

Framework para sistemas adaptativos: um estudo de caso

Elaine H. T. Oliveira¹, Erika H. Nozawa², Rosa M. Vicari³, Luciana F. Costa², Yuri C. Albuquerque²

¹Instituto de Computação – Universidade Federal do Amazonas – Manaus, AM – Brazil

²Instituto Ambiental e Tecnológico da Amazônia (IATECAM) – Manaus, AM – Brazil

³Instituto de Informática – Universidade Federal do Rio Grande do Sul – Porto Alegre – RS – Brazil

elaine@icomp.ufam.edu.br, {erika.handa,luciana.costal887,
yuridenommus@gmail.com}, rosa@inf.ufrgs.br

***Abstract.** In this paper, we propose a conceptual framework to allow the integration of adaptive hypermedia in learning management systems from a cognitive-semantic perspective. The cognitive approach is based on principles of the Relevance Theory. The semantic approach considered the use of a domain ontology as the knowledge structure. The instantiation of this framework took place in the area of preparation for certification of proficiency in Japanese language. The validation of this framework was accomplished through the implementation of an instantiation of this in an existing educational support tool. Data from real users interaction with the tool are presented as partial validation.*

***Resumo.** Neste artigo é proposto um framework conceitual que permite a integração de hipermídias adaptativas em ambientes virtuais de aprendizagem a partir de uma perspectiva cognitivo-semântica. A abordagem cognitiva é baseada nos princípios da Teoria da Relevância. A abordagem semântica considera a utilização de uma ontologia de domínio como estrutura de conhecimento. A instanciação desse framework ocorreu na área de preparação para certificação de proficiência em língua japonesa. A validação do framework foi realizada através da implementação de uma instanciação deste em um ferramenta de apoio educacional. Dados da interação de usuários reais com a ferramenta são apresentados como validação parcial.*

1. Introdução

O uso de computadores na educação tem como objetivo promover a aprendizagem do aluno e ajudá-lo a desenvolver habilidades importantes para conseguir uma participação ativa na sociedade. Um dos problemas, relatados por Sternberg (2000), é que os alunos são tratados como tendo sempre o mesmo perfil, objetivos e conhecimento. O desenvolvimento de sistemas que se adaptam às necessidades de aprendizagem dos alunos é uma tarefa complexa. Aprender envolve vários fatores, incluindo uma comunicação eficaz em um contexto específico. Em várias situações do mundo real, nós sempre fazemos uso de informações de contexto, que nos permitem direcionar nossas

ações e comportamentos. Sistemas convencionais de educação, em geral, não levam em consideração o contexto atual do aluno.

Apesar de contexto poder ser entendido de diferentes maneiras, nesta pesquisa, trata-se de contextos cognitivos. Considerando isso, esta é uma proposta de um framework conceitual, que permite a integração de hiperlinks adaptativas em ambientes virtuais de aprendizagem sob uma perspectiva cognitiva-semântica. O contexto aqui considerado é baseado em princípios da Teoria da Relevância [Sperber e Wilson 1995]. A perspectiva semântica considerou o uso de uma ontologia de domínio como estrutura de conhecimento. O domínio escolhido foi a área de preparação para certificação do JLPT (Japanese Language Proficiency Test).

Esta pesquisa investiga a evolução de um sistema simulador de testes de proficiência existente para um sistema de apoio educacional com a adaptação de navegação, baseado no framework proposto.

Este artigo está organizado da seguinte forma: a próxima seção apresenta alguns trabalhos relacionados à aprendizagem da língua japonesa, mais especificamente em teste de proficiência, considerando suporte inteligente. Na seção seguinte é apresentado o framework conceitual com todos os seus elementos. Este framework foi instanciado usando uma validação funcional de casos de uso para cenários de adaptação identificados no ambiente, que é apresentado na seção 4. Essa instanciação permitiu a interação de usuários reais com o sistema. Essa interação e todo o processo de coleta e análise de dados foram realizados sob a forma de estudo de caso, e estão descritos na seção 5. Os dados coletados e alguns resultados preliminares são apresentados na seção 6. Conclui-se o artigo com algumas considerações finais e sugestões para trabalhos futuros.

