

# UML para Modelagem de Objetos de Aprendizado

Alberto Bastos do Canto Filho<sup>1</sup> & Alexandre Ribas Semeler<sup>2</sup> & Herik Zednik Rodrigues<sup>3</sup> & Liane Margarida Rockenbach Tarouco<sup>3</sup>

<sup>1</sup> Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul - Av. Osvaldo Aranha, 103 - CEP: 90035-190 - Porto Alegre - RS – Brasil (alberto.canto@ufrgs.br)

<sup>2</sup> Instituto de Geociencias da Universidade Federal do Rio Grande do Sul - Av. Bento Gonçalves, 9500 Prédio 43113 91540-000 - Porto Alegre - RS – Brasil (alexspleen@yahoo.com.br)

<sup>3</sup> Programa de Pós-Graduação em Informática na Educação da Universidade Federal do Rio Grande do Sul- Av. Paulo Gama, 110 - prédio 12105 - 3º andar sala 332 90040-060 - Porto Alegre (RS) – Brasil(hzednik@hotmail.com e liane@penta.ufrgs.br)

**Resumo.** O presente artigo apresenta necessidades específicas de projetos de objeto de aprendizado, das quais decorre a necessidade de uma linguagem adequada de projeto. Não obstante diversas propostas venham sendo apresentadas, esta diversidade termina por depor contra o próprio objetivo de estabelecimento de um padrão, utilizado para fins de comunicação e especificação. Apresenta-se a proposta de utilização da UML - *Unified Modelling Language*. É também apresentado um exemplo de objeto de aprendizado cujo desenvolvimento fez uso da UML.

**Palavras-Chave:** Objetos de Aprendizagem, Unified Modeling Language, Design de Interação.

## Introdução

A Engenharia de Software e os paradigmas de Programação Orientada a Objetos vêm sendo trabalhados desde a origem dos dispositivos programados. Possuem uma longa história da qual resulta um determinado estágio de amadurecimento que, embora ainda não tenha alcançado o estado da arte, encontra-se num nível de estabilidade ainda não alcançado pelos Objetos de Aprendizagem.

Apesar de possuírem características próprias, os Objetos de Aprendizagem possuem muitas similaridades com os objetos produzidos dentro do paradigma de Programação Orientada a Objetos, o que proporciona a oportunidade de estabelecer paralelos que permitam a adaptação de conceitos e técnicas amadurecidas durante um ciclo histórico mais longo, da Engenharia de Software.

Estas analogias devem ter seus limites claramente definidos, aproveitando similaridades que permitam valer-se dos conceitos consolidados na Engenharia de Software. Simultaneamente, devem-se destacar as diferenças, de forma a promover as adaptações necessárias, ou até mesmo o descarte de determinados conceitos e práticas que não possuam a aderência necessária.

Estudos mostram que a “maior parte dos erros de um sistema – 64% – está associada às fases de análise de requisitos e projeto, e só são descobertos em etapas mais avançadas como codificação e testes” [1]. Neste sentido, é fundamental para o sucesso de um software uma comunicação eficiente durante a fase inicial do processo de desenvolvimento com a finalidade de evitar falhas nos projetos.

Neste artigo, defende-se o uso da *Unified Modeling Language* - UML como ferramenta de desenvolvimento de Objetos de Aprendizagem com vistas a facilitar a comunicação entre o Projeto Pedagógico e o Projeto de Informática, apresentando um exemplo prático de Objeto de Aprendizagem modelado e planejado utilizando UML.

