

# **LOCPN: REDES DE PETRI COLORIDAS NA PRODUÇÃO DE OBJETOS DE APRENDIZAGEM**

**Maria de Fátima Costa de Souza<sup>1,2</sup>,  
Danielo G. Gomes<sup>2,a</sup> e  
Giovanni Cordeiro Barroso**  
Universidade Federal do Ceará  
Programa de Pós-Graduação em  
Engenharia de Teleinformática - PPGETI  
Campus do Pici, s/n - CEP 60755-640  
Fortaleza - CE - Brasil  
{fatimasouza, dgomes}@great.ufc.br  
gcb@fisica.ufc.br

**Cidcley Teixeira de Souza<sup>3</sup>**  
Centro Federal de Educação  
Tecnológica do Ceará  
Gerência de Telemática  
Av. 13 de Maio, 2081 - Benfica  
CEP:60040-531  
Fortaleza - CE - Brasil  
cidcley@cefetce.br

**José Aires de Castro Filho<sup>1</sup>,  
Mauro Cavalcante Pequeno<sup>1</sup>**  
Universidade Federal do Ceará  
Instituto UFC Virtual  
Campus do Pici, Bl.901 - 1º andar  
CEP 60455-760  
Fortaleza - CE - Brasil  
j.castro@ufc.br  
mauro@virtual.ufc.br

**Rossana M. C. Andrade<sup>2</sup>**  
Universidade Federal do Ceará  
Departamento de Computação  
Bl. 942a - CEP 60455-760  
Fortaleza - CE - Brasil  
rossana@ufc.br

---

**Resumo:** *Objetos de Aprendizagem (OAs) são recursos digitais desenvolvidos para serem utilizados dentro do contexto educacional como ferramenta de apoio ao processo de aquisição de conhecimento do indivíduo. Tais recursos, evidenciados no Brasil por esforços como o do RIVED (Rede Interativa Virtual de Educação), têm mostrado a importância dos OAs, e, particularmente, das simulações e animações, como artefatos de software capazes de auxiliar a prática pedagógica. Contudo, em virtude do grau de complexidade dos requisitos envolvidos na produção desses artefatos, há a necessidade de se contar com diferentes tipos de profissionais, que possam responder, de forma coerente, pelo caráter tanto pedagógico quanto técnico. Entretanto, esse caráter multidisciplinar afeta o processo de desenvolvimento em virtude das imprecisões na comunicação dos diferentes tipos de requisitos entre esses profissionais. Nesse sentido, propomos nesse trabalho, a adoção de um modelo formal, denominado LOCPN (Learning Objects production with Colored Petri Nets), desenvolvido especificamente para auxiliar o processo de desenvolvimento de OAs. A eficácia da utilização de LOCPN é comprovada neste trabalho pela qualidade das implementações e pela redução significativa do tempo de produção de OAs.*

---

**Palavras-chave:** *Objetos de Aprendizagem, Modelo de Processo, Redes de Petri Colorida, Especificação formal.*

---

**Abstract:** *Learning Objects (LOs) are digital resources developed to be used in the educational setting as a tool to support one's learning acquisition process. Such resources, made notorious in Brazil by efforts such as RIVED (Virtual Interactive Educational Network), have proved the significance of LOs, and particularly of simulations and animations, as software artifacts able to help the pedagogical practice. However, due to the high degree of complexity involved in the production of these artifacts, different types of professionals are needed to answer, in a coherent manner, both the pedagogical and technical concerns. In addition, this multidisciplinary characteristic affects the development process due to imprecision in communication among these professionals about the different types of requirements. Therefore, we propose in this work the adoption of a formal model, named LOCPN (Learning Objects production with Colored Petri Nets), specially developed to aid the development process of LOs. The usage efficiency of LOCPN is demonstrated in this work by the quality of implementations and the significant time reduction in the production of LOs.*

---

**Keywords:** *Learning Objects, Process Model, Colored Petri Nets, Formal Specification.*

---

<sup>1</sup> Grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem (PROATIVA)

<sup>2</sup> Grupo de Redes de Computadores, Engenharia de Software e Telecomunicações (GREaT)

<sup>3</sup> Núcleo Avançado em Engenharia de Software Distribuído e Sistemas Hiperfídia (NASH)

