coopera visando atender às necessidades do *Aprendiz*. Em algum momento, entretanto, é possível que a complexidade na interação evolua para uma situação mais extremada na qual os Agentes na *SATA* não conseguem atender à requisição do *Aprendiz*. Nesse caso, o *Agente Supervisor* primeiramente notifica o aprendiz sobre a impossibilidade em atendê-lo, pelo menos momentaneamente, aconselhando-o a retornar noutra ocasião, na qual a *SATA* estará apta a resolver o problema em apreço. Em seguida, passa a consultar sua entidade oracular. Assim, ele informa a *SEH* sobre o ocorrido, através de serviços oferecidos pelo *Agente de Manutenção*. A *SEH*, tendo resolvido o problema, informa o agente supervisor da sua disponibilidade em interagir com o *Aprendiz*, e assim o faz, caso todas as providências tenham sido tomadas. Além disso, caso julgue necessário, *SEH* realiza uma operação de manutenção na *SATA*.

4.5 Considerações sobre as interações no MATHEMA

Na seção anterior, apresentou-se uma idéia do funcionamento geral da arquitetura do MATHEMA. Nesta seção, são feitas algumas considerações sobre as interações no MATHEMA. Ao invés de considerar as interações entre todos os pares indicados na arquitetura do ambiente, são discutidas aqui, por fazerem parte do escopo central desta tese, apenas e sucintamente, os três tipos de interações indicados na Figura 4.5. São eles:

- (i) as interações que ocorrem entre um *Aprendiz Humano* e a *SATA* através de um agente tutor em cada momento;
- (ii) as interações entre os agentes tutores; e
- (iii) entre as duas sociedades: *SATA* e *SEH*.

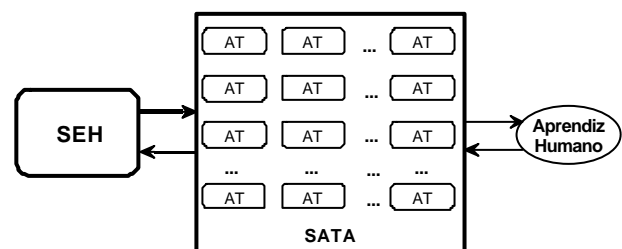


Figura 4.5: Interações no MATHEMA

Os tipos de interações (i) e (ii) são brevemente mencionados aqui, pois são o objeto de estudo dos capítulos seguintes, consti-

tuindo o alvo principal da tese. Já o tipo (iii), vai ser discutido aqui e não mais será tratado, a menos de rápidas referências, no restante do texto, pois está fora do escopo desta tese.

As interações dinâmicas envolvendo um *Aprendiz* e um *Agente Tutor* inciam-se em uma situação de resolução de problemas. A idéia é que a partir de uma atividade de resolução de problema, várias outras funções pedagógicas possam ser executadas, como parte do processo de interação adaptativa do tutor ao *Aprendiz*. Nestas interações, pode acontecer do *Agente Tutor* que participa da interação com o *Aprendiz*, não conseguir realizar determinadas tarefas. Neste caso, ele busca a cooperação com outros agentes em *SATA*. Esses agentes possuem, conforme será visto nos próximos capítulos, linguagem e protocolos que asseguram os seus envolvimento em atividades cooperativas. Com isso, eles podem trabalhar conjuntamente. Caso a cooperação entre os agentes não seja suficiente para alcançar os objetivos pretendidos na interação com o *Aprendiz*, o *Agente Tutor* (no caso, o que está responsável pelo controle na ocasião) recorrerá a um pedido de ajuda do meio exterior, isto é, a *SEH*.

As interações entre as sociedades *SATA* e *SEH* ocorrem por basicamente dois motivos. O primeiro é quando há um pedido de um *AT* à *SEH*. Nesse caso, a *SEH* analisa o pedido, busca solução para o mesmo e, caso seja necessário, realiza uma operação de manutenção em *SATA*. Isso pode implicar inclusão de agente, alteração no conhecimento dos agentes, exclusão de agentes, ou mesmo uma combinação delas. Essas interações acontecem com a intermediação do agente de manutenção, provendo protocolos e ferramentas para edição de agentes. Os protocolos para tais operações são discutidos no Capítulo 7, que trata especificamente de linguagem e protocolos de interação. O segundo motivo é quando a *SEH* observa, através de algum mecanismo (a exemplo de um *log*), o processo de interação entre o *Aprendiz* e *SATA* e constata algo que motive uma operação de manutenção sobre *SATA*. Uma tal operação é também realizada com a ajuda do agente de manutenção.

5. Sociedade de agentes tutores artificiais

Neste capítulo, apresenta-se inicialmente uma visão geral da sociedade de agentes tutores artificiais como um todo. Em seguida, discutem-se as características dos agentes tutores, enfocando principalmente aspectos básicos sobre o modelo do agente e os modelos de cooperação e comunicação entre esses agentes. Prosseguindo-se, apresenta-se a arquitetura desses agentes, discutindo a funcionalidade de cada um de seus componentes.

5.1 Características gerais

A sociedade *SATA* (Sociedade de Agentes Tutores Artificiais) é um Sistema Tutor Multi-Agente (*STMA*) constituído por uma coleção estruturada de agentes capazes de, através de linguagens e protocolos estabelecidos, cooperarem entre si e, em alguns casos, também com a Sociedade de Especialistas Humanos (*SEH*). Visa-se com uma tal cooperação, promover a aprendizagem de um dado estudante, estando envolvido numa situação de resolução de problemas ou numa de suas consequências.

A *SATA* é um *STMA* aberto e dinâmico no sentido de que ele permite entrada e saída de agentes, bem como atualização em algum deles, no momento em que são submetidos a uma determinada operação de manutenção. Esses agentes são concebidos e

mantidos pela *SEH*, sendo preparados para resolver tarefas relativas à competência que lhe foi atribuída.

Tais agentes são do tipo cognitivo e são aqui assumidos como benevolentes, significando que um agente sempre concorda em cooperar com os demais agentes quando for solicitado. Eles gozam de propriedades, tais como: autonomia, orientação a objetivos e habilidade social [FG 96, SD 96]. No contexto desta tese, cada uma dessas propriedades assume um significado, tal como a seguir. Por autonomia, entende-se que os agentes têm seus próprios objetivos, permitindo-lhes lidar com estes sem a intervenção direta de outros agentes. Nesse sentido, um agente não tem *a priori* a necessidade de cooperar com os outros agentes, tendo uma existência própria, bem como controle sobre suas ações e estados internos. Já a orientação a objetivos tem a ver com o fato dos agentes exibirem um comportamento de acordo com suas capacidades. Por fim, a habilidade social diz respeito à possibilidade dum agente poder interagir com os demais e com o mundo externo à *SATA*, via linguagens de interação definidas.

Estando diante de uma tarefa, cada agente é capaz de detectar, quando é o caso, a necessidade de cooperação, a partir da constatação da insuficiência de seus conhecimentos locais para resolvê-la. Nesse processo, um agente toma decisões baseadas em um dos três seguintes níveis de conhecimento [CLF 95]:

- (i) conhecimento: *situação na qual um agente age sobre uma tarefa, certo de que ele pode resolvê-la, sem a necessidade de interagir com os demais agentes.*
- (ii) ignorância parcial: *situação na qual um agente sabe o que ele não sabe com respeito à resolução de uma tarefa, mas pode saber de alguém entre os outros agentes da sociedade com quem pode contar para resolver uma tal tarefa. Esse caso ocorre quando um agente não consegue resolver uma tarefa (total ou parcialmente).*
- (iii) ignorância total: *o agente não sabe como resolver uma tarefa total ou parcialmente, nem conhece algum agente que possa resolvê-la.*

Enfim, com respeito a uma definição para os agentes tutores aqui referidos, adotou-se aquela que foi apresentada no Capítulo 2, devido a [FG 91], pela sua afinidade ao que foi requerido para a descrição do modelo de agente cognitivo pretendido no presente trabalho. Este modelo é dedicado especialmente ao suporte de atividades relacionadas às interações pedagógicas e cooperativas, envolvendo principalmente um Agente Tutor de *SATA* e um *Aprendiz*.

5.1.1 Modelo de agente

Nessa sociedade de agentes tutores artificiais, os agentes interagem entre si, utilizando-se de mecanismos de cooperação e comunicação. Cada um dos agentes é definido como uma entidade especializada em algum domínio específico, tendo o conhecimento necessário para realizar tarefas pedagógicas nesse domínio. Esses agentes desempenham papéis que incluem principalmente os de tutores inteligentes cooperativos, no momento de suas interações com um aprendiz. Nesse sentido, esses agentes são dotados de capacidades que lhes permitem organizarem-se, cooperarem entre si e comunicarem-se, visando resolver uma dada tarefa.

Cada agente possui um conjunto de habilidades utilizadas localmente na execução de tarefas. A habilidade diz respeito a um

método para resolver uma determinada classe de tarefa. Um método corresponde a uma das três categorias seguintes de atividades pedagógicas, quais sejam: resolução de problema, diagnóstico, instrução. Uma tarefa, por sua vez, é composta pela informação da habilidade que pode executá-la e pelos recursos necessários a sua execução.

Com respeito à propriedade de habilidade social atribuída aos agentes tutores em questão, isto é, a possibilidade deles interagirem entre si, há a noção de autoconhecimento e conhecimento social. Isso permite a um agente buscar outros agentes para cooperar sobre alguma tarefa, com base em um tal conhecimento.

Em suma, cada um dos componentes que asseguram a um agente as capacidades supramencionadas em relação ao modelo de agente, começam a ser descritos a seguir. Assim, descrevem-se algo sobre a organização, cooperação e comunicação. Prosseguindo-se, apresentam-se outros componentes já posicionados na arquitetura do agente. Enfim, outras características são descritas em capítulos posteriores, a exemplo da sua característica mais importante, que é a de ser uma entidade especializada¹⁴.

5.1.2 Organização e controle na sociedade de agentes

Para um agente envolver outros agentes numa resolução cooperativa de uma dada tarefa, este lança mão da sua estratégia para formação de coalizão. Uma tal estratégia está relacionada com a noção de organização de grupo de agentes dentro da *SATA*.

A organização social de um sistema multi-agentes é a maneira na qual um grupo é formado, em um dado momento, a fim de resolver uma certa tarefa conjuntamente.

O modelo de organização adotado no presente trabalho para os agentes tutores é baseado num modelo de agentes cognitivos [SDB 92], no sentido de que ele é inspirado em um tipo de organização social (a exemplo de um departamento acadêmico), caracterizado por meio de suas interações sociais. Este modelo se inclui na categoria ascendente (*bottom-up*) apresentada por Conte e Castelfranchi, em 1992 [apud AS 97], segundo a qual os agentes não têm necessariamente um objetivo comum a atingir. Mais especificamente, trata-se de um modelo baseado na complementaridade tal como mostrado por Conte e Sichman, em 1995 [apud AS 97], onde se considera a possibilidade da existência de capacidades complementares entre os agentes, permitindo, desse modo, que um agente possa alcançar seus objetivos pedindo auxílio aos demais. Assim sendo, o modelo de organização de grupo na *SATA* foi definido para ser dinâmico. Isso impõe que o processo de formação de uma coalizão não esteja preestabelecido, mas ocorra dinamicamente em função de uma demanda particular de um agente relativamente a um certo tipo de interação.

O controle das atividades cooperativas entre os agentes na *SATA* é de natureza essencialmente distribuída. A idéia básica do controle pode ser resumida assim: inicialmente o controle fica sob a responsabilidade do *agente supervisor* que, como foi visto no Capítulo 4, trata-se do agente responsável diretamente pela interação com o *Aprendiz*. Esse agente controla as interações com os agentes aos quais ele pediu cooperação. No momento em que um desses agentes precisa de cooperação, ele identifica os agentes com os quais pode cooperar e, nesse caso, passa a ser o controlador dessa instância de cooperação. Esse processo continua de tal maneira que quem pede cooperação passa a ser o controlador

dessa cooperação. Em suma, o controle não está sob a responsabilidade de um agente específico, mas todos podem potencialmente exercê-lo. Portanto, pode-se dizer que o controle, no âmbito da sociedade de agentes, é distribuído e que cada agente possui localmente um mecanismo de controle.

5.1.3 Modelo de cooperação

Como já foi discutido anteriormente, um agente nem sempre consegue por si só resolver uma tarefa ou mesmo uma parte dessa tarefa. Nesse caso, ele busca uma interação cooperativa com outros agentes na sociedade a fim de resolver aquilo que não conseguiu resolver sozinho. Daí, surge a idéia de cooperação¹⁵ e com ela se acha definida a maneira pela qual vários agentes se unem para resolver tarefas coordenadamente. Essa maneira, encontra-se definida num modelo de cooperação.

Um modelo de cooperação permite definir os protocolos e o conjunto das interações entre agentes que podem ocorrer no momento da realização de uma atividade cooperativa. Essas interações podem ser definidas com base nos protocolos utilizados na viabilização do processo cooperativo sob o qual dois agentes podem se relacionar. Esses relacionamentos estão definidos segundo a idéia de modelo multidimensional do conhecimento, conforme será detalhado mais adiante neste documento.

O modelo de cooperação adotado é híbrido, pois se utiliza tanto dum modelo de organização Mestre-Escravo, quanto dum modelo baseado em Licitação. O primeiro concerne a situação na qual um agente (o mestre) dispõe de uma tarefa a ser resolvida e, tendo identificado algum agente (o escravo) com capacidade para executá-la, ele simplesmente envia a tarefa para este agente identificado. Quando o agente requisitado conclui a tarefa, ele retorna os resultados ao agente que lhe requisitou. Já o segundo modelo tem a ver com a seguinte situação: se um agente tiver uma tarefa a ser resolvida e não conhece ninguém habilitado a resolvê-la, mas como sabe dos endereços, esse agente envia um anúncio de tarefa para toda sociedade ou para uma parte dela. Os agentes que se acharem em condições de executá-la, enviam uma proposta de volta. Caso mais de um agente responda favoravelmente, o anunciante escolhe, com base em critérios de afinidade (definidos de acordo com o modelo multidimensional) um dentre os proponentes e ativa o modelo Mestre-Escravo com ele. Caso apenas um agente se proponha, o anunciante ativa o mesmo modelo do caso anterior, diretamente. Por fim, o agente atualiza o seu Conhecimento Social. No caso mais extremado, quando nenhum agente dá um retorno positivo, chega-se numa situação de impossibilidade de cooperação com a sociedade, surgindo daí a necessidade de se recorrer ao mundo externo, neste caso, inicialmente, ao *Aprendiz*, e persistindo o problema, à *SEH*.

5.1.4 Modelo de comunicação

A interação entre os agentes torna-se viável a partir da adoção de algum mecanismo que possa promover a comunicação entre eles. Nesse sentido, a comunicação é aqui colocada como um suporte para o desenvolvimento das atividades interativas.

O modelo de comunicação utilizado pelos agentes na sociedade *SATA* define um mecanismo de comunicação baseado em troca

¹⁴ Esta característica será abordada no Capítulo 8.

¹⁵ Essa é a noção de cooperação aqui adotada, que diz respeito à possibilidade de um agente ter que envolver outros agentes para resolver uma tarefa sua, em que ele não é capaz de realizá-la total ou parcialmente.

assíncrona de mensagens. Esse modelo permite que cada agente possa se comunicar com os demais agentes da sociedade. Nesse sentido, há pelo menos quatro necessidades básicas a serem consideradas:

- um meio de comunicação, no qual as mensagens possam trafegar;
- uma linguagem de comunicação comum que assegure um entendimento mútuo entre o emissor e o receptor de uma mensagem;
- tipos de endereçamento que contemplem as necessidades do sistema. Tais necessidades dependem do modelo de cooperação, descrito na seção anterior (Tabela 5.1);
- a existência de um esquema que esclareça a informação do agente interlocutor que envia uma mensagem.

Tipo	Destino	Representação gráfica (conjunto)
direto	para um agente conhecido	
global	para todos os agentes da sociedade	
complementar	para os agentes pertinentes ao complemento de um subconjunto de agentes da sociedade	

Tabela 5.1: Tipos de endereçamentos

5.2 Arquitetura de um agente tutor

O objetivo desta seção é apresentar a arquitetura de um agente tutor, considerando, para isso, as funcionalidades de cada um dos seus componentes isoladamente, e em seguida uma visão funcional integrada dos sistemas. Essa apresentação está estruturada de forma a oferecer uma visão da arquitetura em dois níveis distintos de abstração: o nível macro e o nível micro, descritos, respectivamente, nas subseções 5.2.1 e 5.2.2. Dentro de cada um desses níveis, são descritos aspectos estáticos e dinâmicos relativos aos componentes da arquitetura.

5.2.1 Nível macro

Aqui, descreve-se modelo conceitual de arquitetura que foi elaborado para um agente tutor. Este modelo define sua estrutura através de uma composição hierárquica de três componentes principais, a saber: os sistemas **tutor**, **social** e **de distribuição**. Cada um desses sistemas está exibido na Figura 5.1, sendo descritos logo em seguida. Observe que se trata de três sistemas conceitualmente independentes, mas guardando uma interdependência funcional quando um agente necessita envolver-se em atividades cooperativas:

- (i) **Sistema Tutor:** este sistema é responsável pela interação direta com o aprendiz humano, cabendo a ele a execução das atividades tutoriais nos termos tratados no âmbito do ambiente MATHEMA. Isoladamente, este sistema pode ser visto como um sistema tutor inteligente. É nesse sistema onde estão os conhecimentos que o agente possui para resolver problemas e para efetuar outras operações peda-

gógicas no domínio de aplicação.

- (ii) **Sistema Social:** este sistema é responsável pela viabilização do comportamento cooperativo entre os agentes tutores. Ele é composto por bases de conhecimento e mecanismos de raciocínio necessários para realizar tal comportamento. Ademais, ele oferece recursos para cooperação com a Sociedade de Especialistas Humanos, através do agente de manutenção.

- (iii) **Sistema de Distribuição:** este sistema é responsável pela manipulação das mensagens enviadas e recebidas pelo agente tutor, através do meio de comunicação. Além disso, também é de sua responsabilidade a função de gerenciamento interno (cumprindo o protocolo no interior do agente) da distribuição das mensagens no agente tutor. Em suma, este sistema viabiliza a execução daquilo que o Sistema Social decide.

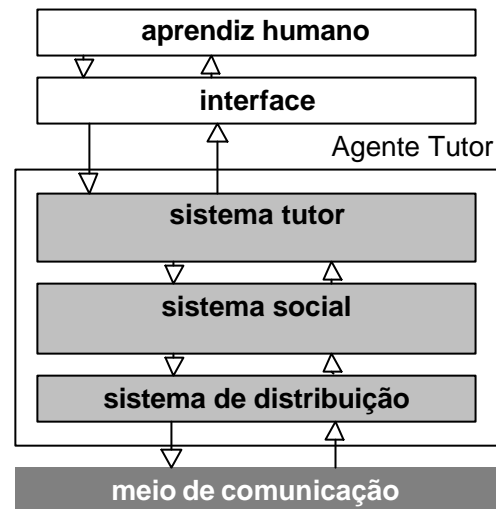


Figura 5.1: Arquitetura de um Agente Tutor: visão externa

Dinâmica da interação

Para dar uma idéia de como esses sistemas se comportam (comunicam) nessa arquitetura, descreve-se a seguir a dinâmica da interação entre eles. Essa dinâmica, colocada em termos sucintos, acontece do seguinte modo: o Sistema Tutor, em meio a uma interação com o Aprendiz, pode esbarrar em uma de suas limitações quando da tentativa de resolver uma dada tarefa e, assim, evidencia a sua necessidade de pedir cooperação a outros agentes. Nesse momento, ele efetua uma requisição de cooperação ao sistema social, tomando como parâmetro uma determinada tarefa proveniente da sua interação com o aprendiz. O Sistema Social, no intuito de efetivar a cooperação, faz um tratamento apropriado sobre esta tarefa, objetivando identificar e selecionar agentes habilitados a lhe ajudar a resolvê-la. De posse dessa seleção e concordância dos agentes escolhidos, o agente através do seu Sistema Social utiliza-se do Sistema de Distribuição como mediador, para interagir com tais agentes. Em suma, a interação entre esses dois sistemas ocorre da seguinte maneira: de cima para baixo, há um pedido de envio de mensagem; de baixo para cima, há uma notificação de recebimento de mensagem. Após o retorno do agente solicitado, o processo de cooperação finalmente chega ao fim com o Sistema Social retornando o resultado da execução da tarefa para o Sistema Tutor, que inclusive pode ser um caso de erro (situação de insucesso durante a cooperação).

5.2.2 Nível micro

Neste nível apresenta-se uma visão interior da arquitetura de um agente tutor, descrevendo os módulos de cada um dos sistemas mencionados, destacando-se, neste momento, apenas os mais relevantes na viabilização da interação entre agentes, e destes com a *SEH*. Estes módulos estão exibidos na Figura 5.2 e descritos logo em seguida. Observe que esta descrição se utiliza do mesmo recurso da seção anterior, ou seja, primeiramente descreve-se cada módulo isoladamente destacando as suas funções, em seguida, descreve-se a dinâmica dos módulos quando operando em conjunto. Essa visão de conjunto, entretanto, está descrita no Capítulo 6 e tem entre seus objetivos, mostrar como um agente trabalha a resolução de suas tarefas, como é invocado para uma interação e como se comunica com os outros agentes e com o mundo exterior.

