

Construindo Mapas Conceituais utilizando a abordagem *iMap*

Wagner de Andrade Perin, Roberto Guimarães Morati Junior, Davidson Cury,
Crediné Silva de Menezes

Departamento de Informática
Universidade Federal do Espírito Santo (UFES) – Vitória, ES – Brasil

wagner.perin@ufes.br, robertomorati@gmail.com, {davidson,
credine}@inf.ufes.br

Abstract. *This article discusses the importance and application of concept maps in teaching and learning, emphasizing the need for tools that give support to teachers in the analysis of maps to track and assess learning. For this, an intelligent tool for navigation and retrieval of information is designed. This tool is able to answer questions about the maps based on their ability to make inferences. A proof of concept is also performed.*

Resumo. *Este artigo discute a importância e aplicações dos mapas conceituais no ensino e aprendizagem destacando a necessidade de ferramentas que deem suporte aos professores na análise dos mapas para acompanhar e avaliar a aprendizagem. Para tanto, é projetada uma ferramenta inteligente de navegação e recuperação de informação, capaz de responder a perguntas sobre os mapas, baseando-se em sua capacidade de realizar inferências. Uma prova de conceito é também realizada.*

1. Introdução

A utilização de recursos computacionais na promoção da educação tem se tornado uma prática recorrente na maioria das escolas e universidades. Diversos estudos já comprovam que o uso bem direcionado deles podem proporcionar resultados positivos à aprendizagem [Silva 2005]. Neste contexto, Carvalho, Nevado e Menezes (2005) apresentam o importante conceito de arquiteturas pedagógicas e estudam metodologias que integrem ferramentas tecnológicas na educação de modo a causar impactos significativos na aprendizagem. Algumas dessas ferramentas tecnológicas são as que dão apoio à construção de mapas conceituais.

Devido à sua versatilidade, é possível construir mapas conceituais para uma única aula, uma unidade de um curso, um capítulo de um livro ou mesmo para um programa educacional completo. Na medida em que os alunos vão construindo conhecimento em sala de aula, eles podem utilizar os mapas conceituais para integrar, reconciliar e diferenciar conceitos, servindo este, portanto, como um recurso de aprendizagem. Dessa forma, os mapas representam a estrutura cognitiva do aprendiz e podem ser, obviamente, valiosas fontes de informação para os professores no acompanhamento e na avaliação da aprendizagem.

O atual cenário educacional demonstra, no entanto, que tal importante ferramenta está sendo subutilizada devido a, principalmente, falta de tempo para análise dos

resultados obtidos. Por se tratar de uma representação da estrutura cognitiva de um indivíduo, a análise de mapas conceituais não é uma atividade trivial. Requer que o professor analise, de maneira profunda, a capacidade do estudante de identificar e estabelecer relações entre os conceitos estudados e aqueles já presentes na sua estrutura cognitiva. Algumas vezes, um único mapa conceitual pode exigir horas para ser analisado por completo. Multiplicando o problema da análise de um mapa conceitual pela quantidade de alunos que um professor possui em sala de aula será possível notar a necessidade de o professor dedicar um dispendioso tempo, tempo este que, na maioria das vezes, eles não possuem.

Torna-se evidente, portanto, a grande necessidade que existe da criação de um aparato ferramental que auxiliem os alunos na criação de mapas conceituais e, ao mesmo tempo, forneça subsídios aos professores na análise e avaliação destes. Muitas iniciativas já foram realizadas neste sentido, de modo que existem, hoje, diversas ferramentas computacionais que auxiliam no processo de criação e edição de mapas conceituais e sua variante próximo, os mapas mentais. Algumas dessas ferramentas são: *ExploraTree*¹, *Mindomo*², *MindMeister*³, *WiseMapping*⁴ e o mais conhecido deles, o *CmapTools*⁵. Outras apoiam a construção automática, ou semiautomática, de mapas conceituais [Kowata, Cury e Boeres 2009].

É importante frisar, no entanto, que todas estas soluções permitem a criação de mapas conceituais e mapas mentais, mas não fornecem mecanismos que ampliem suas funcionalidades permitindo uma busca rápida por conceitos e relações, sem a necessidade de uma análise, ou mapeamento, de seus elementos gráficos. Pode ser que os professores queiram, por exemplo, buscar respostas a questões que geralmente fazem ao analisar um mapa conceitual, questões tais como: “Como se certificar de que os alunos identificaram, em seus mapas conceituais, um determinado conceito?”; “Como verificar se o estudante identificou uma relação entre dois dados conceitos?”; “Quais conceitos o aluno relacionou a um dado conceito?”; e, “Que conceitos um aluno precisa conhecer para o completo entendimento de outro conceito mais abrangente?” As ferramentas supracitadas não possuem mecanismos que facilitem esse processo.