## Representação de Modelos Instrucionais

O Desenvolvimento de Modelos Instrucionais é a região de interface entre a Engenharia de Software e o Desenvolvimento Instrucional, devendo traduzir os anseios do *Instrutor* para uma linguagem adequada para o *Desenvolvedor de Módulos*. Numa analogia com a construção de edifícios, deve existir uma linguagem capaz de expressar a necessidade do usuário, aquele que deseja uma residência que atenda suas necessidades e expectativas, capaz de agregar a experiência do arquiteto, conhecedor dos recursos disponíveis dentro das tecnologias existentes, e capaz de passar ao engenheiro civil uma especificação clara do imóvel desejado, a partir da qual se realize o cálculo das estruturas e se execute a obra. Nesta área da Engenharia Civil e Arquitetura, os proibitivos custos de corrigir um erro após a construção do imóvel levaram ao desenvolvimento de uma linguagem de clara compreensão entre estes três agentes (o contratante, o arquiteto e o engenheiro civil), de forma que, através de uma planta, seja possível fazer uma especificação completa. Apenas na planta arquitetônica se realizam ensaios e mudanças de especificações que levem a uma especificação final. Alterações se realizam apenas *no papel*, pois os custos de mudanças após a conclusão da obra são impraticáveis.

O Desenvolvimento de Software não possui custos de materiais expressivos como a construção civil, o que permitiu o desenvolvimento de técnicas de prototipação rápida como forma de melhor trabalhar o processo de especificação. Observe que a prototipação rápida ainda é uma técnica de especificação, portanto preliminar à etapa de construção de software. Por outro lado, os custos da mão de obra da *construção de software* são expressivamente maiores do que os custos da mão de obra da *construção civil* o que reitera a necessidade de evitar retrabalhos e custos decorrentes de especificações erradas ou incompletas.

A questão que se coloca neste trabalho é: até que ponto estão amadurecidas as formas de comunicação entre os *Desenvolvedores Instrucionais* e os *Desenvolvedores de Software*? Deve-se buscar uma linguagem única que promova a comunicação eficaz entre estes dois agentes? Ou cabe apenas ao Analista de Sistemas a confecção e leitura de diagramas elaborados numa linguagem desconhecida do Desenvolvedor Instrucional?

Projetos de pequeno porte, concebidos e implementados por uma única pessoa poderão dispensar a utilização de modelos. No entanto, projetos de maior porte, que envolvam muitos *HH* (Homens \* Hora) associados a requisitos de prazo exigem o

trabalho em equipe. E o trabalho em equipe somente é possível se houver mecanismos eficientes de documentação e especificação. Na eficiência da comunicação e da especificação, fundamenta-se o desenvolvimento de todas as linguagens de projeto, sejam eles projetos de arquitetura, engenharia mecânica, elétrica, engenharia de software ou até mesmo a elaboração de partituras para uma orquestra sinfônica.

A necessidade de comunicação entre o Desenvolvedor Instrucional e a equipe de Desenvolvimento de Software caracteriza um contexto muito similar àquele que levou à especificação da UML (*Unified Modeling Language - Linguagem de Modelagem Unificada*). Neste contexto envolve-se um grande público já conhecedor desta linguagem (os desenvolvedores de software) e um novo perfil profissional o *Desenvolvedor Instrucional*. Observe que a definição proposta pela OMG (1999) a caracteriza como uma ferramenta adequada para os fins de desenvolvimento de objeto de aprendizado: “*linguagem para especificação, visualização, construção e documentação de artefatos de sistemas de software, assim como para modelagem de negócios e outros tipos de sistemas. A UML representa uma coletânea das melhores práticas de engenharia que se mostraram vitoriosas na modelagem de sistemas grandes e complexos*” [2].

Caracterizado como uma nova área de projeto, o *Desenvolvimento Instrucional* deverá evoluir para uma linguagem própria, ainda em construção. Dada a origem desta nova área, a sua proximidade com os paradigmas de Programação Orientada a Objetos e o perfil de grande parte dos profissionais atuantes, o uso da UML como ponto de partida é uma opção natural.

A seguir, serão apresentadas considerações sobre o uso atual e potencial da UML como linguagem para especificação de Projetos Instrucionais.

### **Diagramas de Classe e Diagramas de Atividades**

Dentre os diagramas da UML, o Diagrama de Classes é o que mais frequentemente tem sido utilizado como ferramenta de modelagem instrucional. “Diagramas de Classe representam a estrutura estática mais importante da modelagem UML, sendo formados por classes de objetos com características e comportamento semelhantes e seus relacionamentos mais comuns, como associações, agregações, heranças e dependências” [1]. A Figura 1.a, a seguir mostra exemplos de utilização de diagramas de classe como ferramentas de modelagem, utilizadas por Katrien Verbert e Erik Duval [3].

