

# Modelagem AMBER-Adp de um Ambiente para Gerenciamento de Ensino a Distância.

Almir Rogério Camolesi<sup>1,2</sup>, João José Neto<sup>1</sup>

<sup>1</sup> Departamento de Engenharia da Computação e Sistemas Digitais (PCS)  
Escola Politécnica da USP (Poli-USP)  
Av. Prof. Luciano Gualberto, Trav. 3 n°158 –09508-900 – São Paulo – SP – Brasil  
almir.camolesi@poli.usp.br, joao.jose@poli.usp.br

<sup>2</sup> Instituto Municipal de Ensino Superior de Assis (IMESA)  
Centro de Pesquisas em Informática (CePeIn)  
Fundação Educacional do Município de Assis (FEMA)  
Caixa Postal 252 – 19807-635 – Assis – SP – Brasil  
camolesi@femanet.com.br

**Resumo.** Neste artigo será apresentado o Modelo AMBER-Adp, utilizado para especificação de sistemas distribuídos. Para ilustrar a utilização do modelo, foi desenvolvido um estudo de caso baseado no desenvolvimento de um Ambiente de Gerenciamento de Ensino a Distância.

**Palavras-Chave:** Amber, Amber Adaptativo, Ensino a Distância, Modelagem Formal, Técnicas Adaptativas.

## 1. Introdução

EAD é o ensino que ocorre quando professor e aluno estão separados no tempo ou no espaço. Segundo CHAVES [CHA 2002], para que possa haver EAD, é necessário que ocorra a intervenção de alguma tecnologia.

A primeira tecnologia que permitiu o EAD foi a escrita. A tecnologia tipográfica, posteriormente, ampliou o alcance do EAD. Recentemente, as tecnologias de comunicação e telecomunicações, especialmente, em suas versões digitais, ampliaram ainda mais o alcance e as possibilidades de EAD.

Conforme apresentado em CHAVES [CHA 2002], o computador permitiu que o texto fosse enviado com facilidade a localidades remotas ou fosse buscado, também com facilidade, nestas localidades. O correio eletrônico permitiu que as pessoas se comunicassem assincronicamente com extrema rapidez. Mais recentemente, o aparecimento de "chats" ou "bate-papos" permitiu a comunicação síncrona entre várias pessoas. E, mais importante, a Web permitiu que fosse agilizado não só o processo de acesso a documentos textuais, mas também possibilitou abranger gráficos, fotografias, sons e vídeos.

Além disso, a Web permitiu ainda, que o acesso a todo esse material fosse feito de forma não-linear e interativa, usando a tecnologia de hipertexto.

O primeiro computador foi revelado ao mundo em 1946, mas foi só depois do surgimento e do uso constante de microcomputadores (que apareceram no final de 1977) que estes começaram a ser vistos como tecnologia educacional. A Internet, embora tenha sido criada em 1969, só causou furor no mercado nos últimos anos, quando foi aberta para uso comercial, pois antes servia apenas à comunidade acadêmica.

Há quem pressuponha que EAD não difere substancialmente do ensino presencial. Esses argumentam que, se o ensino presencial é bom e é possível ensinar a distância, então devemos nos valer dessa oportunidade [CHA 2002]. Por outro lado, há também os que vêem vantagens no EAD em relação ao ensino presencial: maior alcance, razão custo/benefício mais favorável, e, principalmente, maior flexibilidade (tanto para os professores quanto para os alunos).

Atualmente, não se pode conceituar o vocábulo *ensino* sem presumir estar intimamente relacionado à tecnologia. A aplicação de fatores tecnológicos contribuiu para a mudança do atual processo de ensino e aprendizagem. A tecnologia

computacional foi fundamental para a consolidação do ensino a distância, assim como os avanços tecnológicos ocorridos na área de multimídia/hipermídia — quanto a sincronismo e interatividade — ampliaram a influência da informática na educação.

As metodologias para o desenvolvimento de Ensino a Distância costumam empregar linguagens naturais, que são intrinsecamente ambíguas. As especificações informais, freqüentemente, induzem a diferentes interpretações e, por conta disso, implementações incompatíveis podem ser geradas a partir dessas.

Especificações formais, além de eliminar ambigüidades, podem ser úteis também em todo o ciclo de vida (especificação, verificação, implementação e teste) de tais aplicações. Modelos formais constituem-se de uma linguagem textual ou gráfica, com um poder de abstração, geralmente, superior ao encontrado nas linguagens de programação.