Com o objetivo de resolver essa limitação, este artigo objetiva apresentar o *iMap*, um ambiente computacional que vai além de auxiliar os estudantes na construção de seus mapas conceituais. Para os professores, o *iMap* fornece mecanismos que os auxiliam a realizar a verificação do conteúdo dos mapas sem a necessidade de uma análise demorada.

Para apresentar este ambiente, este artigo foi organizado em cinco seções. A Seção 2 objetiva apresentar o ambiente proposto, suas características e principais componentes. A Seção 3 apresenta um exemplo de abordagem pedagógica apoiada por este ambiente. A Seção 4 apresenta as considerações finais e os possíveis trabalhos futuros. Por fim, a Seção 5 lista as referências bibliográficas.

¹ Disponível em: <http://www.exploratree.org.uk/>

² Disponível em: <http://www.mindomo.com>

³ Disponível em: <http://www.mindmeister.com>

⁴ Disponível em: <http://www.wisemapping.com/c/home.htm>

⁵ Disponível em: <http://cmap.ihmc.us/download/>

2. O ambiente proposto

O *iMap* é um protótipo que se encontra em fase de testes. Seu nome é um acrônimo de “Inferência em Mapas” que sintetiza o seu principal objetivo. Ele foi idealizado com o objetivo de formar um ambiente em cuja principal tarefa é fornecer um mecanismo de inferências capaz de responder às perguntas que os professores desejam fazer para extrair informações presentes nos mapas conceituais, de modo simples, direto e rápido.

Sendo assim, o *iMap* fornece duas funcionalidades principais: 1) uma ferramenta gráfica que facilita o processo de criação e edição de mapas conceituais e 2) a geração de um arquivo, a partir do mapa gerado, alimenta uma base de fatos e regras de um sistema capaz de realizar inferências e responder a diversas perguntas sobre o mapa e navegá-lo em busca de conceitos relacionados a um conceito-origem.

2.1. O *iMap* e sua arquitetura geral

O desenvolvimento do *iMap* utilizou uma combinação das seguintes técnicas e soluções: MDD (*Model-Driven Development*) e RBS (*Rule Based Systems*). Sua arquitetura geral está sintetizada na Figura 1. Observa-se nesta figura que a construção da ferramenta para edição de mapas conceituais utilizando a abordagem MDD engloba três subatividades: 1) metamodelagem de mapas conceituais, 2) definição da sintaxe concreta visual e 3) definição de regras de transformação do modelo instanciado para um arquivo de texto. A junção destas três atividades compõe a ferramenta gráfica do *iMap* que pode ser utilizada para construção e definição de instâncias do mapa conceitual. Uma vez gerado uma instância do mapa conceitual, as regras de transformação do *iMap* são executadas, gerando um arquivo de saída contendo os fatos e regras que servirão de entrada para o mecanismo inteligente. A abordagem MDD faz com que o tempo gasto para produção de uma aplicação seja reduzido drasticamente, uma vez que requer pouca edição de códigos-fonte de aplicações, bastando apenas especificar, visualmente, a sintaxe abstrata do domínio da aplicação e a sintaxe concreta gráfica que se deseja obter das instâncias do metamodelo da aplicação. Essas instâncias podem então ser transformadas em arquivo texto cujo formato é especificado pelo desenvolvedor da aplicação [Almeida et al. 2004].

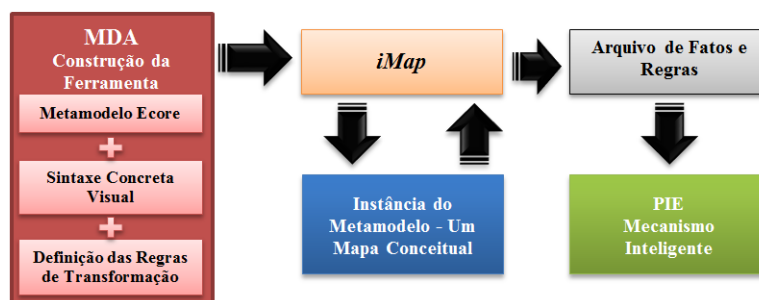


Figura 1. Arquitetura Geral do *iMap*

O arquivo de regras resultante da transformação das instâncias do metamodelo servirá de entrada para um RBS, um sistema que utiliza um método combinado de pesquisa e raciocínio [Simon 1997; Abraham 1995]. Ele será utilizado como uma ferramenta auxiliar e será responsável por fornecer a capacidade de realizar inferências nos mapas conceituais. Esta camada inteligente será útil na criação de regras que permitirão aos professores navegar o mapa conceitual e, mais importante ainda, responder a perguntas extraindo, assim, informações sobre os conceitos e relações presentes nos