Além dos diagramas de classe, também os diagramas de atividades são ferramentas UML utilizadas com certa frequência no contexto do projeto instrucional. “O objetivo do diagrama de atividades é mostrar o fluxo de atividades em um único processo. O diagrama mostra como uma atividade depende da outra. Um diagrama de atividades pode ser regiões denominadas *swimlanes*. Estas regiões são associadas a um objeto do modelo. Desta forma, dentro de cada região, encontram-se as atividades relativas ao objeto da região” [5].

A figura 1.b apresenta um diagrama de atividades relativo a um estudo de caso “Problem Based Learning” [6]. Os arcos funcionam como conexões que evidenciam as dependências entre as atividades. É importante definir os fluxos de eventos

principais quando se efetua um caso de uso, que por sua vez, é representado pelo diagrama de atividades a partir de todos os fluxos de eventos.

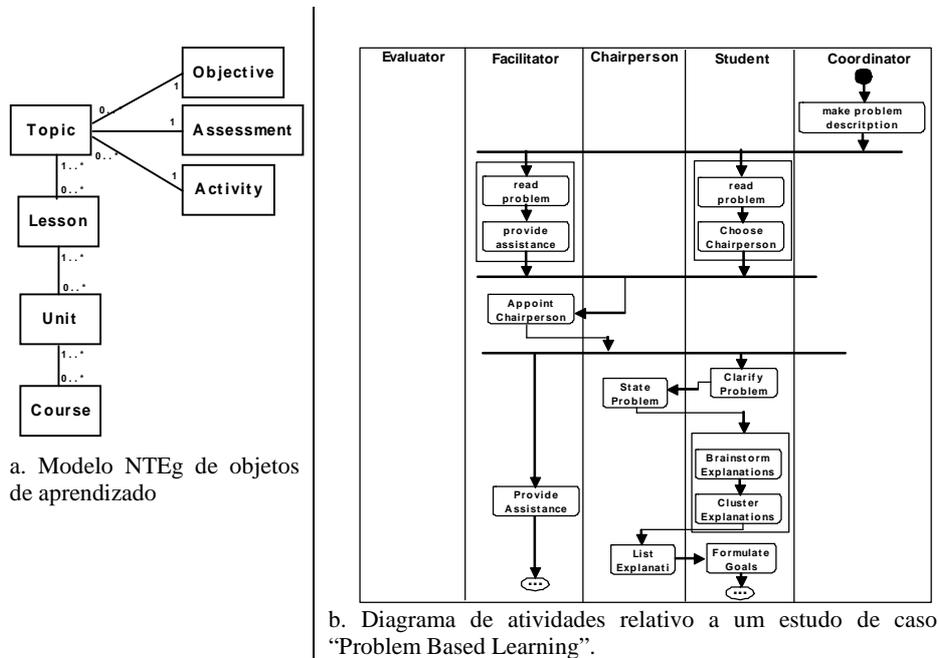


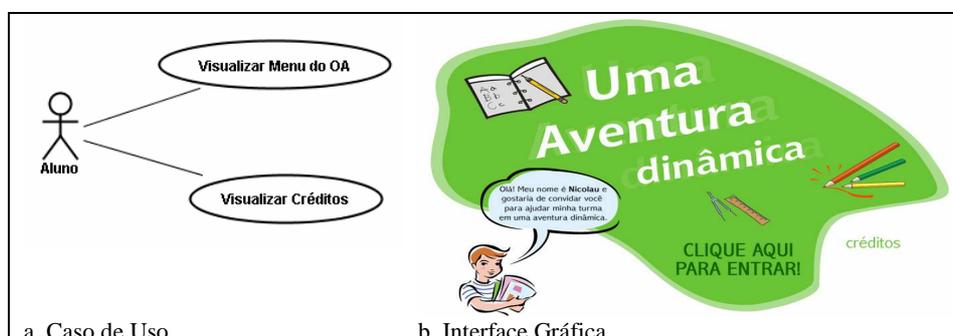
Fig. 1. Diagramas UML em modelos instrucionais

