

Um arcabouço que enfatiza a retroação

Diego Marczal¹, Alexandre Direne¹

¹C3SL – Departamento de Informática – Universidade Federal do Paraná (UPFR)
Caixa Postal 19.081 – 81.531-980 – Curitiba – PR – Brasil

{diego, alexd}@inf.ufpr.br

Abstract. *Initially, the paper presents the main aspects that justify the need for the design and implementation of a generic interactivity controller that can be applied and reused by a wide range of educational objects. The main objectives of the controller are detailed in terms of content access and the way it emphasizes meta-cognitive aspects involved with typical learning tasks. Moreover, it presents a new approach to use the errors made by the apprentice as a effective tool to repair and build your knowledge. At the end, the architecture and implementation of a framework called CARRIE (access Reflective and retroactive controller indexed by error) have their modules described and exemplified.*

Resumo. *Inicialmente, são apresentados os aspectos que justificam a necessidade de se projetar e implementar um controlador genérico de interatividade que pode ser aplicado e reutilizado por diversos objetos de aprendizagem. Os principais objetivos do controlador são detalhados em termos de acesso ao conteúdo e como ele enfatiza aspectos meta-cognitivos envolvidos nas tarefas típicas de aprendizagem. Além disso, apresenta-se uma nova abordagem para utilizar os erros cometidos pelo aprendiz como uma ferramenta eficiente para reparar e construir o seu conhecimento. Ao final, a arquitetura e a implementação do arcabouço denominado controlador de acesso reflexivo e retroativo indexado por erros (CARRIE) têm seus módulos principais descritos e exemplificados.*

1. Introdução

Atualmente, vários objetos de aprendizagem (OA) e simuladores têm sido desenvolvidos para apoiar tarefas de visualização de conceitos e solução de problemas em diversas áreas do conhecimento humano (e.g., Matemática, Física, etc). Por iniciativa coordenada do governo, boa quantidade desses objetos está disponível em bases de acesso público e gratuito do Ministério de Educação [BIOE/MEC 2009]. Esses OA possuem muitas características em comum, sendo que as principais estão relacionadas com a interface gráfica de acesso aos seus conteúdos educacionais. Todavia, a principal carência desses OA está na ausência de uma maneira de utilizar o erro do aprendiz como uma forma eficiente de mecanismo reparador e de aquisição de conhecimento. Outra fragilidade que pode ser destacada está na falta de um controlador genérico de interatividade que se aplique a todos os OA de maneira reutilizável e reconfigurável pelo autor do material de simulação e de solução de problemas.

Assim sendo, no processo de aquisição de conhecimento, mediado ou não pelo computador, é inevitável que o aprendiz cometa alguns erros. Os erros dos aprendizes são

comuns principalmente quando eles estão resolvendo exercícios, por exemplo, quando eles estão solucionando uma expressão aritmética. E é partir desse ponto que pode-se fazer com que o aprendiz reflita sobre como ele está abordando o problema proposto e se ele realmente possui o conhecimento necessário para resolvê-lo. Dessa forma, a partir dos erros é possível que o aprendiz amplie ainda mais seu conhecimento. Exemplos disso podem ser encontrados nos ambientes de aprendizagem baseados na arquitetura ACT* [Anderson 1983]. Nesses ambientes, um erro cometido leva o aprendiz a refletir e entender melhor suas ações e conceituações. A partir de uma abordagem sistemática, o ambiente consegue identificar se algo está incorreto no raciocínio do aprendiz durante a resolução de um problema. Caso algo inexato seja identificado, um *feedback* imediato é mostrado ao aprendiz para que ele então possa tentar novamente e encontrar soluções corretas a partir dos erros vistos.

Com isso, um exemplo de evento meta-cognitivo envolvido com o controlador é a possibilidade de o aprendiz retroceder ao contexto de erros cometidos no passado, uma vez que ele percebe em um dado instante a razão genérica por trás de sua incompreensão. De acordo com estudos recentes [Bull and Kay 2007], isso ajuda a promover condições para a reflexão sobre os passos adotados na solução de problemas e faz com que os mesmos caminhos de solução, ou novos caminhos, sejam tentados.