<sup>a</sup> Bolsista de Desenvolvimento Científico Regional do CNPq/Funcap (processo 350093/2005-9)

## 1. INTRODUÇÃO

A necessidade do uso de novas tecnologias no processo de ensino e aprendizagem vem sendo cada vez mais presente no cotidiano de alunos e professores. Contudo, é preciso ampliar esta discussão com o objetivo de contextualizar as novas tecnologias da informação e da comunicação e suas relações com o ensino e aprendizagem na Educação [5]. Esta discussão surge com o anseio de modificar a forma como a Educação propõe o ensino e como os materiais educacionais são projetados, desenvolvidos e entregues àqueles que desejam aprender.

Atualmente, um dos materiais educacionais que procuram atender à demanda tecnológica para atingir esses objetivos são os Objetos de Aprendizagem (OAs) [4,7].

Na literatura, existem diversas definições para Objeto de Aprendizagem (OA). De acordo com a definição apresentada em [2], OA (*Learning Objects*) podem ser definidos como “qualquer entidade digital, que pode ser usada, reusada ou referenciada durante um processo de aprendizagem suportado pela tecnologia”.

Para o desenvolvimento de OAs, faz-se necessário a utilização de um modelo de processo no qual um grupo multidisciplinar deve estar envolvido. Assim, pedagogos, designers gráficos, engenheiros de software, entre outros, devem interagir de modo a se atingir os objetivos, tanto tecnológicos quanto pedagógicos desse produto. Nesse contexto, a precisão na forma de comunicação das informações relacionadas aos OAs deve ser garantida, a fim de permitir a sua construção com qualidade e minimizando o tempo de produção dos mesmos.

Normalmente, documentos escritos em linguagem natural são utilizados como forma de descrição de OAs e, conseqüentemente, de promover a interação entre os integrantes da equipe multidisciplinar. Esse documento recebe o nome de Roteiro de Atividade. Na seção 4 deste trabalho detalharemos a produção e o objetivo deste documento. Contudo, a complexidade dos aspectos de seqüenciamento, interatividade e navegação nas atividades relacionadas a certos OAs tornam esse mecanismo bastante ineficiente. De fato, diversos OAs acabam passando por um processo enorme de validação e verificação e, em conseqüência, de re-implementação por conta da dificuldade de se detalhar esses aspectos antecipadamente em um documento utilizando somente a linguagem natural.

Por outro lado, técnicas de descrição formal (TDFs) [19] vem sendo utilizadas em diversas áreas, tais como telecomunicações e engenharia de software, para descrever com precisão os requisitos dos usuários minimizando assim os erros que podem ocorrer na fase de validação e verificação e o tempo alocado na fase de implementação e manutenção de código.

Assim, este trabalho propõe a aplicação de uma TDF para formalizar etapas estabelecidas dentro do documento

Roteiro de Atividades de forma que permita a expressão dos aspectos dos OAs de forma correta antes de sua implementação. Para isso, é proposta a inclusão de uma etapa de modelagem formal para a produção de OAs utilizando redes de Petri coloridas.

As redes de Petri coloridas são utilizadas, porque as mesmas permitem uma formalização dos processos existentes dentro do documento Roteiro de Atividades, uma maior inserção das características necessárias para a produção de um OA e além disso, as mesmas podem ser analisadas, facilitando as descobertas de erros ainda na fase de modelagem

O restante deste artigo está dividido em 7 seções. Na segunda seção são apresentadas as redes de Petri coloridas, utilizadas como ferramenta de modelagem no processo de criação dos objetos. Na terceira seção são apresentados os OAs e as formas como eles são desenvolvidos atualmente. Na quarta seção é apresentado o desenvolvimento de um OA, ressaltando as problemáticas dessa produção. Na quinta seção é apresentada a proposta de formalização do processo de desenvolvimento de um OA. Na sexta seção é apresentado um estudo de caso, mostrando a aplicabilidade proposta. Na sétima seção são apresentados os trabalhos relacionados existentes e, por fim, na sétima seção são apresentadas as conclusões do trabalho e as pretensões quanto aos trabalhos futuros.