Com relação à Figura 5.2, convém ressaltar que os módulos destacados com tons de cinza mais escuro dizem respeito às bases de conhecimento de apoio a atividade cooperativa. Já os outros módulos representam mecanismos operacionais.

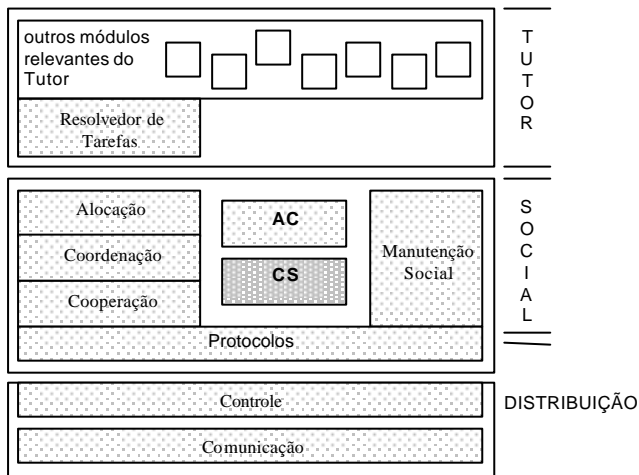


Figura 5.2: Arquitetura de um Agente Tutor: visão interna

A seguir será explicada a necessidade e a funcionalidade interna de cada um dos módulos indicados na Figura 5.2, considerando-os num agente que está buscando uma cooperação com outros agentes. Assim, observa-se todo o processamento desde o pedido de cooperação que sai do Resolvedor de Tarefas até o disparo da mensagem pelo módulo Comunicação.

Resolvedor de Tarefas: trata-se de um módulo representando uma abstração que é aqui destacada para desempenhar um conjunto de tarefas pedagógicas relativas ao Sistema Tutor. Entre essas tarefas, incluem-se: resolução de problemas, diagnóstico cognitivo e instrução. Somente no Capítulo 8, essas tarefas são definidas mais especificamente e localizadas dentro dos módulos que assumem suas responsabilidades.

Os agentes tutores possuem suas especialidades em resolver tarefas em domínios bem específicos. Entretanto, existem certas tarefas em que suas soluções envolvem mais do que a especialidade de um agente tutor. Cabe ao Resolvedor de Tarefas deste agente, realizar as tarefas que são de sua competência e, quando for o caso, identificar as que não o são. Este módulo conta com várias funcionalidades, entre elas inclui-se um método de decomposição de tarefas, que opera sobre tarefas descritas apenas na linguagem algébrica que foi definida para representar os domínios utilizados.

Na Figura 5.3, ilustra-se uma representação gráfica da relação de dependência causal para execução de uma tarefa T decomposta. O significado disso é que a resolução de uma tarefa T decomposta acontece quando se resolve todas as subtarefas dela obtidas, tendo associada a esse processo uma ordem de execução. Os círculos representam a tarefa e as subtarefas, ao passo que os arcos ordenados indicam uma relação de dependência causal entre duas tarefas.

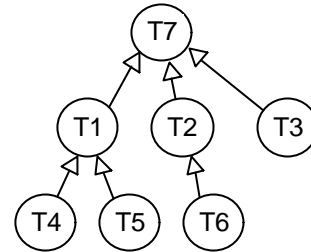


Figura 5.3: Representação gráfica da relação de dependência para execução de uma tarefa T decomposta

Para definir um modelo de organização na formação dinâmica de coalizões, tal como indicado acima, cada agente possui um mecanismo de raciocínio para operar sobre o modelo de suas capacidades e o modelo das capacidades dos outros agentes. Para fazer isso, ele conta com três módulos, que são aqui denominados: **Autoconhecimento**, **Conhecimento Social** e **Alocação**.

Autoconhecimento (AC) é um modelo representando o que o agente sabe sobre as suas próprias habilidades e conhecimentos. Este módulo é usado quando um agente necessita decidir se ele possui conhecimento para resolver uma certa tarefa pedagógica. Formalmente, o AC para um agente pode ser definido¹⁶ assim:

$$AC(j) = \langle \text{AgentId}, \text{Lista de Habilidades} \rangle,$$

onde *AgentId* denota um identificador para o agente j e *Lista de Habilidades* representa uma lista contendo as descrições para cada uma das habilidades que o agente possui.

Conhecimento Social (CS) é um modelo no qual o agente representa explicitamente o conhecimento sobre os outros agentes tutores na sociedade. Este módulo é usado quando o conhecimento do agente é insuficiente, ou mesmo ausente, para resolver uma dada tarefa. Assim, ele usa o CS visando identificar agentes que possam potencialmente engajar-se numa cooperação a fim de resolver a tarefa em questão. O CS é definido pelo conjunto de ACs dos outros agentes tutores. O CS constitui um modelo que um agente possui sobre a SATA. Devido ao dinamismo inerente à sociedade, este modelo pode se alterar no decorrer das interações. Formalmente, o CS pode ser definido assim:

$$CS = \bigcup_{j=1}^m AC(j),$$

onde m = cardinalidade do conjunto de agentes da SATA menos um, que é o próprio agente e j está denotando um índice relacionado ao nome de um dado agente.

Alocação é o módulo responsável pelo processo de seleção dos agentes identificados como aptos a resolver uma determinada tarefa que foi submetida pelo Sistema Tutor para cooperação.

Seu funcionamento básico é o seguinte: dada uma lista de tarefas

¹⁶ Para uma definição mais rigorosa e completa para o AC veja o capítulo 7.

propostas pelo módulo Resolvedor de Tarefas, o módulo Alocação opera sobre ela e produz como resultado uma lista de pares - $\langle t_1, LA_1 \rangle, \langle t_2, LA_2 \rangle, \dots, \langle t_n, LA_n \rangle$ - formada por cada uma das tarefas envolvidas e sua correspondente lista de agentes aptos a executá-la. Essa lista de agentes pode, eventualmente, ser vazia. Formalmente, pode-se definir este módulo como um mecanismo relacional, tal como segue:

$$\begin{aligned} \text{aloc} : \quad T &\rightarrow A \\ t &\mapsto \text{aloc}(t), \end{aligned}$$

onde $T = \{t_1, t_2, \dots, t_n\}$ denota uma lista de tarefas e $A = \{ag_1, ag_2, \dots, ag_m\}$ denota um conjunto de agentes.

Como uma tarefa pode ser apresentada para cooperação de maneira decomposta, o módulo Alocação necessita efetuar um trabalho de seleção para cada uma das subtarefas envolvidas. O mecanismo para realizar uma tal seleção é discutido no próximo capítulo.

Para efeitos de ilustração, na Figura 5.4 apresenta-se um esquema gráfico para as alocações feitas para as subtarefas obtidas na decomposição de uma determinada tarefa T . Os círculos preenchidos representam subtarefas não alocadas (T_5 e T_3), as setas pontilhadas que saem dos círculos não preenchidos indicam os agentes alocados para a sub tarefa em questão. Observe que para as subtarefas T_7 e T_4 , mais de um agente foi alocado. É importante salientar que em T_4 é possível executar a sub tarefa internamente pelo próprio agente tutor, sendo assim priorizada, buscando-se, sobretudo, tirar proveito da redução do tráfego de mensagens no meio de comunicação.

De acordo com a representação apresentada na Figura 5.4, o processo de alocação pode selecionar um agente, um conjunto deles, ou até mesmo nenhum. Cada situação conduz a um tratamento particular, que é abordado no próximo capítulo.

Uma vez o AT tendo identificado as tarefas e tendo realizado o processo de identificação de agentes, ele passa ao trabalho de coordenar e executar as atividades cooperativas. Para isso, o agente dispõe dos módulos **Coordenação** e **Cooperação**.

Coordenação é um mecanismo responsável por interpretar a estrutura de tarefas (grafo com todas as subtarefas), com base numa dependência causal definida, determinando a cada momento a tarefa a ser repassada para o próximo módulo, que no caso é o **Cooperação**, visando a execução desta tarefa. A idéia é assegurar a execução coerente da tarefa completa. Além disso, é este módulo que garante a distribuição e coleta correta dos seus recursos. Para tanto, baseia-se na estrutura da decomposição da tarefa. O módulo **Coordenação** opera sincronizando (paralelizando ou serializando) as execuções das subtarefas.

Seu funcionamento básico é o seguinte: dada uma lista de tarefas disposta numa estrutura de grafo vinda do módulo Resolvedor de Tarefas, o módulo **Coordenação** opera sobre ela e decide circunstancialmente qual a tarefa a ser enviada para o módulo **Cooperação**.

Na Figura 5.4, pode-se observar que a tarefa T_7 só pode ser executada após a realização das subtarefas T_1 , T_2 e T_3 . Da mesma forma, a sub tarefa T_1 só pode ser disparada após o retorno de T_4 e de T_5 ; enquanto o T_2 só após o retorno de T_6 . Pode-se observar também que as subtarefas T_3 , T_4 , T_5 e T_6 podem e devem ser realizadas de forma concorrente, pois são mutuamente independentes. A paralelização destes pedidos é possível devido à natu-

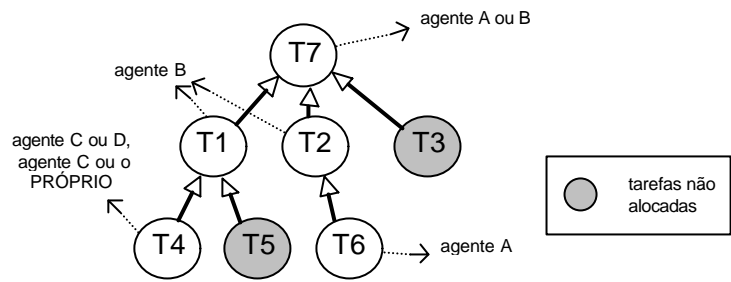


Figura 5.4: Representação gráfica das alocações para uma tarefa decomposta

reza das subtarefas, efetivamente realizável devido a natureza distribuída do sistema, e compensadora pois diminui o tempo de realização da cooperação sobre a tarefa T em questão, como um todo.

O insucesso na cooperação obtido para uma determinada sub tarefa particular T_n acarreta no insucesso na cooperação para a tarefa T , como um todo. Quando isto acontece, o módulo de coordenação espera a finalização das cooperações em andamento, não realiza as pendentes, e aborta o processo.

O módulo **Coordenação** é funcionalmente justificado devido à sua necessidade para tratar os possíveis pedidos de cooperação a partir de tarefas decompostas em subtarefas. Se não existisse tal possibilidade, este módulo não seria necessário na arquitetura. Com isto, fica fácil entender que o trabalho do módulo **Coordenação** é útil apenas no tratamento das situações que envolvem tarefas decompostas; caso contrário, para uma única tarefa, a função do módulo torna-se praticamente inexistente.

Cooperação é o módulo responsável por promover a execução de uma tarefa. Considerando que a execução de uma tarefa não é assegurada diretamente ou mesmo garantida de ser consolidada, cabe a esse módulo exaurir todas as possibilidades de cooperação, até alcançar uma situação conclusiva.

Quando se trata de uma situação na qual mais de um agente está apto a cooperar sobre uma dada tarefa, gera-se então um situação de conflito. Isso porque o agente que solicitou a cooperação terá que escolher um entre os agentes aptos. Assim, é também função do módulo **Cooperação** resolver um tal conflito, isto é, escolher um agente.

Além disso, o módulo **Cooperação** dispõe de um mecanismo para a ativar uma instância de execução de um dado protocolo, no momento do encaminhamento da atividade cooperativa ao agente que vai cooperar.

Esse módulo ativa protocolos adequados dependendo inicialmente da entrada e posteriormente, em certos casos, do desenrolar das suas atividades. Esse trabalho se faz necessário devido a sua função de exaurir todas as possibilidades de cooperação.

Seu funcionamento básico é o seguinte: para cada par envolvendo uma tarefa e uma lista agentes a ela associada vinda do módulo **Coordenação**, o módulo **Cooperação** seleciona, inicialmente, um protocolo apropriado à situação em questão. Porém, há situações em que, por motivos diferentes, outros protocolos poderão ser ativados para essa mesma tarefa. Isso tudo pode ser definido levando-se em conta o resultado obtido do módulo **Alocação**: um agente alocado, vários, ou nenhum.

Uma instância de cooperação encaminha a tarefa para algum agente da sociedade (inclusive ele próprio). Quando uma tarefa for encaminhada para o próprio agente diz-se que houve uma coope-

ração local, acontecendo através de uma chamada direta do módulo Cooperação para o sistema tutor.

Um contrato social, que é efetuado com algum membro da sociedade, pode ser obtido diretamente ou através de licitação. Entretanto, antes de encaminhar uma sub tarefa num contrato social, o módulo Cooperação deve saber qual o agente que será responsável pela sua execução. Nesse instante, o módulo Cooperação pode se deparar com três situações:

- (1) Nenhum agente foi selecionado para a tarefa em questão. *Neste caso, o módulo Cooperação deve realizar o protocolo baseado em licitação, tal como visto anteriormente.*
- (2) O módulo Alocação identificou apenas um agente. *Neste caso, o módulo Cooperação deve automaticamente tentar fechar um contrato diretamente com ele, porém, se não for possível, o módulo volta para a situação 1.*
- (3) O módulo Alocação selecionou mais de um agente. *Neste caso, o módulo Cooperação deve eleger um dentre eles para tentar fechar um contrato direto, mas, se não for possível efetuar o contrato, o agente eleito deve ser retirado da lista de pretendentes e o processo deve ser repetido até não existir mais agentes aptos, se isto acontecer, como em 2, volta-se a situação 1.*

Observe que as situações 2 e 3, por motivo de insucesso, podem acabar gerando a situação 1. Para melhorar o desempenho, quando for necessário negociar com toda a sociedade após tentativas fracassadas de contrato direto, é vantajoso negociar apenas com os agentes da sociedade que não foram detectados como inapropriados, nestes casos, ao invés de realizar uma difusão (*broadcast*), seria necessário apenas uma difusão seletiva (*multicast*) para não mobilizar os agentes inadequados e não gerar tráfego indesejado no meio de comunicação. A esta situação dá-se, neste trabalho, o nome de licitação reduzida.

Uma instância de cooperação esgota todas as possibilidades de cooperação e dispara os protocolos de cooperação necessários. No caso de sucesso, algum agente tutor da sociedade realiza a tarefa e envia o seu resultado até chegar na instância de cooperação. Nos casos de insucesso, quando todas as tentativas de cooperação fracassam, a *SEH* e o aprendiz são avisados da situação ocorrida através de uma notificação. Nos dois casos, algum retorno é devolvido para a instância de coordenação que efetuou o pedido de cooperação (o resultado ou o erro) seguido da identificação do pedido.

Manutenção Social (MS): Um dos requisitos básicos para garantir um funcionamento apropriado do módulo Alocação, é que todos os agentes tutores da sociedade estejam sempre com os seus CSs atualizados. Para manter a completeza do CS, cada agente tutor precisa estar apto a executar os protocolos de manutenção¹⁷ relacionados a operações de entrada, saída e atualização de agentes. O módulo Manutenção Social é responsável por esse tratamento dentro da arquitetura do agente tutor. Para cada situação de manutenção (entrada, saída e atualização) existe um algoritmo específico¹⁸.

Protocolos: Este módulo diz respeito às instâncias de diálogos

de interação em execução. Existe, conforme será visto no Capítulo 7, um conjunto preestabelecido de protocolos que trata as situações de interação entre os agentes tutores na *SATA*. O módulo Protocolos é o responsável pela criação e ativação de um diálogo com base nos protocolos disponíveis. Um protocolo define um comportamento interativo, enquanto que uma instância de diálogo é uma entidade com um contexto particular responsável pela sua execução.

O tratamento final da mensagem é feito no sistema de distribuição que é encarregado de realizar as atividades de trocas de mensagens propriamente ditas (envio e recebimento) entre as entidades envolvidas (dois agentes tutores ou um agente tutor e a *SEH*) numa determinada interação. Para isso, ele é constituído por dois módulos, a saber: os módulos **Controle** e **Comunicação**.

Controle é o módulo responsável pela intermediação entre o Sistema Social (através do módulo responsável pelas instâncias de diálogo em execução) e o Sistema de Distribuição. Ele conta com um mecanismo de apoio à tomada de decisão para saber se uma dada mensagem está relacionada a uma atividade de cooperação ou de manutenção. Uma vez tomada a decisão, ele encaminha a mensagem ao destino apropriado.

Suas funções são as seguintes:

- (i) *encaminhar as mensagens recebidas para serem tratadas pela instância de diálogo apropriada;*
- (ii) *checar a consistência das mensagens recebidas;*
- (iii) *eliminar as mensagens atrasadas que: não precisam mais de tratamento ou não existe mais o protocolo apropriado.*

O módulo *Controle* tem a sua importância nos dois sentidos. Quando uma mensagem é recebida, ela precisa ser encaminhada para a instância de diálogo apropriada, cabendo ao módulo *Controle* efetuar isto. Para tal trabalho, as mensagens que trafegam pelo ambiente de comunicação devem ser dotadas de informações adicionais. O módulo *Controle* também é dotado de informações e mecanismos que permitem desconsiderar o recebimento de uma determinada mensagem. Isto acontece especificamente devido ao atraso de sua chegada, sendo uma das seguintes situações: ou a instância de diálogo responsável pelo seu tratamento não existe mais; ou o tempo estabelecido para recebê-la, após uma difusão (*broadcast*), expirou.

No outro sentido, de uma instância de diálogo para o módulo *Comunicação*, a função do *Controle* é bastante simples, significando apenas colocar na mensagem a ser enviada as informações de controle sobre a instância em questão, visando o controle adequado no outro lado do diálogo.

Comunicação, por sua vez, é o módulo encarregado da distribuição e coleta das mensagens, fazendo assim, a mediação entre o sistema de distribuição e o meio de comunicação. A visão do módulo Comunicação cria uma abstração importante na independência de todo o mecanismo existente acima dele com relação ao meio de comunicação utilizado.

O módulo *Comunicação* implementa um modelo de comunicação que é responsável pelas atividades de envio e recebimento das mensagens através do ambiente de comunicação, comum a todos os ATs da sociedade.

5.3 Formalização do modelo de agente tutor

Uma vez descritas algumas das características fundamentais na

¹⁷ Esses protocolos estão descritos no Capítulo 7.

¹⁸ Os detalhes sobre o funcionamento deste módulo estão apresentados na seção 6.5, onde se trata das três situações relacionadas à manutenção.

composição do modelo de agente, convém agora apresentar uma definição formal para esse modelo.

O modelo de agente tutor adotado no presente trabalho, tal como já foi mencionado, é baseado num modelo de agentes cognitivos [SDB 92], no sentido de que ele é inspirado em um tipo de organização social, caracterizado por meio de suas interações. Entre as suas capacidades, tal como já mencionado, inclui-se: a de ser uma entidade especializada, podendo organizar-se, cooperar e comunicar-se com os demais agentes.

Antes de apresentar a definição para o modelo de agente, são dadas algumas definições a seguir, formalizando algumas das noções introduzidas anteriormente e que servem de suporte à definição de agente.

DEFINIÇÃO 5.1: Seja AC o autoconhecimento de um agente tutor AT , definido por $AC = \langle AgentId, LH \rangle$, onde: $AgentId$ é um identificador único para um agente e LH é uma lista indicando um conjunto finito de suas habilidades, sendo dada por $LH = \langle Hab_1, Hab_2, \dots, Hab_n \rangle$.

Considerando que um agente tutor AT em um dado domínio pedagógico pode resolver problemas, diagnosticar, e/ou instruir, pode-se definir uma habilidade como segue.

DEFINIÇÃO 5.2: Uma habilidade em LH de um agente tutor AT é definida por $Hab = \langle Tipo, Identificações, Representações \rangle$, onde:

1. $Tipo$ refere-se às tarefas pedagógicas, tais como: **Resolução, Diagnóstico e Instrução**.
2. $Identificações$ é um conjunto definido por $Identificações = \langle Ident_1, Ident_2, \dots, Ident_i \rangle$, onde cada $Ident_i \in Identificações$ objetiva atribuir um nome para a habilidade Hab .
3. $Representações$ é dada por $Representações = \langle Rep_1, Rep_2, \dots, Rep_i \rangle$, onde cada $Rep_k \in Representações$, define um formato de entrada e saída pelo qual a habilidade pode ser acessada, e é dada por $Rep = \langle Entrada, Saída \rangle$.

DEFINIÇÃO 5.3: Seja CS o conhecimento social de agente tutor AT , definido por $CS = \langle AC_1, AC_2, \dots, AC_n \rangle$, onde cada $AC_k \in CS$ é um autoconhecimento de cada um dos outros agentes tutores em $SATA$.