2. Trabalhos relacionados

Alguns candidatos ao JLPT relataram a dificuldade em encontrar material de apoio específico para estudo na Web. Existem muitos softwares com material sobre a língua japonesa, especialmente sobre *kanji*, o que pode tornar a preparação deficiente ou tendenciosa pela falta de orientação e material direcionado. No entanto, há softwares disponíveis na Internet que dão suporte à preparação para o JLPT. O Quadro 1 apresenta algumas características consideradas importantes neste tipo de software.

Quadro 1. Comparação entre sistemas de apoio ao JLPT

Sistema	Características				
	Conteúdo específico para o JLPT	Voltado só para o tópico <i>kanji</i>	Quantidade satisfatória de questões	Construto cognitivo	Recursos de adaptatividade
Site oficial do JLPT	X				
JLPT Study Page	X				
Study Japanese	X	X	X	X	
Renshuu	X		X	X	X
JLPT Kanji Project	X	X	X		
Sistema e-JLPT	X		X	X	X

A quantidade de questões é uma característica importante, para evitar que o candidato exercite sempre com as mesmas questões. O sistema e-JLPT conta com mais de 300.

A utilização de um construto cognitivo ou a existência de tutoria mostra interesse em personalizar o processo de aprendizagem, exercícios e preparação para o teste. Existem três sistemas que utilizam esses recursos: Study Japanese, Renshuu e sistema e-JLPT. Os primeiros dois utilizam a metodologia de estudo de repetição espaçada [Balota *et al.* 2006]. O sistema e-JLPT utiliza a Teoria da Relevância como suporte cognitivo.

Dentre os softwares comparados, apenas o Renshuu e o e-JLPT possuem recursos de adaptatividade. O Renshuu adapta o conteúdo apresentado de acordo com o nível de conhecimento do aluno. O sistema e-JLPT adapta a navegação através do método de sugestão de tópicos de estudo relevantes. Pela tabela comparativa, o sistema e-JLPT mostra-se uma opção interessante a ser considerada e explorada por candidatos ao JLPT.

3. Framework

Frameworks são técnicas utilizadas no projeto de softwares orientados a objetos. A grande vantagem de seu uso é o potencial de reutilização de partes de software já implementadas e o desenvolvimento de novos componentes de software com a possibilidade de reutilização futura. O conceito de framework é implementado em diferentes paradigmas de programação, com muitas possíveis definições:

- Projeto de subsistema consistindo de um conjunto de classes abstratas e concretas e a interface entre elas [Wirfs-Brock e Johnson 1990];
- Estrutura genérica que pode ser expandida para criar um subsistema ou uma aplicação mais específica [Sommerville 2000];
- Conjunto de classes que colaboram para assumir a responsabilidade para um domínio de um subsistema da aplicação [Fayad *et al.* 1999];
- Arquitetura desenvolvida para atingir a máxima reutilização, representada por um conjunto de classes abstratas e concretas, com um grande potencial para especialização [Mattson 1996].

O cerne da construção de um framework é capturar o comportamento geral de um domínio de aplicação e modelar uma estrutura de controle capaz de representá-lo, ao invés de programação orientada a objetos com foco no desenvolvimento de uma solução para uma determinada aplicação específica. O desenvolvimento de software para uma implementação específica consiste em instanciar o framework referido, através da especialização de seus componentes.

3.1. O framework conceitual adaptativo

A proposta de elaborar um framework conceitual veio da necessidade de organizar o sistema e-JLPT, integrando os vários componentes desenvolvidos. Um dos objetivos de um framework conceitual é o de proporcionar um diagrama de classes que pode ser usado como a base para a adaptação de um ambiente virtual de aprendizagem, sob uma

perspectiva semântico-cognitiva. Um framework conceitual, orientado a objetos, de acordo com [Johnson 1997], não implica necessariamente em um produto acabado e executável, mas um esquema de dados conceituais utilizados para resolver um problema em um domínio específico.