Desse modo, o desenvolvimento de software educacionais, geralmente é fundamentado em teorias que investigam como as pessoas aprendem e adquirem competências, além de investigar também como e porque elas cometem erros. Dentre essas teorias, as que mais se destacam são a ACT (*Adaptive Control of Thought*) [Anderson 1983], REPAIR [Brown and VanLehn 1988], e STEP [Vanlehn 1987]. Essas teorias foram tradicionalmente utilizados para lidar com erro em software de interface ativa. Diferentemente do proposto por este trabalho onde, a abordagem é de interface passiva, mas que traz recursos que apoiam a retroação do aprendiz a contextos de erros por meio da influência de parâmetros do próprio erro na ordem com que o acesso é oferecido, por meio de links, e suas prioridades estimadas.

Com base na busca bibliográfica realizada, verificou-se que ainda existe uma carência de abordagens e ferramentas de software destinadas a este propósito específico, o qual torna o contexto do erro uma forma rica de aprendizagem exploratória livre (*i.e.*, dos micromundos), sem a intervenção frequente das máquinas de tutoria tais como os STI. Além disso, também ressalta-se aqui a carência de enfoque da aplicação dos conceitos de Múltiplas Representações Externas (MREs) para a revelação dos erros. Por si só, essa conjugação de conceitos aponta indícios de originalidade do presente projeto de pesquisa e desenvolvimento de Mestrado.

2. Resenha literária

As linguagens de autoria podem ser classificadas como linguagens de programação peculiares. Elas são projetadas com o objetivo de proporcionar formas claras e de rápida assimilação para a construção de software educacional. Essas formas claras, na verdade, são Múltiplas Representações Externas (MRE) [Ainsworth 2006], conhecidas por elevarem o grau de abstração e de cobertura da descrição de conceitos de um domínio específico.

Criadas a partir de uma linguagem rica em MRE, as ferramentas de autoria são a implementação de compiladores ou interpretadores acoplados a ambientes de editoração.

Em geral, tais ferramentas são utilizadas por meio de técnicas de programação visual em um WYSIWYG de tal maneira que o autor não precise usar nenhum tipo de programação por de linhas de comandos.

Apesar de essa abordagem ser mais interessante do que escrever o código linha por linha, e limita o usuário ao projeto do sistema [Batista et al. 2003], pois todas as decisões de como desenvolver um OA já foram tomadas por que projeto a ferramenta. Como exemplos de sistemas de autoria voltados à criação de STI (Sistemas Tutores Inteligentes), é possível citar os ambientes EON [Murray 1998], RUI [Direne 1997] e SIMQUEST [van Joolingen et al. 1997].

Alguns ambientes de autoria do passado, como o RUI, tiveram o objetivo de ensinar apenas conceitos de sub-domínios visuais especializados (*e.g.*, Radiologia médica). Devido a isso, eles implementam apenas os modelos de domínio e pedagógico da arquitetura funcionalista interna de um STI. Outros foram mais genéricos, tal como o EON, e implementaram todos os quatro modelos especialistas da referida arquitetura de STIs, incluindo os modelos do aprendiz e da interação. O SIMQUEST, por sua vez, foi destinado à criação de simuladores baseados em ambientes exploratórios de aprendizagem.

Apesar das facilidades que esses sistemas de autoria oferecem, eles apresentam extrema rigidez gramatical em certas situações de editoração. No ambiente RUI, por exemplo, há a necessidade do autor conhecer sentenças em Lógica de Predicados, o que significa uma dificuldade para leigos. Adicionalmente, pouco foi relatado sobre o funcionamento do mecanismo essencial da *shell* interpretadora do modelo de aprendiz do ambiente EON. Quanto ao SIMQUEST, nenhuma pesquisa evidenciou o potencial dos seus aspectos meta-cognitivos que afetam o controle que o usuário final (aprendiz) tem sobre o acesso aos conteúdos inseridos pelo autor das simulações.

Os STIs como ambientes inteligentes que, mesmo realizando diagnóstico ativo, não possuem nenhum registro de sucesso em abordagens de acompanhamento (monitoramento) de mais longo prazo do aprendiz. De fato, pode-se dizer que os STIs desenvolvidos baseados nas teorias ACT, REPAIR e STEP atuam de uma forma que fornecem apenas ajuda ao aprendiz em formato textual a curto prazo.

2.1. Ambientes exploratórios para aprendizagem (AEA)

No que diz respeito aos AEAs, pode-se defini-los como ambientes de auto-estudo, onde a aquisição de conhecimento se dá por atividades de exploração investigação e descoberta. Um exemplo, deste tipo de ambiente é o LOGO [Papert 1980]. O qual enfatiza que o erro do aprendiz não é algo negativo mas sim uma forma de aprendizagem onde o conhecimento é construído com bases em experiências no passado.

