

---

# UTILIZAÇÃO DE UMA FERRAMENTA INDEPENDENTE DO DOMÍNIO PARA DIAGNÓSTICO DO COMPORTAMENTO DO ALUNO

**Vanessa Lindemann e Ana Lúcia Cetertich Bazzan**

{vanessal, bazzan}@inf.ufrgs.br

Universidade Federal do Rio Grande do Sul – UFRGS  
Instituto de Informática  
Programa de Pós-Graduação em Computação  
Caixa Postal 15064 CEP 91501-970  
Porto Alegre - RS

## RESUMO

A evolução da Informática na Educação exige ambientes de ensino capazes de se adaptarem ao contexto de acordo com as características individuais do aluno, que permitam interatividade e gerem um diagnóstico do comportamento cognitivo desse aluno. Com base nestes argumentos, o objetivo deste trabalho é propor um sistema de diagnóstico independente do domínio capaz de analisar o comportamento do aluno na resolução de problemas. O professor organiza o material em estruturas de tarefas *TÆMS* (uma linguagem independente do domínio para descrição de planos de resolução de tarefas), gerando uma biblioteca de planos que deverão ser executados pelo aluno. As informações referentes a navegação do aluno pelo material são gravadas em um *log*. O processo de diagnóstico ocorre através do confronto entre as informações do *log* e os planos gerados pelo professor.

**Palavras-Chaves:** Informática na Educação, Inteligência Artificial, Sistemas Multiagentes, diagnóstico cognitivo.

## ABSTRACT

The evolution of Informatics in Education demands environments capable of adapting to the context according to the student's individual characteristics. This permits interactivity and creates a diagnosis of this student's cognitive behavior. Based on these facts, this work proposes a diagnosis systems independent of the domain, which is capable of analyzing the student's behavior in solving problems. The teacher creates the material in *TÆMS* tasks structures (a domain-independed language for description of plans and tasks), generating a plan library that should be executed by the student. The referring information the student's navigation for the material is recorded in a log. The diagnosis process happens when the information of the log are confronted with the tasks generated by the teacher.

**KeyWords:** Informatics in Education, Artificial Intelligence, Multiagent Systems, cognitive diagnosis.

## 1. 1. INTRODUÇÃO

A Ciência da Computação, juntamente com a Psicologia e a Educação, tem buscado aperfeiçoar ferramentas computacionais de ensino, voltadas principalmente para o ensino individualizado. Novas abordagens do uso da Informática na Educação têm trazido boas perspectivas para esta área. A utilização pura e simples de recursos computacionais costuma resultar na construção de ambientes de aprendizagem estáticos que, ou não condizem com as modernas teorias educacionais, ou apresentam ao aluno uma avalanche de material instrucional, deixando totalmente sob a sua responsabilidade a ação a ser tomada no processo de aprendizagem. É necessário criar ambientes que sejam capazes de se adaptarem ao contexto de acordo com as capacidades individuais deste aluno, que permitam a interatividade e o diagnóstico do comportamento do aluno

---

na resolução de problemas. Portanto, além de recursos computacionais, é indispensável o uso de recursos propiciados pela Inteligência Artificial (IA).

Nas pesquisas de IA na educação, o diagnóstico cognitivo é considerado algo extremamente difícil [KON 99]. Uma das razões é que, geralmente, o diagnóstico do comportamento é baseado em catálogos de erros, que são difíceis de criar e aplicáveis somente a domínios específicos. Quando o assunto muda, é necessário desenvolver e implementar um novo catálogo de erros. Na tentativa de resolver esses problemas, são utilizados métodos de gerar erros dinamicamente, porém, estes também apresentam dificuldades. Self [SEL 93] afirma que o diagnóstico cognitivo, sendo um problema complexo, apresenta um grande desafio: expressar os aspectos cognitivos não cobertos pelo diagnóstico de falhas e caminhar para um *framework* padrão para a área de diagnóstico. Com este trabalho, pretende-se contribuir exatamente nesta direção. A seção 2 descreve a estrutura de tarefas *TÆMS*, utilizada no processo de diagnóstico. O estado da arte e a proposta de um sistema de diagnóstico independente do domínio são descritos na seção 3, enquanto que a seção 4 apresenta algumas considerações finais.