## 2. REDES DE PETRI

Rede de Petri é uma TDF que faz uso de uma modelagem matemática e gráfica desenvolvida por Carl Adam Petri com o intuito de representar sistemas concorrentes, controle, conflitos sincronização e compartilhamento. Pode-se afirmar que o uso de um modelo facilita e justifica o prosseguimento de novas etapas de projeto e análise antes da etapa final de avaliação. Entre estas etapas podemos citar a fase de medição, a qual depende diretamente da existência física do sistema e que muitas vezes só deve ser realizada após uma fase anterior de validação, antes do protótipo ser construído.

Uma rede de Petri ordinária é composta pelos seguintes elementos [8]: Lugares, que representam uma condição, uma atividade ou um recurso; Fichas, que representam o estado de um sistema; Transições que representam a ocorrência de um evento e Arcos que indicam os lugares de entrada ou saída das transições.

Atualmente, existem diversas extensões de redes de Petri (RdP), dentre elas podem ser citadas [8]: as coloridas, as temporizadas e as estocásticas.

O presente trabalho tem como objetivo abordar somente as RdP coloridas por estas serem capazes de modelar sistemas grandes e complexos, devido a quantidade de recursos que dispõem, possibilitando uma redução nos tamanhos dos modelos.

## 2.1. REDES DE PETRI COLORIDAS (RdPC)

O principal objetivo das RdPC [20] é a redução do tamanho do modelo, permitindo que fichas individualizadas (coloridas) representem diferentes processos ou recursos em uma mesma sub-rede. Em RdPC as fichas são representadas por estruturas de dados complexas. Deste modo, as fichas podem conter informações. Além disso, cada lugar armazena fichas de um certo tipo definido e arcos realizam operações sobre elas. As transições determinam a dinâmica da RdPC e podem apresentar “expressões de guarda”. Estas, por sua vez, indicam os tipos de fichas que possibilitam ativar uma transição

Uma RdPC é composta por três partes: estrutura, inscrições e declarações. A estrutura é um grafo direcionado, com dois tipos de nós (lugares e transições), com arcos interconectando nós de tipos diferentes. As inscrições são associadas aos lugares, transições e arcos. As declarações são tipos, funções, operações e variáveis. Quando a expressão do arco é avaliada, ela gera um multi-conjunto de fichas coloridas. Expressões podem conter constantes, variáveis, funções e operações definidas nas declarações, e não produzem efeito colateral.

**Definição 1.** Uma Rede de Petri Colorida é definida por  $RdPC = (\Sigma, L, T, A, N, C, G, E, I)$ , em que:  $\Sigma$  é um conjunto de tipos não-vazios chamados cores;  $L$  é um conjunto finito de lugares;  $T$  é um conjunto finito de transições;  $A$  é um conjunto finito de arcos tais que  $L \cap T = L \cap A = T \cap A = \emptyset$ ;  $N$  é uma função de nó, definida por  $N: A \rightarrow L \times T \cup T \times L$ ;  $C$  é uma função de cor, definida por  $C: L \rightarrow \Sigma$ ;  $G$  é uma função de guarda definida por  $G: T \rightarrow exp$ , em que  $exp$  é uma expressão tal que:  $\forall t_i \in T \mid Type(G(t_i)) = Boolean \wedge Type(Var(G(t_i))) \subseteq \Sigma$ ;  $E$  é uma função de expressões de arco definida por  $E: A \rightarrow exp$ , tal que:  $\forall a \in A \mid Type(E(a)) = C \wedge Type(Var(E(a))) \subseteq \Sigma$ ;  $I$  é uma expressão de inicialização, tal que  $\forall l_i \in L \mid Type(I(l_i)) = C(l_i) \wedge Var(I(l_i)) = \emptyset$ .  $Var(t)$  é o conjunto de variáveis associadas à transição  $t$ . Essas variáveis podem ser associadas às transições de duas maneiras: nas expressões de guarda ( $G(t)$ ,  $t \in T$ ) e nas expressões dos arcos ( $E(a)$ ,  $a \in A$ ) que interconectam lugares e transições.

Dessa forma, podemos verificar que as redes de Petri coloridas possuem uma grande quantidade de recursos que podem ser utilizados para especificar detalhadamente as ações de um sistema, permitindo a formalização na sua produção. Na próxima seção apresentaremos como é realizado o processo de produção dos OAs sem a inserção da fase de formalização com a utilização das RdPC.