mapas. Na escolha deste mecanismo, diversos *RBSs* foram analisados (*Drools*⁶, *JESS*⁷, *RuleML*⁸ e *PIE*⁹). Devido às características singulares deste projeto o *RBS* escolhido foi o *PIE* (*Prolog Inference Engine*).

A Figura 2 apresenta um mapa conceitual que pode ser construído utilizando o *iMap* e o arquivo gerado pela transformação deste. Ela ajuda a entender melhor o processo de transformação do mapa conceitual e a composição dos fatos contidos no arquivo de saída, que segue a sintaxe esperada pelo *PIE*.

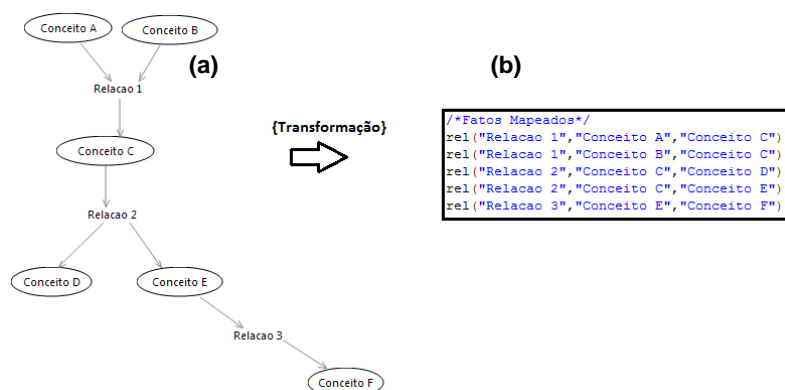


Figura 2. (a) Mapa conceitual construído no *iMap* / (b) Arquivo gerado

Os fatos contidos no arquivo de saída são, na realidade, um mapeamento das relações definidas no mapa conceitual. Como as relações podem possuir mais de um conceito-destino e também muitos conceitos-origem, as relações são decompostas em tuplas do tipo “conceito-relação-conceito”, fazendo, assim, uma permutação dos conceitos-origem com os conceitos-destino. Sendo assim, todas as possíveis combinações são geradas formando um conjunto de tuplas que abrangem todo o mapa.

2.2. Requisitos e desenvolvimento

As três subatividades mencionadas na seção anterior podem ser resolvidas com o uso de um único *framework*, denominado *Obeo Designer*¹⁰. Este *framework* fornece todas as facilidades características da abordagem MDD. A principal característica desta abordagem é a redução da complexidade no desenvolvimento de ferramentas de modelagem de linguagens específicas de domínio, neste caso, dos mapas conceituais. A Figura 3 sintetiza a organização e relação entre essas três subatividades. A etapa 1 é a especificação de um metamodelo, a sintaxe abstrata, que representa a definição do domínio da aplicação, seus conceitos, relações e propriedades. A etapa 2 consiste na descrição da sintaxe concreta visual, ou seja, na definição da forma como cada um dos componentes será representado graficamente. Por último, na etapa 3, é especificado um conjunto de regras de transformação que serão aplicadas ao mapa instanciado a fim de gerar o arquivo de saída.