Observe que no CS de um certo AT não se inclui o seu autoconhecimento, representado aqui por AC_i .

DEFINIÇÃO 5.4: Seja MAG o modelo de um agente tutor AT , definido por $MAG = \langle ST, CS, AC, ME \rangle$, onde ST é um sistema tutor, sendo um componente especializado num domínio, CS é o conhecimento social, AC é o autoconhecimento, e ME é o mundo externo à $SATA$, definido por $ME = \langle AgInt, AgMnt \rangle$, onde: $AgInt$ denota o agente de interface, e $AgMnt$ denota o agente de manutenção.

Na definição acima, para o modelo agente, deixou-se implícito alguns componentes, tais como: mecanismos de entrada e saída, mecanismo de raciocínio social. Essa opção foi feita devido ao fato de tratar-se de componentes invariantes, em relação à sociedade de agentes.

6. Visão funcional das atividades cooperativas

A idéia aqui é discutir o comportamento dinâmico (*modus operandi*) de um agente no momento em que ele necessita envolver outros agentes para resolução cooperativa de uma dada tarefa. Pretende-se com isso, apontar as funcionalidades que emergem

no nível interno de cada agente quando eles interagem. Para isso, foram localizadas quatro etapas envolvidas no processo de interação entre dois agentes, quais sejam: *mecanismo de raciocínio social, coordenação e cooperação, tratamento dos protocolos e tratamento das mensagens*.

Cada uma destas etapas é explicada e visualizada com o acompanhamento das interações existentes entre os módulos envolvidos, considerando também os algoritmos utilizados para implementar os seus comportamentos.

Observe que a etapa relativa ao *mecanismo de raciocínio social*, bem como a de coordenação e cooperação, são utilizadas exclusivamente no processo cooperativo. Estas etapas acontecem apenas no agente que pede a cooperação. No agente cooperador, estas etapas não figuram. As demais etapas são também utilizadas em outro contexto, diferente do relativo ao processo cooperativo. Trata-se do contexto relacionado às situações de manutenção.

6.1 Mecanismo de raciocínio social

Como já foi mencionado anteriormente, quando o sistema tutor detecta a necessidade de cooperação, ele efetua um pedido de viabilização dessa cooperação ao sistema social (Figura 6.1). O parâmetro que acompanha este pedido é uma tarefa pedagógica (problema, diagnóstico, instrução). Esta tarefa é preparada pelo Resolvedor de Tarefas antes de efetuar-se o pedido. O preparo consiste em descrevê-la num padrão entendido pelo sistema social. Isso é importante, principalmente, quando se trata de uma tarefa decomposta. Uma tarefa é descrita numa estrutura de lista tal como mostra o esquema a seguir:

$\langle tarefa_1 \rangle, \langle tarefa_2 \rangle, \dots, \langle tarefa_n \rangle$

Esta lista representa uma estrutura bem mais elaborada do que aparenta. Ela é tratada, conforme mostrado anteriormente, pelo módulo Alocação.

Como há a possibilidade de existirem relações de dependência entre as tarefas envolvidas, embutidas em cada tarefa, na verdade, o que ela representa é uma estrutura de grafo. Essas relações de dependência são estabelecidas pelo Resolvedor de Tarefas.

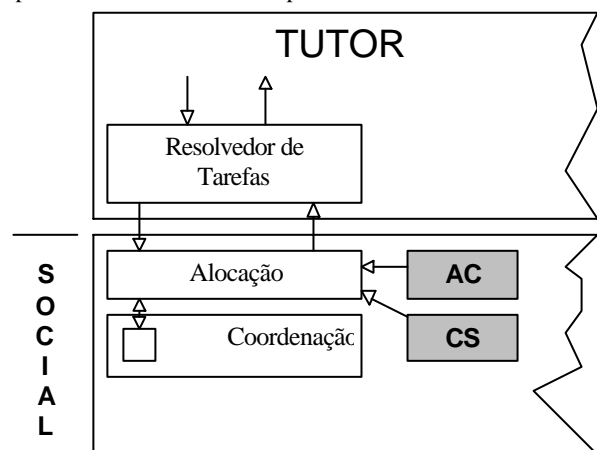


Figura 6.1: Comunicação interna entre os módulos na etapa de Raciocínio Social

O módulo *Alocação* realiza o seu trabalho para todas as tarefas apresentadas pelo Resolvedor de Tarefas. O algoritmo para realizar tal operação é descrito a seguir.

```
alocação.executa( lista de tarefas )
para cada tarefa[i]
```

```

se casa( AC, tarefa[i] )
  aloca o próprio agente para a tarefa[i]
fim-se
para cada AC-j do CS
  se casa( AC-j, tarefa[i] )
    aloca o agente do AC-j para a
    tarefa[i]
  fim-se
fim-para
fim-para

```

O algoritmo, como se pode observar acima, é relativamente trivial. Entretanto, pode haver alguma complexidade envolvida no procedimento de *casamento*, que dependendo das necessidades do sistema pode se tornar mais elaborado com a utilização de recursos mais avançados para o casamento de padrão, como por exemplo: biblioteca de sinônimos e regras de equivalência. O *casamento* é realizado sobre a habilidade de cada $AC \in CS$ e a habilidade necessária para realizar uma tarefa.

Uma vez concluído o processo de alocação, a estrutura abaixo, representando a tarefa alocada, é repassada para o módulo Coordenação. Esta estrutura em forma de tupla possui a seguinte configuração:

```

<tarefa1> <lista-de-agentes1>,
<tarefa2> <lista-de-agentes2>, ...,
<tarefan> <lista-de-agentesn>

```

Entretanto, quando a tarefa não se apresenta na forma decomposta, a situação fica reduzida a um caso particular, no qual a lista possui apenas um elemento. Portanto, recai-se no esquema mostrado abaixo.

```

<tarefa> → alocação → <tarefa> <lista-de-
agentes>

```

O resultado final do processo de coordenação (isto é, considerando o retorno do agente que cooperou) é repassado para o sistema tutor, podendo ser a tarefa resolvida ou mesmo um indicativo de insucesso.

6.2 Coordenação e Cooperação

O funcionamento integrado desses dois módulos (Coordenação e Cooperação) possui, tal como ilustrado na Figura 6.2, uma particularidade. Ele pode ocorrer de forma concorrente e sincronizada.

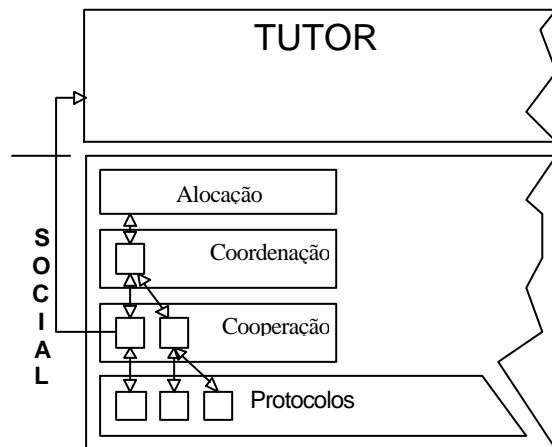


Figura 6.2: Visão integrada envolvendo Coordenação e Cooperação

Quando uma tarefa alocada é tratada pelo módulo Coordenação, cria-se uma instância de coordenação com um contexto particular associado, relativamente à tarefa em questão. A criação de instân-

cias é necessária devido à possibilidade da manipulação de vários pedidos de cooperação ao mesmo tempo pelo sistema social. Isso gera a possibilidade de coordenações concorrentes e, assim, a necessidade de um controle coerente.

Uma instância de coordenação realiza o seu trabalho com base no algoritmo de execução inicial, tal como definido abaixo.

```

coordenação.executa( lista de tarefas alocadas )
coordenando = VERDADEIRO
insucesso = FALSO
para cada tarefa[i]
  tarefa[i].condição = NÃO EXECUTADA
fim-para
coordenação.dispara() /* ativa o algoritmo de disparo */
espera enquanto coordenando = VERDADEIRO

```

Cada tarefa possui uma das três condições seguintes, importantes para o processo de coordenação:

- (i) **NÃO EXECUTADA**: significa que a tarefa ainda não foi executada pela sociedade ou internamente.
- (ii) **EXECUTADA**: a tarefa já foi executada e o resultado obtido já foi tratado e mantido.
- (iii) **EM ANDAMENTO**: existe uma instância de cooperação em andamento operando para viabilizar a sua execução.

O algoritmo descrito acima, simplesmente realiza as inicializações necessárias para o correto funcionamento do processo de coordenação, atribuindo a condição de **NÃO EXECUTADA** para todas as tarefas, sendo uma das inicializações; e ativa o algoritmo de disparo, cuja operacionalidade é discutida mais adiante. Após a execução do algoritmo de disparo, que inicia as primeiras cooperações em paralelo, o algoritmo passa para um estado de espera, verificando a finalização da operação de coordenação que prossegue enquanto não se chegar a um resultado final para a tarefa global. O algoritmo de disparo é o seguinte:

```

coordenação.dispara()
para cada tarefa[i]
  se tarefa[i].condição = NÃO EXECUTADA
  se tarefa[i] estiver pronta para ser executada
    cria uma instância de cooperação para encaminhá-la
  fim-se
fim-se
fim-para

```

O algoritmo de disparo analisa as tarefas **NÃO EXECUTADAS**, sendo que inicialmente todas estão com essa condição. Durante o processo de análise, se uma determinada tarefa está pronta para execução (sendo assim a tarefa da vez), cria-se uma instância de cooperação para tratá-la. Uma tarefa está pronta para ser executada, quando ocorre uma das duas situações seguintes:

- (i) ela não possui dependências, ou
- (ii) ela possui dependências, mas todas as tarefas das quais ela depende já foram executadas.

As instâncias de cooperação são criadas de forma concorrente. A idéia é a de se tirar proveito do aspecto de paralelismo inerente às aplicações para se realizar cooperações paralelas, desde que elas sejam mutuamente independentes. Cada instância, tal como ocorre com as instâncias de coordenação, possui um contexto particu-

lar a ela associado. Quando uma instância de cooperação é criada, a seguinte tupla é repassada para essa instância:

`< tarefa > < lista-de-agentes > < id. da tarefa >`

Como visto anteriormente, uma instância de coordenação pode criar várias instâncias de cooperação simultaneamente. Quando uma instância de cooperação finaliza o seu trabalho, ela notifica a instância de coordenação que a criou. Neste momento, há a possibilidade de existirem várias outras instâncias de cooperação em andamento. É necessário que a instância de coordenação saiba a qual tarefa com a condição de EM ANDAMENTO corresponde a notificação recebida. O campo `< id. da tarefa >`¹⁹ existe para viabilizar este controle.

A função para exaurir todas as possibilidades de cooperação, que é de competência do módulo Cooperação, é desempenhada pelo seguinte algoritmo:

```
cooperação.executa( tarefa alocada, identificador da tarefa )
se há algum agente alocado para a tarefa
  ativa o algoritmo encaminhamento passando a tarefa e a lista de alocados como parâmetro
  se algum agente executou a tarefa com SUCESSO
    comunica este sucesso para a instância de coordenação que o criou
    finaliza a execução do algoritmo
  fim-se
fim-se

ativa uma instância do diálogo de licitação passando como parâmetro a habilidade necessária para a execução da tarefa e a lista dos agentes já contactados

se algum agente se propuser a executar a tarefa anunciada
  ativa o algoritmo de encaminhamento passando a tarefa e a lista dos proponentes como parâmetro
  se algum agente executou a tarefa com SUCESSO
    comunica este sucesso para a instância de coordenação que o criou
    finaliza a execução do algoritmo
  fim-se
senão
  comunica a situação de INSUCESSO para a instância de coordenação que o criou
  notifica a situação de INSUCESSO para SEH
  finaliza a execução do algoritmo
```

O algoritmo **cooperação.executa** descrito acima, realiza o trabalho de exaustão descrito anteriormente quando se apresentou o módulo Cooperação. Em suma, ele consta das seguintes etapas:

- (1) Realização de uma tentativa de encaminhamento com a lista de alocados. Em caso de sucesso (algum agente cooperou), comunica-se o fato à instância de coordenação e finaliza-se a execução do algoritmo.
- (2) Caso não se obtenha sucesso na etapa 1, acontecerá uma licitação. O mesmo acontece, caso a etapa 1 não ocorra; esse

caso pode decorrer do fato da lista está vazia.

- (3) Se o resultado da etapa 2 for bem sucedido (algum agente se propôs a cooperar), ocorre uma tentativa de encaminhamento, só que agora com a lista de proponentes. Em caso de sucesso, procede-se como na etapa 1.
- (4) Caso não se obtenha sucesso na etapa 3, ou caso ela não aconteça (nenhum agente se propôs), comunica-se o insucesso para a instância de coordenação e para a *SEH*, logo após, a execução é finalizada.

A função de encaminhamento, usada para contactar agentes potencialmente cooperantes, referenciada no algoritmo anterior, é realizada através do seguinte algoritmo:

```
cooperação.encaminhamento( tarefa, lista de agentes )
se há mais de um agente na lista
  ordena a lista de agentes segundo o critério de afinidade
fim-se
para cada agente[i] da lista
  se o agente[i].apelido = AC.apelido
    efetua um pedido interno de cooperação
  senão
    cria uma instância do diálogo mestre-escravopassando como parâmetro o apelido do agente[i] e a tarefa
  fim-se
se houve sucesso em um dos dois casos
  retorna SUCESSO
senão
  notifica a situação de INSUCESSO para SEH
  fim-se
fim-para
retorna INSUCESSO
```

O algoritmo **cooperação.encaminhamento** foi referenciado no algoritmo **cooperação.executa** em dois lugares: antes e após a licitação. De forma sucinta, o seu funcionamento é o seguinte:

Inicialmente acontece um tratamento para uma situação de conflito que ocorre quando o encaminhamento pode ser efetuado para mais de um agente da sociedade. O conflito acontece quando há mais de um agente alocado para a tarefa (antes da licitação), ou quando há mais de um agente proponente (após a licitação).

Após a ordenação dos agentes, caso se constate a necessidade, o algoritmo passa a realizar as tentativas de encaminhamento da tarefa tomando como base a lista de agentes passada. Quando o encaminhamento é social, isto é feito através da criação de uma instância de diálogo *mestre-escravo*. Quando o encaminhamento é local, acontece um pedido direto para o sistema tutor. Quando uma cooperação é bem sucedida, decorrente de alguma das duas situações acima, o algoritmo finaliza a sua execução e retorna um sinal de sucesso para o algoritmo **cooperação.executa**. Quando acontece um insucesso, tal situação provoca uma notificação à *SEH*. O insucesso total, decorrente do encaminhamento mal sucedido com todos os agentes da lista, provoca a notificação de insucesso.

Antes da finalização do seu trabalho, uma instância de cooperação deve notificar a instância de coordenação, no algoritmo **cooperação.executa**, a ocorrência de uma dentre as duas situações seguintes: insucesso (conforme etapa 4) e sucesso (con-

¹⁹ O id. da tarefa é um valor numérico obtido do índice da tarefa na lista.

forme etapas 1 e 3).

Quando uma notificação de insucesso chega à instância de coordenação, ela aparece no formato da tupla abaixo.

`<id. da tarefa> <resultado da execução>`

Quando isto acontece, é possível que haja algumas cooperações em andamento. Neste caso, adota-se a estratégia²⁰ de esperar a finalização dessas tarefas. Tal estratégia tem a prerrogativa de ser prática, embora apresente a desvantagem de gerar um consumo desnecessário de recursos nos agentes escravos, que porventura estejam executando as tarefas em andamento. O algoritmo abaixo apresenta o tratamento da situação particular de insucesso no nível de coordenação.

```

coordenação.insucesso( id. da tarefa,
descrição do erro )
se ainda não houve insucesso
    insucesso = VERDADEIRO
    armazena a descrição do erro
    tarefa[id. da tarefa].condição =
                                                NAO EXECUTADA
fim-se
para cada tarefa[i] da lista
    se tarefa[i].condição = EM ANDAMENTO
        finaliza a execução do algoritmo
    fim-se
fim-para
coordenando = FALSO

```

O algoritmo **coordenação.insucesso** faz o tratamento adequado para viabilizar a estratégia de espera mencionada anteriormente. Daí, armazena a descrição do erro para servir de realimentação ao aprendiz e muda a condição da tarefa indicada para NAO EXECUTADA. A última operação efetuada é a verificação se há alguma tarefa em andamento. Caso não haja mais nenhuma nessa condição, o processo de coordenação é finalizando. Do contrário, ele continua a esperar as suas finalizações.

Quando uma notificação de sucesso chega à instância de coordenação, ela aparece no formato da tupla abaixo.

`<id. da tarefa> <descrição do erro>`

Neste momento, o algoritmo **coordenação.sucesso** é executado. Note que os tratamentos necessários para implementar a estratégia de espera estão colocados de forma análoga.

```

coordenação.sucesso( id. da tarefa,
resultado da execução )
se ainda não houve insucesso
    armazena o resultado da execução
    tarefa[id. da tarefa].condição = EXECUTADA
    ativa o algoritmo de disparo
fim-se
para cada tarefa[i] da lista
    se tarefa[i].condição = EM ANDAMENTO
        finaliza a execução do algoritmo
    fim-se
fim-para
coordenando = FALSO

```

Se não ocorreu nenhum insucesso anteriormente, o algoritmo **coordenação.sucesso** armazena o resultado da execução da

tarefa indicada, e muda a sua condição para EXECUTADA. Logo após, o algoritmo de disparo é invocado para instanciar, caso haja tarefas prontas para execução, novas cooperações que darão seguimento ao trabalho integrado dos módulos de coordenação e cooperação.

O funcionamento integrado dos dois módulos pode ser concluído devido a um dos dois motivos seguintes:

- (i) Quando houver algum insucesso de cooperação e todas as tarefas EM ANDAMENTO finalizarem a sua execução (estratégia de espera); neste caso, o que é repassado para o módulo Alocação é a descrição do erro que conduziu a esse resultado.
- (ii) Quando todas as tarefas forem executadas com sucesso; neste outro caso, o resultado é passado para o módulo Alocação através do seguinte formato:

`<tarefa1> <resultado1>, ...,`
`<tarefan> <resultadon>`

6.3 Tratamento dos Protocolos

No Capítulo 2, foram apresentados alguns aspectos ligados a protocolos de interação. No capítulo 5, discutiu-se o modelo Cooperação entre agentes concebido neste trabalho, o qual conduz à necessidade de estabelecimento de um conjunto de protocolos para disciplinar as atividades interativas envolvidas. Uma instância de diálogo é ativada no módulo *Protocolos*. O seu funcionamento é o objeto de discussão desta seção.

A existência simultânea de instâncias de diálogos é motivada pela concorrência inerente ao sistema. Cada instância é dotada de um contexto particular (Figura 6.3), útil para a sua execução coerente. Dentre as inúmeras possibilidades, algumas situações podem servir de exemplo, a saber: duas licitações referentes a duas tarefas paralelas, entrada de dois ATs simultaneamente na sociedade.

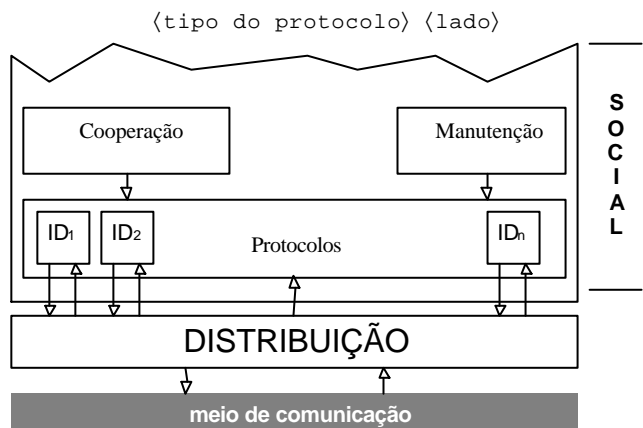


Figura 6.3: Instâncias de Diálogos

Para cada um dos protocolos preestabelecidos, existem duas implementações envolvidas: uma para o lado que inicia o protocolo e a outra para o lado que responde à requisição de interação. No primeiro caso, as instâncias de diálogo são sempre ativadas pelo módulo *Cooperação* ou pelo módulo *Manutenção Social*. Já no segundo, elas são ativadas através do módulo *Controle* após receber uma primeira mensagem enviada pela instância de diálogo que requisitou a interação. O formato do pedido de ativação é o seguinte:

O algoritmo utilizado pelo módulo *Protocolos* para ativar uma instância de diálogo é o seguinte:

²⁰ Em [UG 92] adota-se a estratégia de anulação das cooperações em andamento. Isto implica a necessidade de interações (*Goal Anulation*) com os agentes escravos com o objetivo de anular as execuções das tarefas em andamento.

```

protocolos.ativa( tipo do protocolo, lado )
se conseguir cria uma instância de diálogo
com base no
tipo de protocolo apropriado e no lado indi-
cado
    gera um id único associado à instância
    retorna o valor desse id.
fim-se
retorna um id inválido para sinalizar um er-
ro

```

O seu funcionamento é bastante simples: com base nas informações passadas para a ativação, é criada uma instância de diálogo adequada; caso a criação ocorra sem problemas, um identificador, gerado para a instância em questão, é retornado para a entidade que invocou a ativação. O algoritmo referente à implementação de cada protocolo, nos dois lados, é definido no Capítulo 7.