Neste contexto, será apresentado o Modelo Adaptive 'Architectural Modelling Box for Enterprise Redesign' (AMBER-Adp) que é uma extensão do modelo AMBER [QUA 1996] [FAR 2001] aos conceitos de tecnologias Adaptativas apresentadas em [NET 1993], [IWA 2000] e [NET 2001]. Um exemplo será ilustrado para demonstrar sua utilização no desenvolvimento de um Ambiente de Gerenciamento para Ensino a Distância (AGEAD).

Neste trabalho, na seção 2, apresenta-se um breve resumo do modelo AMBER. Em seguida, na seção 3, foi ilustrada a extensão do modelo AMBER aos conceitos de adaptatividade constituindo o AMBER-Adp. Na seção 4 foi ilustrada a modelagem abstrata de um Ambiente, utilizando-se do AMBER-Adp. Finalmente, na seção 5, puderam ser expostas algumas conclusões, inclusive, motivadoras de trabalhos futuros.

## 2. Modelo AMBER

Entre 1981 e 1986, especialistas da *International Organization for Specification* (ISO), dentro do ISO/TC97/SC21/WG1/FDT Subgrupo C desenvolveram a *Language of Temporal Ordering Specification (LOTOS)* [ISO 1987], que é uma Técnica de Descrição Formal (TDF), desenvolvida com o intuito de permitir a especificação e o desenvolvimento de protocolos e sistemas distribuídos abertos. A TDF LOTOS

atingiu o status de Padrão Internacional ISO em 1989. Maiores detalhes sobre o desenvolvimento de LOTOS podem ser encontrados em [VIS 1989].

Baseados na experiência adquirida no desenvolvimento de ferramentas para a TDF LOTOS, um grupo de pesquisadores da Universidade de Twente (Holanda), desenvolveu o modelo AMBER [QUA 1996] [FAR 2001] que se constitui numa base de conceitos arquitetônicos com o objetivo de facilitar o desenvolvimento de sistemas distribuídos. O modelo AMBER, originalmente foi desenvolvido para suportar o projeto de sistemas telemáticos, como, por exemplo, serviços e protocolos do modelo OSI. Uma metodologia para o projeto de serviços e protocolos baseado nesse modelo pode ser encontrada em QUARTEL [QUA 1997] e VISSERS [VIS 1998]. Em CAMOLESI [CAM 2000], foi desenvolvida uma metodologia para design de serviços de TV-Interativa, utilizando-se do modelo AMBER em uma de suas fases.

### 2.1. Conceitos básicos

Quando nos referimos a um sistema, de maneira geral, estamos interessados no que este pode nos proporcionar. De forma análoga, este modelo arquitetônico foi estruturado como *entidades funcionais*, as quais modelam a identidade de sistemas ou partes em sistemas maiores; e *comportamentos*, os quais modelam as funções das entidades funcionais.

Um sistema interage com o seu ambiente através de mecanismos que são desenvolvidos especialmente para tal fim. Um ponto de interação representa, de modo abstrato, mecanismos que suportam a interação entre sistemas. O comportamento de uma entidade funcional só pode ser acessado por outra entidade funcional através de um de seus pontos de interação.

Num modelo, cada entidade funcional tem um comportamento que representa sua funcionalidade. Um comportamento representa uma coleção de atividades que a entidade pode executar isoladamente ou em cooperação com outras entidades. Conceitos arquitetônicos, tais como ação e interação, permitem-nos representar essas atividades, constituindo os blocos básicos para a definição de comportamentos do modelo.

Atividades são executadas visando a obtenção de um resultado ou produto. Uma ação é um

conceito abstrato introduzido para representar uma atividade executada por uma única entidade em um dado nível de abstração. Embora exista uma variedade infinita de atividades, um único conceito de ação é suficiente para modelar todas essas atividades, uma vez que as características essenciais de uma atividade são representadas através dos atributos da ação que modela esta atividade: *informação*, *tempo* e *local*. O atributo de informação representa o produto obtido na execução da atividade que está sendo modelada. O atributo de tempo representa o momento em que a atividade faz seu produto tornar-se disponível, enquanto o atributo de local representa o local (físico ou lógico) no qual este produto está disponível.

Uma ação ocorre, ou não, uma única vez, sendo sua ocorrência a representação com sucesso do término de uma atividade. Uma ação é uma unidade indivisível de atividade num determinado nível de abstração (uma ação é atômica), sendo portanto, considerada uma unidade de comportamento.