⁶ Mais informações em: <http://www.jboss.org/drools/>

⁷ Mais informações em: <http://www.jessrules.com/>

⁸ Mais informações em: <http://www.ruleml.org/>

⁹ Mais informações em: <http://www.visual-prolog.com/>

¹⁰ *Obeo Designer* é uma ferramenta proprietária desenvolvida pela empresa *Obeo*, uma companhia especializada em construção de ferramentas *MDA* (*Model-Driven Architecture*) baseada em *Eclipse*. Mais informações podem ser obtidas através do site: www.obeodesigner.com

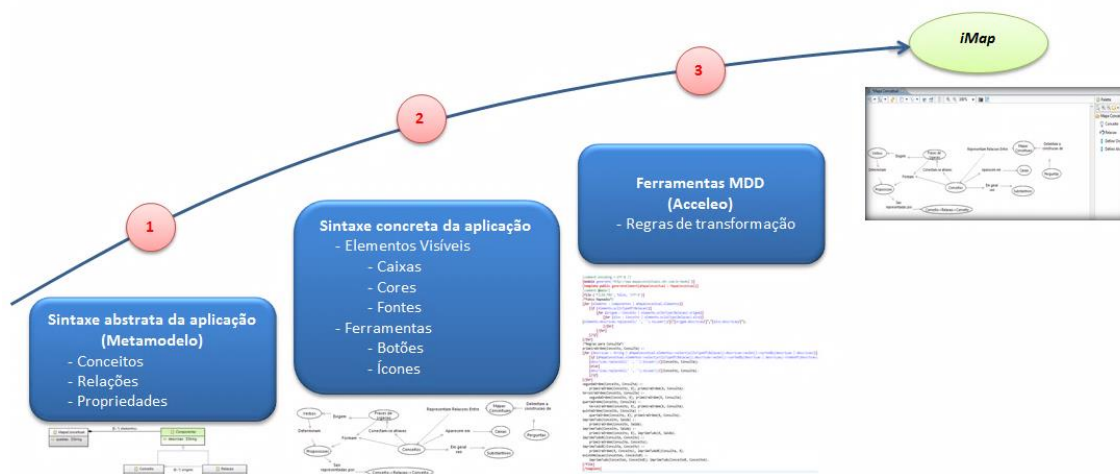


Figura 3. Etapas de desenvolvimento do *iMap* no *Obeo Designer*

A seguir serão apresentados os resultado obtidos com a realização de cada uma dessas etapas na construção da ferramenta gráfica do *iMap*.

2.2.1. O metamodelo da aplicação

O metamodelo visa descrever todos os componentes presentes em um mapa conceitual, bem como suas características e relações. A Figura 4 apresenta o metamodelo construído para o *iMap*. A metaclassa “*MapaConceitual*” é considerada um *container* onde são armazenados os componentes (“*Conceito*” e “*Relacao*”). Outra característica que se pode destacar é a possibilidade de especificar a questão de investigação do mapa. Notamos ainda os conceitos e relações presentes em um mapa conceitual possuem, como propriedade, uma descrição.

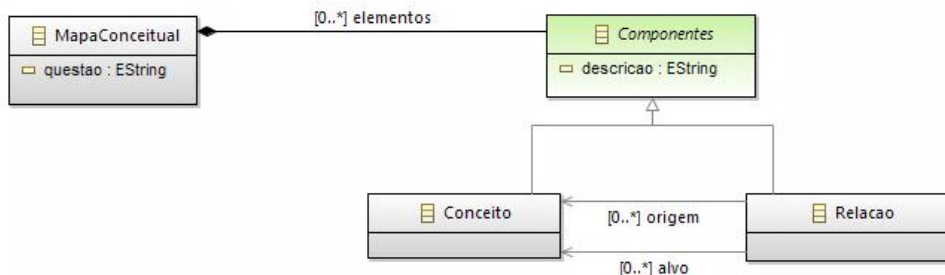


Figura 4. Metamodelo do *iMap*

2.2.2. A sintaxe concreta visual

A segunda etapa para a construção do *iMap* consistiu na definição da sintaxe concreta visual, ou seja, o layout da aplicação. A sintaxe concreta visual permite especificar, por meio do *framework* utilizado, a forma como os componentes presentes no metamodelo serão representados graficamente. Partindo das características de mapas conceituais, que podem ser observadas na Figura 2(a), notamos que os conceitos presentes num mapa conceitual são representados graficamente por meio de uma elipse contendo, no seu interior, uma breve descrição do conceito. As relações são representadas por setas e possuem, de maneira similar, uma descrição, que representa o significado da ligação entre

os dois, ou mais, conceitos relacionados. A Figura 5 apresenta parte do arquivo que especifica a sintaxe concreta visual do *iMap*.