## 3. OBJETOS DE APRENDIZAGEM

Os OAs são um tipo de tecnologia muito recente que utiliza os recursos digitais para promover a aprendizagem. Como apresentado em [21], seu surgimento é bem mais fácil de ser compreendido do que sua própria definição.

Assim, há uma disseminação do termo entre autores, professores, educadores e profissionais da área de informática e novas tecnologias. Muitos começaram a usar o termo e criar OAs indistintamente até que surgiu a necessidade de definir e padronizar essa nova tecnologia. Diante dessa situação, o IEEE Learning Technology Standardization Committee (LTSC) [3] propôs uma definição para esse termo. Assim, segundo [3], OA é qualquer entidade, digital ou não, que pode ser utilizada, re-utilizada ou referenciada durante o aprendizado apoiado pela tecnologia.

Há diversos fatores que favorecem o uso de OAs na área educacional, como por exemplo: a flexibilidade, a facilidade para atualização, a customização, a interoperabilidade, o aumento do valor de um conhecimento e, por fim, a indexação e procura. Assim, todas estas vantagens são suficientes para justificar a utilização dos OAs dentro do contexto educacional.

### 3.1. PROCESSO DE DESENVOLVIMENTO DE OAS

São vários os processos utilizados para se desenvolver OAs. No Brasil, podemos citar basicamente dois modelos de processos. O primeiro deles é o desenvolvido pelo Laboratório Didático Virtual (Labvirt) [6], que é um projeto criado através da parceria realizada entre a Microsoft do Brasil e a Escola do futuro da Universidade de São Paulo para o desenvolvimento e armazenamento de OAs. Nele, professores e alunos podem disponibilizar e utilizar uma grande quantidade de OAs desenvolvidos para o ensino médio, nas áreas de química e física.

O modelo de desenvolvimento adotado pelo Labvirt é composto da seguinte forma: os professores, que formam a equipe pedagógica, determinam o conteúdo a ser abordado, em conjunto com os alunos das escolas públicas. Em seguida a equipe responsável pelo *design gráfico* contextualiza o conteúdo, a partir da criação de roteiros e cenários. Por fim, uma equipe técnica produz o objeto, utilizando processos oriundos da engenharia de software.

Um segundo modelo importante é o utilizado pela Rede Interativa Virtual de Educação (RIVED) [9]. O RIVED é um programa da Secretaria de Educação a Distância - SEED, que tem por objetivo a produção de conteúdos pedagógicos digitais, na forma de OAs baseados em simulações e animações. Tais conteúdos primam por estimular o raciocínio e o pensamento crítico dos estudantes, associando o potencial da informática às novas abordagens pedagógicas [9].

O modelo de processo proposto pelo RIVED, apresentado na Figura 1, é baseado na junção de um conjunto de características que são pertinentes ao aprendizado do indivíduo. Essa junção dá origem aos documentos *General Design (GD) ou Design Pedagógico, Roteiro de Atividades e Guia do Professor*, que são gerados e utilizados nas fases apresentadas na Figura 1.

Na Fase 1, os objetivos e as atividades são descritas no documento intitulado *General Design* (GD). Na Fase 2, o GD é submetido às outras equipes para obter críticas e feedback. Na Fase 3, os especialistas de conteúdo revisam o *design* original após receber *feedback* das outras equipes. Nesse momento, passam a descrever as especificações para cada OA para que a equipe técnica implemente os produtos desejados, dando origem ao documento Roteiro de Atividades. Na Fase 4, a equipe técnica produz os OAs. Na Fase 5, os especialistas em conteúdo criam o Guia do Professor para cada OA. Por fim, na Fase 6, os OAs são organizados nos módulos e publicados na *web*.

Todas as etapas do modelo de processo atual do RIVED são baseadas em documentos descritos em linguagem natural. Além dessa forma de comunicação estar sujeita a erros de

interpretação, ela é vaga em termos de detalhamento que permita ao grupo multidisciplinar discutir e interagir de modo a tornar clara as especificações. Essa comunicação clara é imprescindível, uma vez que a multidisciplinaridade das equipes pode provocar um desentendimento nos conceitos apresentados pelo grupo. De fato, no modelo atual, a abordagem de apenas permitir a interação entre os integrantes da equipe, de forma a elucidar requisitos, não é suficiente para assegurar uma correta interpretação das informações por parte da equipe técnica, visto que a comunicação entre a equipe é realizada de modo não sistemático e não documentado, o que dificulta a integração de novos requisitos ao software.