### Concepção e Modelagem do Objeto de Aprendizagem “Uma Aventura Dinâmica”

Nesta seção será apresentada uma experiência de concepção e modelagem UML do objeto de aprendizagem *Uma Aventura Dinâmica*. Este objeto foi desenvolvido a partir do planejamento inicial no qual a UML foi utilizada como linguagem de projeto. O papel de Desenvolvedor Instrucional foi exercido por uma equipe de analistas de sistemas e profissionais da área pedagógica. A seguir serão apresentadas algumas características deste objeto e a aplicação de *Diagramas de Caso de Uso* no processo de desenvolvimento.

O objeto de aprendizagem (OA) apresentado neste trabalho foi denominado de “*Uma Aventura Dinâmica*”. Trata-se de um jogo-simulação que tem como proposta inicial trabalhar conceitos de Dinâmica aplicados à aprendizagem de Física no Ensino Médio.

O caso de uso da figura 2.a descreve a funcionalidade do sistema apresentando a interação do usuário com a tela inicial implementada através da interface gráfica mostrada na figura 2b.



**Fig. 2.** Tela Inicial

Este caso de uso refere-se aos recursos básicos disponíveis ao usuário na entrada do ambiente, onde o usuário aluno interage com os recursos do sistema, conforme as descrições apresentadas nas Tabelas 1 e 2.

**Tabela 1.** Caso de Uso referente a Visualizar Menu e Créditos.

No	Caso de Uso	Descrição	Pré-Condições	Pós-Condições
1	Visualizar Menu	Permite ao usuário visualizar o Menu de opções com os recursos disponíveis no OA.	O usuário deve selecionar a opção selecionar Menu.	Será exibida uma tela com os recursos do OA.
2	Visualizar Créditos	Permite ao usuário visualizar os Créditos de desenvolvimento do OA.	O usuário deve selecionar a opção visualizar créditos.	Será exibida uma tela com os créditos de desenvolvimento

**Tabela 2.** Caso de Uso: Visualizar Menu e Créditos – Sequência Típica de Eventos

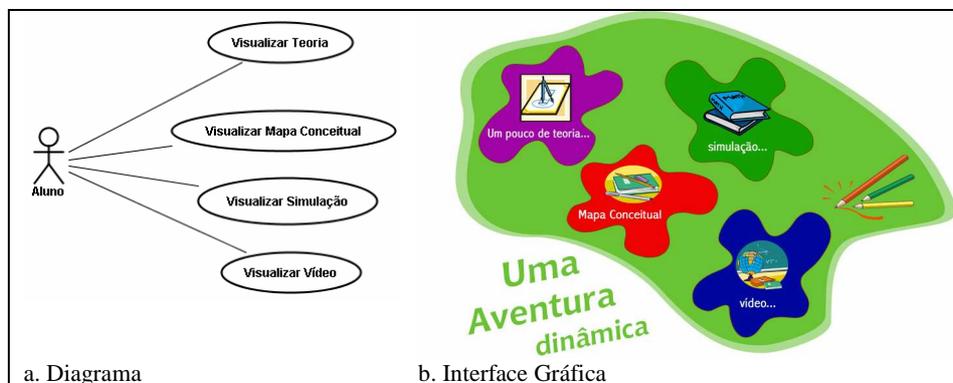
AÇÃO DO ATOR ALUNO	AÇÃO DO ATOR SISTEMA
Clica na opção “Clique Aqui Para Entrar”.	O sistema exibe a tela de opções com os recursos principais do OA.
Clica na opção “Créditos”.	O sistema exibe uma tela com os créditos de desenvolvimento do OA.
Não clica em alguma opção	O sistema permanece na tela de entrada.

### Caso de Uso “Menu Principal”

Esta interface se trata do Menu Principal de opções do OA, logo após sua tela de entrada. Para o diagrama de caso de uso apresentado na figura 3.a, o ator aluno interage com os recursos disponíveis para a manipulação no objeto.

A figura 3.b apresenta a interface gráfica implementada do Menu Principal. Nesta interface verifica-se a presença de cada uma das opções descritas anteriormente.