6.4. Tratamento das Mensagens

O tratamento das mensagens realizado no sistema de distribuição é explicado tomando como base os dois sentidos de uma mensagem: de cima para baixo e de baixo para cima, conforme ilustrado na Figura 6.4.

No sentido de cima para baixo, quando uma mensagem é enviada ao módulo *Controle*, este coloca na mensagem a informação sobre a instância de diálogo que lhe repassou a mensagem e a envia para o módulo *Comunicação*. Esse módulo, por sua vez, desempenha três atividades: coloca a informação sobre o agente que está enviando a mensagem, põe a mensagem na sintaxe da linguagem de interação padrão²¹ e efetiva o envio da mensagem para os agentes indicados como destinatários pela instância de diálogo.

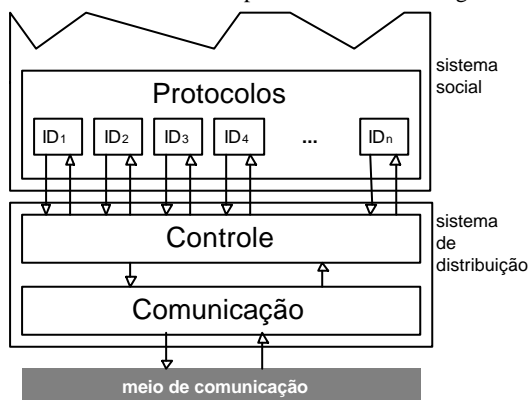


Figura 6.4: Arquitetura funcional do sistema de distribuição

No sentido de baixo para cima, quando uma mensagem é recebida, o módulo *Comunicação* verifica se a sintaxe está correta e nesse caso passa esta mensagem para o módulo *Controle*. Esse módulo, analisa para qual instância de diálogo a mensagem deve ser dirigida, conforme mostrado no algoritmo abaixo.

```

controle.trata_chegada( mensagem )
se a mensagem não tem instância de
diálogo destino determinada
    se conseguir criar uma instância de
    diálogo apropriado
        passa a mensagem para ser tratada por
    ela
retorna OK

```

```

fim-se
retorna PROBLEMA NA CRIAÇÃO
fim-se

para cada instância de diálogo[i]
    se mensagem.idprotdest = protocolo[i].id
        passa a mensagem para ser tratada por
    ela
        retorna OK
    fim-se
fim-para
retorna CHEGADA_ATRASADA

```

Esse algoritmo de controle trata a chegada de uma mensagem como segue. Se a mensagem não foi endereçada para nenhuma instância de diálogo específica (primeira interação), então é necessária a criação de uma instância, antes de repassar a mensagem para tratamento, construída a partir de informações contidas na própria mensagem. Se, por outro lado, a mensagem foi endereçada para uma instância de diálogo conhecida, verifica-se a sua existência antes de repassar a mensagem.

7. Interações entre Agentes Tutores

Neste capítulo, apresentam-se as linguagens utilizadas nas atividades interativas entre os agentes de *SATA*, enfatizando posteriormente os protocolos que são utilizados nas interações entre esses agentes. Nas seções 7.1 e 7.2, define-se formalmente, respectivamente, a linguagem social dos agentes e a linguagem de interação desses agentes, expondo o formato das interações juntamente com a linguagem adotada para expressá-las. Finalmente, na seção 7.3, descrevem-se os protocolos de cooperação e manutenção.

7.1 Linguagem Social

A linguagem social (Figura 7.1) tem o propósito de padronizar e precisar o vocabulário relativo aos elementos tratados no sistema Social, a fim de que eles possam ser corretamente operados nas situações de interação descritas acima. Existem duas situações de interação que justificam a existência de uma linguagem social: uma no nível interno, entre os sistemas Social e Tutor; e a outra no nível externo, entre os sistemas sociais de agentes distintos.

Com base na definição de autoconhecimento apresentada no Capítulo 5, descreve-se a seguir, através de uma *BNF*²², a gramática da linguagem de formalização deste componente, a fim de utilizá-la como parte da linguagem de interação entre agentes. Com esse propósito, descrevem-se também os recursos: *habilidade*, *tarefa*, *resposta* e *parecer*. Para tanto, foram usadas as estruturas sintáticas apresentadas na Figura 7.1.

Cada sub tarefa possui atributos extras que indicam as suas relações de dependência dentro da estrutura de decomposição.

Autoconhecimento: é utilizado no protocolo de entrada, onde os agentes se apresentam trocando os seus ACs, e quando o Sistema Tutor mapeia o conhecimento do tutor para o AC.

Habilidade: No processo de alocação, há um casamento (*matching*) entre as habilidades dos agentes com a habilidade necessária na realização da tarefa.

²¹ A sintaxe desta linguagem de interação é apresentada no Capítulo 7.

²² BNF vem de Backus-Naur Form, correspondendo a uma variante duma gramática livre de contexto

```

<autoconhecimento> ::= <id-do-agente> <habilidades>
<id-do-agente> ::= 'id' ':' '(' <id> ',' <apelido> ')'
<habilidades> ::= 'habilidades' ':' '(' <habilidade> ')' [ ',' '(' <habilidades> ')' ]
<id> ::= 'AT' '(' <natural*> ',' <natural*> [ ',' <natural*> ] ')'
<apelido> ::= cadeia com caracteres indicando um nome significativo para o agente

```

(a) BNF para o Autoconhecimento de um agente

```

<habilidade> ::= <tipo> <identificação> <representações>
<tipo> ::= 'tipo' ':' '(' <tipo-da-habilidade> ')'
<identificação> ::= 'identificação' ':' '(' <nomes> ')'
<representações> ::= 'representações' ':' '(' <representação> ')' [ ',' '(' <representações> ')' ]
<tipo-da-habilidade> ::= 'resolução' | 'instrução' | 'diagnóstico'
<nomes> ::= <nome> [ ',' <nome> ]
<representação> ::= 'entrada' ':' '(' <parâmetros> ')'
<parâmetros> ::= <parâmetro> [ ',' <parâmetro> ]
<parâmetro> ::= <tipo-do-parâmetro> <nome>
<tipo-do-parâmetro> ::= 'número' | 'texto'
<nome> ::= cadeia de caracteres

```

(b) BNF para descrever uma habilidade de um agente

```

<tarefa> ::= 'tarefa' ':' <subtarefa> [ ',' <subtarefas> ]
<subtarefa> ::= <nomesb> '(' <habilidade> ')' '(' <dados> ')'
<dados> ::= 'dados' ':' '(' <dado> [ ',' <dado> ] ')'
<dado> ::= <variavel> '=' <argumento>
<argumento> ::= <valor> | <string> | <referência>
<valor> ::= valor numérico
<string> ::= ''' cadeia de caracteres '''
<referência> ::= '(' <nomesb> <variavel> ')'

```

(c) BNF para descrição de uma tarefa

```

<resposta> ::= <sucesso> | <insucesso>
<sucesso> ::= 'resposta' ':' '(' <dado> [ ',' <dado> ] ')'
<insucesso> ::= 'erro' ':' <motivo>
<motivo> ::= cadeia de caracteres

```

(d) BNF para descrever o formato de uma resposta de um agente

```

<parecer> ::= 'parecer' ':' '(' <fator> ')'
<fator> ::= um número no intervalo 0 - 100

```

(e) BNF para descrever um parecer de um agente

Figura 7.1: Linguagem social

Tarefa: No protocolo de mestre-escravo. Quando o sistema Tutor submete um pedido de cooperação, a tarefa é mapeada para esse formato.

Resposta: No protocolo de mestre-escravo. Retorno da cooperação.

Parecer: Somente no protocolo de licitação.

7.2 Linguagem de Interação

As interações entre os agentes tutores em *SATA*, ocorrem utilizando-se um mecanismo de comunicação através de troca de mensagens em situações de diálogos. A estrutura das mensagens trocadas entre os agentes é fixa. Isso facilita a composição e a interpretação das mesmas.

Uma mensagem para interação, tal como ilustrado a seguir, é composta por três grandes componentes, a saber: um para distribuição, um para social (controle) e o outro para tutor (aplicação). Seu formato é o apresentado na Figura 7.2, sendo inicialmente inspirado nos trabalhos da equipe coordenada por Yves Demazeau, a exemplo do que consta em [Dem 95].

DISTRIBUICAO				SOCIAL		TUTOR	
TAM	REM	DEST	IDIPT REM	IDIPT DEST	PRO T	PRIM	CONTEÚDO

Figura 7.2: Formato de uma mensagem na interação.

Note que a figura está dividida em áreas com tons acinzentados. Cada área é composta por atributos, sendo esses voltados propósitos particulares. Os elementos indicados nas áreas, juntamente com seus atributos, são descritos a seguir:

- a) **DISTRIBUIÇÃO:** É composta por atributos de endereçamento do agente e identificação de diálogo, tanto para o emissor quanto para o receptor. Tais campos são úteis nos serviços de comunicação e controle do Sistema de Distribuição. Segue uma descrição de cada um deles:

Tamanho da Mensagem (TAM): Indica o tamanho da mensagem, para que o módulo Comunicação tenha controle sobre o tamanho das mensagens recebidas, e para todos os módulos acima que utilizem tal campo.

Remetente (REM): Deve ser preenchido com a identificação do agente tutor que remete a mensagem. Quem remete uma mensagem precisa se identificar, pois o destinatário geralmente precisa interagir de volta com ele.

Destinatário (DEST): Deve ser preenchido com a identificação do Agente Tutor destino. Este campo pode ser preenchido também com a palavra chave *TODOS*, nas situações de difusão, ou com a palavra *EXCETO* seguida de uma lista de nomes de agentes, sendo este preenchimento relativo respectivamente às situações de endereçamento *global* e endereçamento *complementar* mencionadas no modelo de comunicação visto no Capítulo 5.

Identificação da instância de diálogo do remetente (IDIPTREM): Deve ser preenchido com a identificação da instância de diálogo que remete a mensagem. Este atributo precisa ser identificado tomando como base a mesma justificativa utilizada para o atributo REM.

Identificação da instância de diálogo do destinatário (IDIPTDEST): Deve ser preenchido com o número identificador da instância de diálogo do destinatário ao qual a mensagem também pertence.

- b) **SOCIAL:** É composto por atributos úteis para a interpretação do conteúdo da mensagem dentro do diálogo.

Tipo do Protocolo (PROT): Deve ser preenchido com o tipo de protocolo utilizado na interação. Seu valor pode ser: MESTRE-ESCRAVO, LICITAÇÃO (Protocolos de Coope-

ração), ENTRADA ou SAIDA (Protocolos de Manutenção).

Primitiva (PRIM): Diz respeito aos tipos de interação entre agentes. Nesse sentido, **PRIM** é utilizado para contextualizar a interação dentro do protocolo. Observe a Tabela 7.1 com as primitivas utilizadas, estando cada uma com a sua respectiva semântica.

c) **TUTOR:**

Conteúdo da Aplicação (CONTEÚDO): O conteúdo da mensagem, dependendo da semântica, possui um formato especial com um significado particular. O tamanho desse conteúdo pode ser obtido a partir do tamanho da mensagem (TAM), retirando-se o espaço preenchido para os campos anteriores.

A BNF apresentada na Figura 7.3, descreve a sintaxe da linguagem de interação utilizada entre os agentes tutores.

PRIMITIVA	SEMÂNTICA
APRESENTAÇÃO	os agentes se apresentam
INFORMAÇÃO	um agente informa que está saindo
ANÚNCIO	anúncio da tarefa
PROPOSTA	proposta de resolução de tarefa
PERGUNTA	pedido de execução de uma tarefa
RESPOSTA	retorno de execução

Tabela 7.1: Primitivas de Interação

```

<interação> ::= <distribuição> <social>
               <tutor>
<distribuição> ::= <tamanho> <remetente>
                  <destinatário> <id-diálogo-remetente>
                  <id-diálogo-destinatário>
<tamanho> ::= 'tamanho' ':' '(' <número> ')'
<remetente> ::= 'remetente' ':' '(' <nome> ')'
<destinatário> ::= 'destinatário' ':' '('
                  <desc-destino> ')'
<desc-destino> ::= <nome-agente> | 'todos' |
                  <complementar>
<complementar> ::= 'exceto' <nome-agente> [ ','
                  <complementar> ]
<id-diálogo-remetente> ::= 'id_dlg_rem' ':' '('
                  <número> ')'
<id-diálogo-destinatário> ::= 'id_dlg_dest' ':'
                  '(' <número> ')'
<social> ::= <tipo-do-protocolo> <primitiva>
<tipo-do-protocolo> ::= 'protocolo' ':' '('
                  <nome-do-protocolo> ')'
<nome-do-protocolo> ::= 'licitação' | 'mestre-
                  escravo' | 'entrada' | 'saída'
<primitiva> ::= 'primitiva' ':' '('
                  <nome-da-primitiva> ')'
<nome-da-primitiva> ::= 'apresentação' |
                  'informação' | 'anuncio' | 'proposta' |
                  'pergunta' | 'resposta'
<nome-agente> ::= <string>
<tutor> ::= <conteúdo>
<conteúdo> ::= 'conteúdo' ':' cadeia de caracteres finalizada por
                  EOF

```

Figura 7.3: BNF da linguagem de Interação

7.3 Protocolos de interação

Os protocolos de interação regem as instâncias de diálogos responsáveis pelo tratamento das situações de interação entre os ATs na sociedade e destes com a SEH. No presente trabalho, tais protocolos estão classificados em duas categorias: protocolos de cooperação e protocolos de manutenção. A seguir, cada protocolo é descrito e ilustrado com o uso de figura, destacando-se apenas os módulos necessários aos propósitos imediatos, descrevendo-se sequencialmente os passos representados, em suas figuras associadas, por setas indexadas.

7.3.1 Protocolos de Cooperação

Os protocolos de cooperação regem as instâncias de diálogos que promovem as situações de cooperação. Com relação à cooperação, existem duas situações a considerar: *mestre-escravo* e *licitação*. Os protocolos de cooperação, no lado do agente que pede cooperação, são ativados pelo módulo *Coop* (através de alguma instância de cooperação)

Quando há vários agentes aptos a cooperar, provenientes do módulo *Coord* ou do processo de licitação com a sociedade, a questão fica reduzida a uma situação de mestre-escravo, devendo este ser ativado pelo módulo *Coop* após a eleição (através de um mecanismo de resolução de conflitos) de um agente, dentre os habilitados.

Mestre-Escravo

O Mestre-Escravo ocorre quando se sabe *a priori* com quem cooperar. Neste protocolo existem dois papéis: o do mestre (algo como contratante) e o do escravo (algo como contratado). Na Figura 7.4, ilustra-se uma situação de uso do protocolo Mestre-Escravo. O processo acontece como segue:

1. No lado do mestre, uma determinada instância de cooperação cria uma instância de diálogo, rotulada na Figura 7.4 por ID_1 , relativa à situação de mestre-escravo. Os parâmetros utilizados na criação são os seguintes: a TAREFA para a qual se deseja cooperação e a IDENTIFICAÇÃO do agente que deve funcionar como escravo.
2. O primeiro passo do protocolo, executado por ID_1 , é o envio de um **PEDIDO** para o agente que desempenhará o papel de escravo. A mensagem, antes de ser enviada, é preenchida com a TAREFA que foi passada como parâmetro na criação de ID_1 .
3. No agente escravo, uma outra instância, rotulada na mesma Figura 7.4 por ID_2 e responsável pelo tratamento da situação de mestre-escravo neste lado da interação, é criada após a recepção do **PEDIDO**. Logo após a criação, o sistema de distribuição repassa a mensagem recebida para ser tratada adequadamente por IP_2 .
4. O único tratamento realizado por ID_2 , depois da recepção da mensagem, é o encaminhamento da TAREFA para o sistema Tutor.
5. O sistema Tutor tenta executar a TAREFA. O PRODUTO obtido da execução é retornado para ID_2 , podendo este ser tanto o RESULTADO requerido pelo lado mestre, quanto a descrição de algum ERRO que porventura tenha acontecido durante o processamento.
6. ID_2 envia uma **RESPOSTA** para o agente mestre. A mensagem enviada é preenchida com o PRODUTO obtido da exe-

cução da tarefa pelo sistema Tutor. Após o envio, ID_2 finaliza a sua execução.

7. No agente mestre, ID_1 recebe a **RESPOSTA** enviada pelo agente escravo através de ID_2 .
8. ID_1 encaminha para a instância de cooperação apropriada, aquela que a criou, o PRODUTO da cooperação e finaliza a sua execução. Neste ponto, a instância de cooperação reassume o controle e continua o seu processamento.

Licitação

A Licitação é aqui definida como um mecanismo que visa estabelecer uma interação dialógica de um agente com um grupo de agentes. Tal mecanismo é utilizado quando não se sabe *a priori* um agente com quem se possa cooperar. Portanto, uma Licitação acontece quando o conjunto de agentes informado pelo módulo *Coord* para a tarefa em questão é vazio ou ainda quando há um insucesso numa situação em que o agente alocado não corresponde às expectativas indicadas pelo CS do agente que o invocou (uma tal situação está prevista, pois assume-se neste trabalho que o CS pode não estar correto e/ou completo).

Como se pode observar no esquema da Figura 7.5, no protocolo de Licitação existem dois papéis: o de gerente, que inicia a licitação e o de subordinado, que é o contratado pelo gerente. O processo acontece como segue:

1. No lado gerente, uma determinada instância de cooperação cria uma instância de diálogo, rotulada na Figura 7.5 por ID_1 , relativa a situação de licitação. Os parâmetros utilizados na criação são: a TAREFA para a qual se quer cooperação e uma lista com as IDENTIFICAÇÕES dos agentes que devem ser utilizados na licitação (*multicast*), caso esta lista seja vazia, todos os agentes da sociedade devem ser contatados (*broadcast*);
2. Aqui, o primeiro passo do protocolo, executado por ID_1 , é o envio de um **ANÚNCIO** para os agentes necessários. A mensagem, antes de ser enviada, é preenchida com a TAREFA em questão.
3. Em todos os agentes destinatários do **ANÚNCIO**, uma instância de diálogo (ID_2 , ..., ID_n , conforme Figura 7.5), relativa ao tratamento da situação de licitação no lado subordinado, é criada após a recepção da mensagem. Logo após a criação, em cada agente, o sistema de distribuição repassa o **ANÚNCIO** recebido para ser tratado pela ID_i recém criada.
4. No intuito de analisar a sua própria competência sobre a

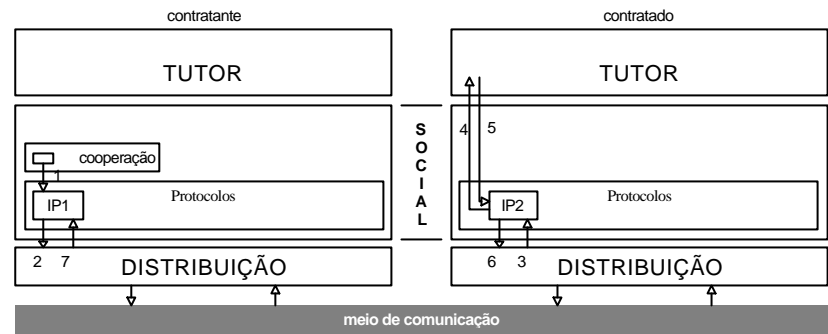


Figura 7.4: Protocolo Mestre-Escravo

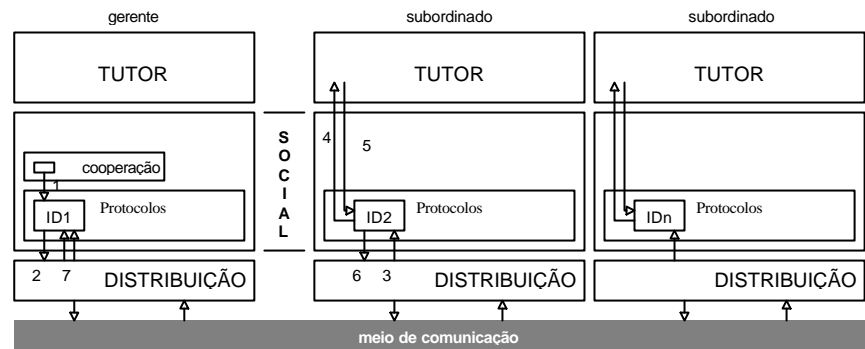


Figura 7.5: Protocolo de Licitação



Figura 7.6: Protocolo de Entrada

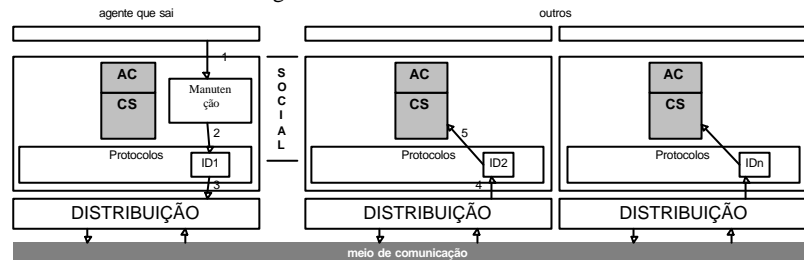


Figura 7.7: Protocolo de Saída

TAREFA anunciada, cada ID_i extrai a TAREFA inserida na mensagem para então encaminhá-la ao sistema Tutor.