Para que se possa entender melhor os problemas relacionados à interpretação da documentação de um OA, é apresentado na seção 4, um exemplo prático.

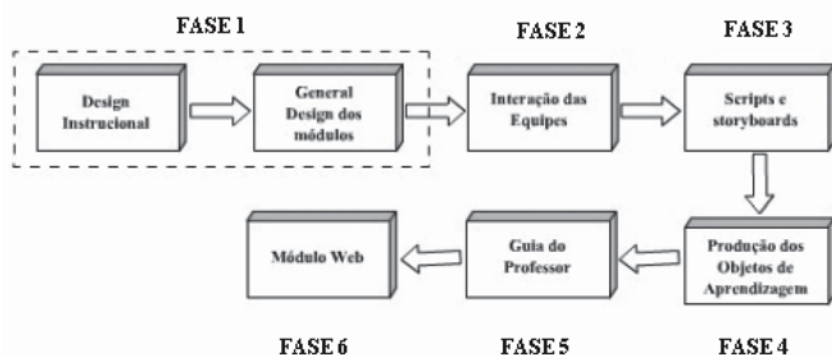


Figura 1: Modelo de Processo RIVED [11]

#### 4. EXEMPLO DE ESPECIFICAÇÃO DE UM OA

Para exemplificar a forma como são descritos as relações de seqüenciamento, interatividade e navegação em um Roteiro de Atividades, segue abaixo um trecho de especificação de um OA, denominado Pontos em Batalha (PemB) [12]. Esse OA foi desenvolvido pelo grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem (PROATIVA) da Universidade Federal do Ceará, e tem por objetivo auxiliar o aprendizado dos conceitos de distância entre dois pontos e distância entre um ponto e uma reta. O PemB está dividido em duas fases. Na primeira fase, o objetivo é trabalhar a contextualização da distância entre dois pontos. Nessa fase, o OA disponibiliza aos usuários, 4 níveis de navegação, os quais possuem graus crescentes de dificuldades ou seja, quanto maior o nível selecionado pelo usuário maior será a exigência no domínio do conhecimento sobre o assunto tratado pelo objeto. Já na segunda fase, o foco principal é trabalhar a contextualização da distância entre um ponto e uma reta. Nessa fase, existem três níveis de navegação diferentes, em que da mesma forma como na primeira fase, quanto maior o nível maior o grau de exigência sobre o domínio do assunto tratado por essa fase do OA.

Como forma de incentivar os usuários na aprendizagem dos conceitos formais da geometria analítica, esse OA foi

desenvolvido baseado na estrutura e ludicidade proporcionada pelo jogo Batalha Naval. É importante ressaltar, que o OA é desenvolvido para auxiliar o professor no processo de ensino-aprendizagem, portanto, sua eficácia só é validada na escola se houver sintonia entre professor e OA.

Na Figura 2 é apresentada uma tela do PemB. Essa tela faz parte da segunda fase do jogo, onde são trabalhados os conceitos de distância entre um ponto e uma reta.

A produção desse objeto só foi possível devido a criação de uma equipe multidisciplinar composta por professores de matemática, pedagogos, desenvolvedores, além de designers gráficos. Nessa equipe, os professores de matemática, que no caso são os especialistas da área, e os pedagogos compõem a chamada equipe pedagógica. Já os desenvolvedores e os designers gráficos compõem a equipe técnica.

A comunicação entre os membros da equipe pedagógica com a equipe técnica se dá através da produção de um documento desenvolvido denominado Roteiro de Atividades.

Dessa forma, neste trabalho, damos ênfase ao documento Roteiro de Atividades, devido à nossa preocupação com o grande volume de informações que a equipe pedagógica repassa à equipe técnica, tornando-o

passível de erro de interpretação por parte dos desenvolvedores quando da implementação do objeto, podendo acarretar uma demora na produção desse produto e o comprometimento de sua qualidade.