Uma das fragilidades encontras nesses ambientes está no fato que não possuem nenhum tipo de mecanismos de recuperação de erros, alguns deles chegam a identificar os erros dos aprendizes mas, nem como interface passiva são capazes de dar mensagens mais elaboradas aos aprendizes. O que de fato daria um maior potencial de flexibilidade ao aprendiz [Puntambekar and Hübscher 2005, Puntambekar and Stylianou 2003, Puntambekar et al. 2003].

2.2. Limitações do OAs para ciências físicas e matemáticas

Atualmente muitos dos OAs atuam de maneira puramente somativa [Márcia G. de Oliveira 2009] e são raros os que atuam de um maneira somativa [Pimentel et al. 2007, Zanella 2005].

De forma resumida, OAs com caráter de avaliação somativa buscam auxiliar o julgamento de classificação dos aprendizes no final de uma determinada aprendizagem. Já os OAs com caráter formativo, acompanham todo o processo de aprendizagem fornecendo *feedback* sempre que acharem necessário. Com isso, os de caráter formativo induzem a reflexão do aprendiz sobre seu aprendizado e conduzem-no a um melhor desempenho [Pimentel et al. 2007]. Dessa forma so OAs formativos, apesar de raros, são os que possuem maior potencial de utilidade para as áreas de ciências físicas e matemáticas.

3. Conceitos fundamentais da solução

A *abstração* é o processo por meio do qual os aprendizes criam entidades mentais que servem como base para novos procedimentos e a formação de conceitos [Ainsworth 2006]. Nesse sentido, os aprendizes podem construir referências através das MREs para que a estrutura subjacente do domínio representado seja eventualmente exposta e tratada como um novo conhecimento, ainda mais genérico. Estudos passados demonstram que a aprendizagem realizada com pares de representações externas leva a uma aquisição de conhecimento mais abstrato [Schwartz 1995].

Neste projeto de pesquisa e desenvolvimento, a construção de conhecimento aprofundado foi planejada para ocorrer por meio de abstração por generalização. A partir disso, as MREs selecionadas nos diversos software do projeto são pelos menos em pares. Por exemplo, para o software de progressões geométricas (PG), elas são: (a) expressões analíticas organizadas passo-a-passo; (b) figuras da *geometria fractal*, as quais guardam correspondência com as várias versões de expressões do item anterior. As figuras de fractais possuem estágios de formação que podem ser chamados de passos. Em cada um desses passos, as alterações geométricas provocam crescimento ou decréscimo em medidas de tamanho do menor lado, perímetro, sub-áreas circunscritas e alguns outros aspectos com potencial para refletir regularidades matemáticas tais como as propriedades de PG. Além disso, por os fractais não possuírem um número finitos de passos seu uso pode-se auxiliar na aquisição de conceitos mais abstratos.

Nessa sequencia de ideias, o aspecto mais importante para o desenvolvimento da capacidade de abstração está na maneira com que um aprendiz é levado a generalizar cada vez mais as expressões analíticas para que elas reflitam as grandezas de cada passo do fractal. Dessa forma, argumenta-se que um software pode levar à boa prática do desenvolvimento da capacidade de abstração do aprendiz se este último for envolvido com o reconhecimento e a descrição analítica precisa de representações passo-a-passo, tais como as da formação de fractais.

Mais especificamente, o presente projeto investigou como a generalização por indução poderia ser atingida tomando-se os passos de formação de figuras fractais como sendo compostas de progressões geométricas. Por definição uma PG é uma seqüência numérica em que cada termo, após o segundo, é igual ao produto do termo anterior por uma constante q . Por exemplo, a seqüência $(1, 2, 4, 8, 16, 32, \dots)$ possui razão igual a 2. Com





| PASSO | NÚMERO DE TRIÂNGULOS | TAMANHO DO LADO DO TRIÂNGULO (16) |
|---|----------------------|-----------------------------------|
|  | 1 | 3 |
|  | 2 | 9 |
|  | 3 | 27 |
|  | 4 | 81 |

Tabela 1. Passos de formação do Triângulo de Sierpinsky

essa definição em mente, pode-se observar a Tabela 1, a qual apresenta quatro passos da formação do fractal Triângulo de Sierpinsky. Tanto a quantidade de triângulos formados nos diferentes passos como o tamanho do menor lado de cada triângulo formam PG. Na verdade, mais elementos matemáticos desse fractal formam PG ao longo das transformações entre os passos. O objetivo da Tabela 1 é mostrar parte do processo de indução do aprendiz por meio de uma aprendizagem passo-a-passo que pode ser precisamente guiada com a ajuda de um software.