O modelo computacional proposto utiliza ontologias de domínio como estrutura de conhecimento. O uso de ontologias está amplamente relacionado à portabilidade, à capacidade de adicionar novos conceitos e características, bem como à clareza, à reutilização, à extensibilidade, entre outros. A ideia de usar ontologias na área da educação está bem fundamentada, como pode ser visto em [Mohammed e Mohan 2007] [Mizoguchi e Bourdeau 2000] [Isotani *et al.* 2011]. O framework proposto pode ser visto na Figura 1. Seus principais componentes são:

Ambiente Virtual de Aprendizagem – esse componente é um subsistema contendo a base de hipermídia, as classes de alunos e professores. É a camada de interface do ambiente com os atores envolvidos, permitindo a definição de regras de perfis para adaptação às necessidades do aluno, visualização das hipermídias adaptadas semântico-cognitivamente, bem como das funcionalidades que o ambiente dispõe a cada ator envolvido no sistema.

Modelo de Adaptação – modelo que estabelece as diretivas para a adaptação do ambiente, especificando os componentes necessários para a base do mecanismo de adaptação. É formado por dois componentes: o método de sugestão de tópicos de estudo relevantes e pela ontologia de domínio sob a perspectiva da Teoria da Relevância.

Mecanismo de Adaptação – formado pelas bases de usuário e de domínio, obtidas a partir dos serviços de ontologias. É responsável por armazenar as ontologias (repositório semântico), extrair os dados do arquivo XML e armazenar nas bases de dados. Associada a estas bases, estão as bases específicas, com as regras de adaptação dos alunos, composta pela base Perfil Aluno e base Cognitiva do Aluno.

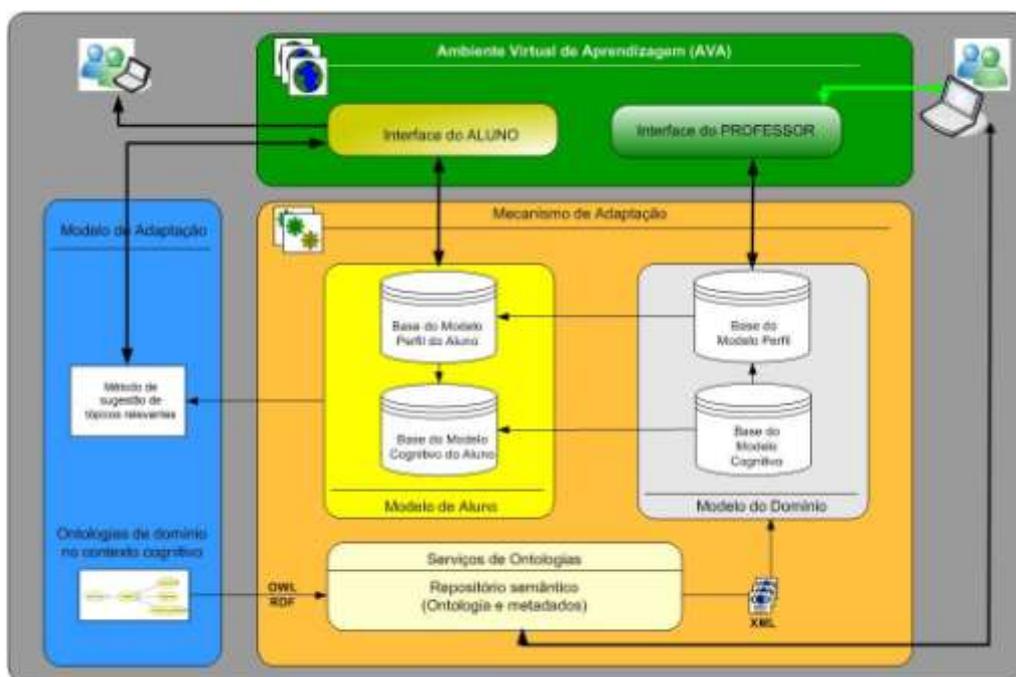


Figura 1. Framework conceitual

O framework proposto foi projetado para ter a participação de uma equipe interdisciplinar. Ontologias, por exemplo, podem ser desenvolvidas por engenheiros de ontologia, mas devem ser avaliadas por especialistas de diferentes áreas, como ciências cognitivas, educação e especificamente das áreas de domínio. Este envolvimento é definido como evidente e imprescindível por vários pesquisadores na área de engenharia de ontologias, sendo identificado por alguns como um fator de insucesso nos casos em que esse envolvimento não existe. O objetivo é, pois, envolver pesquisadores de várias áreas, refletindo a natureza interdisciplinar do desenvolvimento de um sistema para aplicar numa área que não é determinística, justificada por lidar com pessoas e com a forma como estas expandem seu conhecimento.