A seguir, devido a limitação de espaço, apresentamos apenas algumas instruções existentes no documento Roteiro de Atividades do OA PemB, de modo a explicitar seu conteúdo. Esse documento é criado para estabelecer uma comunicação entre a equipe pedagógica e a equipe de implementação, de modo que as atividades a serem implementadas nesse OA sejam explicitadas minuciosamente.

- Quando o usuário acessar o OA PemB, primeiramente será aberta a caixa de instrução que

ficará no centro da tela com o seguinte texto: “O seu objetivo será atingir o barco inimigo, para isso localize os barcos no mar e calcule corretamente a distância que a bala deverá percorrer para atingi-lo.” A caixa de instrução terá um botão para ser fechada;

- No canto superior a direita da tela do OA teremos a ajuda que será identificada como uma “?”. A “ajuda” fornecerá as fórmulas matemáticas necessárias para se trabalhar tanto a distância entre dois pontos quanto a distância entre ponto e reta;
- A localização do barco que irá interceptar os outros dois será dada, cabendo ao usuário identificar os pontos das outras duas embarcações;

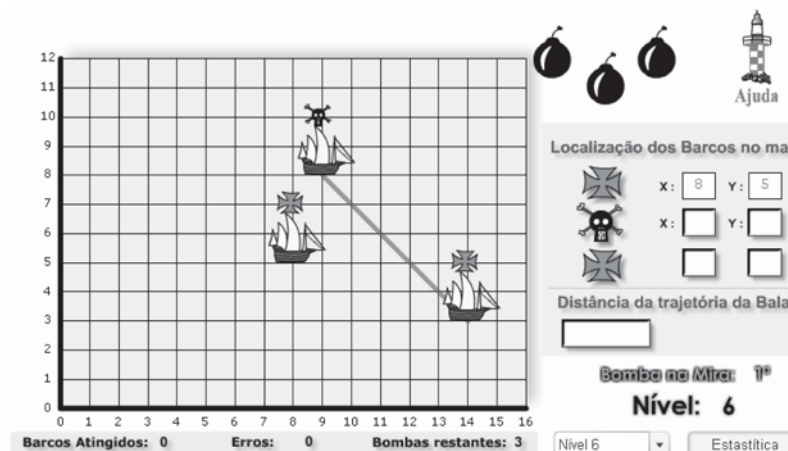


Figura 2: Tela do Nível 6 do Objeto de Aprendizagem Pontos em Batalha

- Em cada nível de dificuldade o usuário terá três bombas para atingir dois barcos inimigos (como na primeira etapa da atividade);
- Caso o usuário acerte o alvo aparecerá uma animação onde o barco de sua esquadra vai explodir e naufragar. Caso ele erre, a animação será de uma explosão no meio da trajetória;

Juntamente com as especificações acima, é possível relatar algumas das problemáticas presentes no Roteiro de Atividades desse OA. São elas: uma vez fechada a caixa de instrução, o usuário só terá possibilidade de vê-la novamente se reabrir o objeto; a ajuda disponível não oferece o feedback necessário para o usuário utilizar o OA, além de não existir nenhuma instrução de como o usuário deverá proceder durante as atividades.

Todas essas problemáticas apresentadas poderiam ser sanadas com a inclusão do processo de formalização, visto que seria passível de se melhorar a qualidade dos produtos desenvolvidos além de reduzir o tempo de implementação do mesmos.

Na próxima seção apresentaremos de que modo podem ser inseridas as redes de Petri coloridas e, conseqüentemente, a formalização dentro do processo de produção dos OAs.

## 5. ESPECIFICAÇÃO FORMAL COM RdPC

Propomos nesse trabalho, uma abordagem formal para tratar os aspectos de interpretação livre, que são associados aos documentos relacionados ao processo de desenvolvimento de OAs. Essa liberdade, por um lado facilita a integração das equipes de desenvolvimento, visto que estas se caracterizam por serem multidisciplinares. Por outro lado, conforme apresentado na seção 4, essa informalidade introduz diversas dificuldades na implementação do OA, fato que pode colocar em risco a própria qualidade pedagógica do produto gerado.