Uma interação representa uma atividade que é executada por dois ou mais sistemas em cooperação. Conseqüentemente, uma interação é comum às entidades que representam os sistemas que participam da interação. Por serem representações de atividades, as interações têm os mesmos atributos das ações. A diferença entre interação e ação está nos participantes de uma interação, que podem influenciá-la de formas distintas, o que não é possível em uma ação.

A Figura 1 ilustra a representação gráfica de uma ação. A ação *login* representa uma atividade na qual um usuário estabelece uma conexão. Uma ação é representada por um círculo, enquanto o identificador da ação é colocado dentro desse círculo ou em um quadrado de texto ligado ao círculo por uma linha. Atributos da ação são representados dentro de um quadrado de texto. Os símbolos  $\iota$ ,  $\tau$  e  $\lambda$  representam os atributos de informação, tempo e local de uma ação, respectivamente.

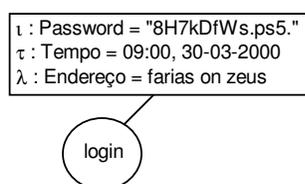


Figura 1 - Representação de uma Ação.

A Figura 2 ilustra a representação gráfica da ação *login* como uma interação entre o usuário e o sistema. Uma interação é representada ressaltando-se a distribuição de responsabilidades entre as entidades participantes na sua execução.

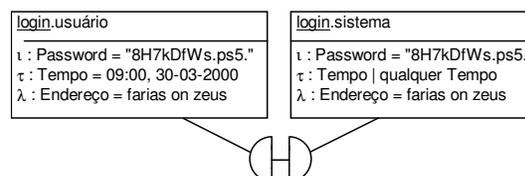


Figura 2 - Representação de uma interação.

## 2.2. Relações de causalidade

Uma ação pode ocorrer tanto sem restrições a partir do início da execução de um comportamento, chamada de *ação inicial*, quanto pode ser habilitada através da satisfação de uma condição, envolvendo a ocorrência ou não de outras ações, durante a execução de um comportamento. Uma relação de causalidade, que define a condição para a ocorrência de uma ação, consiste em:

- uma *condição de causalidade*, que define como a ocorrência de uma ação depende da ocorrência ou não de outras ações;
- *restrições dos atributos de ação*, que definem como os valores dos atributos de informação, tempo e local, estabelecidos pelas ações da condição de causalidade, influenciam a ocorrência da ação e os respectivos valores de seus atributos, e;
- um *atributo de probabilidade*, que define a probabilidade de ocorrência de uma ação, caso a condição de causalidade e as restrições dos atributos de ação sejam satisfeitas na execução de um comportamento. Neste trabalho, assumimos o valor 1 para o atributo de probabilidade, i.e., caso as duas condições anteriores sejam satisfeitas, a ação deverá ocorrer.

Um comportamento consiste numa ou mais ações e as relações entre as mesmas. Como as relações de causalidade são usadas para modelar as relações entre ações, um comportamento pode ser definido como um conjunto de relações de causalidade; uma para cada ação do comportamento.

Uma relação de causalidade permite a modelagem da ordem temporal das ocorrências de ações.

Considerando-se duas ações  $a$  e  $b$ , as quais ocorrem nos instantes  $\tau_a$  e  $\tau_b$ , respectivamente, as seguintes condições básicas de causalidade podem ser definidas:

- *habilitação* ( $a$  habilita  $b$  -  $\tau_a < \tau_b$ ), na qual a ocorrência da ação  $a$  é uma condição para a ocorrência da ação  $b$ ;
- *desabilitação* ( $a$  desabilita  $b$  -  $\tau_a > \tau_b$ ), na qual a não ocorrência da ação  $a$  é uma condição para a ocorrência da ação  $b$ , até que  $b$  ocorra. Assumindo-se que  $a$  ocorra e  $b$  ainda não ocorreu,  $b$  é desabilitada;
- *sincronização* ( $a$  sincroniza-se com  $b$  -  $\tau_a = \tau_b$ ), na qual a ocorrência da ação  $a$  é uma condição para a ocorrência da ação  $b$ , tal que  $a$  deve acontecer simultaneamente com  $b$ .