A próxima seção apresenta uma instanciação do framework proposto. Cada componente do framework conceitual apresenta um correspondente no trabalho desenvolvido. O processo de desenvolvimento do software e as ferramentas utilizadas serão apresentados. É também apresentado o processo de avaliação, que contou com a colaboração de especialistas e deu a perspectiva cognitiva à ontologia.

4. Instanciação

A instanciação do framework ocorreu na área de preparação para certificação de proficiência em língua japonesa. Uma instanciação do framework foi implementada usando uma validação funcional parcial de casos de uso para cenários de adaptação identificados no ambiente. A validação foi realizada através de um estudo de caso. Os principais elementos do framework instanciado são:

Ambiente virtual de aprendizagem – O sistema e-JLPT não é um ambiente virtual de aprendizagem completo, mas tem as principais funcionalidades para simular uma prova do JLPT.

Modelo de Adaptação – Modelo que estabelece as diretrizes para a adaptação do ambiente está baseado no método de sugestão de tópicos relevantes e na ontologia de domínio JLPT, sob a perspectiva da Teoria da Relevância.

Mecanismo de Adaptação – foi modelado no ambiente do e-JLPT a base de domínio e a base de perfil, absorvendo as características associadas ao JLPT. A proposta é modelar o domínio de preparação para um teste proficiência da língua japonesa levando-se em consideração as várias possibilidades de se navegar no contexto.

Alguns cenários de adaptação à simulação JLPT foram identificados:

- **Adaptação pelo aluno** – isso pode ocorrer através de dois casos de uso: (a) seleção dos contextos, tais como conhecimento da língua (gramática e vocabulário), compreensão auditiva e leitura, ou (b) perfis específicos, tais como metas relacionadas aos tópicos de estudo e de desempenho. Até o momento, somente a adaptação por perfis foi desenvolvida.
- **Adaptação pelo sistema** – isso pode ocorrer por um dos três casos de uso: (a) contextos que precisam de mais estudo, (b) relevância dos tópicos de estudo, ou (c) contextos relevantes. Até o momento, apenas a adaptação por relevância foi implementada.

A subseção seguinte descreve o processo de desenvolvimento do sistema e-JLPT e as ferramentas utilizadas no processo.

4.1. Processo de desenvolvimento e ferramentas

Para implementar o framework, ocorreram cinco estágios de desenvolvimento. Cada estágio teve a sua fase de testes, com a ajuda da ferramenta *Test::Unit*. Essas fases são descritas a seguir:

a) Migração da Linguagem de Programação

O framework proposto foi originalmente construído em ASP, mas não foi satisfatório porque a estrutura de diretórios do sistema era confusa, e as ligações entre os arquivos eram difíceis de controlar. O sistema e-JLPT passou por um processo de migração para Ruby on Rails, recebendo um código um pouco mais simples e mais ágil do que o original, executando as mesmas funções que tinha anteriormente. Rails veio como uma opção devido à sua filosofia. Essa filosofia diz que é melhor agir de acordo com as convenções do framework do que com as configurações do usuário. Isso melhorou a organização estrutural do sistema, e forneceu uma base de padrão de projeto para se trabalhar, o MVC (Model-View-Controller).

MVC é um padrão de projeto estrutural. Ele organiza o projeto em três camadas:

- Modelo – uma abstração em forma de objeto dos dados a serem apresentados, como um registro no banco de dados. Por exemplo, cada registro na tabela Aluno tem uma instanciação da classe Aluno, e cada campo na tabela representa um atributo do objeto. Essa abstração permite uma facilidade inata para implementar métodos específicos para cada tabela ou entidade do projeto.
- Controlador – ele faz a comunicação entre um modelo e suas visões. Normalmente realiza tarefas simples, como buscar um registro na base de dados de acordo com um URL, de modo que a visão correspondente tenha os dados para mostrar a página. Cada método do controlador tem uma visão correspondente.
- Visão – é a representação visual do objeto. É geralmente uma página HTML, mas também pode ser XML, JSON ou outros formatos.