## 2. 2. *TÆMS* e *DTC*

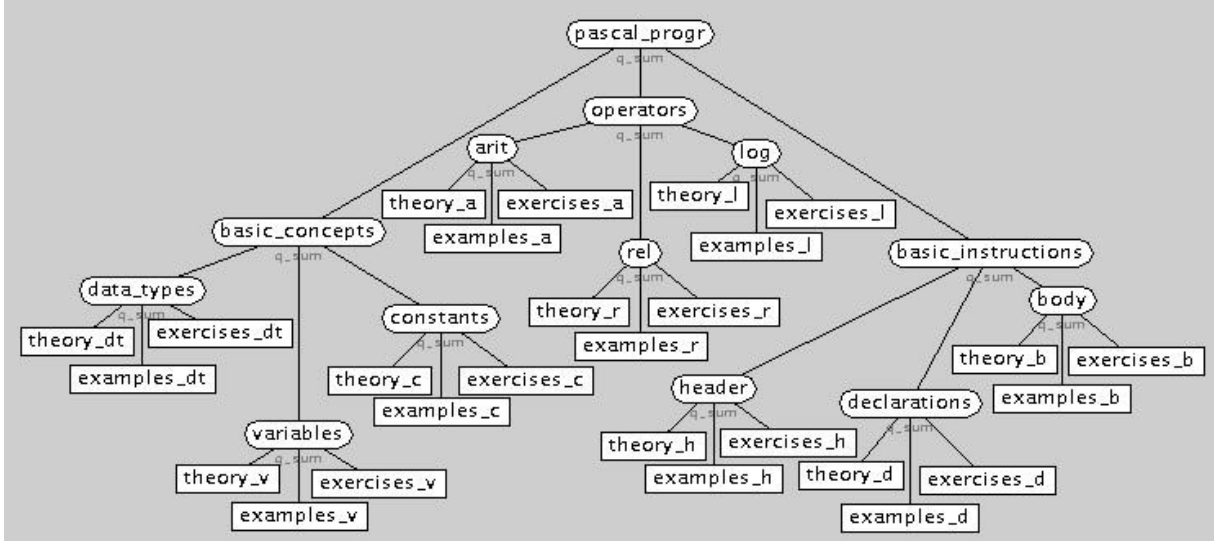
O *TÆMS* – *Task Analysis, Environmental Modeling, and Simulation* ([DEC 93], [LES 98], [HOR 00b]) é um *framework* independente do domínio usado para representar formalmente aspectos de coordenação. Representa tarefas hierárquicas, planeja ações, atividades candidatas e caminhos de soluções alternativos numa perspectiva quantificada. Todas as ações são descritas através de distribuições estatísticas discretas em três dimensões (qualidade, custo e duração). Considerando essas distribuições e as condições do ambiente, o melhor caminho de solução é selecionado.

O material (páginas *web*) criado pelo professor para ser apresentado ao aluno é representado no *TÆMS*, gerando uma estrutura de tarefas que é, essencialmente, uma árvore de decomposição do objetivo destas tarefas, onde os nodos folhas representam métodos primitivos executáveis e os nodos internos provem uma organização hierárquica (veja FIGURA 2.1). Cada tarefa está associada a uma *QAF* (*quality-accumulation function*) que indica como a qualidade de suas subtarefas deve ser calculada. Cada método é associado a uma descrição baseada na distribuição das medidas de qualidade, custo e duração esperadas (veja FIGURA 2.2). Os inter-relacionamentos que aparecem entre tarefas, métodos e recursos, são usados para indicar interações, como por exemplo *habilitar* (tarefa A habilita tarefa B), *facilitar* (tarefa A facilita tarefa B), *impedir* (tarefa A impede tarefa B) ou *consumir* (tarefa A consome recurso X) (veja FIGURA 2.2). Combinadas, essas capacidades oferecem flexibilidade para modelar uma gama enorme de características, desde características de baixo nível de uma única ação à representação em alto nível do controle hierárquico do sistema.