5. O sistema tutor analisa a **TAREFA** e retorna um **PARECER**. Este parecer, é um valor percentual que indica a porção da **TAREFA** que pode ser executada através do agente. Quando o agente pode executar a **TAREFA** completamente, o valor é 100; quando não pode realizar nada, o valor é 0. Um valor entre 0 e 100 indica uma competência parcial e a existência de uma decomposição durante o processo de análise. Quando isto acontece, o **PARECER** é obtido através da fórmula abaixo:
- $$parecer = \frac{n^{\circ} \text{tarefas alocadas com sucesso}}{n^{\circ} \text{tarefas resultantes da decomposição}} * 100$$
6. Se o **PARECER** obtido no processo de análise for maior que zero, o ID_i do agente subordinado envia para o agente gerente uma **PROPOSTA** com este valor. Após isso, cada ID_i fina-

liza a sua execução.

7. Após o envio do ANÚNCIO inicial, o gerente, através de ID_1 , passa a receber as **PROPOSTAS** enviadas pelos agentes voluntários (que se manifestaram aptos a cooperar). A IDENTIFICAÇÃO destes agentes e o PARECER dado por cada um são inseridos numa lista destinada à catalogação das propostas. O gerente continua a receber as PROPOSTAS, durante um determinado intervalo de tempo.
8. Quando o tempo estabelecido para a recepção expira, ID_1 deixa de receber as **PROPOSTAS** e encaminha para a instância de cooperação apropriada, a lista dos agentes que se propuseram a cooperar. Com isso, ID_1 finaliza a sua execução e a instância de cooperação continua com o seu processamento.

7.3.2 Protocolos de Manutenção

Os protocolos de manutenção regem as instâncias de diálogos que promovem as situações de manutenção. A manutenção é necessária, pois é desejável que todos os agentes estejam sempre com os seus CSs consistentes. Três situações devem ser tratadas: entrada, saída e atualização de um agente. Tanto na entrada quanto na saída de um agente é necessário que toda a sociedade seja notificada. Os protocolos de manutenção são ativados pelo módulo de Manutenção Social.

Entrada

O processo de entrada estabelece uma interação dialógica de um agente com a sociedade. Na Figura 7.6, durante o processo de entrada acontece uma troca de ACs entre o agente que está entrando e o restante da sociedade. O protocolo de Entrada deve ser o primeiro a ser executado num agente e todos os pedidos de interação devem ser recusados antes da finalização da entrada. Basicamente, a situação consiste no seguinte:

1. Quando um novo agente entra na sociedade, o sistema Tutor pede ao sistema Social, através do módulo Manutenção, que inicie o protocolo de entrada;
2. O módulo *MS* cria uma instância de diálogo, rotulada por ID_1 , conforme Figura 7.6, responsável pelo tratamento da situação no lado do agente que está entrando. O AC e uma referência para o CS, são passados como parâmetros na criação de ID_1 .
3. ID_1 prepara uma mensagem para enviar a sociedade. O AC passado como parâmetro na criação é colocado no campo o envio APLICAÇÃO da mensagem.
4. ID_1 envia a mensagem preparada, uma **APRESENTAÇÃO**, para todos os agentes da sociedade.
5. Em todos os agentes da sociedade, uma instância de diálogo (ID_2, \dots, ID_n , conforme Figura 7.6), relativa ao tratamento da situação de entrada nos agentes que já estão na sociedade, é criada após a recepção da mensagem. Logo após a criação, em cada agente, o sistema de distribuição repassa a APRESENTAÇÃO recebida para ser tratada pela ID_i , recém-criada.
6. Cada agente, através do seu ID_i , cadastra em seu CS, o AC do agente que está entrando e que foi colocado na APRESENTAÇÃO recebida.
7. Na seqüência, o próximo passo executado por cada ID_i , é a preparação de uma mensagem que é enviada de volta à sociedade. O AC do agente em questão é colocado no campo A-

PLICAÇÃO da mensagem.

8. Envia-lo ao agente que entra (**APRESENTAÇÃO**);
9. Após o envio da **APRESENTAÇÃO** inicial, o agente que entra, através de ID_1 , passa a receber as **APRESENTAÇÕES** enviadas pelos agentes da sociedade. O AC dos agentes são cadastrados em seu CS. O agente continua a receber as APRESENTAÇÕES durante um determinado intervalo de tempo.
10. Cadastra-os em seu CS.

Saída

O processo de saída estabelece também uma interação dialógica de um agente com a sociedade. O agente que sai deve avisar a todos da sua saída. A interação de saída tem uma característica particular. Uma instância de diálogo sua, só pode ser executada de forma exclusiva, ou seja, quando nenhuma outra estiver “rodando”. Caso alguma instância de diálogo esteja executando é necessário esperar a sua finalização. Durante a execução da interação de saída, todos os pedidos de criação de uma nova instância devem ser recusados. Um outro detalhe, após a finalização da instância de diálogo de saída, nenhuma outra instância deve ser executada no agente. O processo acontece tal como se encontra ilustrado na Figura 7.7:

1. Quando um agente precisa sair da sociedade, antes de se efetuar a saída real do agente, o sistema Tutor pede ao sistema Social, através do módulo Manutenção, que inicie o protocolo de saída;
2. O módulo *MS* cria uma instância de diálogo, rotulada na Figura 7.7 por ID_i , nas condições necessárias para a sua execução.
3. O primeiro passo efetuado por ID_i , é o envio de uma **INFORMAÇÃO** para todos os agentes da sociedade. Esta **INFORMAÇÃO** serve para sinalizar a sua saída.
4. Em todos os agentes da sociedade, uma instância de diálogo (ID_2, \dots, ID_n , conforme Figura 7.7), relativa ao tratamento da situação de saída nos agentes que permanecerem na sociedade, é criada após a recepção da mensagem. Logo após a criação, em cada agente, o sistema de distribuição repassa a **INFORMAÇÃO** recebida para ser tratada pela ID_i , recém-criada.
5. O único tratamento feito pela ID_i , é o descadastramento do autoconhecimento do agente que está saindo. Este descadastramento é feito através da remoção do CS do AC relativo ao agente identificado no campo remetente da mensagem recebida.

Atualização

A situação de atualização de um agente tutor é caracterizada neste trabalho através de uma saída seguida de uma reentrada. Devido a isso, a sua operacionalização é obtida tirando-se proveito dos passos descritos para os dois protocolos (entrada e saída) descritos anteriormente.

7.3.3 Síntese das interações ao nível dos protocolos

De forma resumida, a Tabela 7.2 mostra o preenchimento das interações nas situações existentes nos quatro protocolos apresentados, a saber: Entrada, Saída, Mestre-Escravo e Licitação.

	Id. destinatário	Protocolo	Primitiva	Aplicação
1	-	cooperação	pergunta	Tarefa
2	-	cooperação	resposta	Erro
3	-	cooperação	resposta	Resposta
4	GRUPO	cooperação	anúncio	Habilidade
5	-	cooperação	proposta	Parecer

	Id. destinatário	Protocolo	Primitiva	Aplicação
1	TODOS	manutenção	apresentar	Descrição do agente
2	TODOS	manutenção	informar	Id. do agente
3	-	manutenção	apresentação	Descrição do agente

Tabela 7.2: Resumo das interações utilizadas

7.3.4 Modelagem dos Protocolos

Apresentam-se nas Figuras 7.8 e 7.9, os autômatos (diagramas de estado) que modelam os protocolos de cooperação (Licitação e Mestre-Escravo) e de manutenção (Entrada e Saída). Aqui, uma transição representa uma interação ou um processo interno.

8. Interações entre um aprendiz e um agente tutor

Neste capítulo, apresenta-se o componente sistema tutor de um agente, definindo-o, caracterizando-o, e, sobretudo, especificando as suas interações com o agente aprendiz humano. Inicia-se, entretanto, por uma discussão sobre o modelo de ensino e aprendizagem adotado como referência para orientar a interação Tutor-Aprendiz. Prosseguindo-se, discute-se o modelo de interação entre um Aprendiz e um Agente Tutor, e mais a arquitetura de um Sistema Tutor. Finalmente, estende-se a discussão para a interação entre um Aprendiz e a Sociedade de Agentes Tutores Artificiais.

8.1 Modelo de Ensino-Aprendizagem

O suporte teórico na concepção do ambiente MATHEMA, no que diz respeito às suas atividades pedagógicas, inspira-se numa abordagem construtivista [Pia 77], possibilitando ao aprendiz uma interação com o sistema tutor que permita a sua aquisição/construção do conhecimento. Aliada à abordagem construtivista há também uma inspiração na teoria de Vigotsky [Vyg 91], no que se refere principalmente aos aspectos sociais envolvidos no processo de interação. A idéia é a de habilitar o Aprendiz a adquirir conhecimento como resultado de sua interação com o Sistema Tutor. Isto, valorizando uma postura e participação ativa do Aprendiz, proporcionando-lhe meios para a construção do seu próprio conhecimento.

Nesta perspectiva, adotou-se aqui um modelo de ensino-aprendizagem cooperativo para dar suporte ao processo de aquisição de conhecimento pelo aprendiz humano. Este modelo combina aspectos das modalidades de aprendizagem: aprendizagem pela ação (*learning by doing*) e aprendizagem por instrução (*learning by being told*) [Cos 91]. A primeira considera uma abordagem enfatizando características exploratórias em situações de resolução de problemas, conduzindo a uma forma de aprendizagem por descoberta a partir da ação, sendo inspirada, de um certo modo, na mencionada categoria Micromundo. Por outro lado, a segunda modalidade focaliza sobre aspectos mais instrucionais provenientes dos STIs clássicos, operando sob uma forma de aprendizagem através de instrução.

Na combinação mencionada acima, a aprendizagem passa a ser decorrente do processo de interação entre as entidades envolvi-

das, tomando a atividade de resolução de problemas como básica. Isso porque é a partir dessa atividade, que se assume aqui, que outras atividades passam a emergir. A idéia aqui é, portanto, o sistema tutor engajar o aprendiz em situações de resolução de problema. Isso deverá ocorrer de acordo com um modelo cooperativo de ensino-aprendizagem, no qual tanto o sistema tutor quanto o aprendiz trabalham com o mesmo objetivo, qual seja: a aquisição de conhecimento do aprendiz. Trata-se, portanto, da aprendizagem viabilizada através da integração de resolução de problema e um modelo de aprendizagem cooperativa.

A atividade de resolução de problemas ocorre segundo um dos dois modos: o aprendiz resolve problemas colocados pelo sistema ou o contrário. No primeiro modo, o sistema tutor, mediante um certo critério, escolhe e apresenta um problema ao aprendiz. Nisso, o aprendiz tenta resolvê-lo, podendo resultar em situações bem distintas, desde solucionar o problema completamente até não apresentar nenhuma reação. Em qualquer caso, o sistema acompanhará o processo e dará o *feedback* necessário, em meio a uma interação dialógica. No segundo modo, o aprendiz coloca um problema para o sistema tutor. Daí, o sistema procura resolver e, se for demandado, apresenta explicações sobre a sua solução.

8.2 Modelo de interação

Com este modelo, pretende-se descrever as categorias de situações de interações pedagógicas que ocorrem entre o Sistema Tutor e o Aprendiz Humano.

O Sistema Tutor, conforme colocado anteriormente, é responsável pelas atividades de interações tutoriais adaptativas, tomando por base uma aprendizagem calcada em resolução de problemas, nos termos já declarados. É através desse sistema que um Agente Tutor interage com o Aprendiz. A situação mais comum é a do Aprendiz ser colocado numa situação de resolução de problemas em interação com o Agente Tutor. Nesse processo, o Agente Tutor acompanha o Aprendiz durante a interação, monitorando suas ações e provendo-lhe uma realimentação (*feedback*) adequada.

Os tipos de interações entre o Aprendiz e o Agente Tutor, incluem aspectos didáticos, através de atividades pedagógicas vistas num esquema de troca de mensagens, onde o Agente Tutor envia ao Aprendiz uma mensagem com conteúdo *x* e este retorna uma outra com conteúdo *y*, definidas assim:

```

<X> ::= <problema> |
        <explicação ou justificativa> |
        <crítica> |
        <solução> |
        <suprimento de conhecimento:
        help, dicas, instrução...> |
        <controle>

<Y> ::= <solução> |
        <consulta> |
        <argumentação> |
        <problema> |
        <justificativa>

```

Os tipos de mensagens envolvidas são as seguintes: Problema, Solução, Justificativas, Crítica, Consulta, Suprimento de Conhecimento e Controle. Daí, as intervenções tanto do Aprendiz quanto do Sistema Tutor ocorrem de acordo com o conjunto de mensagens disponíveis para cada um deles.

As interações pedagógicas e dinâmicas entre o Aprendiz e o Sistema Tutor são inicializadas através de atividades de resolução de

problemas. Isso significa que o sistema apresenta, inicialmente, um conjunto de problemas para o Aprendiz solucionar. É, portanto, a partir do processo de solução que podem emergir outras atividades, a depender do retorno do Aprendiz com respeito a um problema que lhe foi posto. Esse retorno do Aprendiz é dado, portanto, com base em um dos tipos de mensagens indicados acima. Consequentemente, as intervenções por parte do Sistema Tutor são elaboradas e realizadas com base naquilo que o Aprendiz lhe apresenta e também baseadas nos seus tipos de mensagens, já apresentados. As intervenções do Sistema Tutor ocorrem tanto por demanda do Aprendiz, quanto por iniciativa sua, visando explorar situações pertinentes em relação ao sucesso do Aprendiz.

8.3 Arquitetura de um sistema tutor

Para dar suporte às interações pedagógicas referidas acima, foi definida uma arquitetura para o Sistema Tutor em apreço, tal como representada na Figura 8.1.

O Sistema Tutor está definido formalmente a seguir e as funcionalidades dos seus componentes estão descritas logo em seguida. Antes, porém, convém destacar o Aprendiz, que representa um papel desempenhado por um humano, em interação com o Sistema Tutor através do módulo *Interface*. A *Interface* provê, além do serviço de comunicação, informações acerca dos recursos disponíveis na Sociedade de Agentes Tutores Artificiais, mais especificamente, em cada um dos Sistemas Tutores que compõem os agentes.

DEFINIÇÃO 8.1: Um **Sistema Tutor** é definido pela tripla seguinte: $ST = \langle Med, Raciocinadores, Bases de Conhecimento \rangle$, onde *Med* é o módulo mediador que corresponde a um mecanismo pedagógico, definido para reagir a cada ação *Y* vinda do aprendiz, devolvendo-lhe uma reação com conteúdo *X*, assim:

$$\begin{aligned} med: Y &\rightarrow X \\ y &\mapsto x = med(y), \text{ sendo descrito a seguir.} \end{aligned}$$

Raciocinadores representam os mecanismos provedores de funções pedagógicas pertencentes ao conjunto das intervenções, tais como: resolução de problemas, diagnóstico cognitivo, explicação, instrução e **Bases de Conhecimento** é o módulo responsável por manter os conhecimentos de suporte à operacionalização dos raciocinadores.

8.3.1 Componentes do Sistema Tutor

Apresenta-se a seguir a descrição de cada um dos componentes do Sistema Tutor, indicados na Figura 8.1, enfatizando o papel que cada um deles desempenha.

8.3.1.1 Bases de Conhecimento

Neste módulo estão incluídos os conhecimentos relacionados à estrutura pedagógica (*curriculum*), ao domínio em si e ao Aprendiz.

Cada agente tutor, conforme foi apresentado no Capítulo 4, possui um domínio de conhecimento *d*. Este domínio é olhado internamente como constituído por um conjunto de unidades pedagógicas, definidas em função de objetivos de ensino/aprendizagem específicos que estão associados a um *curriculum*.

DEFINIÇÃO 8.2: Cada um dos domínios *d* é definido de acordo com a seguinte estrutura pedagógica:

- (i) ao domínio de conhecimento *d*, associa-se um *curriculum*;

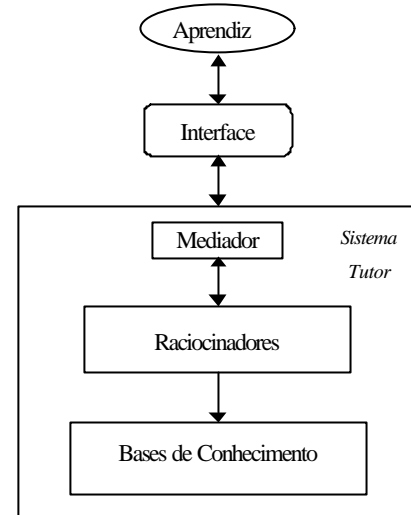


Figura 8.1: Arquitetura de um Sistema Tutor

- (ii) ao *curriculum*, associa-se um conjunto de unidades pedagógicas, dispostas segundo uma ordem definida e relacionada a pré-requisito e nível de dificuldade;
- (iii) cada unidade pedagógica é constituída por um conjunto de problemas;
- (iv) a cada problema está associado um conjunto de unidades de conhecimento que servem de apoio à solução deste problema.
- (v) a cada unidade de conhecimento estão associados conhecimentos de suporte à resolução de problemas, tais como: conceitos, resultados e exemplos (que são as particularidades dos resultados), contra-exemplos, dicas, problemas análogos (da mesma classe).

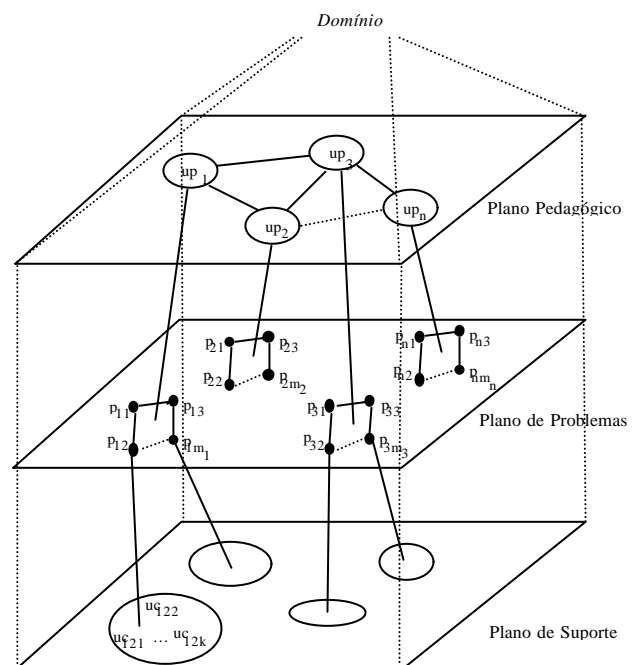


Figura 8.2: Estrutura pedagógica do conhecimento do domínio

Para ilustrar a estrutura pedagógica do domínio, observe a Figura 8.2. Nela, percebem-se três planos fundamentais, que são aqui denominados de plano pedagógico, plano de problemas e plano de suporte.

No plano pedagógico (plano superior) estão as unidades pedagógicas relacionadas por uma ordem definida em função de pré-

requisitos e níveis de dificuldades, servindo para indicar uma ordem de execução das atividades pedagógicas.

No plano de problemas (plano intermediário) encontram-se conjuntos de problemas, estando cada um deles associado a uma unidade pedagógica, servindo como atividade pedagógica básica no processo de ensino-aprendizagem. Estes problemas também estão relacionados por uma ordem parcial, definida de acordo com o nível de dificuldade envolvido em suas resoluções.

Finalmente, no plano de suporte (plano inferior) acha-se definido o conhecimento de apoio à resolução dos problemas. Desse modo, tal como já mencionado, incluem-se: conceitos, resultados, exemplos (que são as particularidades dos resultados), contra-exemplos, dicas, problemas análogos (da mesma classe), catálogo de erros e maus entendimentos. Além disso, há os problemas resolvidos e os problemas a serem resolvidos. Quanto aos resolvidos, a cada um deles existe um conjunto de soluções, constando de caminhos diferentes.

No que toca ao conhecimento do domínio em si, estão duas bases de conhecimento importantes: uma sobre o conhecimento assumido pelo sistema tutor como correto (*BCO*) e a outra sobre o conhecimento assumido como ruidoso (*BCR*). Por último, há uma base de conhecimento construída dinamicamente sobre o Aprendiz (o modelo do aprendiz), servindo para armazenar o seu estado cognitivo.

8.3.1.2 Módulo Raciocinadores

Trata-se de um grande módulo que representa e envolve os mecanismos provedores das principais funções pedagógicas, tais como: resolução de problemas, diagnóstico cognitivo, explicação, instrução. Essas funções são conceitualmente desempenhadas por três sub-módulos: Especialista, Tutor e Modelagem do Aprendiz.