Para o início do projeto, o banco de dados SQLite foi escolhido por ser simples de usar. Posteriormente foi adotado como banco de dados de teste para o desenvolvimento, deixando o MySQL como ambiente de produção, por ser mais robusto e preencher os requisitos da aplicação.

b) Primeira versão – novo layout e melhorias de funcionalidades

Após a migração, uma versão foi criada com um novo layout, para melhorar a usabilidade do sistema. Essa primeira versão se beneficiou de novas versões de tecnologias web, como HTML 5, que trouxe a possibilidade de usar o aplicativo de áudio, de forma inata no navegador, evitando o uso de recursos externos para sua implementação. Funcionalidades, tais como processo de registro e primeiro acesso ao sistema, foram simplificadas e toda a aplicação foi estabilizada e mantida durante todo o curso de desenvolvimento de e-JLPT.

c) Segunda versão – Módulo não-adaptado e função de importação da Ontologia

Com a versão estável já em funcionamento, o framework recebeu a implementação da função de importação da ontologia de domínio JLPT, em formato RDF, começando a transformação do sistema em um ambiente virtual de aprendizagem, com o Modelo de Adaptação proposto. A biblioteca utilizada foi a *RDF.rb*, uma biblioteca Ruby para lidar com os arquivos RDF, baseada na RDF Raptor, a partir da linguagem C.

Embora RDF tenha sido um dos pontos centrais do e-JLPT, não houve necessidade de uma grande quantidade de funcionalidades. Dessa forma, buscou-se por uma biblioteca minimalista, que não afetasse muito o fluxo de trabalho do framework. *RDF.rb* cumpriu este requisito.

Mas como era necessário vincular o Raptor RDF (cujo código era apenas para Linux), foi necessário migrar a aplicação para um ambiente Linux. Isso não gerou grandes problemas, uma vez que Ruby on Rails também é executado neste sistema operacional.

Com a ontologia de domínio integrada ao sistema, sob a forma de tópicos de estudo, o módulo que não possui adaptatividade foi desenvolvido, com a criação de testes JLPT a partir de um banco de questões de livre escolha do estudante.

d) Terceira versão - módulo Adaptado

No desenvolvimento do Módulo Adaptado, duas novas funcionalidades foram criadas:

- Permissão de especialistas na língua japonesa para definir a relevância de cada um dos tópicos de estudo apresentados na ontologia;
- Adaptação da geração de testes, baseada na Teoria da Relevância, com sugestões de tópicos relevantes para o aluno em cada momento de interação com o sistema, antes ou após a execução de uma simulação.

e) Implantação e Manutenção.

Primeiramente, SVN (Subversion Apache) foi usado para gerenciar as versões e-JLPT. Mas ocorreram vários problemas por causa das características internas do SVN. Decidiu-se, então, migrar para o Git.

O principal problema era o modelo de versões do SVN. Como o repositório do sistema não é organizado de acordo com a estrutura padrão deste controlador de versão, a dificuldade de implantação de versões no SVN, a fim de isolar o código das funcionalidades novas, seria mais custoso do que sem tal isolamento.

No Git, uma versão é apenas um ponteiro para um *commit*, que representa a árvore do estado de diretórios e arquivos naquele momento. Quando se cria um novo *commit*, a versão está apontando para o novo, que aponta para o *commit* anterior. Portanto, se houver mais de uma versão no projeto, é possível que a estrutura de *commit* se pareça com uma árvore. Quando duas versões diferentes estão ligadas (através de um comando chamado integração), um novo *commit* é criado, apontando para o último *commit* de cada ramo. Essa estrutura faz com que o gráfico de versões fique muito mais simples e mais facilmente manipulado.