A FIGURA 2.1 apresenta um exemplo da estrutura de tarefas *TÆMS*, cujo domínio é o aprendizado da Linguagem de Programação Pascal. As elipses representam as tarefas e os retângulos os métodos. A FIGURA 2.2 apresenta uma parte da estrutura de tarefas da FIGURA 2.1 com a distribuição de qualidade (Q), custo (C) e duração (D) de cada método e os inter-relacionamentos entre eles. As distribuições de qualidade e tempo variam de acordo com o método e a distribuição do custo foi ignorada pois não é necessária neste caso.

Utilizando o *TÆMS*, é possível construir estruturas de tarefas para a resolução de problemas de uma situação, mesmo que essa mude com o tempo. A estrutura real,

chamada de modelo *objetivo* do ambiente, é inacessível aos agentes. Porém, cada agente tem duas versões diferentes da estrutura de tarefas local: *subjetiva* e *condicional* [HOR 01]. A visão subjetiva contém tarefas e relacionamentos que o agente acredita ser o modelo completo das alternativas de execução local. A visão condicional é uma cópia da subjetiva, modificada pela adição, modificação ou remoção de tarefas e



relacionamentos, dependendo de alguns processos de condicionamento.

FIGURA 2.1 – Exemplo de uma estrutura de tarefas *TÆMS*

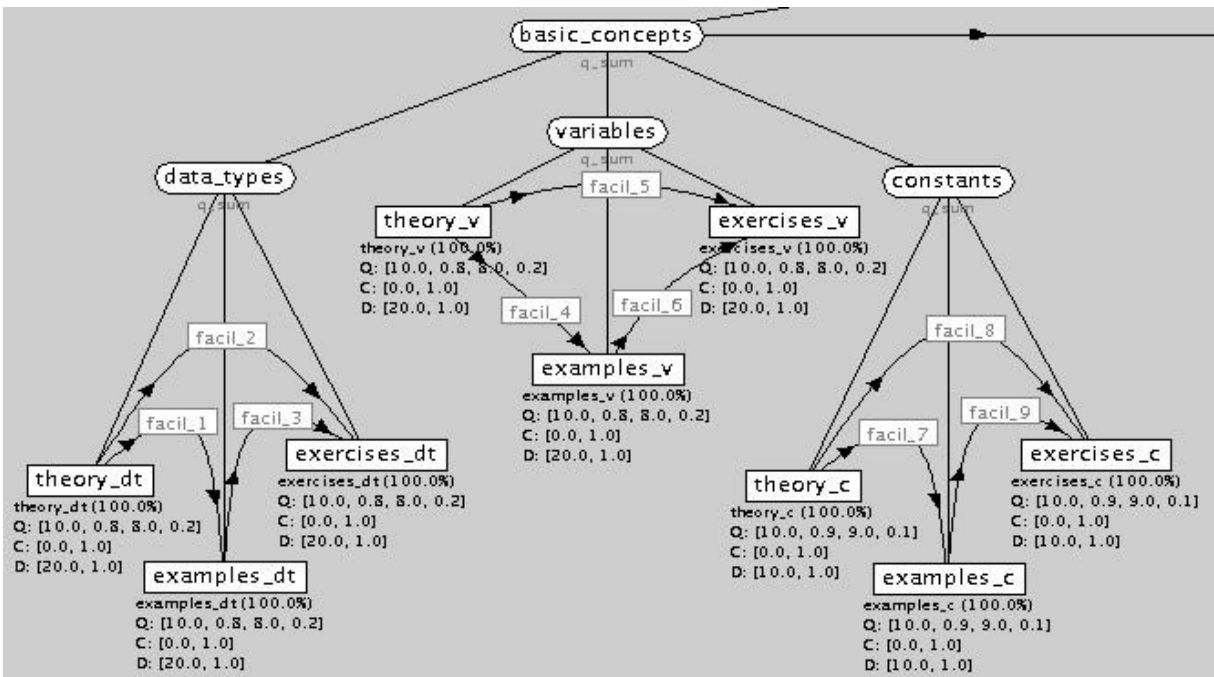


FIGURA 2.2 – Parte de uma estrutura de tarefas *TÆMS* com a distribuição de qualidade (Q), custo (C) e duração (D) de cada método e os inter-relacionamentos entre eles.