Especialista

É um sistema especialista dotado de componentes que desempenham papéis que incluem: resolução de problemas, explicações, diagnóstico cognitivo e remediação. O mecanismo **Resolvedor de Problemas** (RP) resolve tanto os problemas que o sistema coloca para o aprendiz, quanto os problemas que o aprendiz lhe propõe. As soluções apresentadas pelo RP podem ser justificadas para o aprendiz através de um mecanismo que provê explicações (**Explicador**). Finalmente, o módulo **Diagnóstico Cognitivo** analisa as soluções do Aprendiz, identificando os acertos, erros e os maus entendimentos apresentados por este durante o processo de resolução de problemas. Ademais, há o módulo **Remediação** que é responsável pelo provimento de *feedback* ao Aprendiz, em função dos resultados oferecidos pelo módulo Diagnóstico.

Tutor

Este módulo é responsável pela seleção de recursos pedagógicos, escolhidos sobre o *currículum*. A seleção dos recursos pedagógicos levam em conta o contexto da interação, o modelo do aprendiz e o modelo do domínio. Além disso, é também encarregado da execução de atividades instrucionais.

Modelagem do aprendiz

Cada agente possui um módulo constando dos elementos necessários à realização de uma modelagem local do aprendiz, sendo este definido de acordo com o domínio de conhecimento em questão. Assim, ele é responsável por construir e manter o mo-

delo do Aprendiz. Para isso, conta diretamente com resultados do módulo de diagnóstico cognitivo, já descrito, que atua segundo os limites de sua percepção do conhecimento em pauta. Cabe a este, a identificação dos acertos e conhecimentos adquiridos, erros e os maus entendimentos apresentados pelo aprendiz durante o processo de resolução de problemas.

A abordagem adotada para modelagem do Aprendiz é, como já mencionado, focalizada sobre a estrutura do domínio. Assim, o modelo do Aprendiz é construído a partir de uma combinação dos modelos de sobreposição (*overlay*) e de perturbação, com a vantagem e a diferença, em relação a uma abordagem clássica, de tratar-se de conhecimento que pode evoluir com o desenvolvimento das interações entre entidades principais do sistema. Essa abordagem aqui definida conta com a base de conhecimento assumida como correta para trabalhar a idéia de sobreposição e com uma base de conhecimentos ruidosos para trabalhar a questão da perturbação.

Como se trata de um modelo baseado no domínio, pode-se, com efeito, identificar uma relação de conjuntos [Sel 94], onde: o componente de sobreposição do modelo é um subconjunto do domínio representado, no caso, BCO. Enquanto que o componente de perturbação é um subconjunto de BCR. Formalmente, o modelo do Aprendiz com respeito à sobreposição e à perturbação, pode ser definido separadamente, respectivamente, pelos mecanismos *sob* e *pert*, como seguem:

DEFINIÇÃO 8.3: O componente do modelo do Aprendiz relativo à sobreposição (*overlay*) é definido pelo mecanismo seguinte:

$$\begin{aligned} \text{sob} : \text{BCO} &\rightarrow \text{MAO} \\ d &\mapsto \text{mao} = \text{sob}(d), \end{aligned}$$

onde BCO = Base de Conhecimento Oracular e MAO = Componente do modelo relativo a BCO.

DEFINIÇÃO 8.4: O componente do modelo do Aprendiz relativo à perturbação é definido pelo mecanismo seguinte:

$$\begin{aligned} \text{pert} : \text{BCR} &\rightarrow \text{MAR} \\ d &\mapsto \text{mar} = \text{pert}(d), \end{aligned}$$

Assim, o modelo do aprendiz resultante é definido pelo mecanismo *map*, como segue:

DEFINIÇÃO 8.5: O modelo do Aprendiz é definido pelo mecanismo seguinte:

$$\begin{aligned} \text{map} : \quad D &\rightarrow \text{BCA} \\ d &\mapsto \text{ma} = \text{map}(d), \end{aligned}$$

onde $D = \text{BCO} \dot{\cup} \text{BCR}$ e BCA = Base de Conhecimento com o Modelo do Aprendiz.

8.3.1.3 Módulo Mediador

Este módulo é encarregado do controle geral da execução das funções pedagógicas no Sistema Tutor, ao nível dos Raciocinadores. Ele é responsável por realizar interpretações sobre as ações do Aprendiz. Ele desempenha o papel de decidir sobre o tipo de intervenção a ser realizada para atender ao Aprendiz. O Mediador possui conhecimento sobre os Raciocinadores e sobre as Bases de Conhecimento. No mais, cumpre a este módulo o papel de interação com o Sistema Social do Agente.

8.4 Sobre o funcionamento básico do sistema tutor

A seguir, apresenta-se um algoritmo geral para a atividade de

resolução de problemas e seus desdobramentos, destacando-se a atividade de controle desempenhado por um agente tutor.

8.5 O sistema tutor multi-agentes

Nesse ponto, discute-se o Sistema Tutor Multi-Agentes definido por uma Sociedade de Agentes Tutores Artificiais que permite realizar uma espécie comportamento inteligente de forma distribuída. Trata-se da existência e engajamento de um grupo de agentes nas atividades de resolução de problemas visando favorecer a aprendizagem de um aprendiz humano. A idéia com esta sociedade e o ferramental nela disponível é poder facilitar a colaboração com o Aprendiz durante o processo de realização de uma tarefa pedagógica.

Como já foi mencionado no Capítulo 4, a partir de considerações sobre o modelo do domínio, com vistas a melhorar o processo de modelagem do aprendiz e de interação Tutor-Aprendiz, definiu-se um sistema tutor multi-agentes. Um tal sistema, desempenha, como já foi discutido, o papel de uma sociedade de agentes tutores artificiais. Sobre este sistema, são feitos a seguir, alguns comentários relacionados aos mecanismos encarregados das funções pedagógicas mencionadas anteriormente, desempenhadas, por exemplo, pelos módulos Especialista, Tutor e Modelagem do Aprendiz. Portanto, tais funções são agora tratadas com a possibilidade de serem realizadas de maneira distribuída.

A resolução distribuída de tarefas pedagógicas, a exemplo de solução de problemas, diagnóstico cognitivo distribuído, explicação e instrução, está relacionada com o desempenho das atividades de interação cooperativa envolvendo a plataforma multi-agentes, tal como foi definida através dos recursos apresentados nos Capítulos 5, 6 e 7. Naquela ocasião, apresentaram-se todos os recursos que permitem aos agentes da sociedade trabalharem cooperativamente. Portanto, são estes recursos que dão a funcionalidade do Sistema Tutor Multi-agentes, aqui referido. Nesse ponto convém dizer o seguinte: cada agente tutor possui um sistema tutor inteligente associado e este sistema é responsável por interagir diretamente com o Aprendiz, promovendo atividades pedagógicas. Entretanto, cada agente pode precisar envolver outros agentes, utilizando-se do seu conhecimento social, para atender aos seus objetivos locais e, assim o faz, devido à existência do sistema tutor multi-agentes. Enfim, como já foi definido no Capítulo 4, cada agente é dotado de um pedaço de conhecimento sobre o domínio alvo, sobre o qual é especializado e, com base neste, desempenha as suas funções pedagógicas. Portanto, é a capacidade de integração destes agentes que permite ao sistema interpretar e tratar de forma adequada às ações do Aprendiz.

8.5.1 O modelo distribuído do domínio

O modelo do domínio é um dos módulos mais críticos de um STI, sendo considerado a base para se buscar melhorias no processo de interação entre um Aprendiz e um Tutor. Foi com esta perspectiva, que se investiu e se chegou à definição do já apresentado modelo multidimensional do domínio de conhecimento, constituindo-se na sua representação externa. Isso conduziu à possibilidade de definir um domínio de conhecimento por vários pedaços de conhecimento obtidos disciplinadamente, sendo cada um desses pedaços representados e tratados por um agente tutor que lhe é associado. Daí, as consequências mais imediatas ocorrem no sentido de assegurar relações e cálculos de dependência, interessantes, entre os agentes, servindo para promover o desenvolvimento de atividades cooperativas.

Esse modelo global do domínio, através de sua estruturação e das representações pelos agentes de suas bases de conhecimentos, tem permitido aos módulos Raciocinadores dos agentes a realização de tarefas pedagógicas, de maneira distribuída, tais como: resolução de problemas, explicações, diagnóstico cognitivo e instrução (ensino, dicas, etc.). Isso porque o conhecimento passou a ser melhor elaborado e distribuído entre diversos agentes especializados e com capacidades para cooperarem entre si. Isso tudo é viabilizado com a plataforma de apoio à cooperação e comunicação de agentes, apresentada nos capítulos anteriores. Além disso, convém destacar um outro resultado fundamental obtido nesta pesquisa e facilitado pela estruturação do domínio, com respeito à viabilização das atividades cooperativas. Trata-se da operação de decomposição de tarefas. Para viabilizar tal operação, definiu-se, para descrição de tarefas nos domínios, um arcabouço de uma linguagem algébrica, na qual a base de sua gramática é: $\langle \text{termo operador termo} \rangle$, onde as noções de termo e operador coincidem com as da lógica clássica. Um exemplo direto de um domínio onde suas tarefas são naturalmente expressas por essa linguagem é o de equações, onde o termo é uma expressão algébrica e o operador é o de igualdade ($=$). Sobre esse domínio, fez-se um experimento com essa linguagem em relação à descrição e à operação de decomposição de suas tarefas. Sem dúvidas, a exigência de que tarefas nos domínios sejam descritas por uma tal linguagem é uma restrição considerável, uma vez que não são muitos os domínios com suas tarefas passíveis de serem descritas por esta linguagem.

Neste cenário, o já definido arcabouço para interação entre agentes munido dos mecanismos de decomposição de tarefas e de recombinação dos resultados obtidos, habilita os módulos Raciocinadores presentes nos agentes a desempenharem suas atividades pedagógicas de forma distribuída. Assim, por exemplo, asseguram-se a distribuição das atividades: resolução de problemas, explicação, diagnóstico cognitivo e instrução. Isso tudo, facilita a colaboração do Sistema Tutor Multi-agentes com o Aprendiz durante o processo de realização de uma dada tarefa pedagógica.

8.5.2 A modelagem distribuída do aprendiz

A modelagem do aprendiz em relação ao domínio alvo é construída a partir dos resultados obtidos pela participação de vários agentes operando de maneira distribuída. Portanto, o modelo do Aprendiz construído passa a ser existir distribuído nos agentes tutores. Trata-se, desse modo, de um modelo com representação distribuída.

A construção do modelo do Aprendiz, agora distribuído, é realizada através da construção individual de cada agente, com base na combinação definida a partir das noções contidas no modelo de sobreposição (vide def. 8.3) e no modelo de perturbação (vide def. 8.4), resultando no modelo map (vide def. 8.5), tal como discutido anteriormente. Para a realização dessa construção, há um vínculo entre o diagnóstico cognitivo distribuído e os mecanismos *maps* distribuídos. Na verdade, o distribuído é para significar a combinação dos vínculos locais diagnóstico-map, em cada agente. Isso subsidia a representação global do modelo do Aprendiz.

O processo do diagnóstico cognitivo distribuído ocorre de acordo com a funcionalidade já apresentada para os agentes tutores diante da execução de uma tarefa pedagógica. Assim, o funcionamento básico é o seguinte: o Sistema Tutor encaminha um problema

escolhido criteriosamente para o Aprendiz. Este último, por sua vez, reage devolvendo, por exemplo uma solução. Assim sendo, o módulo de diagnóstico de um agente tutor avalia esta solução, podendo recair num das três situações já discutidas: ele sabe fornecer o diagnóstico por si só, não o sabe sozinho, mas conhece alguns agentes com quem possa cooperar ou por fim, não conhece ninguém na sociedade que possa lhe ajudar.

8.5.3 O modelo de instrução distribuída

Além das tarefas de resolução de problemas, explicação e diagnóstico cognitivo e remediação, foi feita menção à tarefa de instrução. Essa tarefa surge de alguma necessidade por parte do Aprendiz provocada no processo de resolução de problemas. Uma instrução inclui recursos pedagógicos, tais como: explanação sobre algum assunto, provimento de dicas e sugestões.

Mais uma vez pode-se dizer que é através da estrutura baseada na existência de múltiplos agentes, construída para permitir a interação entre esses agentes, que se pode realizar uma tal tarefa de maneira distribuída, tal como já se colocou em relação às atividades de, por exemplo, resolução de problemas e diagnóstico cognitivo, realizados cooperativamente de maneira distribuída.

9. Aspectos de implementação e aplicações

Nesse capítulo, descrevem-se mais alguns detalhes sobre a implementação do modelo de agente apresentado. Inicialmente, entretanto, são descritas as classes concebidas para implementar tal modelo e daí, é ilustrada a sua utilização através de um exemplo. Em seguida, introduz-se um exemplo e mostram-se as classes que foram definidas para sua implementação. Enfim, apresenta-se um exemplo ilustrativo no domínio da álgebra.

9.1 Aspectos de implementação

A descrição a seguir considera, inicialmente, a noção geral de um agente tutor e, em seguida vai detalhando os seus componentes e acessórios. Para tal, utiliza-se de uma especificação baseada na metodologia orientada a objeto OMT (Object Modeling Technique [RBP 91]). Essa especificação foi implementada com a linguagem JAVA e contando com um suporte operacional para comunicação entre agentes através do uso de *sockets*. Nesse sentido, foram implementados alguns elementos do sistema tutor (resolvedor de problemas, diagnosticador e decomposição de tarefas). Os sistemas social e de distribuição foram implementados completamente. O investimento em implementação foi feito unicamente com a intenção de validar alguns dos elementos concebidos, ao mesmo tempo buscando subsídios para melhoria de certos conceitos. Por conta disso, não houve grandes preocupações com uma escolha tão criteriosa de um ferramental para implementação que se alcançasse mais eficiência, ao invés disso, buscou-se algumas conveniências técnicas.

9.1.1 A classe Agente

A classe agente reflete o que foi descrito para a arquitetura de um agente, tal como se encontra no Capítulo 5. Assim, um agente é implementado a partir de três objetos: o *tutor*, o *social* e o *distribuição*, conforme indica a Figura 9.1.

A classe que descreve um agente tutor particular no MATHEMA é uma especialização da classe **Agente Tutor**. Essa classe possui dois métodos abstratos, *resolver* e *propor*, que necessitam de uma implementação em cada classe especializada no intui-

to de viabilizar a interação que existe do sistema social para o sistema tutor. A classe **Agente Tutor** agrega, entre outros elementos, um objeto da classe **Social**. Com esse elemento e através do método *cooperar*, o agente tutor submete os pedidos de cooperação. A classe social define mais dois métodos, *entrar* e *sair*, que disparam respectivamente os protocolos de entrada e saída.

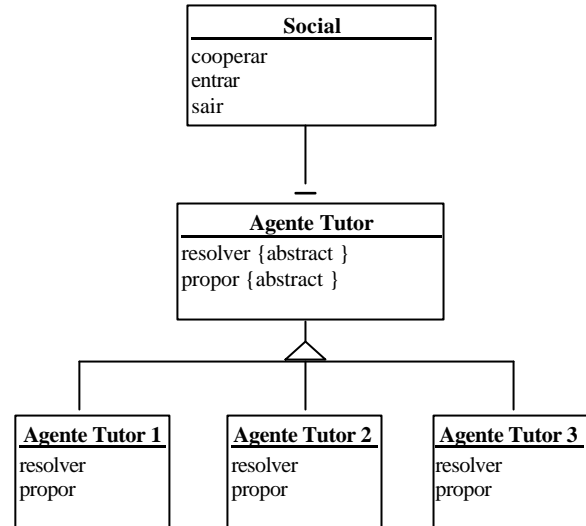


Figura 9.1: Modelo objeto para o agente tutor

9.1.2 A classe Tutor

Essa classe, tal como ilustrada na Figura 9.2, é composta de objetos que representam ou uma base de conhecimento ou um mecanismo de raciocínio ou ainda uma entidade mediadora. As bases de conhecimento são relativas ao domínio de aplicação e ao modelo do aprendiz. Já os mecanismos de raciocínio correspondem a um resolvedor de problemas, um diagnosticador, um instrutor e um controlador. Finalmente, o mediador controla as atividades interativas entre o aprendiz e o Sistema Tutor.

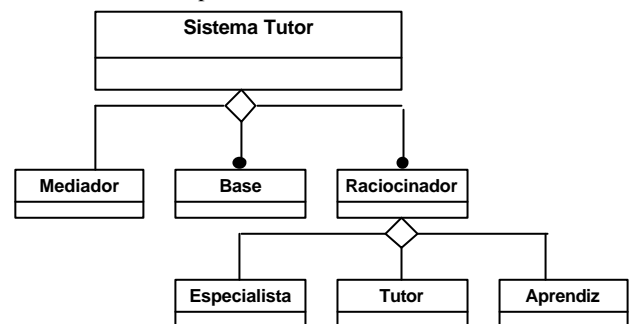


Figura 9.2: Modelo objeto para o Sistema Tutor

9.1.3 A classe Social

A classe social é inicialmente construída a partir de duas bases de conhecimento, uma coleção de protocolos e os mecanismos de coordenação e cooperação. Essas bases representam o autocohecimento e o conhecimento social de um agente. Além disso, indicam-se os seus agregados principais, que são a noção de habilidade e a de tarefa, ambas já descritas e relacionadas anteriormente.

No modelo objeto, exibido na Figura 9.3, a classe social é agregada com várias instâncias de protocolos, um CS e um AC. Um CS é constituído por vários ACs. Um AC possui um nome e é definido por várias habilidades. Uma habilidade possui um nome, um tipo e duas representações (saída e entrada). Uma representação

tótipo cujo objetivo primeiro foi as interações cooperativas entre os agentes tutores [CBF 97].

Quanto ao domínio da Álgebra, mais especificamente o estudo das equações de grau 2, investiu-se, como será mostrado a seguir, mais detalhadamente no desenvolvimento de um protótipo com uma certa abrangência. Tal protótipo foi implementado com a linguagem JAVA, no ambiente operacional Windows 95. Mas, devido às características de JAVA, pode-se assegurar a portabilidade do protótipo para plataformas diferentes (UNIX, Windows NT) e, especialmente, sua execução na *World Wide Web* (WWW). Nesse sentido, a implementação ocorreu envolvendo os três sistemas que compõem um agente tutor. Aqui, destacam-se especialmente a implementação de forma distribuída do resolvidor de tarefas pedagógicas. Assim, desenvolveu-se o modelo de diagnóstico distribuído e o módulo de resolução distribuída de problemas.

9.2.1 Um exemplo ilustrativo em Álgebra

Visando ilustrar todo o arcabouço conceitual apresentado para a definição do modelo do domínio, para sociedade de agentes e para os modelos de interação envolvendo o Aprendiz, SATA e a SEH, segue um exemplo focalizando um domínio particular da álgebra: o estudo das equações de grau 2. A escolha desse domínio se deveu basicamente a sua adequação a dois aspectos: o primeiro é devido a sua característica de enfatizar as atividades de resolução de problemas e diagnóstico, e o segundo está ligado ao fato de, apesar de simples, este domínio servir para ilustrar apropriadamente o que foi concebido em relação ao comportamento do sistema.

9.2.1.1 Descrição do domínio

O Domínio utilizado para ilustração se insere no campo da álgebra. Definindo este domínio de acordo com o modelo de domínio baseado nas dimensões de conhecimento propostas no MATHEMA, tem-se, conforme já destacado no Capítulo 4, a seguinte configuração:

⇒ *Domínio*

D = estudo de equação de segundo grau

⇒ *Contextos*

C_1 = através de métodos algébricos de resolução canônicos (fórmula de Bhaskara e método da redução)

C_2 = através de métodos de resolução do cálculo numérico

Para ilustrar as demais noções, sem ser exaustivo, considera-se apenas a utilização do contexto C_1 . Sobre ele foram definidas as profundidades que seguem, tomando o universo de trabalho onde se buscam as raízes, como parâmetro de definição dos níveis de profundidades. Convém esclarecer, entretanto, que esse é apenas um parâmetro a ser considerado, tendo muitos outros que poderiam ter sido considerados.

⇒ *Profundidades*

P_{11} = no domínio dos reais

P_{12} = no domínio dos complexos

Baseado no contexto fixado e nas suas profundidades associadas, podem-se definir os seguintes subdomínios para D :

⇒ *Subdomínios*

$d_{11} = \langle C_1, P_{12} \rangle \therefore d_{11}$ é um estudo da equação de se-

gundo grau no domínio dos reais.