Na referida tabela, é possível observar a formação de ao menos duas PGs: uma crescente e outra decrescente. A primeira está relacionada com a quantidade de triângulos e a segunda com o tamanho do lado do menor triângulo, ambas consideradas em cada passo. Observe que, se o tamanho do lado do triângulo inicial for definido com o valor genérico L , a divisão sucessiva dos lados em seus pontos médios leva gradualmente à sequência $(\frac{L}{2}, \frac{L}{4}, \frac{L}{8}, \frac{L}{16}, \dots)$. Esse processo pode ser repetido infinitamente sendo que em cada passo, uma figura com triângulos cada vez menores será formada. Além disso, o importante mesmo do ponto de vista de generalização é que o aprendiz deverá descrever por si só o tamanho do lado no n -ésimo passo (passo de ordem n) como sendo $\frac{L}{2^n}$.

Essa representação passo-a-passo exige do aprendiz um desenvolvimento gradual de sua capacidade de abstração. Os conhecimentos matemáticos intrínsecos de um fractal fazem com que o aprendiz construa uma referência sobre os conceitos visualmente estruturados por bases analíticas. Isso parece tornar intuitivo o entendimento do assunto através do uso de MREs adequadas para serem manipuladas tanto por um software como por um humano, mantendo ainda a ligação direta com elementos concretos da rotina didática (*i.e.*, de PG).

4. Arquitetura

Para consolidar os fundamentos da aprendizagem adotados até o momento no presente projeto de pesquisa e desenvolvimento, diversas ferramentas de software educacional estão sendo implementadas e validadas no ambientes escolar. A abordagem arquitetural de todas elas é baseada na existência de um núcleo comum de software, o qual é um arcabouço chamado aqui de Controlador de Acesso Reflexivo e Retroativo Indexado por Erros (CARRIE).

4.1. Organização funcionalista do CARRIE

Conforme representado na Figura 1, quatro módulos principais compõem a arquitetura funcionalista do CARRIE, são eles: Módulo de acesso ao conteúdo, Módulo indexador

de erros, Módulo Facilitador e a Interface com o aprendiz. Todos esses módulos serão detalhados na sequência.

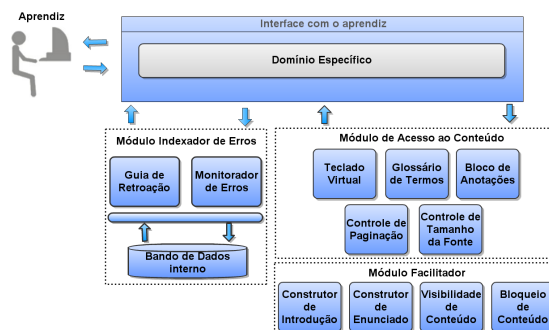


Figura 1. Arquitetura Funcionalista do CRI

Foram implementados, completamente, quatro OAs que fazem uso do arcabouço CARRIE, sendo eles para o domínio de Progressões Geométricas em Fractais, o segundo para o domínio de Funções de Primeiro Grau, o terceiro para o domínio da Matemática Financeira por fim o último destina ao ensino de Funções Periódicas. Os de quatro software educacionais fazem parte da fase inicial de construção e aplicação de apoio computacional a atividades laboratoriais do ensino escolar de conceitos matemáticos do nível médio. A iniciativa se destina ao cumprimento de metas no âmbito do Projeto XXX [REF. OMITIDA], financiado com recursos do Edital conjunto MCT/MEC 001-2007.

Para a implementação do CARRIE, foi utilizada a plataforma Java seguindo as técnicas do paradigma de programação Orientada a Objetos. O software foi projetado para ser executado em qualquer navegador *Web*, independente de sistema operacional ou hardware. Além disso, o código é divulgado como código livre sobre licença GPL (Licença Pública Geral).

4.2. Módulo indexador de erros

Este módulo é composto por 2 partes principais, o monitorador de erros e o guia de retroação.