Em alguns comportamentos, múltiplas condições básicas envolvendo ações diferentes devem ser satisfeitas para que uma determinada ação possa ocorrer. Tal situação pode ser representada através da conjunção de duas ou mais condições de habilitação, desabilitação ou sincronização envolvendo diferentes ações. Em outros comportamentos, pelo menos uma condição básica ou uma conjunção de condições deve ser satisfeita para que uma determinada ação possa ocorrer. Tal situação pode ser representada através da disjunção de duas ou mais condições de causalidade básicas ou conjunções de condições.

A Figura 2 ilustra a conjunção e a disjunção de condições de causalidade. Na Figura 3a a ocorrência da ação  $c$  é habilitada após a ocorrência de ambas as ações  $a$  e  $b$ , enquanto na Figura 3b, a ocorrência da ação  $c$  é habilitada após a ocorrência de pelo menos uma das ações  $a$  ou  $b$ .

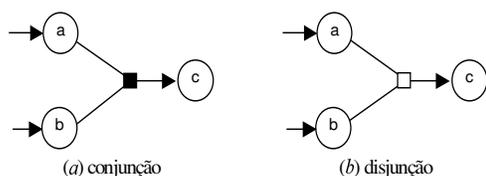


Figura 3 – Con(Dis)junção de condições de causalidade.

A Figura 4 apresenta algumas relações comumente encontradas entre duas ações  $a$  e  $b$ . Comportamentos compostos por múltiplas ações podem ser definidos através da conjunção e disjunção destas relações.

Relação entre ações	Representação gráfica
$a$ habilita $b$	
Escolha	
$a$ desabilita $b$	
Entrelaçamento	
Sincronização	

Figura 4 - Relações comuns entre ações.

### 2.3 Estruturação de comportamentos

Um comportamento complexo pode ser melhor analisado e entendido se o este for definido como uma estrutura que consiste numa composição de comportamentos mais simples. Existem duas técnicas básicas para a estruturação de um comportamento no AMBER: uma, usando relações de causalidade entre comportamentos e outra, usando restrições aplicadas a ações comuns a comportamentos. Comportamentos podem ser estruturados de forma arbitrária através da combinação dessas duas técnicas.

A estruturação usando causalidade é baseada na decomposição de uma relação de causalidade, de tal modo que, uma ação e a condição para a sua ocorrência, são definidos em comportamentos distintos. Neste sentido, os seguintes elementos sintáticos foram definidos:

- *ponto de entrada*, que consiste num ponto de um comportamento a partir do qual ações deste comportamento podem ser habilitadas através de condições envolvendo ações de outros comportamentos;
- *ponto de saída*, que consiste numa condição de causalidade em um comportamento que pode ser utilizado para habilitar ações em outro comportamento.

Na estruturação de um comportamento usando restrições, este é definido como uma composição de comportamentos mais simples, com ações comuns (interações), em que cada comportamento contribui separadamente para a ocorrência das ações.

### 3. AMBER Adaptativo

De forma similar aos Autômatos Adaptativos [NET 1993], [RAD 1995], [IWA 2000] e [NET 2001], pode-se definir o modelo AMBER

**Adaptativo** (*AMBER-Adp*) como sendo constituído por:

- $\omega$ , como a seqüência dos eventos externos independentes a serem tratados;
- $A$ , o conjunto de ações adaptativas, responsáveis pela dinâmica da topologia do modelo *AMBER-Adp*;
- $Comp_0$ , o comportamento *AMBER* convencional que representa o comportamento *AMBER-Adp* no início da operação;
- $Comp_n$ , o comportamento final do modelo *AMBER-Adp* no final da operação;
- $Comp_i$ , como o comportamento *AMBER-Adp* após  $i$  transições adaptativas.

Os comportamentos  $Comp$  são constituídos no modelo *AMBER-Adp* por ações convencionais (denominadas neste trabalho simplesmente por ações), suas condições básicas de causalidade, as relações entre ações e as ações adaptativas. Uma ação *AMBER-Adp* pode ser representada por  $\langle a, I, T, \Lambda, \zeta \rangle : \mathcal{A} \rightarrow \langle a', I', T', \Lambda', \zeta' \rangle : \mathcal{B}$ . Onde:

- $\langle a, I, T, \Lambda, \zeta \rangle$  representa a situação do comportamento  $Comp_i$  antes da execução de uma ação;
- $a \in \mathcal{A}$ , é o nome da ação que identifica unicamente seu nome;
- $I \subseteq I$ , é o valor de domínio da informação;
- $T \subseteq T$ , é o valor de domínio de tempo;
- $\Lambda \subseteq L$ , é o valor de domínio de localização da informação;
- $\zeta \subseteq I \times T \times \Lambda$ , é a combinação dos valores dos domínios de informação.