O escalonamento, executado pelo *DTC (Design-to-Criteria)*, é um processo que ocorre em tempo real, avalia possíveis caminhos de ações para um agente e escolhe o caminho que melhor se ajusta as circunstâncias atuais. Para isso, são consideradas as

distribuições de qualidade, custo e duração associados a cada método e as condições atuais do ambiente. Considerando uma estrutura de tarefas *TÆMS*, que modela vários planos, a função do *DTC* é interpretá-los para 1) decidir qual conjunto de tarefas executar, 2) decidir qual a ordem que as tarefas devem ser executadas, 3) executar as duas primeiras funções para enviar limitações, (por exemplo prazo final para uma determinada tarefa) e encontrar um equilíbrio entre o projeto e os critérios do objetivo especificados pelo cliente. A FIGURA 2.3 apresenta uma lista com os dados contidos no arquivo gerado pelo *DTC* (escalonamento das tarefas) para a estrutura de tarefas *TÆMS* da FIGURA 2.1.

theory_dt	0.000000	20.000000	9.600000	0.000000	20.000000
theory_a	20.000000	28.080000	14.651999	0.000000	8.080000
examples_dt	28.080000	42.320004	13.286402	0.000000	14.240000
theory_v	42.320004	62.320004	9.600000	0.000000	20.000000
examples_v	62.320004	76.547203	13.286402	0.000000	14.240000
theory_c	76.547203	86.547203	9.900000	0.000000	10.000000
examples_c	86.547203	93.581955	13.820399	0.000000	7.030000
examples_a	93.581955	99.295456	20.325521	0.000000	5.667406
theory_r	99.295456	107.380989	14.646606	0.000000	8.056179
examples_r	107.380989	112.940521	20.347174	0.000000	5.702903
theory_l	112.940521	161.349426	13.510775	0.000000	48.337078
examples_l	161.349426	195.765381	17.865017	0.000000	35.340164
theory_h	195.765381	200.765366	9.899997	0.000000	5.000000

**FIGURA 2.3** – Saída gerada pelo *DTC*

### 3. 3. DIAGNÓSTICO COGNITIVO

Nos sistemas gerais de diagnóstico há um interesse em tentar entender a tarefa em si. A maioria destes sistemas tem codificado o que pode ser chamado conhecimento associativo, isto é, partes de conhecimento que vão dos dados relativos ao domínio a conclusões parciais de interesse. Assim, os sistemas de diagnóstico têm a maior parte do seu conhecimento no forma “observações  $\rightarrow$  hipóteses de diagnóstico”. Frequentemente, tal conhecimento pode não estar disponível, ou tornar-se muito extenso em domínios complexos, ou a base de conhecimento pode estar incompleta. Nestes casos, é útil se ter métodos pelos quais o sistema possa usar um modelo do domínio ou do sistema, e raciocinar com este modelo para gerar informação sobre o comportamento esperado do sistema [BAZ 92]. Geralmente, um diagnóstico parte de uma observação de algum comportamento que é reconhecido como um desvio sobre uma situação desejada ou prevista, ou seja, observa-se um comportamento de mau funcionamento. Para resolver este problema, são geradas algumas hipóteses sobre a