Monitorador de Erros: O monitorador de erros funciona com base em pontos de observação definidos pelo autor do conteúdo. Mais precisamente, o autor precisa necessariamente marcar, através da chamada de um método, onde o aprendiz comete um erro. Os parâmetros desse método são o título do erro, o texto explicativo do erro e uma variável booleana. Quando essa variável for verdadeira, uma janela *popup* será mostrada, alertando o aprendiz para o erro. A área visual do *popup* na tela tem como conteúdo o título e o texto explicativo passado como parâmetro para o método. Caso seja falsa, nada será mostrado.

No momento que esse método é chamado, o monitorador de erros recebe uma mensagem indicando um desvio conceitual cometido pelo aprendiz. Após o recebimento dessa mensagem, o monitorador salva o erro e o estado da aplicação em que o erro ocorreu em um banco de dados interno. Com essas informações salvas, o CARRIE estrutura o seu comportamento para que o aprendiz possa retroceder ao momento exato em que o erro

foi cometido. Sendo assim, é apresentado ao aprendiz um guia de retroação, por meio do qual o aprendiz pode retroceder a qualquer erro cometido durante seu aprendizado.

Guia de Retroação: O guia de retroação permite que o aprendiz inspecione momentos exatos de erros passados e decida a quais deles vale a pena retroceder. Este guia faz a leitura do banco de dados interno de erros e cria um menu contendo um *link* para cada erro armazenado no banco de dados. Esse menu é mostrado ao aprendiz só depois de existir pelo menos um erro cometido. Quando o menu está visível, o aprendiz pode visualizar o título do erro e o seu texto explicativo. Passando-se o *mouse* sobre o *link* do erro, a qualquer momento pode-se retroceder ao erro cometido apenas clicando no *link*.

Quando o aprendiz retrocede ao erro, o CARRIE possibilita que ele refaça todo o exercício em que o erro foi cometido. Tal abertura de tarefas promove fundamentalmente o que é chamado de atividade reflexiva e permite que o aprendiz tente revisitar os aspectos que o levaram ao erro. Caso o aprendiz não queira refazer todo o exercício novamente (*e.g.*, por sentir que ainda tem dúvidas), ele também pode desistir da tentativa. Vale a pena ressaltar aqui que, ao contrário dos tradicionais recursos de desfazer simplesmente uma tarefa (*undo*, em inglês), o guia de retroação atinge um potencial meta-cognitivo muito mais adequado à finalidade pedagógica do CARRIE. Acredita-se que essa característica lhe confere um bom grau de originalidade em relação aos ambientes existentes.

4.3. Módulo de acesso ao conteúdo

É um conjunto de módulos responsáveis por apresentar o conteúdo ao aprendiz. Ele está dividido nos seguintes sub-módulos:

Glossário de termos: Através do CARRIE, sempre que o autor achar necessário que um glossário de termos seja disponibilizado ao aprendiz, basta definir um arquivo respeitando uma indentação pré-estabelecida. Quando o CARRIE identificar que um arquivo que segue essa estrutura foi criado, ele automaticamente disponibilizará a interface responsável pela apresentação do glossário por meio do *link* de acesso. Além disso, o glossário ainda conta com uma busca por palavra-chave baseada no recurso de *auto-completar*.

Controle de tamanho de fonte: Um aspecto importante para os softwares atuais é a acessibilidade. Devido a isso, o CARRIE oferece um controle para que o aprendiz atue sobre o tamanho de letras de qualquer texto do OA.

Controle da paginação: O CARRIE tem como um dos objetivos fazer com que o aprendiz desenvolva seu conhecimento passo-a-passo para atingir a abstração por generalização. Para que isso ocorra, há um módulo que é responsável por oferecer paginação de todo e qualquer conteúdo a ser apresentado.

Teclado Virtual: Alguns OAs necessitam manipular uma entrada de dados do tipo numérico-analítica. Para isso, é necessário um estilo diferente de mecanismo de interação. Esse mecanismo é um teclado virtual, que é disponibilizado pelo CARRIE para o autor do conteúdo fazer uso em seu OA.

Bloco de anotações: Um aspecto que pode ser considerado importante para o aprendizado é o bloco de anotações. Nele o aprendiz pode realizar anotações de todo o conteúdo que achar importante e, dessa forma, pode construir sua própria percepção do tema em foco. A partir disso, o CARRIE oferece um bloco de anotações ao aprendiz com recursos para destacar o texto em diversas cores.