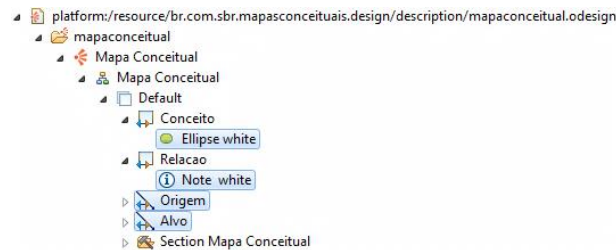


Figura 5. A sintaxe concreta visual do *iMap*

Os elementos selecionados na Figura 5 descrevem a forma como as instâncias de “Conceito” e “Relação” devem ser representadas graficamente. Percebe-se que as instâncias de conceitos serão representadas por elipses brancas enquanto as de relações serão representadas por uma nota com descrição e setas apontando para os conceitos-origem e conceitos-alvo. Todas estas especificações são realizadas visualmente, com poucas interferências de código, sendo esta uma característica da abordagem MDD.

2.2.3. As regras de transformação

A última etapa para a construção do *iMap* consiste na definição de um conjunto de regras de transformação. A ferramenta gráfica do *iMap* cria e manipula uma instância da metaclassa “MapaConceitual” que é usada como um *container* de instâncias de “Conceito” e “Relação”. Para que seja possível manipular os conceitos presentes no mapa conceitual instanciado através do *PIE*, é preciso mapear os conceitos e relações para o formato de proposições aceito por este mecanismo. As proposições de entrada para o *PIE* devem seguir o padrão presente na Figura 2 (b), ou seja, as relações entre os conceitos precisam ser mapeadas na forma de tuplas no seguinte formato:

$$rel (< \text{“Descrição_da_Relação”} >, < \text{“Conceito_Origem”} >, < \text{“Conceito_Alvo”} >)$$

Este processo de transformação é possível com a utilização de um gerador de código, muito utilizado na abordagem *MDA* e presente no *Obeo Designer*, denominado *Acceleo*¹¹. Ele permite a definição de regras de transformação, utilizando sintaxe própria, que são aplicadas às instâncias do *container* do metamodelo.

No *iMap*, a execução das regras de transformação irão gerar um arquivo contendo as proposições no formato esperado pelo *PIE*. Este arquivo inclui, ainda, as seguintes regras de consulta que serão utilizadas pelos professores pra realizar perguntas ao mapa conceitual:

- **primeiraOrdemDireta(ConceitoA, ConceitoB)**: Auxilia na verificação de existência de relação direta entre o “ConceitoA” e o “ConceitoB”, sendo estes origem e destino da relação, respectivamente. Ou seja, o “ConceitoA” leva ao “ConceitoB”.

- **primeiraOrdemInversa(ConceitoA, ConceitoB)**: Auxilia na verificação de existência de relação direta entre o “ConceitoA” e o “ConceitoB”, sendo estes destino e origem da relação, respectivamente. Ou seja, o “ConceitoA” é inferido pelo “ConceitoB”.

¹¹ Mais informações em: <http://www.acceleo.org>

- **todoDestino(Conceito, Saida)**: Imprime todos, e quantos são, os conceitos possíveis para “Saida” descritos no mapa conceitual, navegando a partir do “Conceito” sendo, este ultimo, origem da navegação. Permite verificar, então, todos os conceitos que podem ser inferidos no mapa conceitual tendo por base o “Conceito” de origem.

- **todaOrigem(Saida, Conceito)**: Imprime todos os conceitos possíveis para “Saida” descritos no mapa conceitual, navegando até o “Conceito” sendo, este ultimo, destino da navegação. Permite, portanto, verificar todos os conceitos que são pré-requisitos para o entendimento do “Conceito” dado como alvo no mapa conceitual.

- **existeRelacaoDireta(ConceitoA, ConceitoB)**: Verifica a existência de uma relação direta entre os conceitos A e B.

- **existeRelacao(ConceitoA, ConceitoB)**: Verifica a existência de relação entre os conceitos A e B, independente de ser uma relação direta ou não. Ou seja, podendo haver um ou vários conceitos entre os conceitos A e B.

- **quaisRelacoes(Saida, Conceito)**: Informa, através de “Saida”, todas as descrições das relações na qual o “Conceito” está presente.

- **quaisRelacoes(Saida, ConceitoA, ConceitoB)**: Informa, através de “Saida”, todas as descrições das relações que possuem o “ConceitoA” e “ConceitoB” relacionados.

2.3. Os passos para utilização

Uma vez que o *iMap* é, na realidade, um ambiente composto por várias ferramenta onde cada uma possui um propósito específico, a Figura 6 sintetiza o fluxo de utilização deste ambiente.