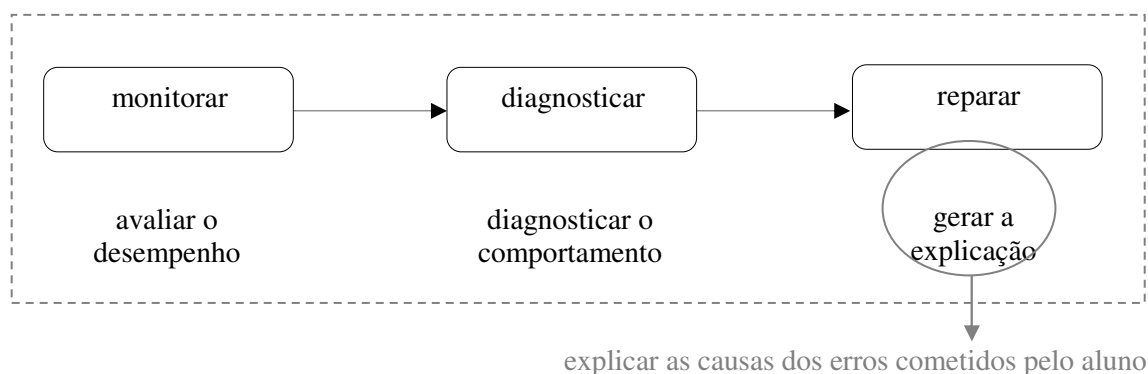
---

causa do mau funcionamento. Esta tarefa de geração de hipóteses pode ser mais ou menos complexa e mais ou menos controlada, dependendo do domínio e do conhecimento que se tem. Não importa o método, todos partem de uma observação comportamental (valores de testes, sinais, sintomas, etc.) para chegar a um número de hipóteses, preferencialmente ordenadas [BAZ 92].

Diagnóstico cognitivo é o processo de dedução do estado cognitivo das observações de desempenho de uma pessoa e, como tal, é um componente indispensável em qualquer sistema que se propõe a construir um modelo dinâmico do seu usuário. Segundo Self [SEL 93], os métodos de diagnóstico cognitivo podem ser divididos em duas classes: métodos baseados em regras usando associações entre sintomas e diagnóstico, como os métodos utilizados em sistemas especialistas padrões; métodos baseados em modelos usando descrições de componentes e estruturas de um dispositivo. Os métodos baseados em regras são criticados por várias razões como por exemplo: como são específicos para o sistema que está sendo diagnosticado, não podem ser aplicados diretamente a outros sistemas e são restritos a determinados erros. Os métodos baseados em modelos, independentes de uma descrição de dispositivos particulares, podem superar estas dificuldades [KON 99].

As teorias de Reiter [REI 87] e de Kleer [KLE 87] proveram *frameworks* gerais para diagnóstico que, na realidade, são interessantes para diagnóstico de dispositivos físicos como um circuito lógico, por exemplo. Porém, o diagnóstico cognitivo tem problemas para identificar o conhecimento defeituoso (erros, falsas concepções) na base de conhecimento do aluno. Para resolver isto, Self [SEL 93] propõe algumas extensões aos *frameworks* gerais, como utilizar modelos hierárquicos, modelos de erros, evidências do diagnóstico, erros sistemáticos e deslizes. Mesmo assim, alguns dos principais problemas dos métodos gerais de diagnóstico permanecem, entre eles, o fato de que o diagnóstico cognitivo é interativo, pois o objeto a ser diagnosticado é um agente ativo no processo de diagnose, o que é imensamente complicado de tratar. É difícil imaginar como tratar estes problemas sem levantar quase todas as questões de representação do conhecimento em IA e ciência cognitiva. Porém, como o diagnóstico cognitivo baseado em modelo não tem uma metodologia própria e segue os princípios básicos dos métodos gerais de diagnóstico, o desafio é ser independente do domínio e, consequentemente, desenvolver descrições cognitivas válidas [LIN 00].

O ambiente de aprendizagem deve ser capaz de adaptar a interação à situação atual, referente tanto ao assunto específico quanto ao aluno com quem está interagindo [VIC 90]. Independente da direção escolhida, esta requer conhecimento sobre o aluno e, para isto, é necessário avaliar seu comportamento. Assim, um sistema de ensino deve diagnosticar o comportamento deste na resolução de problemas. O processo de diagnóstico, segundo Koning et al. [KON 99], é dividido em três funções principais: monitoramento, diagnóstico e reparo (FIGURA 3.1).