Em outras, todas essas funcionalidades estão disponíveis, através do CARRIE, ao autor de material eletrônico não precisa se preocupar com esse tipo de código.

4.4. Módulo Facilitador

Os sub-módulos contidos no módulo facilitador são destinados a tornar o desenvolvimento de um OA, com o uso do CARRIE, mais simples e com menos preocupação em codificação. Para isso, o CARRIE oferece três submódulos: (a) Construtor de Introdução; (b) Construtor de Enunciado; (c) Bloqueio de conteúdo e Visibilidade de Conteúdo.

Os construtores de introdução e enunciado permitem o autor do conteúdo ampliar a formatação de imagens e textos através da codificação em html. Além disso, também permitem ao autor definir links de palavras, que quando clicadas abrem o glossário de termos com a sua respectiva definição em foco.

Por outro lado, certos conteúdos podem ser bloqueados ou escondidos pelo autor para que o aprendiz os acesse somente quando for apropriado. Isso oferece controle indireto do professor sobre os alunos espalhados no laboratório.

4.5. Interface com o aprendiz

Uma interface com o aprendiz é pré-estabelecida pelo CARRIE e está dividida em três partes principais. A primeira é destinada ao título do OA, juntamente com o título da página atual. Em seu canto superior direito, também estão as opções para aumentar e diminuir o tamanho das letras. A segunda parte é destinada ao domínio específico. A outra é reservada às opções da dinâmica de controle oferecida ao aprendiz. Exemplos de tais controles são: pagnar para frente ou para trás, botão de acesso ao glossário, botão de acesso ao guia de retroação e o botão de acesso ao bloco de anotações.

A Figura 2 mostra aspectos gráficos da interface em um certo instante da interação com o OA sobre força e deslocamento de molas. No quadro estão ressaltados como as constantes elásticas das molas influenciam os gráficos das funções de primeiro grau de cada uma delas.

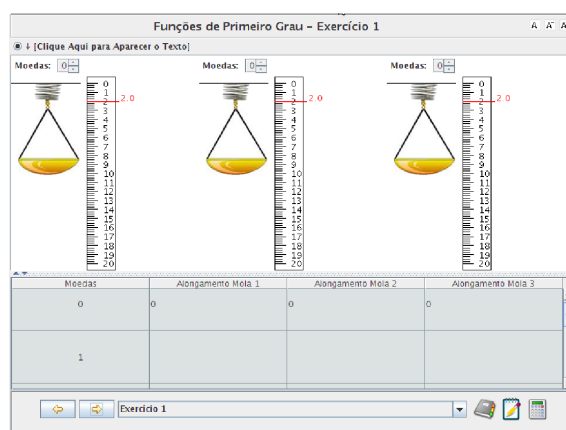


Figura 2. Interface do OA sobre funções de primeiro grau

5. Conclusão e trabalhos futuros

Neste trabalho foram apresentadas tanto a necessidade como a proposta de um controlador de interatividade genérico que evidencie os aspectos meta-cognitivos envolvidos em

tarefas típicas de aprendizagem. Dentre os quais destaca-se a retroação a contexto de erros cometidos no passado. Elucidou-se também a carência de pesquisas que que diz respeito a este nicho específico.

Destacou-se na resenha literária os aspectos que dizem respeito às linguagens e ferramentas de autoria juntamente com as limitações dos objetos de aprendizagem e dos ambientes exploratórios de aprendizagem. Da mesma forma, as representações externas foram elucidadas como diretrizes fundamentais para o desenvolvimento de interfaces educacionais. Para demonstrar o conceito de conhecimento aprofundado através da abstração por generalização, foi descrito o uso de fractais para o ensino de progressões geométricas e sua extensão para outros domínios por meio de uma arquitetura genérica.

Finalmente as perspectivas futuras apontam naturalmente para um maior aprofundamento das formas de retroação às situações onde o aprendiz cometeu erros. A ideia de registrar e restaurar os quadros de erro ainda é feita de maneira quase linear no arcabouço atual do controlador CARRIE. Adicionalmente, a iniciativa mais próxima de pesquisas do passado que ofereceram ao aprendiz uma visão de ambientes de aprendizagem com recursos para os usuários inspecionarem o que o software assumiu sobre eles foi chamada de modelos abertos de aprendizes (*open student models*) [Zapata-Rivera and Greer 2002]. No entanto, tais iniciativas não contemplaram a criação e o uso de linguagens de descrição da configuração com que os erros do aprendiz ocorreram para que tais erros pudessem ser revisados em outras condições no futuro.