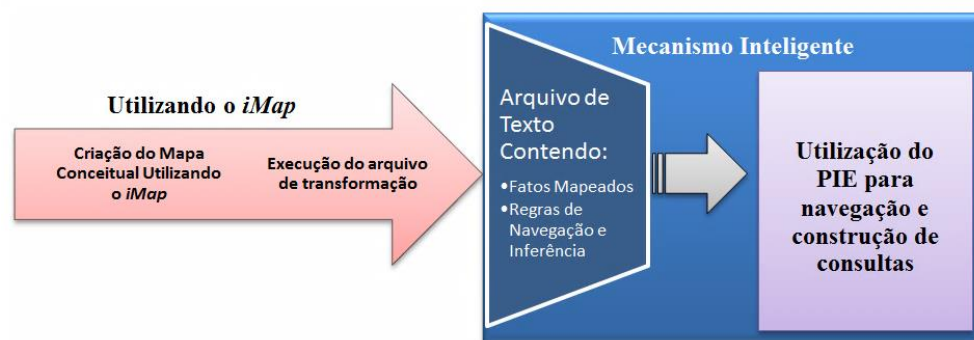


Figura 6. Fluxo de utilização do *iMap*

Percebe-se, portanto que, o *iMap* é utilizado nas etapas iniciais, ou seja, na criação e transformação do mapa conceitual num arquivo que servirá de entrada para o mecanismo inteligente, o *PIE*. Este mecanismo é, então, “alimentado” pelo mapa conceitual e pode ser utilizado pelo professor para extrair as informações que deseje.

3. Uma abordagem apoiada pelo *iMap*

Existem diversas abordagens possíveis onde o *iMap* poderia ser utilizado como ambiente de apoio à aprendizagem. Um dos cenários possíveis pode ser o modelo descrito no passo-a-passo abaixo:

- **Passo 1**: O professor sugere aos alunos que, a partir de um artigo que explica, por exemplo, o que são mapas conceituais, os estudantes devem construir um mapa,

utilizando o *iMap*, que responda a seguinte questão: “O que são mapas conceituais?”. Um possível resultado obtido por um desses estudantes é apresentado na Figura 7.

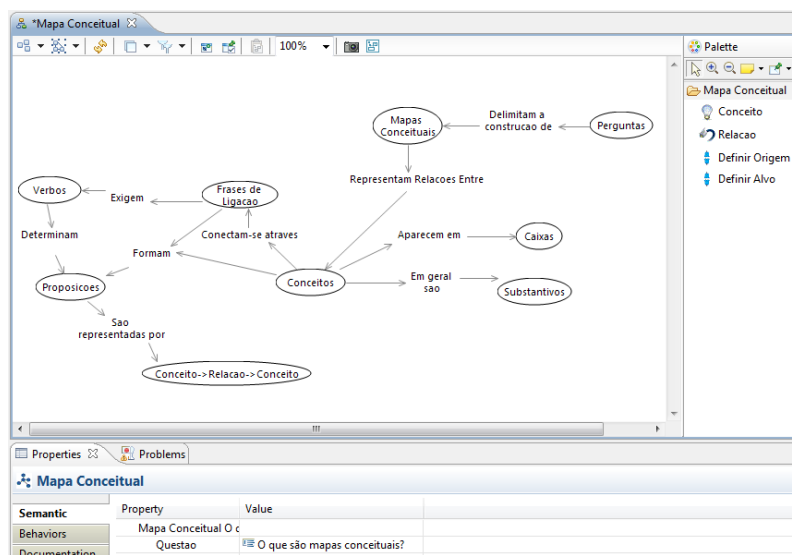


Figura 7. Um exemplo de mapa conceitual construído no *iMap*

- **Passo 2:** O professor realiza a transformação do mapa através da execução do arquivo de transformação do *iMap*. Esta execução é simples, bastando informar dois parâmetros: 1) a instância de “MapaConceitual” que deseja transformar e 2) o local de saída do arquivo transformado.

- **Passo 3:** O professor inicializa o *PIE* que estará previamente configurado para, na inicialização, abrir o arquivo gerado pelo *iMap*. A Figura 8 apresenta a tela logo após a inicialização do *PIE*, com parte do arquivo gerado pelo *iMap* a esquerda.