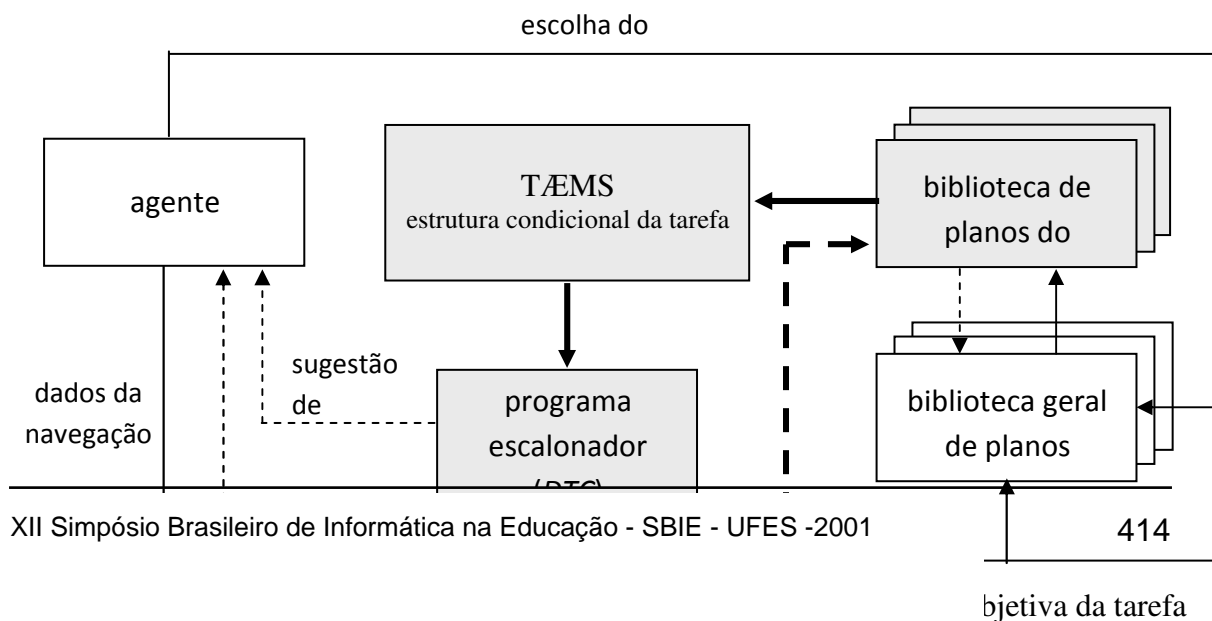


**FIGURA 3.1** – Decomposição funcional do diagnóstico cognitivo

O modo como o aluno interage com o ambiente de aprendizagem reflete seu comportamento e, portanto, o sistema deve monitorar estas interações e diagnosticar divergências, baseando-se em erros cometidos pelo aluno. Nos métodos existentes, o propósito do diagnóstico é visto frequentemente como uma explicação ou indicação das causas dos erros cometidos pelo aluno. Estas causas podem ser erros ou falsas concepções. Este é um dos motivos pelos quais o diagnóstico do comportamento na resolução de problemas é considerado extremamente difícil. É complicado construir catálogos de erros ou falsas concepções consistentes e, além disso, estes catálogos servem apenas para um domínio específico e, quando este muda, um novo catálogo deve ser implementado [KON 99].

Uma abordagem diferente para diagnóstico cognitivo é apresentada na FIGURA 3.2. Além de ser genérica e independente do domínio, não requer catálogos de erros explícitos, mas a representação do comportamento correto do estudante. Na verdade, esta também é uma suposição delicada: nem sempre o que o professor considera a “representação correta do comportamento do aluno” é a mais adequada. Entretanto, o sistema encontra uma explicação para o comportamento “incorreto” e adapta a estrutura organizacional de tarefas a fim de que esta seja mais adequada para o aluno em interações futuras.

Para diagnosticar o comportamento do aluno, estruturas de tarefas são criadas no TÆMS e o comportamento é monitorado (um log das ações atuais do aluno é gravado). Desta forma, diferenças entre o modelo especificado e o comportamento atual do aluno podem ser diagnosticadas, concluindo-se eventualmente porque o aluno não realizou determinada tarefa como o esperado. A diferença entre o modelo especificado e o comportamento atual do aluno constitui um sintoma.