$d_{12} = \langle C_1, P_{12} \rangle \therefore d_{12}$ é um estudo da equação de segundo grau no domínio dos complexos

Por agora, somente o d_{11} está sendo focalizado. As unidades pedagógicas a serem trabalhadas em d_{11} são:

- (i) *Definição*: equação algébrica de grau 2 (equação do segundo grau) é toda sentença do tipo $P(x) = Q(x)$, onde P e Q são funções polinomiais de grau menor ou igual a 2, mas pelo menos uma delas deve ter grau igual a dois.
- (ii) *Redução*: toda equação do segundo grau é redutível à forma $ax^2 + bx + c = 0$, com $a \neq 0$.
- (iii) *Classificação*: equação completa e incompleta.
- (iv) *Resolução*: resolver uma equação quadrática é obter o seu conjunto verdade que é o conjunto de todas as suas raízes. Dizemos que um número real α é raiz de $F(x) = 0$, se e somente se $F(\alpha) = 0$.
- (v) *Teorema da decomposição*: toda equação do segundo grau pode ser decomposta num produto de dois fatores do primeiro do tipo $x - r$. $F(x) = a(x - x')(x - x'')$.
- (vi) *Obtenção das raízes*:
 - método 1: fórmula de Bhaskara;
 - método 2: Decomposição.
- (vii) *Relações de Girard*: $x' + x'' = -(b/a)$ e $x' \cdot x'' = (c/a)$.

Feito isso, volta-se a considerar a seguir a dimensão de lateralidade, a partir da qual pode-se obter novos domínios para serem tratados relativamente ao domínio das equações.

Continuando a ilustração, considere fixado o subdomínio d_{11} . Daí, precisa-se agora estabelecer o conhecimento de suporte necessário para que se trabalhe adequadamente nele. A seguir, estão alguns dos conhecimentos de suporte associados as suas visões, acompanhados de suas unidades pedagógicas:

⇒ *Lateralidades*

$d_{111} = L_{111}$: expressões algébricas - valor numérico, polinômios, redução de termos semelhantes, grau.

$d_{112} = L_{112}$: operações com polinômios - adição e subtração, multiplicação, divisão por um monômio, potência de polinômios.

$d_{113} = L_{113}$: produtos notáveis / fatoração - quadrado de um binômio, produto da forma $(a + b)(a - b)$, fatoração de polinômios (trinômio do quadrado perfeito, e diferença de dois quadrados), fator comum (agrupamento), fatoração por agrupamento, fatoração completa.

$d_{114} = L_{114}$: frações algébricas - simplificações, redução ao mesmo denominador, adição e subtração de frações, multiplicação, divisão e potenciação.

$d_{115} = L_{115}$: equação do primeiro grau - fracionária, definição, resolução.

9.2.1.2 Definição da Sociedade

Ao subdomínio d_{11} está associado um agente tutor denotado por AT_{11} , tendo este os conhecimentos e habilidades necessários para tutorar sobre o domínio d_{11} .

A cada lateralidade L_{11k} está associado um agente tutor AT_{11k} , tendo este os conhecimentos e habilidades para atuar sobre a lateralidade em questão.

Baseado nas considerações acima, a sociedade de agentes tutores no domínio em questão fica definida parcialmente assim:

SATA = {AT₁₁, AT₁₁₁, AT₁₁₂, ..., AT₁₁₇}

9.2.1.3 Descrição dos Agentes

Como foi visto no Capítulo 5, um agente tutor é caracterizado por vários elementos. Por exemplo, a definição do agente AT₁₁ pode ser dada como segue:

```
id: ( AT11, pardal )
habilidades : (
  tipo: ( instrução )
  identificação: ( equação de segundo grau )
)
representações: ( ),
( tipo: ( resolução )
  identificação: ( Bhaskara )
  representações: (
    entrada: ( número a, número b, número c )
  ),
  saída: ( número raízes, número x1,
            número x2 ) ),
( tipo: ( resolução )
  identificação: ( redução )
  representações: (
    entrada: ( número a, número b ),
              ( número a, número c )
    saída: ( número raízes, número x1,
             número x2 ) ) )
```

9.2.1.4 Situações de Interação

Para ilustrar as interações entre um aprendiz e a sociedade de agentes tutores artificiais, considere os exemplos seguintes, ilustrados num esquema de diálogos.

Exemplo 1

Situação 1: Instrução (dica)

AT₁₁ Resolva a equação $25 = x^2 + (x + 1)^2$ no conjunto dos reais \mathbb{R}

Aprendiz Pede uma consulta (**dica**) sobre como resolver $(x + 1)^2$

AT₁₁ Reconhece que $(x + 1)^2$ não é assunto de sua responsabilidade. Logo após uma consulta ao seu conhecimento social, ele identifica que essa expressão é uma instância do padrão $(a + b)^2$. Isso é relativo ao assunto de Produtos Notáveis, que tem o agente AT₁₁₃ como o seu responsável. Daí, o AT₁₁ parte para uma cooperação para tratar a questão colocada pelo aprendiz, com o agente AT₁₁₃. Como resultado da cooperação, o agente AT₁₁₃ apresenta ao Aprendiz a **dica**:

$$\text{Lembre que } (a + b)^2 = a^2 + 2ab + b^2$$

que é retornada para o AT₁₁, para então ser exibida ao Aprendiz.

Detalhes sobre a cooperação na Situação 1

A cooperação entre os agentes AT₁₁ e AT₁₁₃ na situação 1, ocorre de acordo com os passos seguintes:

```
tipo : ( dica )
identificação : ( transformação )
representações: (
  entrada: ( texto expressão )
  saída: ( texto dica ) )
dados: ( expressão = "(x + 1)^2" )
```

resposta: (dica = "(a + b)^2 =
(a^2 + 2ab + b^2)")

Situação 2: Instrução

Imediatamente após a **situação 1**, pode-se ter a seguinte interação:

Aprendiz Indica que não entendeu a dica apresentada.

AT₁₁ Pergunta ao Aprendiz se ele deseja uma explicação sobre a Dica ou uma Instrução sobre o assunto.

Aprendiz Opta por uma Instrução

AT₁₁ Ativa novamente agente AT₁₁₃ para cooperar sobre esse assunto, relativamente à tarefa Instrução.

AT₁₁₃ Apresenta uma Explicação, dentro de um processo interativo sobre o assunto, para o Aprendiz.

Exemplo 2

Situação 1: Diagnóstico

Agora a situação apresentada diz respeito a um **diagnóstico** a ser realizado sobre a solução esboçada pelo Aprendiz.

AT₁₁ Resolva a equação $25 = x^2 + (x + 1)^2$ em \mathbb{R}

Aprendiz Devolve a solução

$$25 = x^2 + (x + 1)^2 \rightarrow 25 = x^2 + x^2 + 1 \rightarrow 25 = 2x^2 + 1 \rightarrow \dots \rightarrow x = \pm 2\sqrt{3}$$

AT₁₁ Avalia o resultado e conclui que o mesmo não está correto. Novamente AT₁₁ seleciona e invoca o AT₁₁₃.

AT₁₁₃ percorre o trecho da solução e encontra que a falha se deu na transformação de $(x + 1)^2$ em $x^2 + 1$. Daí, apresenta uma mensagem advertindo:

$$(x + 1)^2 \text{ não é, em geral, igual a } x^2 + 1$$

Exemplo 3

Situação 1: Resolução de Problema

Aprendiz Resolva a equação

$$\frac{2x}{2x - 1} + \frac{(x + 1)^2}{4x^2 - 1} = 1$$

AT₁₁ ao tentar resolver, constata a sua insuficiência, mas reconhece a natureza do problema e a necessidade da simplificação do polinômio:

$$\frac{2x}{2x - 1} + \frac{(x + 1)^2}{4x^2 - 1}$$

Daí, invoca um pedido de cooperação para tentar simplificá-la.

AT₁₁₄ é eleito e contratado para simplificar o polinômio em questão. Durante a sua operação, ele reconhece a necessidade de uma cooperação que fature a expressão algébrica $4x^2 - 1$. Daí, ele invoca um pedido de cooperação para tentar fatorá-la.

AT₁₁₃ é eleito e contratado para fatorar a expressão dada e devolve o resultado $(2x + 1)(2x - 1)$.

AT₁₁₄ continua a sua operação, acha o mínimo múltiplo comum (mmc) correspondente, simplifica o polinômio e devolve para AT₁₁ o polinômio

10. Conclusões e Perspectivas Futuras

Neste capítulo, apresentam-se as conclusões da presente pesquisa, considerando uma reflexão global sobre os resultados obtidos na solução dos problemas identificados e que passaram a ser abordados como integrantes dos objetivos desta tese. Finalmente, discutem-se as principais contribuições da tese e sugerem-se alguns caminhos para futuras pesquisas a serem desenvolvidas com base no presente trabalho.

10.1 Situação do Trabalho

O trabalho de pesquisa apresentado e desenvolvido nessa tese teve como objetivo global a elaboração dum modelo de ambiente interativo de apoio ao processo de ensino-aprendizagem, a partir do qual se possa conceber e desenvolver aplicações particulares. Tal como já foi visto anteriormente, trata-se de um ambiente baseado numa arquitetura multi-agentes. Nesse sentido, este envolveu um estudo relacionado com a integração das áreas de STI e SMA, rumo à concepção do referido ambiente.

Este ambiente foi concebido a partir, inicialmente, do estabelecimento de um modelo abstrato de interação entre duas entidades, onde uma delas é uma máquina desempenhando o papel de tutor e a outra é um ser humano desempenhando o papel de aprendiz. Neste ponto, levou-se em conta o entendimento de que o sucesso de um tal modelo, olhando o lado do sistema tutor, depende fortemente da capacidade desse sistema atender satisfatoriamente às necessidades do aprendiz, nos termos inseridos no problema da adaptação. Assim, foi feita uma análise de elementos que deveriam ser trabalhados para se alcançar um processo de interação suficientemente satisfatório. Desse modo, para tal fim, o primeiro elemento escolhido para investimento, foi o modelo do domínio de conhecimento, seguido pelos modelos do aprendiz e pedagógico.

Com respeito ao modelo do domínio, foi definido, conforme apresentado no Capítulo 4, um modelo multidimensional do conhecimento. Isso foi feito em resposta a uma pretensão de trabalhar o domínio buscando disponibilizá-lo com riqueza e estruturação. Pretende-se assim contar com um recurso que contribua fortemente para sua qualidade no processo de ensino-aprendizagem.

A partir da elaboração do modelo do domínio, definiu-se um sistema tutor multi-agentes, conforme apresentado no Capítulo 4, especializados em partes do conhecimento deste domínio.

Quanto ao modelo do aprendiz, definiu-se um mecanismo de modelagem para realizar diagnóstico distribuído, tomando por base a solução multi-agentes apresentada, assegurando uma atualização dinâmica deste modelo. Isso permite contribuir significativamente para o problema da adaptação.

Por fim, partiu-se para a construção de um protótipo sobre o domínio da álgebra, em particular da resolução de equações de grau 2. Outros investimentos também foram feitos, tal como mencionado anteriormente, nos domínios da Geometria Plana e Harmonia Musical.

10.2 Principais Contribuições da Tese

Com a elaboração desta tese, contribuiu-se para a área de IA-ED, oferecendo uma abordagem original e relevante na concepção de sistemas STI/ILE. A arquitetura do MATHEMA representa um avanço em termos de abrangência na proposta de distribuição de

papéis e conhecimentos em múltiplos agentes. A concepção do ambiente MATHEMA conduz a uma contribuição relevante ao problema da adaptação, ao investir num compromisso adequado entre as qualidades de conhecimento e da capacidade de interação. Assim, revisou-se, a luz de novos entendimentos, os principais modelos de um STI/ILE, quais sejam: modelo do domínio, modelo do aprendiz e modelo pedagógico. O modelo do domínio foi revisto de forma a contemplar mais “riqueza” e “estruturação” de conhecimento, incluindo a possibilidade de evolução deste conhecimento durante o processo interativo e de representações múltiplas do mesmo. Nesse sentido, apareceu a primeira grande contribuição dessa tese que foi a proposta do modelo multidimensional do domínio de conhecimento, importante para a engenharia do conhecimento envolvido. Desse modelo decorreram outros resultados importantes deste trabalho. Assim, surgiu a definição do sistema tutor multi-agentes e, concomitantemente, elaborou-se um mecanismo para construção do modelo do aprendiz, através de uma modelagem distribuída. Esse modelo foi trabalhado no sentido de considerar uma combinação envolvendo, por um lado um processo de modelagem distribuída para o diagnóstico cognitivo, resultante das interações dos agentes tutores, e por outro a possibilidade futura de inspeção e possível modificação do modelo, via um mecanismo de negociação, envolvendo o aprendiz e o tutor, e em certos casos envolve também a SEH. Já o modelo pedagógico está sendo definido dentro de um esquema interativo, com a intenção de contemplar, posteriormente, vários estilos de ensino. Pode-se entender tais contribuições como envolvidas exclusivamente em Sistemas Tutores Inteligentes.

As contribuições em Sistemas Multi-Agentes (SMA) foram também significativas, a começar pela própria base para alcançar tal sistema, que foi a elaboração do modelo multidimensional. Esse modelo permitiu definir um SMA bastante apropriado para funcionar como um Tutor Inteligente, sendo dotado das propriedades interessantes apresentadas no Capítulo 5. Com isso, foi possível reduzir a complexidade de alguns problemas considerados críticos em SMA. Por exemplo, foram solucionadas questões referentes aos seguintes aspectos: decomposição de tarefas, desenvolvimento do mecanismo de raciocínio social, definição do conhecimento social, mecanismo de resolução de conflitos e suporte ao desempenho de atividades concorrentemente, dentre outros. Isso tudo ficou bem resolvido no âmbito do protótipo desenvolvido.

Enfim, a contribuição dada para a concepção de sistemas IA-ED com base em múltiplos agentes foi de alta relevância, considerando o que se anunciou como desejável no atual estado da arte no assunto.

10.3 Algumas Reflexões sobre a Tese

O presente trabalho sobre a concretização total do ambiente MATHEMA não se esgotou, nem ao menos o pretendeu, considerando o assunto abordado, que é a concepção de sistemas IA-ED. Muito pelo contrário, ele apenas acena com um nível considerável de concretizações, tanto do ponto de vista conceitual, quanto do ponto de vista de realização. Mas, sem dúvidas, há consideráveis aspectos desta pesquisa a serem explorados.

Há o que melhorar desde o modelo multidimensional do domínio, modelo de diagnóstico distribuído até aspectos do modelo do agente tutor. Isso parece natural, pois dificilmente o ser humano constrói algo e deixa de perceber maneiras de ainda fazer melhor.

Apesar disso, os resultados previstos e alcançados foram bastante satisfatórios e, sem o nível atingido, dificilmente estaríamos vislumbrando certos melhoramentos futuros. Na seção seguinte, são apresentados alguns aspectos, que já foram antevistos durante esta pesquisa, que apontam para a evolução do que se alcançou na realização desta tese.

A realização plena do MATHEMA é bem maior do que um trabalho de pesquisa de tese de doutorado. Mas, um dos aspectos interessantes no investimento na concepção do MATHEMA, foi, por exemplo, o fato de se ter elaborado um projeto integrador de problemas e soluções (algumas ainda não atingidas), que à medida que forem se concretizando vão chegando mais próximo a uma solução cada vez mais sofisticada e eficaz para o, mencionado e desafiador, problema da adaptação.

10.4 Perspectivas Futuras

De fato, ainda há questões inexploradas ou pouco trabalhadas rumo à consolidação plena do MATHEMA. Portanto, a partir dessa tese, abre-se um caminho ainda longo para ser explorado. Isso tanto num nível conceitual, no sentido de ainda completar o MATHEMA; quanto num nível tecnológico, pela falta de um ferramental que sirva para agilizar o desenvolvimento de sistemas com aplicações específicas, seguindo a arquitetura e princípios do MATHEMA.

Portanto, alguns dos investimentos já planejados para avançar no MATHEMA, incluem os seguintes:

- (i) melhorar a modelagem das interações entre *SEH* e *SATA*, providenciando por um lado, o desenvolvimento de linguagem e protocolos mais apropriados, além de um ferramental que inclui, por exemplo, um editor sofisticado de agentes para ser incorporado no Agente de Manutenção. Por outro lado, será investido no desenvolvimento de um módulo a ser incorporado no agente tutor, permitindo-lhe adquirir conhecimento de forma automática. Isso certamente contribuirá para melhorar a qualidade do conhecimento em *SATA*.
- (ii) melhorar a estrutura do agente, incluindo-lhe uma medida de crença mais elaborada, servindo para tornar mais sofisticada as interações entre os agentes. Isso virá a ter reflexos imediatos em soluções mais bem elaboradas com respeito aos problemas de resolução de conflitos na escolha de agentes para cooperar e, no próprio mecanismo de raciocínio social.
- (iii) estender o modelo do aprendiz, incorporando um módulo que permita a inspeção do modelo pelo aprendiz, podendo dialogar com ele, com vistas a se chegar a uma negociação sobre um modelo resultante com a cumplicidade de um e de outro. Isso permitirá melhorar a qualidade do modelo do aprendiz, pois o sistema de modelagem pode tirar conclusões equivocadas sobre o aprendiz. Imagine, por exemplo, situações em que o aprendiz erra algo por simples distração, cansaço, etc., e o sistema infere algo que pode não ser o caso. Por esses e outros motivos, é importante que o aprendiz possa inspecionar o que o sistema está concluindo a seu respeito.
- (iv) investigar a possibilidade de investimento em tratar a estrutura interna de cada agente tutor, tal como definida, como uma idéia de microsociedade de agentes. Isso trará benefí-

cios em termos, por exemplo, da administração da complexidade dos conhecimentos internos envolvidos numa tal estrutura, além de permitir reusabilidade e oferecer modularidade.

- (v) continuidade na investigação e elaboração dum modelo formal para caracterização dos sistemas IA-ED, que já teve os seus primeiros resultados divulgados em [CL 97], lançando mão da teoria matemática de problemas proposta por Lopes [Lop 81]. Uma das idéias chave aqui é poder precisar a questão da aprendizagem, e daí partir para descrição dos elementos de um tal sistema, com a linguagem da referida teoria.
- (vi) desenvolvimento de ferramentas que permitam agilizar o desenvolvimento de sistemas IA-ED específicos. Um dos investimentos nessa direção é com respeito à elaboração de uma ferramenta de autoria.
- (vii) estudar mais amiúde alternativas para tornar os agentes de interface e manutenção mais robustos. Por exemplo, o agente interface deverá evoluir para tornar-se uma interface dinamicamente adaptativa ao aprendiz.
- (viii) estudar alternativas de empregar a arquitetura do MATHEMA em outras áreas, a exemplo do que já foi feito em [CPF 96].
- (ix) estudar, isso a mais longo prazo, a possibilidade de evoluir para a interação de vários aprendizes simultâneos trabalhando com a *SATA*, permitindo assim um suporte para realizar a idéia de trabalho cooperativo e em grupo.

Por fim, convém ressaltar que, especialmente, os trabalhos para serem realizados em (i) e (ii), já foram inicializados, a partir da definição de propostas para dissertações de mestrado, tendo o autor desta tese como um dos principais colaboradores.