As tabelas 3 e 4 apresentam as especificações de ação ao selecionar o menu.



**Fig. 3.** Caso de Uso: Menu Principal

**Tabela 3.** Caso de Uso referente a Visualizar Menu e Créditos.

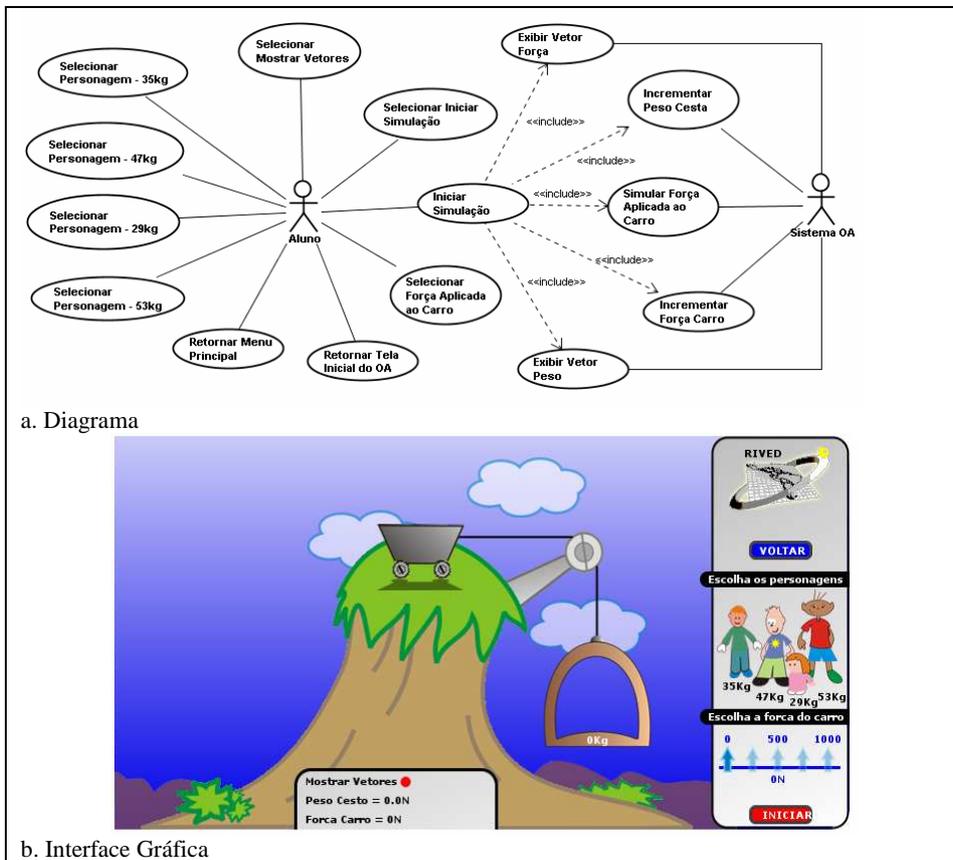
No	Caso de Uso	Descrição	Pré-Condições	Pós-Condições
1	Visualizar Teoria	Permite ao usuário visualizar a teoria disponível no OA referente às Leis de Newton.	O usuário deve selecionar a opção <i>Um pouco de teoria...</i>	Será exibida uma tela com teorias referentes as Leis de Newton.
2	Visualizar Mapa Conceitual	Permite ao usuário visualizar um Mapa Conceitual referente a teoria apresentada no OA.	O usuário deve selecionar a opção <i>Mapa Conceitual</i>	Será exibida uma tela com um Mapa Conceitual.
3	Visualizar Simulação	Permite ao usuário visualizar uma Simulação que aborda uma aplicação das Leis de Newton.	O usuário deve selecionar a opção <i>Simulação...</i>	Será exibida uma tela com uma situação desafio por meio de uma simulação.
4	Visualizar Vídeo	Permite ao usuário visualizar um conjunto de Vídeos referentes a teoria apresentada no OA.	O usuário deve selecionar a opção <i>Vídeo...</i>	Será exibida uma tela com um conjunto de Vídeos.