**FIGURA 3.2** – Arquitetura conceitual do sistema de diagnóstico

A FIGURA 3.2 apresenta uma visão geral do sistema de diagnóstico proposto. Inicialmente, o professor cria um material específico para disponibilizar na *web*. Este material representa o que chamamos de *estrutura objetiva da tarefa* (definida na seção anterior). Baseado nessa estrutura de tarefas, planos são criados (inicialmente, sem considerar as entradas do aluno, existirá um único plano com todo o material descrito pelo professor). Esses planos formam a biblioteca geral de planos do sistema, que pode conter planos para qualquer domínio possível.

O aluno, que é o agente ativo no sistema, faz suas escolhas de acordo com seu objetivo. Através de um formulário (FIGURA 3.3), ele seleciona um plano e indica se quer ver todo o conteúdo ou apenas parte deste. O sistema oferece duas formas de restrição: restrição de tempo ou restrição de conteúdo. Caso o aluno tenha restrição de tempo, este informa o tempo máximo disponível para estudar o plano escolhido e alguns planos são gerados satisfazendo essa condição. A outra possibilidade, seria escolher alguns dos conteúdos que fazem parte do plano selecionado (FIGURA 3.4). Neste caso, o sistema gera um plano contendo os conteúdos selecionados e todos que forem pré-requisito destes.

**FIGURA 3.3** – Formulário para seleção do plano

---

### FIGURA 3.4 – Formulário para seleção de conteúdos relacionados os plano

Qualquer plano gerado a partir de seleções feitas pelo agente representa o que chamamos de *estrutura condicional da tarefa*. Esta estrutura é interpretada pelo *DTC* que retorna um ou mais *schedules* para o conjunto de tarefas (um exemplo da saída gerada pelo *DTC* pode ser visto na FIGURA 2.3). Baseado nesse escalonamento de tarefas, o sistema sugere ao agente um “roteiro de estudo” (indicando quais as páginas *web* que devem ser exploradas). Enquanto o agente está navegando no material e executando as tarefas necessárias para atingir seu objetivo, informações referentes a navegação vão sendo gravadas em um *log* (semelhante ao descrito em [AZE 00]). O processo de diagnóstico ocorre quando as informações gravadas no *log* são confrontadas com o escalonamento gerado pelo *DTC*. Tal comparação é baseada em um modelo causal geral que pode ser utilizado para diagnosticar diferenças entre quaisquer estruturas *TÆMS* ([BAZ 98], [HOR 00a]), tendo sido empregado em outros domínios como o *Intelligent Home* [HOR 00a] e *Producer-Consumer-Transporter* [HOR 01].

Se forem detectadas divergências no processo de diagnóstico é preciso verificar se estas foram geradas por algum problema associado ao agente ou por uma falha no plano. O problema é associado ao agente quando este não segue a sugestão gerada a partir da estrutura condicional da tarefa. Quando isso ocorre o agente é avisado e tem a chance de reiniciar o processo. Se o agente não quiser reiniciar, este é um bom sinal de que ele está certo daquilo que ele fez. Neste caso, o sistema irá criar e incluir um novo plano na biblioteca de planos *para este aluno*. Entretanto, se o agente seguir a sugestão gerada a partir da estrutura condicional da tarefa e sintomas de falha forem encontrados, isto significa que o plano deve ser alterado (este processo é representado pelos retângulos em cinza na FIGURA 3.2). Esse laço deve ser repetido até que o diagnóstico seja satisfatório.

## 4. 4. CONSIDERAÇÕES FINAIS

Diagnóstico cognitivo é um componente indispensável em ambientes de aprendizagem, porém, difícil de implementar. Além do fato de estarem baseados em catálogos de erros e serem aplicáveis a domínios específicos, estes são interativos, o que dificulta a identificação do conhecimento defeituoso na base de conhecimento do aluno.