Os atributos de informação, tempo, localização e a combinação destes atributos para o domínio da ação  $a$  são denotados como  $I_a, T_a, \Lambda_a$  e  $\zeta_a$ , ou alternativamente como  $I(a), T(a), \Lambda(a)$ , e  $\zeta(a)$ , para representação textual destes, quando requeridos.

A definição de um atributo de ação é opcional. No caso de um atributo de ação deixar de ser definido, o correspondente valor do domínio pode

ser removido ou um símbolo específico pode ser inserido para explicitar a não definição.

$\langle a', I', T', \Lambda', \zeta' \rangle$  representa a situação do comportamento *AMBER-Adp* depois da execução de uma ação.

As ações adaptativas associadas a uma ação no comportamento *AMBER-Adp* são  $\mathcal{A}$  e  $\mathcal{B}$  (opcionais), e são executadas pelo comportamento *AMBER-Adp* em resposta a execução da ação à qual estão associadas, onde  $\mathcal{A}$  e  $\mathcal{B} \in \mathcal{A}$ .

$\mathcal{A}$  representa a ação adaptativa a ser executada antes da mudança de estado;

$\mathcal{B}$  representa a ação adaptativa a ser executada depois da mudança de estado.

As ações adaptativas elementares utilizadas para o *AMBER Adaptativo* são semelhantes as utilizadas nos *Autômatos Adaptativos*, ou seja, têm o formato *prefixo* [padrão da produção], no qual o *prefixo* representa um dos três tipos de ações adaptativas elementares a serem executadas: “?” (ação de inspeção), “-” (ação de eliminação) e “+” (ação de inserção), enquanto o padrão da produção representa uma produção parametrizada à qual devem ser aplicadas as ações.

A Figura 5 ilustra a representação de uma ação e suas ações adaptativas anteriores ( $\mathcal{A}$ ) e posteriores ( $\mathcal{B}$ ). As ações adaptativas são inseridas em caixas superiores e inferiores à ação tradicional do modelo *AMBER*.

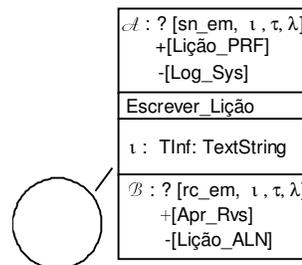


Figura 5 - Representação de ações adaptativas.

A evolução do modelo *AMBER-Adp* ocorre por meio da modificação das ações pertencentes a um comportamento pela evolução gradual do seu conjunto de regras, com a adição ou remoção de ações efetuadas por meio da execução das ações adaptativas.

#### 4. Modelagem AMBER-Adp do AGEAD

No projeto EAD-IMESA [FAB 2000] foi proposto um Ambiente de Ensino a Distância, composto por 3 componentes principais: Aluno, Professor e um Ambiente de Gerenciamento de Ensino a Distância (AGEAD). Alunos e Professores são entidades que interagiram diretamente com o AGEAD que, por sua vez, é composto de vários módulos:

**Gerenciamento Acadêmico:** Este módulo é responsável pelo gerenciamento do controle acadêmico (notas, faltas, turmas, assiduidade, etc.).

**Controle de Acesso e Gerenciador de Interfaces:** Controla o acesso/validação dos Professores e Alunos ao Ambiente. Gerencia as interações entre o Ambiente e as entidades externas (Alunos/Professores) em um determinado momento.

**Ferramenta de Autoria:** Provê recursos aos autores de aulas (geralmente um Professor), para que estes tenham subsídios para elaborar aulas, cursos, material de apoio, listas de exercícios, etc.

**Gerenciamento de Aulas/Cursos Virtuais:** Gerencia os cursos e as aulas virtuais elaboradas pelos autores.

**Gerenciamento de Laboratórios Virtuais:** Gerencia os fluxos de informações entre Professores, Alunos e Ambiente.

**Trabalhos Cooperativos:** Gerencia recursos, geralmente síncronos, para trabalhos cooperativos entre Alunos e Alunos/Professores. Estes recursos podem ser caracterizados por editores gráficos e textuais, *chats*, correio eletrônico, entre outros.

**Biblioteca Virtual:** Gerencia o acesso à biblioteca do Ambiente e propicia o acesso a diversas outras bibliotecas.

Uma síntese dos módulos inclusos no Ambiente EAD-IMESA é ilustrada a seguir, na Figura 6.