**Tabela 4.** Caso de Uso: Menu Principal

AÇÃO DO ATOR ALUNO	AÇÃO DO ATOR SISTEMA
Clica na opção “Um pouco de teoria...” presentes no Menu Principal do OA.	Exibe uma tela com a Teoria abordada no OA.
Clica na opção “Mapa Conceitual” presentes no Menu Principal do OA.	Exibe uma tela com um Mapa Conceitual.
Clica na opção “Simulação...” presentes no Menu Principal do OA.	Exibe uma tela com uma Simulação
Clica na opção “Vídeo...” - Menu Principal	Exibe uma tela com Vídeos.
Não clica em alguma opção	Permanece no Menu Principal

### Caso de Uso Desafio

No diagrama de caso de uso apresentado na figura 4, o ator aluno interage com os recursos disponíveis na simulação presente no ator Sistema OA. Neste exemplo, são utilizadas as seguintes opções: *Selecionar Personagem* (35, 47, 29 e 53 kg), *Selecionar Mostrar Vetores*, *Selecionar Iniciar Simulação*, *Iniciar Simulação*, *Selecionar Força Aplicada ao Carro*, *Retornar Tela Inicial*, *Retornar Menu Principal*. O ator Sistema OA também interage com o ator Aluno a partir dos seguintes recursos: *Exibir Vetor Força*, *Incrementar Peso Cesta*, *Simular Força Aplicada ao Carro*, *Incrementar Força Carro* e *Exibir Vetor Peso*.



**Fig. 4.** Caso de Uso: Desafio

A Tabela 5, a seguir apresenta as descrições, pré-condições e pós-condições de algumas das opções apresentadas no diagrama de casos de uso (Figura 4).

**Tabela 5.** Caso de Uso referente a Visualizar Menu e Créditos.

No	Caso de Uso	Descrição	Pré-Condições	Pós-Condições
1	Selecionar Personagem	Permite ao usuário selecionar a quantidade de personagens e o peso de cada um. (35, 47, 29 e 53kg).	O usuário deve clicar em cima de um personagem desejado.	Será inserido o personagem selecionado para o interior da cesta.
2	Selecionar Força Aplicada ao Carro	Permite ao usuário visualizar a Força Aplicada ao Carro para tentar levantar a cesta.	O usuário deve selecionar a opção Força Aplicada ao Carro.	Após a definição do valor da Força aplicada ao Carro, dependendo do Peso presente na cesta, ela poderá ser erguida.
3	Selecionar Exibir Vetor Força	Permite ao usuário selecionar a opção exibir vetor força.	O usuário deve selecionar a opção Exibir Vetor Força.	Será exibido graficamente o vetor força presente no Carro.
4	Selecionar Exibir Vetor Peso.	Permite ao usuário visualizar o vetor peso presente na cesta.	O usuário deve selecionar a opção Exibir Vetor Peso.	Será exibido graficamente na tela o vetor Peso presente na cesta.

Além das telas apresentadas neste trabalho, o OA apresenta também as telas Mapa Conceitual, vídeos, Simulação e Teoria. Esta é a última interface disponível no OA e apresenta sua simulação interativa.

## Conclusão

À medida que se pense em formação de uma massa crítica de profissionais voltados para o Projeto Instrucional, cria-se também a necessidade de amadurecimento de uma linguagem própria de projeto, necessária para fins de comunicação e especificação. A busca de um padrão recai num paradoxo, onde o objetivo de padronização é prejudicado pela diversidade de alternativas oferecidas. A formação do *Desenvolvedor Instrucional* passa hoje por uma diversidade de linguagens, que congrega não apenas a falta de unidade sobre a melhor forma de representação a ser adotada na Engenharia de Software, mas também a multiplicidade de formas de expressão adotada em cada uma das áreas que se aglutinam de forma multidisciplinar. A própria expressão *modelar* assume significados diferentes conforme a origem profissional seja Tecnologia da Informação, Educação, Psicologia, ou qualquer outra das diversas classes profissionais que estão compondo esta nova área de conhecimento. A narrativa bíblica sobre a Torre de Babel expressa de forma magistral este conceito: a diversidade de línguas foi a alternativa divina para impedir que o homem construísse uma torre que chegasse ao céu. À medida que a utilização de uma linguagem única seja aceita como uma das premissas necessárias para a consolidação desta área de conhecimento, será possível dar início ao desenvolvimento de cursos nos quais os *Designers Instrucionais*, novos profissionais da área de ensino / aprendizagem, recebam uma formação adequada para o entendimento desta

linguagem, acelerando assim os processos de comunicação e a consolidação da área como um todo.