A proposta de um sistema de diagnóstico baseado em modelo apresentada neste trabalho é genérica e independente do domínio. Divergências entre o comportamento do aluno e os modelos (gerados através do *TÆMS* e do modelo causal) podem ocorrer tanto por um erro provocado pelo aluno quanto por um erro no plano. Quando o aluno, como um agente ativo no processo de diagnóstico, não seguiu a sugestão de navegação gerada a partir da estrutura condicional de tarefa, as informações gravadas no *log* serão completamente diferentes das informações geradas pelo *DTC*, apresentando divergências. Nesse caso, o aluno é avisado e deve reiniciar o processo de navegação. Senão, quando o problema estiver no plano, este deve ser especificado novamente e algumas funções do processo devem ser repetidas, até que o diagnóstico seja satisfatório.

Esta abordagem pode ser aplicada com sucesso a fim de resolver significativos problemas da IA em ambientes de ensino, provendo um método genérico para o processo de diagnóstico cognitivo. É necessário ampliar as teorias gerais de diagnóstico para, desta forma, cobrir completamente o diagnóstico cognitivo.



---

## 5. 5. BIBLIOGRAFIA

- [AZE 00] AZEVEDO, L.A.O. **Análise de Atividades de Alunos de Sistemas de Ensino via Internet**. Porto Alegre: PPGC da UFRGS, 2000. Dissertação de Mestrado.
- [BAZ 92] BAZZAN, A.L.C. **Um Sistema Baseado em Conhecimento Para Fechamento de Balanço de Massa na Produção de Celulose**. Porto Alegre: CPGCC da UFRGS, 1992. Dissertação de Mestrado.
- [BAZ 98] BAZZAN, A.L.C.; LESSER, V.; XUAN, P. **Adapting an Organization Design through Domain-Independent Diagnosis**. [S.l.]: University of Massachusetts, 1998. (Technical Report 98-014).
- [DEC 93] DECKER, K.S; LESSER, V.R. Quantitative modeling of complex environments. **International Journal of Intelligent Systems in Accounting, Finance and Management**, 2(4):215-234, December 1993.
- [HOR 00a] HORLING, B.; LESSER, V.; VICENT, R.; BAZZAN, A.; XUAN, P. Diagnosis as an Integral Part of Multi-Agent Adaptability. In: DARPA INFORMATION SURVIVABILITY CONFERENCE AND EXPOSITION. **Proceedings...** South Carolina: 2000.
- [HOR 00b] HORLING, B. et al. **The TÆMS White Paper**. Disponível por WWW em <http://mas.cs.umass.edu/res-earch/TÆMS/shite>. (Agosto 2000).
- [HOR 01] HORLING, B.; BENYO, B.; LESSER, V. Using Self-Diagnosis to Adapt Organizational Structure. Amherst: University of Massachusetts. In: Autonomous Agents 2001. **Proceedings...** Montreal: 2001.
- [KLE 87] de KLEER, J. Diagnosis Multiple Faults. **Artificial Intelligence**, Amsterdam, v.32, n.1, p.97-130, April 1987.
- [KON 99] KONING, K.; BREDEWEG B.; BREUKER, J.; WIELINGA B. Model-Based Reasoning About Learner Behaviour. **Artificial Intelligence**, Amsterdam, n.117, p.173-229, 1999.
- [LES 98] LESSER, V. et al. **A Multi-Agent System for Intelligent Environment Control**. [S.l.]: University of Massachusetts, 1998. (Technical Report 98-40).
- [LIN 00] LINDEMANN, Vanessa. **Um Estudo sobre Diagnóstico: do Diagnóstico de Falhas em Dispositivos Físicos ao Diagnóstico Cognitivo Baseado em Modelo**: trabalho individual. Porto Alegre: PPGC da UFRGS, 2000.
- [REI 87] REITER, R. A Theory of Diagnosis from First Principles. **Artificial Intelligence**, Amsterdam, v.32, n.1, p.57-95, April 1987.
- [SEL 93] SELF, J. Model-Based Cognitive Diagnosis. In: **User Modeling and User-Adapted Interaction**, 1993, v.3, p.89-106.
- [VIC 90] VICCARI, R.M. **Um Tutor Inteligente para a Programação em Lógica – Idealização, Projeto e Desenvolvimento**. Coimbra: Universidade de

---

Coimbra, 1990. Tese de Doutorado.