Referências bibliográficas

- [Abr 91] D'Abreu, J.V.V. "Construção e Interfaceamento de Dispositivos com Computadores para Fins Educacionais", *Anais do II Seminário Nacional de Informática Educativa*, Macaé (AL), outubro 1991.
- [AFA 95] Aimeur, E.; Frasson, C.; Alexe, C. "Towards New Learning Strategies in Intelligent Tutoring Systems", *Proceedings of the 12th Brazilian Symposium on Artificial Intelligence, Campinas, Brazil*, pp. 121-130, Wainer J.; Carvalho A. (eds), Volume 991 of Lecture Notes in Artificial Intelligence, Springer-Verlag, outubro 1995.
- [Agh 86] Agha, G. *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, Cambridge, Massachusetts, 1986.
- [AS 97] Álvares, L.O., Sichman, J.S. "Introdução aos Sistemas Multiagentes", *XVI Jornada de Atualização em Informática - JAI'97*, pp. 1-39, Brasília (DF), agosto 1997.
- [Bae 93] Baecker, R. *Groupware and Computer-Supported Cooperative Work*, Morgan Kaufmann Publishers, (USA), 1993.
- [Bau 89] Baulac, Y. "Cabri-géomètre, présentation fonctionnelle d'un outil logiciel d'aide à l'apprentissage de la géométrie élémentaire", *Actes du Congrès Education et Informatique*, Vol. 2, pp. 467-472, UNESCO, Paris (França), 1989.
- [Bau 90] Baulac, Y. *Un micro-monde de géométrie, Cabri-*

- géomètre, Tese de Doutorado, IMAG/LSD2, Université Joseph Fourier, Grenoble (França), fevereiro 1990.
- [BF 82] Barr, A.; Feigenbaum, E.A. "Applications-oriented AI research", In: *The Handbook of Artificial Intelligence*, Vol. 2, Cap. 9, pp. 283-290 Barr, A.; Feigenbaum, E.A. (eds), Addison Wesley, Reading (EUA), 1982.
- [BG 88] Bond, A.; Gasser, L. (eds) "Readings in Distributed Artificial Intelligence", Morgan Kaufmann, 1988.
- [Bod 89] Boden, M.A. *Artificial Intelligence in Psychology: Interdisciplinary Essays*, Bradford Books, 1989.
- [BP 95] Bull, S.; Pain, H. "Did I say what I think I said, and do you agree with me?": Inspecting and questioning the student model. *Proceedings of AI-ED'95 - World Conference on Artificial Intelligence in Education*, pp. 501-508, Washington (EUA), agosto 1995.
- [BPB 93] Bull, S.; Pain, H.; Brna, P. "Collaboration and Reflection in the Construction of a Student Model for Intelligent Computer Assisted Language Learning." *Proceedings of the Seventh International PEG Conference, Edinburgh*, julho 1993.
- [Bro 91] Brooks, R.A.; "Elephants don't play chess." In: *Designing Autonomous Agents - Theory and Practice from Biology to Engineering and Back*, pp. 3-15, P. Maes (ed.). The MIT Press, 1991.
- [Bru 69] Bruner, L.S.; *Uma Nova Teoria da Aprendizagem*. Editora Bloch, 1969.
- [Car 89] Carbonell, J.G. "Paradigms for Machine Learning", *Artificial Intelligence*, 40(1-3): 1-10, setembro 1989.
- [Car 70] Carbonell, J.R. "AI in CAI: an artificial intelligence approach to computer-assisted instruction", *IEEE Transactions on Man-Machine Systems*, 11(4): 190-202, 1970.
- [CBF 97] Costa, E.B.; Bittencourt, G.; Flemming, E. "Um Sistema Tutor Multi-Agentes em Geometria", *Anais do VIII Simpósio Brasileiro de Informática na Educação - SBIE'97, Volume II*, São José dos Campos (SP), novembro 1997.
- [Cla 93] Clancey, W.J.; "Guidon-Manager revisited: A socio-technical systems approach", *Journal of Artificial Intelligence in Education*, 4(1), 5-34, 1993.
- [CL 97] Costa, E.B.; Lopes, M.A. "Ambientes de Aprendizagem Interativos: Uma Abordagem Teórica", *Anais do I Encontro Nacional de Inteligência Artificial - ENIA'97*, Brasília (DF), agosto 1997.
- [CLF 95] Costa, E.B.; Lopes, M.A.; Fereda, E. "MATHEMA: A Learning Environment Based on a Multi-Agent Architecture", *Proceedings of the 12th Brazilian Symposium on Artificial Intelligence, Campinas, Brazil*, pp. 141-150, Wainer J.; Carvalho A. (eds), Volume 991 of Lecture Notes in Artificial Intelligence, Springer-Verlag, outubro 1995.
- [CMF 97] Casas, L.A.; Maia, L.F.J., Fialho, F.A.P. "Modelagem de um Ambiente Inteligente para a Educação Baseado em Realidade Virtual", *Anais do I Encontro Nacional de Inteligência Artificial - ENIA'97*, Brasília (DF), agosto 1997.
- [Cor 94] Corrêa Filho, M. "A Arquitetura de Diálogos entre Agentes Cognitivos Distribuídos. Tese de Doutorado. UFRJ/COOPE, Rio de Janeiro 1994.
- [Cos 91] Costa, E. "Artificial Intelligence and Education: the role of knowledge in teaching", In: *Machine and Human Learning*, pp. 249-258, 1991.
- [Cos 92] Costa, M.M.C. *Introdução à Lógica Modal Aplicada à Computação*, VIII Escola de Computação, Gramado (RS), agosto 1992.
- [CP 96] Costa, E.B.; Perkusich, A. "Modelling the Cooperative Interactions in a Teaching/Learning Situation", *Proceedings of Third International Conference on Intelligent Tutoring Systems - ITS'96*, Montreal (Canadá), pp. 168-176, Frasson, C.; Gauthier, G.; Lesgold, A. (eds), Volume 1086 of Lecture Notes in Computer Science, Springer, June 1996.
- [CP 97a] Costa, E.B.; Perkusich, A. "A Multi-Agent Interactive Learning Environment Model", *Proceedings of the 8th World Conference on Artificial Intelligence in Education / Workshop on Pedagogical Agents*, Kobe (Japão), agosto 1997.
- [CP 97b] Costa, E.B.; Perkusich, A. "Designing a Multi-Agent Interactive Learning Environment", *Proceedings of the International Conference on Computer in Education - ICCE'97*, (Malásia), dezembro 1997.
- [CPF 96] Costa, E.B.; Perkusich, A.; Figueiredo, J.C.A. "A Multi-Agent Based Environment to Aid in the Design of Petri Nets Based Software Systems", *Proceedings of the 8th International Conference on Software Engineering and Knowledge Engineering - SEKE'96*, Illinois-EUA, junho 1996.
- [CPI 97b] Costa, E.B.; Perkusich, A.; Jatobá, A.A. "Linguagem e Protocolos para Interação entre Agentes Tutores", *Anais do I Encontro Nacional de Inteligência Artificial - ENIA'97*, SBC, Brasília (DF), agosto 1997.
- [CS 91] Cumming, G.; Self, J. "Learner Modelling in Collaborative Intelligent Educational Systems", In: *Teaching Knowledge and Intelligent Tutoring*, Peter Goodyear (ed.), Ablex, Norwood (EUA), 1991.
- [CTF 97] Costa, E.B.; Teixeira, L.de M.; Fereda, E. "Um Sistema Tutor Inteligente Multi-Agentes em Harmonia Musical", *Anais do VIII Simpósio Brasileiro de Informática na Educação - SBIE'97, Volume I*, São José dos Campos (SP), novembro 1997.
- [DCFP 93] Deng, Y.; Chang, S.K.; Figueiredo, J.C.A.; Perkusich, A. "Integrating Software Engineering Methods and Petri Nets for the Specification and Prototyping of Complex Software Systems", In: *Application and Theory of Petri Nets 1993*, 206-223, Marsan, M.A. (ed.), Volume 691 of Lecture Notes in Computer Science, Springer-Verlag, Chicago (USA), 1993.
- [Dem 95] Demazeau, Y. "From Cognitive Interactions to Collective Behaviour in Agent-Based Systems", *Proceedings of the European Conference on Cognitive Science*, Saint Malo (França), abril 1995.
- [Dil 93] Dillenbourg, P. "Evolution épistémologique en EIAO", *Ingenierie Educative. Sciences et Techniques Educatives*, 1(1):39-52, 1993.
- [DM 90] Demazeau, Y.; Muller, J.P. "Decentralized artificial intelligence" In: *Decentralized AI*, pp. 3-13, Demazeau, Y.; Muller, J.P. (ed.), Amsterdam (Holanda), Elsevier Science Publishers, 1990.
- [DS 92] Dillenbourg, P.; Self J. "A computational approach to socially distributed cognition", *European Journal of Psychology of Education*, 7(4): 353-372, 1992.

- [FG 91] Ferber, J.; Gasser, L. "Intelligence Artificielle Distribuée", *Proceedings of the International Workshop on Expert Systems and Their Applications*, Avignon (França), 1991.
- [FG 96] Franklin, S.; Graesser, A. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.
- [FPRS 92] Ferneda, E.; Py, M.; Reitz, Ph.; Sallantin, J. "L'agent rationnel SAID: une application en géométrie", *Proceedings of the First European Colloquium on Cognitive Science*, Orsay (França), junho 1992.
- [Fut 89] Futersack, M.; Vivet, M. "QUIZ: Une architecture multi-experts pour un tuteur intelligent enseignant les enchères au bridge", *Actes du Congrès AFCET RF-IA, tome 1, Paris* (França), décembre 1989.
- [GB 95] Goblet, X.; Benslimane, B. "Conception d'un tuteur intelligent: Représentation centrée agents du domaine d'enseignement et étude des interactions.", *Quatrièmes Journées EIAO de Cachan, Tome 2*, editora Eyrolles, 1995.
- [GC 93] Gonçalves, P.F.; Costa, E.B. "KIDLINK: Uma Estrutura para o Desenvolvimento de Trabalhos Cooperativos Interdisciplinares em Educação via Redes de Computadores", *Anais do 4º. Simpósio Brasileiro de Informática na Educação*, Recife (PE), dezembro 1993.
- [GGL 92] Girard, J.; Gauthier, G.; Levesque, S. Une architecture multiagent. *Proceedings of the International Conference on Intelligent Tutoring Systems - ITS'92*, Montreal (Canadá), junho, 1992.
- [GML 96] Giroux, S.; Marcenac, P.; Leman, S. Representing organisational student models: a generic concurrent coordination. *Workshop of the International Conference on Intelligent Tutoring Systems - ITS'96*, Montreal (Canadá), junho, 1996.
- [GS 88] Gilmore, D.; Self, J. The application of Machine Learning to Intelligent Tutoring Systems. In Self, J.A.(Ed.), *Artificial Intelligence and Human Learning*. Capman and Hall. Londres (RU), 1988.
- [GV 94] Gomes, J.; Velho, L. *Computação Gráfica: Imagem*, Instituto de Matemática Pura e Aplicada e Sociedade Brasileira de Matemática - IMPA / SBM, Rio de Janeiro, 1994.
- [Her 96] Hernández-Dominguez, A. "Specification of an Adapted Training Service in a Virtual Class Architecture", *Anais da VI Semana de Informática da UFBA*, Salvador-BA, maio 1996.
- [HP 88] Hannafin, M.J.; Peck, K.L. The Design, Development, and Evaluation of Instructional Software, MacMillan Publishing company, New York (EUA), 1988.
- [Hun 87] Huns, M. (Ed.) *Distributed Artificial Intelligence*, Morgan-Kaufman, 1987.
- [IS 97] Ito, M.; Sichman, J.S. "Uma Análise do Fluxo de Comunicação em Organizações Dinâmicas de Agentes", *Anais do I Encontro Nacional de Inteligência Artificial - ENIA'97*, SBC, Brasília (DF), agosto 1997.
- [JJ 95] Joolingen, W.R. van; Jong, T. "Design and implementation of simulation-based discovery environments: the SMISLE solution", *Proceedings of AI-ED 95 - World Conference on Artificial Intelligence in Education*, pp. 99-106, Washington (EUA), agosto 1995.
- [JW 92] Jennings, N.R. van; Wittig T. "ARCHON: Theory and Practice", In: *Distributed Artificial Intelligence: Theory and Praxis*, Avouris, N.M.; Gasser L. (Eds), pp. 179-195, Kluwer Academic Publishers, Dordrecht (Holanda), 1992.
- [KD 95] Koning, J.L.; Demazeau, Y. "Collaborative Learning Using DAI Interaction Protocols in a Telecommunication Setting", *Proceedings of the ITP'95: Interacting Agents*, Plovdiv, junho 1995.
- [Kea 87] Kearsley, G. (eds). *Artificial Intelligence and Instruction: applications and methods*, Addison-Wesley Publishing Company, 1987.
- [Kea 93] Kearsley, G. Intelligent Agents and Instructional Systems: Implications of a New Paradigm. *Journal of Artificial Intelligence in Education*, 4(4) 295-304.
- [LGM 95] Leman, S.; Giroux, S.; Marcenac, P. "A Multi-Agent Approach to Student Model Student Reasoning Process", *Proceedings of the AI-ED 95 - World Conference on Artificial Intelligence in Education*, pp. 258-265, Washington (EUA); agosto 1995.
- [LGM 96] Leman, S.; Giroux, S.; Marcenac, P. "A Generic Architecture for ITS Based on a Multi-Agent Approach", *Proceedings of the International Conference on Intelligent Tutoring Systems - ITS'96*, Montreal (Canadá), pp. 75-83, Frasson, C.; Gauthier, G.; Lesgold, A. (eds), Volume 1086 of Lecture Notes in Computer Science, Springer, junho, 1996.
- [Loi 94] Loia, V. "Distributed Diagnostic Reasoning: a new approach to student modeling", *Proceedings of the Fourth International Conference on User Modeling*, Massachusetts (USA), pp. 37-41, agosto 1994.
- [Lop 81] Lopes, M.A. Introdução a uma Teoria Geral de Problemas, Tese de doutorado, Departamento de Informática da PUC/RJ, Rio de Janeiro 1981.
- [LW 91] Lave, J.; Wenger, E. "Situated learning: Legitimate peripheral participation. Cambridge: Cambridge University Presss, 1991.
- [ME 92] Moyse, R.; Elsom-Cook, M. *Knowledge Negotiation*, Academic Press, Londres (RU), 1992.
- [Min 85] Minsky, M. *The Society of Mind*, Simon and Schuster, New York (EUA), 1985.
- [Mur 89] Murata, T. "Petri nets: Properties, Analysis and Applications", *Proceedings of the IEEE*, 77(4):541-580, abril 1989.
- [MV 95] Moussalle, N.M.; Vicari, R.M. "Observação dos Estados Mentais de Agentes Durante um Diálogo", *Anais do VI Simpósio Brasileiro de Informática na Educação, Florianópolis, Brasil*, pp. 282-294, novembro 1995.
- [MVC 96] Moussalle, N.M.; Vicari, R.M.; Corrêa, M. "Intelligent Tutoring Systems Modelled through Mental States", *Proceedings of the 13th Brazilian Symposium on Artificial Intelligence, Curitiba, Brazil*, pp. 221-230, Borges, D.L.; Kaestner, C.A.A. (eds), Volume 1159 of Lecture Notes in Artificial Intelligence, Springer-Verlag, outubro 1996.
- [Nil 82] Nilsson, N. *Principles of Artificial Intelligence*, Springer-Verlag, 1982.
- [Oli 94] Oliveira, F.M. *Critérios de Equilíbrio para Sistemas Tutores Inteligentes*, Tese de Doutorado, UFRGS, Porto Ale-

- gre (RS), 1994.
- [Oli 95] Oliveira, F.M. "Measuring Agreement and Harmony in Multi-Agent Societies: A First Approach", *Proceedings of the 12th Brazilian Symposium on Artificial Intelligence, Campinas, Brazil*, Wainer J.; Carvalho A. (eds), Volume 991 of Lecture Notes in Artificial Intelligence, Springer-Verlag, outubro 1995.
- [OV 92] Oliveira, F.M.; Viccari, R.M. "Belief Logics and Student Modeling in Intelligent Tutoring Systems", *Proceedings of the Iberoamerican Symposium on Informatics and Education*, Santo Domingo (República Dominicana), junho 1992.
- [OV 96] Oliveira, F.M.; Viccari, R.M. "Are Learning Systems Distributed or Social Systems?", *Proceedings of the European Conference in Artificial Intelligence and Education*, Lisboa (Portugal), 1996.
- [OVC 94a] Oliveira, F.M.; Viccari, R.M.; Coelho, H. "An Equilibration-based Conceptual Beliefs Representation", *Anais do II Congresso Iberoamericano de Informática na Educação*, Lisboa (Portugal), Outubro 1994.
- [OVC 94b] Oliveira, F.M.; Viccari, R.M., Coelho, H. "A topological approach to equilibration of concepts", *Proceedings of the XI Brazilian Symposium on Artificial Intelligence*, Fortaleza (CE), outubro 1994.
- [Pap 85] Papert, S. *LOGO: Computadores e Educação*, Ed. Brasiliense, São Paulo (SP), 1985.
- [Pap 87] Papert, S. "Microworlds: Transforming education", In: *Artificial Intelligence and Education - Learning Environments and Tutoring Systems*, Vol. 1, Lawer R.W.; Yazdani M. (eds.), 1987.
- [PC 92] Pontes, E.V., Costa E.B. et alli "HIPERPLAN: Um ambiente de aprendizagem baseado em hipertextos e planos", *Proceedings of the Iberoamerican Symposium on Informatics and Education*, Santo Domingo (República Dominicana), junho 1992.
- [Pea 84] Pearl, J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, 1984.
- [Pia 77] Piaget, J. *A Psicologia da Inteligência*, Zahar, Rio de Janeiro, 1977.
- [PSH 95] Paiva, A.; Self, J.; Hartley, R. "Externalising Learner Models", *Proceedings of AI-ED 95 - World Conference on Artificial Intelligence in Education*, pp. 509-516, Washington (EUA), 1995.
- [RBP 91] Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorensen, W. *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs (EUA), 1991.
- [Rei 92] Reitz, Ph. *Contribution à l'étude des environnements d'apprentissage. Conceptualisation, Spécification et Prototypage*, Tese de Doutorado, LIRMM, Université de Montpellier II, Montpellier (França), fevereiro 1992.
- [RY 87] Lawer, R.W.; Yazdani, M. (eds) *Artificial Intelligence and Education*, Ablex Publication, Norwood (EUA), 1987.
- [SB 82] Sleeman, D.; Brown, J.S. (eds) *Intelligent Tutoring Systems*, Academic Press, Londres (RU), 1982.
- [SDB 92] Sichman, J.S.; Demazeau, Y.; Boissier, O. "When can knowledge-based systems be called agents?", *Anais do Simpósio Brasileiro de Inteligência Artificial*, Rio de Janeiro (RJ), outubro 1992.
- [SD 96] Sichman, J.S.; Demazeau, Y. "A model for the decision phase of autonomous belief revision in open multi-agent systems.", *Journal of the Brazilian Computer Society*, 3(1):40-50, ISSN 0104-6500, julho 1996.
- [Sel 74] Self, J. A. "Student Models in Computer-aided Instruction", *International Journal of Man-Machine Studies*, 6:261-276, 1974.
- [Sel 88a] Self, J. "Bypassing the intractable problem of student modelling", *Proceedings of Conference on Intelligent Tutoring Systems*, pp. 18-24, Montreal (Canadá), 1988.
- [Sel 88b] Self, J.A. "Student Models: what are they?" In: *Artificial Intelligence Tools in Education*, Ercoli, P.; Lewis, R. (eds.), Amsterdam (Holanda), 1988.
- [Sel 92] Self, J. "Computational Mathematics: the Missing Link in Intelligent Tutoring Systems" In: *New Directions in Intelligent Tutoring Systems*, pp. 38-56, E Costa (ed.), NATO ASI, Sintra (Portugal), Series F, Vol. 91, Springer, 1992.
- [Sel 94] Self, J.A. Formal Approaches to Student Modelling. In Greer, J. and McCalla, G. (Eds.), In: *Student Modelling: The Key to Individualized Knowledge-Based Instruction*. pp. 295-352. Springer-Verlag, 1994.
- [SF 97] Souza e Silva, M.E.; FERNEDA E. "De Agentes Racionais a Agentes Semióticos", *Anais do I Encontro Nacional de Inteligência Artificial - ENIA '97*, SBC/UnB, Brasília (DF), 1997.
- [Sho 93] Shoham, Y. "Agent-Oriented Programming. Artificial Intelligence", 60:51-92, 1993.
- [Sic 95] Sichman, J.S. "Du raisonnement social chez les agents: une approche fondée sur la théorie de la dépendance.", Thèse de Doctorat de l'INPG, Grenoble, France, 1995.
- [Ski 58] Skinner, B.F. "Teaching Machines", *Science*, Vol. 128, 1958.
- [Sta 90] Stahl, M.M. "Software Educacional: Características dos tipos Básicos", *Anais do I Simpósio Brasileiro de Informática na Educação*, Rio de Janeiro (RJ), novembro 1990.
- [Tay 80] Taylor, R.P. *The computer in the school: tutor, tool, tutee*, Teacher's College Press, New York (EUA), 1980.
- [Ted 97] Tedesco, P.C.A.R., "SEI - Sistema de Ensino Inteligente", *Dissertação de Mestrado em Informática*, Universidade Federal de Pernambuco, junho 1997.
- [Tei 97] Teixeira, L. de M. Da representação do conhecimento musical ao esboço conceitual de uma sociedade de agentes em Harmonia. *Dissertação de Mestrado em Informática*, CCT, UFPB, Campina Grande, 1997.
- [VG 96] Viccari, R.M.; Giraffa, L.M.M. "Sistemas Tutores Inteligentes: abordagem tradicional x abordagem de agentes", *Anais do XIII Simpósio Brasileiro de Inteligência Artificial*, Tutorial, SBC, Curitiba (PR), outubro 1996.
- [UG 92] Urzelai, K.; Garijo, F.J. "MAKILA: A Tool for the Development of Cooperative Societies", In: *Proceedings of the MAAMAW'92, S. Martino al Cimino, Italy*, pp. 311-323, Castelfranchi, C.; Werner, W. (eds), Volume 830 of Lecture Notes in Artificial Intelligence, Springer-Verlag, (Itália), julho 1992.
- [VO 92] Viccari, R.M.; Oliveira, F.M. *Sistemas Tutores Inteligentes*, Universidade Federal do Rio Grande do Sul, Instituto de Informática, setembro 1992.

-
- [Vyg 91] Vygotsky, L.S. *A Formação Social da Mente*, Livraria Martins Fontes Editora Ltda., 1991.
- [Wen 87] Wenger, E. *Artificial Intelligence and Tutoring Systems: Computational and cognitive approaches to the Communication of Knowledge*. Los Altos, CA: Morgan Kaufmann, 1987.
- [WJ 95] Wooldridge M.; Jennings, N.R. van “Agent Theories, Architectures, and Languages: A Survey”, In: *Intelligent Agents*, pp. 1-22, Springer-Verlag, Berlin (Alemanha), 1995.