O EAD-IMESA tem como uma de suas principais preocupações a qualidade do serviço, tanto em nível computacional como pedagógico, ou seja, há uma preocupação com a qualidade e a utilidade da interação entre os usuários do Ambiente desde o momento da elaboração do

material didático até a utilização do material/Ambiente pelos usuários.

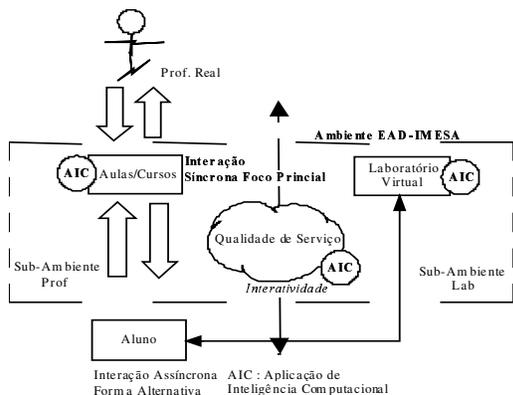


Figura 6 - Interação entre Professores/Alunos e o Ambiente EAD-IMESA.

#### 4.1 Modelagem do AGEAD

Após o levantamento dos requisitos dos componentes básicos do AGEAD, é realizada a modelagem abstrata da aplicação. Inicialmente são identificadas as entidades funcionais que compõem o AGEAD no seu mais alto nível de abstração.

A Figura 7 apresenta as entidades funcionais do AGEAD e seus usuários. Entidades funcionais são representadas por retângulos com bordas entrecortadas, enquanto pontos de interação são representados por elipses. O AGEAD é composto pelas seguintes entidades funcionais: ALUNO, Sistema de Gerenciamento de Ensino a Distância (SGEAD) e PROFESSOR. As entidades funcionais  $Aln_{1..n}$  e  $Prof_{1..n}$  representam os usuários do AGEAD.

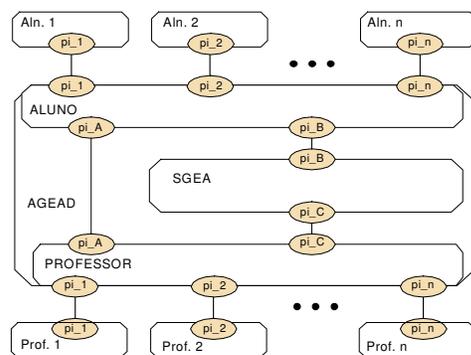


Figura 7 - Entidades representando o AGEAD e seus usuários.

Embora os pontos de interação  $pi_1$ ,  $pi_2$  e  $pi_n$  sejam representados por elementos gráficos distintos (cada um por uma elipse), estes modelam mecanismos comuns às entidades funcionais que interagem através deles pr. Um único ponto de interação suporta a ocorrência de múltiplas interações (acesso ao AGEAD, participar das aulas, acessar os laboratórios e bibliotecas, realizar avaliações, etc.) entre as entidades associadas.

Na Figura 8, é apresentada uma pequena parte da modelagem do AGEAD, responsável pela representação das lições realizadas pelos alunos e enviadas aos professores para correção. O subcomportamento Lição\_ALN recebe os dados a serem digitados para uma determinada lição através da interação (Fz Liç.ALN). Ao ocorrer a ação Fz Liç.ALN, é habilitada a execução Escreve\_Lição, responsável pela digitação das informações necessárias à lição a ser executada. Na seqüência, as informações são enviadas ao subcomportamento Lição\_PRF por meio das interações Env Liç.ALN e Rcb Liç.PRF, respectivamente, dos subcomportamentos Lição\_ALN e Lição\_PRF. No subcomportamento Lição\_PRF, ao ser recebida uma mensagem de habilitação da ação Corrigir\_Lição, é realizada a correção da lição enviada pelo aluno e, em seguida, é habilitada a ação Escrever\_Revisão, que permite ao professor informar as correções necessárias à lição. Após ter sido preparada a mensagem referente às correções necessárias da lição, esta é enviada ao aluno por meio das interações Env Rvs.PRF e Rcb Rvs.ALN dos subcomportamentos Lição\_PRF e Lição\_ALN.