Defende-se neste trabalho que a UML seja a linguagem adotada como ponto de partida. As vantagens desta abordagem passam pela massa crítica já existente entre os profissionais oriundos da área de Programação Orientada a Objetos, e pela questão técnica de se tratar de uma linguagem “de projeto”. Há ainda um terceiro motivo, referente à utilização de outros diagramas da UML que, embora ainda pouco explorados no contexto do desenvolvimento instrucional, apresentam um grande potencial de uso. Numa área em que os psicólogos falam de “Estado Mental”, os pedagogos citam “Percurso de Aprendizado”, e na qual a “Interatividade” é uma palavra de ordem, há que se avaliar os potenciais de uso dos “Diagramas de Estado”, “Diagramas de Sequência”, e outras ferramentas que permitam melhor expressar os eventos de comunicação Homem-Máquina e de comunicação entre diversos objetos, questões estas endereçadas dentro da UML.

O OA *Uma Aventura Dinâmica* modelado através da UML mostrou-se eficiente e de qualidade pedagógica, reforçando a confiança de que a UML possa ser utilizada como ferramenta adequada para fins de especificação do projeto e comunicação, capaz de traduzir de forma clara e eficaz os anseios do designer instrucional para o desenvolvedor de software.

Finalmente, cabe lembrar que, associados às Linguagens de Projeto, existem *métodos de projeto*. Como prazo, custo e qualidade são os fatores críticos de qualquer projeto, e o Desenvolvimento Instrucional é uma classe específica de projetos, a formação de profissionais na área de *Design Instrucional* deverá associar estas duas classes de formação: a representação do projeto e o método de desenvolvimento. Há aqui uma nova vertente para processos de padronização – as metodologias de projeto. E novamente, constata-se a proposição de uma infinidade de métodos, muitos dos quais denominados “ágeis”, tendo em vista a necessidade de desenvolvimento em prazos mínimos, o que, evidentemente, não poderá ser feito sem a observância de critérios de custo e qualidade. A padronização do método de projeto se caracteriza como uma questão menos crítica, desde que, o resultado do método utilizado seja uma documentação de projeto expressa de uma forma adequada para a implementação e comunicação. Em outras palavras, embora o método de desenvolvimento não seja uma questão crítica em termos de padronização, o resultado da aplicação do método o é, pois este deve ser expresso através de uma linguagem amplamente aceita e entendida.

## Referências

1. DANTAS, Alexandre Ribeiro. Oráculo: um sistema de críticas para a UML. Disponível em: < [http://reuse.cos.ufrj.br/files/publicacoes/graduacao/PF\\_Dantas\\_Oraculo.pdf](http://reuse.cos.ufrj.br/files/publicacoes/graduacao/PF_Dantas_Oraculo.pdf) > Acesso em: 22 de 5 de 2011.
2. OMG. OMG Unified Modeling Language Specification. [Versão 1.3] 1999. Verbert, K., Duval, E.: ALOCOM: A Generic Content Model for Learning Objects, International
3. VERBERT, K.; DUVAL, E. ALOCOM: A Generic Content Model for Learning Objects, International Journal on Digital Libraries, Vol. 9. 2008

4. LAFORCADE, Pierre. Visualization of Learning Scenarios with UML4LD, 2007, Journal of Learning Design, Vol 2 – No 2
5. UML Necessidade de uma modelagem visual. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/motivacao/motivacao1.htm> acesso em: 10 de maio de 2011.
6. Laforcade, Pierre. Visualization of Learning Scenarios with UML4LD, 2007, Journal of Learning Design, Vol 2 – No 2