Além disso, como perspectivas futuros podemos citar expansão do mecanismo de retroação para não apenas dar a possibilidade do aprendiz voltar ao erro. Mas também para direcionar o aprendiz em quais assuntos, sejam mais específicos ou mais gerais, ele deve ter mais atenção para obter um compreensão clara do assunto abordado. No mesmo sentido, busca-se também validar e aperfeiçoar essa pesquisa por meio de uma aplicação da ferramenta em ambiente de ensino.

Referências

- Ainsworth, S. (2006). Deft: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, 16(3):183–198.
- Anderson, J. R. (1983). *The architecture of cognition / John R. Anderson*. Harvard University Press, Cambridge, Mass. :.
- Batista, L., Direne, A., Trindade, J., Gimenes, I., and Taha, O. (2003). Autoria de diretrizes pedagógicas destinadas ao treinamento das múltiplas capacidades da perícia em conceitos visuais. In *SBIE2003 - Simpósio Brasileiro de Informática na Educação*, pages 573–582.
- BIOE/MEC (2009). Ministério da educação, banco internacional de objetos educacionais. <http://objetoseducacionais2.mec.gov.br>.
- Brown, J. S. and VanLehn, K. (1988). Repair theory: A generative theory of bugs in procedural skills. In Collins, A. and Smith, E. E., editors, *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, pages 338–361. Kaufmann, San Mateo, CA.

- Bull, S. and Kay, J. (2007). Student models that invite the learner in: The SMILI open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120.
- Direne, A. (1997). Designing intelligent systems for teaching visual concepts. *International Journal of Artificial Intelligence in Education*, (8):44–70.
- Murray, T. (1998). Authoring knowledge based tutors: tools for content, instructional strategy, student model, and interface design. *Jnl. of the Learning Sciences*, 7(1):5–64.
- Márcia G. de Oliveira, Elias Oliveira, R. Z. M. (2009). Um qasystem para interação de alunos em avaliações somativas a distância. In *XX SBIE - Simpósio Brasileiro de Informática na Educação (SBIE-2009)*, Florianópolis-SC, Brasil. SBC.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.
- Pimentel, E., Alves¹, A. S., Oliveira, B. W. R., Ikebara, D. M., Bottaro, P. A., and Lopes, R. (2007). Avaliações adaptativas baseadas no nível de aquisição de conhecimentos do aprendiz. In *XVIII SBIE - Simpósio Brasileiro de Informática na Educação (SBIE-2007)*, São Paulo, Brasil. SBC.
- Puntambekar, S. and Hübscher, R. (2005). Tools for scaffolding students in complex learning environments: What we have gained and what we have missed? *Educational Psychologist*, 40:1–12.
- Puntambekar, S. and Stylianou, A. (2003). Designing metacognitive support for learning from hypertext: What factors come into play? *AIED*.
- Puntambekar, S., Stylianou, A., and Hübscher, R. (2003). Improving navigation and learning in hypertext environments with navigable concept maps. *Hum.-Comput. Interact.*, 18(4):395–428.
- Schwartz, D. L. (1995). The emergence of abstract representations in dyad problem solving. *The Journal of the Learning Sciences*, 4(3):321–354.
- van Joolingen, W. R., King, S., and Jong, T. (1997). The simquest authoring system for simulation-based discovery learning. In du Boulay, B. and Mizoguchi, R., editors, *Frontiers in Artificial Intelligence and Applications: Proceedings of World Conference on Artificial Intelligence in Education (AI-ED 97)*, pages 79–86. IOS Press.
- Vanlehn, K. (1987). Learning one subprocedure per lesson. *Artif. Intell.*, 31(1):1–40.
- Zanella, R. (2005). Easy : sistema de avaliações via web baseado no hyper-autonaton. Master's thesis, Universidade Federal do Rio Grande do Sul. Instituto de Informática. Programa de Pós-Graduação em Computação.
- Zapata-Rivera, D. and Greer, J. E. (2002). Exploring various guidance mechanisms to support interaction with inspectable learner models. In *Proceedings of the Conference on Intelligent Tutoring Systems (ITS-2002)*, pages 442–452.