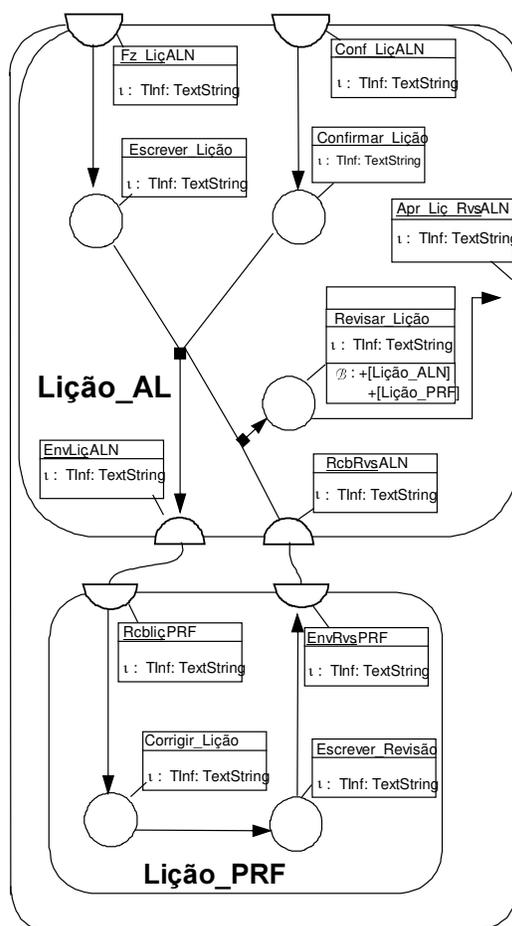


Figura 8 - Modelagem AMBER-Adp Lição\_ALN e Lição\_PRF.

Ao receber uma mensagem de habilitação da ação Revisar\_Lição é apresentado ao aluno as devidas correções por meio da interação Apr Liç Rvs.ALN e, após a execução da ação Revisar\_Lição, são executadas as ações adaptativas posteriores  $B$ , responsáveis pela execução das ações adição (+) que acrescenta as ações desempenhadas pelos subcomportamentos [Lição\_ALN] e [Lição\_PRF]. Assim, a cada instante, o número de lições (cada uma representada por um subcomportamento Lição\_ALN) é exatamente o necessário para cada aluno. Dessa forma, cada aluno tem um conjunto diferente de lições em sua modelagem, representando o ritmo individual de estudo.

A Figura 9 ilustra o comportamento do AGEAD após a realização de uma tarefa por um aluno. Novos subcomportamentos foram adicionados de forma adaptativa, permitindo que o aluno realize uma nova tarefa.

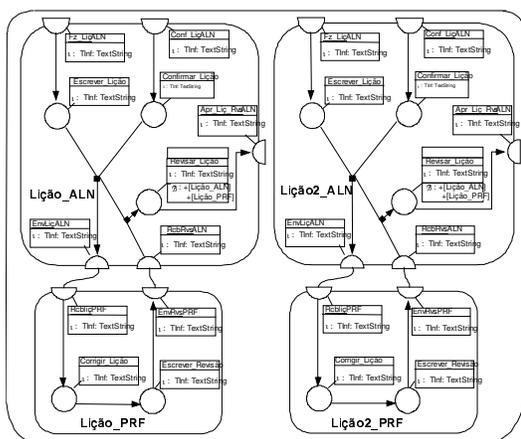


Figura 9 - Representação dos subcomportamentos após execução das ações adaptativas.

## 5. Conclusão

Neste trabalho foram discutidos alguns conceitos básicos referentes à área de Ensino a Distância. Também foi apresentado o Modelo AMBER-Adp, que está sendo desenvolvido a partir de estudos realizados no modelo AMBER e a aplicação de tecnologias adaptativas neste modelo. O modelo AMBER-Adp tem por finalidade facilitar o projeto de sistemas distribuídos que apresentam características adaptativas em sua constituição.

No modelo AMBER-Adp, uma entidade modela um objeto e é caracterizada por seu comportamento (métodos) e estado (atributos). Isso permite, por exemplo, que se estruture um sistema como uma hierarquia de abstrações (entidades) e que se encapsule o comportamento de entidades, pois o comportamento de uma entidade só é acessível por meio dos pontos de interação definidos em sua interface.

Por sua vez, a definição de ações adaptativas na estruturação de comportamentos permite o uso de adição e subtração de comportamentos e ações de forma dinâmica, o que torna o modelo AMBER mais flexível e com maior poder de expressão para problemas que apresentam características dinâmicas durante a execução. Utilizando-se do modelo AMBER-Adp o projetista pode modelar comportamentos que tenham seu conjunto de ações modificado durante a execução do sistema. Os recursos adaptativos permitem que sejam inseridas ou eliminadas ações a cada execução do comportamento, o que permite que aplicações com grande complexidade de representação sejam representadas utilizando-se deste modelo.