Cada versão, por conseguinte, é completamente independente uma da outra (até que a integração seja feita). Se houver um estado testado e funcional do projeto e for necessário aplicar uma nova funcionalidade que pode tornar o sistema instável, uma versão é ideal para ter um ambiente isolado, que não afeta o estado estável do sistema. Se a funcionalidade é implementada com sucesso, só é necessário executar uma integração com a versão principal do projeto.

No final do processo de desenvolvimento, o e-JLPT estava disponível na Internet e liberado para o uso. Correção de pequenos erros e melhorias, resultado de sugestões e observações, foram desenvolvidas sem qualquer interferência nos objetivos principais do framework, para uma melhor interação entre o usuário e o ambiente.

5. Metodologia científica empregada

Segundo a tipologia apresentada por [Cervo e Silva 2007], esta foi uma pesquisa exploratório-descritiva, sob a forma de estudo de caso [Yin 2005]. Buscou-se observar os processos de interação dos alunos que utilizaram o sistema e-JLPT, com um mecanismo de sugestão de relevância, como recurso adicional para se preparar para certificação em língua japonesa. Quanto aos procedimentos e ao objeto, segundo Andrade (2009), esta foi uma pesquisa de campo, onde a coleta de dados é feita através de observação indireta, formulários e entrevistas. Este não foi um estudo onde se procurou encontrar causas ou consequências, mas sim, responder às questões de pesquisa aplicados os princípios da Teoria da Relevância em sistemas de hipermídia adaptativa, com estudos na influencia da navegação. Ainda, desejou-se investigar, através de um indicador de análise, as diferenças das navegações livres e guiadas.

A pesquisa aconteceu em duas etapas: (1) intervenção e coleta de dados e, (2) análise e interpretação dos dados. A intervenção foi realizada com quatorze alunos voluntários que podiam usar o sistema com flexibilidade. Sendo assim, o número de interações e a duração das mesmas foram muito diversos de indivíduo para indivíduo. Os dados foram coletados de várias fontes: arquivo de *log*, banco de dados, questionários e relatos. Após a coleta de dados, esses foram analisados e interpretados à luz da Teoria da Relevância. Com as múltiplas fontes de evidências, foi possível fazer a triangulação dos dados.

A unidade de análise adotada neste estudo foi a trajetória de aprendizagem do usuário aluno, ou seja, o caminho percorrido pelo usuário na ontologia. Para isso, todas as trajetórias de aprendizagem foram observadas. As trajetórias de aprendizagem do aluno com todos os dados associados a ela levam a uma melhor compreensão do tipo de navegação realizada no sistema, em termos de número de tópicos visitados.

6. Resultados iniciais

Na intenção de observar o comportamento do usuário nas duas versões do sistema, a adaptativa e a não-adaptativa, através da sua navegação pelos tópicos, foram monitorados os tópicos de estudo visitados.

Quatorze alunos voluntários utilizaram ambos os sistemas livremente e visitaram alguns tópicos de estudo. Um tópico de estudo era considerado visitado quando fazia parte da prova de simulado que o aluno realizara. Notou-se uma considerável diferença

entre os números de tópicos visitados nos dois sistemas. A Tabela 1 apresenta esses dados coletados pelo sistema para todas as interações realizadas.

Tabela 1 – Total de tópicos visitados

e-JLPT		e-JLPT Adapt	
Aluno	Total de tópicos	Aluno	Total de tópicos
1	5	1	8
2	0	2	8
3	2	3	9
4	0	4	4
5	1	5	13
6	2	6	8
7	32	7	1
8	40	8	0
9	41	9	16
10	41	10	2
11	24	11	0
12	7	12	3
13	22	13	4
14	1	14	0
Média	14,6	Média	4,67

Pelo formulário de avaliação juntamente com as informações obtidas pelas comunicações escritas, nota-se que apresentação dos tópicos e as relações entre os tópicos parecem estar adequadas em ambos os sistemas. A maioria dos alunos não achou que a quantidade de informação foi excessiva. Quanto à organização do conteúdo, a maioria relatou que não se sentiram perdidos em meio ao conteúdo.