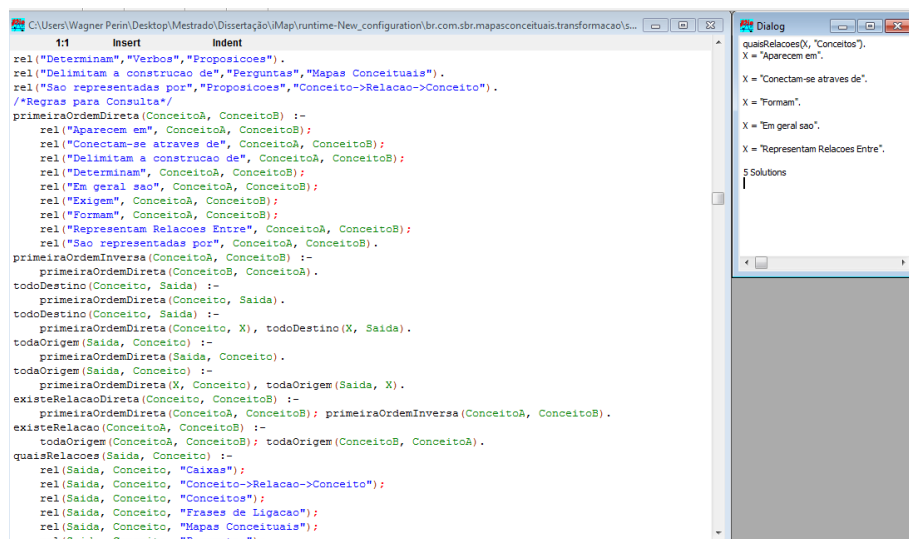


Figura 8. Tela de inicialização do *PIE*, o mecanismo de inferência.

- **Passo 4:** O professor solicita ao *PIE* que atualize sua base de fatos e regras e, a partir de então, poderá realizar perguntas utilizando a caixa de diálogo. Estas perguntas devem ser feitas utilizando a sintaxe definida no arquivo de entrada que se encontra descrita na Seção 2.2.3 deste artigo. Para exemplificar, a Figura 9 apresenta respostas às seguintes perguntas: (a) Existe relação direta entre “Verbos” e “Proposições”? (b) Existe

relação entre “Caixas” e “Substantivos”? (c) Quais e quantos conceitos o aluno identificou partindo do conceito “Verbos”? (d) Quais conceitos levam ao completo entendimento do conceito “Frases de Ligacao”? (e) De quais relações o conceito “Conceitos” participa? (f) Através de qual(is) relação(ões) os conceitos “Conceitos” e “Proposições” estão relacionados? A Figura 9 apresenta as respostas dadas a cada uma dessas perguntas.

<p>Dialog</p> <p>existeRelacaoDireta("Verbos", "Proposicoes")</p> <p>True</p> <p>1 Solution</p> <p>(a)</p>	<p>Dialog</p> <p>existeRelacao("Caixas", "Substantivo")</p> <p>No solutions</p> <p>(b)</p>	<p>Dialog</p> <p>todoDestino("Verbos", X)</p> <p>X = "Proposicoes".</p> <p>X = "Conceito->Relacao->Conceito".</p> <p>2 Solutions</p> <p>(c)</p>	<p>Dialog</p> <p>todaOrigem(X, "Frases de Ligacao")</p> <p>X = "Conceitos".</p> <p>X = "Mapas Conceituais".</p> <p>X = "Perguntas".</p> <p>3 Solutions</p> <p>(d)</p>	<p>Dialog</p> <p>quaisRelacoes(X, "Conceitos").</p> <p>X = "Aparecem em".</p> <p>X = "Conectam-se atraves de".</p> <p>X = "Formam".</p> <p>X = "Em geral sao".</p> <p>X = "Representam Relacoes Entre".</p> <p>5 Solutions</p> <p>(e)</p>	<p>Dialog</p> <p>quaisRelacoes(X, "Conceitos", "Proposicoes").</p> <p>X = "Formam".</p> <p>1 Solution</p> <p>(f)</p>
--	--	---	---	---	--

Figura 9. Respostas às perguntas submetidas ao PIE

Como pode ser observado, o mecanismo inteligente permite que, de maneira rápida, o professor extraia informações do mapa, ou mesmo navegue em seus conceitos. Desta forma, o professor pode acompanhar e avaliar a aprendizagem do estudante certificando-se que estes foram capazes de estabelecer relações entre conceitos presentes no artigo proposto sem depender de um dispendioso tempo para análise e mapeamento dos conceitos presentes nestes mapas, reduzindo drasticamente, assim, o tempo gasto neste processo.