O estudo de caso fundamentado em Ensino a Distância serviu para ilustrar a utilização do modelo no projeto de aplicações distribuídas. Por meio do estudo de caso, foi verificado que o modelo permite representar especificações que possuem características de adaptatividade, como pudemos observar através da especificação dos comportamentos Aluno e Professor.

Como continuação deste trabalho, estão sendo realizados estudos para completar a extensão do modelo AMBER em todos os seus componentes aos conceitos de Adaptatividade. Tal extensão permitirá ao projetista maior facilidade no projeto de suas especificações e possibilitará a obtenção de especificações com maior consistência e de fácil entendimento. Também estão sendo realizados estudos referentes ao desenvolvimento de uma ferramenta que utilizará o novo modelo para dar suporte ao projetista na realização de suas especificações.

## Referências Bibliográficas

- [RAD 1995] RADY, Jorge. STAD-Uma ferramenta para representação e simulação de sistemas através de statecharts adaptativos. Tese de Doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo, 1995.
- [CHA 2002] CHAVES. E. Ensino a Distância Conceitos Básicos. Disponível na Internet no endereço [http://www.edutecnet.com.br/..](http://www.edutecnet.com.br/)
- [CAM 2000] CAMOLESI, Almir. Uma metodologia para o Design de Serviços de TV-Interativa. Dissertação de Mestrado, PPG-CC, UFSCar, Fevereiro, 2000.
- [CAM 2000a] CAMOLESI, Almir; FABRI, José; FORNAZARI, Fabio. Uma Metodologia para o Desenvolvimento de um Ambiente de Gerenciamento Inteligente para Ensino a Distância. III Fórum Nacional de Tecnologia, URI Campi Santo Ângelo, Santo Ângelo-RS, Agosto, 2000.
- [FAB 2000] FABRI, Jose; CAMOLESI Almir; CUNHA, Douglas; FORNAZARI, Fabio; MACHADO, Osmar. Projeto EAD-IMESA. Um Ambiente de Gerenciamento Inteligente para Ensino a Distância, I Workshop de Informática Aplicada à Educação, Uniara, Araraquara, Setembro, 2000.
- [FAR 2001] FARIAS, Clever; PIRES, Luis. AMBER: Uma linguagem para desenvolvimento

de sistemas distribuídos. XIX Simpósio Brasileiro de Redes de Computadores, Florianópolis, Maio, 2001.

[ISO 1987] ISO/IEC 8807. LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behavior. Novembro, 1987.

[IWA 2000] IWAI, Margarete. Um formalismo gramatical adaptativo para linguagens dependentes de contexto. Tese de Doutorado. Escola Politécnica, Universidade de São Paulo, São Paulo, 2000.

[LIM 2002] LIMA, Alexandre. Modelagem AMBER de um Ambiente de Gerenciamento de Ensino a Distância. I Workshop Informática IMESA, Assis-SP, Agosto, 2002.

[NET 2001] NETO, João. Adaptive rule-driven devices - general formulation and case study. CIAA 2001 - Sixth International Conference on Implementation and Application of Automata, Universidade de Pretoria, Pretoria-África do Sul, Julho, 2001.

[NET 1993] NETO, João. Contribuição à metodologia de construção de compiladores. Tese de Livre Docência, Escola Politécnica, Universidade de São Paulo, São Paulo, 1993.

[QUA 1996] QUARTEL, Dick. Actions Relations – Basic design concepts for behavior modelling and refinement. Ph.D Thesis, Netherlands, 1997.

[QUA 1997] QUARTEL, Dick; PIRES Luis, SINDEREN. Mark; FRANKEN Herman; VISSERS, Cris. On the role of basic design concepts in behaviour structuring. Computer Networks and ISDN Systems, 1997, pp. 413-436.

[VIS 1989] VISSERS, Cris. LOTOS Backgrounds The Formal Description Technique LOTOS, North-Holland, 1989, pp.15-22.

[VIS 1998] VISSERS, Cris; PIRES, Luis; QUARTEL, Dick; SINDEREN, Mark. The architectural design of distributed systems. Lecture notes for The design of telematic systems, University of Twente, Enschede - Netherlands, 1998.