Analisando os dados coletados pelo sistema é possível identificar uma diferença em termos do número de tópicos visitados entre o sistema e-JLPT tradicional e o e-JLPT Adapt – a média foi maior no sistema tradicional. Alguns alunos utilizaram objetivos de aprendizagem com muitos tópicos, o que levou a essa diferença.

Neste caso, parece que a sugestão de relevância influencia a navegação, levando a um número menor de tópicos escolhidos para os simulados. Nessa versão do sistema, vários usuários se mostraram mais focados, selecionando poucos tópicos, ao invés de selecionar todos os tópicos disponíveis.

7. Considerações finais

Este artigo apresentou a evolução de um sistema que incluía uma proposta de framework conceitual adaptativo. Sua primeira instanciação foi realizada no estudo de caso JLPT. Os principais componentes deste framework instanciado são: um sistema educacional chamado e-JLPT, um modelo de adaptação com base na ontologia de domínio JLPT e um mecanismo de adaptação com base no método de sugestão relevância proposto. Este estudo de caso permitiu uma validação parcial funcional do framework proposto com alguns dados provenientes da interação de usuários reais com o sistema.

A principal contribuição deste trabalho é a proposta de um framework para a integração de ambientes virtuais de aprendizagem e hipermídia adaptativa com base em

um contexto semântico-cognitivo e uma validação funcional parcial satisfatória em um software educacional existente.

Um possível trabalho futuro seria executar uma validação completa do contexto cognitivo, com ênfase nos princípios da Teoria da Relevância, através da realização de experimentos controlados. Além disso, também seriam necessárias a modelagem e a implementação de novos estudos de caso (instanciações) para obter a validação completa.

Referências

- Andrade, M. M. (2009) “Introdução à metodologia do trabalho científico: Elaboração de trabalhos na graduação”, 9 ed. São Paulo: Atlas.
- Balota, D.A., Duchek, J.M. and Logan, J.M. (2006) “Is expanded retrieval practice a superior form of spaced retrieval? A critical review of the extant literature”, In: J. Nairne (Ed.), *Foundation of Remembering: Essays in Honor of Henry L. Roediger, III*.
- Cervo, A. L., Bervian, P. A. and Silva, R. (2007) “Metodologia Científica”. São Paulo: Pearson Prentice Hall.
- Fayad, M.E., Schmidt, D.C. and Johnson, R.E. (Ed.) (1999) “Implementing Application Frameworks: Object-Oriented Foundations of Framework Design,” New York: J. Wiley, p. 345.
- Isotani, S., Bourdeau, J., Mizoguchi, R., Chen, W., Wasson, B. and Jovanovic, J. (Org.) (2011) “IEEE Transactions on Learning Technologies - Special Issue on Intelligent and Innovative Support Systems for CSCL”, 1st ed. IEEE Press, v. 4, 112 p.
- Johnson, R. E. (1997) “Frameworks = (Components+Patterns)”, *Communications of the ACM*, New York, v. 40, n.10, Oct., p. 39-42.
- Mattson, M. (1996) “Object-Oriented Frameworks: a survey of methodological issues”, PhD thesis. Dept. of Computer Science, UCK.
- Mizoguchi, R. and Bourdeau, J. (2000) “Using ontological engineering to overcome common AI-ED problems”, *International Journal of Artificial Intelligence in Education*, v.11, p.1-12.
- Mohammed, P. and Mohan, P. (2007) “Contextualizing Learning Objects Using Ontologies”, *Computational Intelligence*. Blackwell Publishing, v.23, n.3, p.339-355.
- Sommerville, I. (2000) “Software Engineering”, 6th ed. Boston: Addison Wesley.
- Sperber, D. and Wilson, D. “Relevance: Communication and cognition”, Oxford: Blackwell, 1995.
- Sternberg, R. J. (Ed.) (2000) “Handbook of Intelligence”, Cambridge: Cambridge University Press.
- Wirfs-Brock, R.J. and Johnson, R.E. (1990) “Surveying Current Research in Object-Oriented Design”, *Communications of ACM*, v.33, n.9, Sept., p. 104-1124.
- Yin, R. K. (2005) “Estudo de caso: planejamento e métodos”, 3 ed. Porto Alegre: Bookman, 2005.