4. Considerações finais e Trabalhos futuros

Diversos estudos comprovam os ótimos resultados alcançados por abordagens pedagógicas apoiadas pelo uso dos mapas conceituais. No entanto, é preciso destacar que a utilização desta técnica é limitada, principalmente, devido ao dispendioso tempo gasto pelos professores no acompanhamento e na avaliação destes.

Constando a ausência total de aparato ferramental que auxilie os docentes neste processo, este artigo apresentou o ambiente *iMap*. Esta ambiente permite que os professores alimentem um sistema inteligente com fatos gerados a partir das relações entre os conceitos especificados pelos estudantes e, a partir de então, utilize sua sintaxe para realizar perguntas ao mecanismo que pode, por meio de inferência, respondê-las. O *iMap* permite, portanto, o acompanhamento e a avaliação de mapas conceituais e reduz o dispendioso tempo que os professores gastariam ao mapear e verificar a existência de relações entre conceitos presentes nos mapas gerados por seus alunos.

Os resultados obtidos até então, por meio de experimentos conduzidos pelos autores, demonstram que é possível desenvolver um ambiente completo para criação, análise e avaliação de mapas conceituais com agilidade.

No entanto, há ainda algumas limitações e novas funções do *iMap* que podem ser exploradas e diversos trabalhos futuros desenvolvidos. Outras extensões de suas funcionalidades podem permitir averiguar quais conceitos o estudante ainda não conseguiu identificar por meio, talvez, de uma comparação entre mapas conceituais. Além disto, pode incluir funcionalidades como: geração automática de mapas a partir de textos e mesclagem de mapas conceituais, que podem gerar bases de fatos e regras bem abrangentes.

Uma limitação está nas perguntas utilizadas no mecanismo inteligente que exigem do professor conhecer a sintaxe das regras para consulta definidas. Desta forma, outro trabalho futuro pode estar empenhado em como tornar o mecanismo capaz de interpretar questões feitas pelo professor em linguagem natural.

Outro trabalho futuro identificado pretende integrar a ferramenta gráfica do *iMap* com o *CmapTools*, por ser uma ferramenta amplamente utilizada e possuir um vasto repositório, permitindo que estas compartilhem mapas entre si. Ou mesmo, definir novas regras de transformação no *iMap* de modo a torna-lo capaz de gerar base de fatos e regras a partir de mapas construídos no próprio *CmapTools*.

Pode-se pensar, por último, no desenvolvimento de uma *RBS* que componha, de maneira integrada, o ambiente *iMap* a fim de evitar a necessidade da inicialização de uma segunda ferramenta, neste caso, o *PIE*.

5. Referências

- Abraham, A. (2005) “130: Rule-based Expert Systems”. Oklahoma State University, Stillwater, USA. One Problem Solving Using Heuristics.pp.909-919
- Almeida, J. P. A. ; Dijkman, R. ; van Sinderen, M. ; Ferreira Pires, L. (2004) On the Notion of Abstract Platform in MDA Development. In: 8th IEEE International Enterprise Distributed Object Computing Conference, 2004, Monterrey, USA. Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International. Washington, DC: IEEE Computer Society Press, 2004. p. 253-263.
- Carvalho, M. J. S., Nevado, R. A., Menezes, C. S. (2005) “Arquiteturas Pedagógicas para educação a distância: concepções e suporte telemático”. Anais do XVI Simpósio Brasileiro de Informática na Educação, Juiz de Fora – MG – Brasil.
- Kowata, J. H., Cury, D., Boeres, M. C. S. (2009) “Caracterização das Abordagens para Construção (Semi) Automática de Mapas Conceituais”. Anais do XX Simpósio Brasileiro de Informática na Educação, Florianópolis – SC – Brasil.
- Schank, R. and Birnbaum, L. (1996) "Aumentando a inteligência". In A natureza da inteligência, Edited by Jean Khalfa. São Paulo: Ed. UNESP, p.77-109.
- Silva, E. T. (2005) Revalorização do livro diante das novas mídias. Veículos e linguagens do mundo contemporâneo: a educação do leitor para as encruzilhadas da mídia. In: Integração das Tecnologias na Educação. Salto para o Futuro. Almeida, Maria Elizabeth Bianconcini de., Moran, José Manuel.
- Simon, H. A. (1997) “Models of Bounded Rationality”, Vol. 3. MIT Press. Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.