A utilização de uma linguagem formal pode trazer diversos benefícios à produção de OAs. Podemos citar a redução no tempo de desenvolvimento, ocasionada pela clareza gerada por uma descrição formal, a possibilidade de encontrar inconsistências no projeto antes mesmo de se iniciar a implementação, como por exemplo, telas que não possuem caminhos de acesso, e, principalmente, uma melhoria na qualidade dos produtos desenvolvidos.

É válido ressaltar nesse momento que, embora possa parecer estranho uma proposta de introdução de um modelo formal em um processo de desenvolvimento que conta com uma equipe de áreas como pedagogia e psicologia, e que, possivelmente, não possuem conhecimentos sobre esses modelos, essa experiência é

importante, principalmente no caso da adoção de um modelo formal que possui uma representação gráfica, e cujas ferramentas permitem a realização de simulações de comportamento, como é o caso das RdPC. A clareza dessa representação gráfica permite uma excelente comunicação entre a equipe pedagógica e a equipe técnica ou de desenvolvimento, sem, no entanto, afetar a qualidade das representações.

Adicionalmente, é importante considerar que o produtor e principal consumidor dessas especificações serão os desenvolvedores dos OAs, ou seja, possivelmente um grupo formado por profissionais ou estudantes de computação ou engenharia, e que já possuem, normalmente, algum tipo de familiaridade com representações gráficas para o projeto de software, como por exemplo, através de experiências adquiridas em linguagens como UML [22]. Contudo, essa restrição não é obrigatória, visto que temos em trabalhos como [14,15] a utilização de uma extensão de redes de Petri para o apoio à produção de material educacional, atividade desenvolvida, normalmente, por profissionais de áreas como pedagogia.

Para esse trabalho, estamos tratando, particularmente, aspectos de comunicação entre a equipe pedagógica, responsável pelo desenvolvimento do Roteiro de Atividades, com a equipe técnica, responsável por implementar esse documento. Desse modo, para efeitos de organização, dividiremos os aspectos a serem implementados, e portanto passíveis de modelagem, em três, a saber:

- Modelagem de Fluxo: Indica relação entre as telas em um OA. Por exemplo, a tela de exercícios deve ser apresentada depois da tela de conteúdo instrucional.
- Modelagem de Interação: Indica as possibilidades de interação do usuário, como por exemplo, o botão “X” só vai estar ativo se for inserido algum valor na caixa de entrada “Y”.
- Modelagem de Navegação: Indica as ações associadas ao fluxo de um OA, como por exemplo, pressionando o botão “ajuda” é apresentado um *pop-up* com o conteúdo da ajuda.

Para a realização da modelagem desses aspectos, propomos um novo modelo usando redes de Petri coloridas, que consiga representar mais facilmente essas informações. Esse novo modelo, que denominamos de LOCPN (*Learning Objects production with Colored Petri Nets*), é apresentado na próxima seção.

### 5.1. O MODELO LOCPN

Para a realização da modelagem dos OAs será adotado nesse trabalho as redes de Petri coloridas tradicionais, com alguns refinamentos e adições.

**Definição 2.** Uma *Learning Objects production with Colored Petri Net* é definida como LOCPN = (S, SN, AS, PN, PT, PA, AT), onde:

- S é um conjunto finito de páginas, onde cada  $i \in S$  é uma RdPC ( $\sum_i, Li, Ti, Ai, Ni, Ci, Gi, Ei, Ii$ ), como apresentado na Definição 1, com as seguintes alterações:
  - $\sum_i = \langle C_0, e \rangle$  representando um conjunto não vazio de fichas (cores), no qual  $C_0$  representa cores ordinárias que não carregam nenhuma informação e  $e$ , quando presente em um lugar que represente uma tela/janela de um OA, simboliza a ativação da tela/janela.
  - $L_i = \{l_1, l_2, \dots, l_m\} \cup \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de lugares, que é formado por dois subconjuntos, o primeiro, representa locais comuns (simbolizado graficamente por círculos/elipses com bordas simples), o segundo é formado por lugares que representam telas/janelas de um OA (simbolizado graficamente por círculos/elipses com bordas grossas)
- SN é um conjunto de nós de substituição, onde  $SN \subseteq T$ .
- SA é uma função de atribuição de páginas
- PN é um conjunto de nós que representam portas
- PT é uma função de tipo de porta, definida como  $PT: PN \rightarrow \{in, ou, in/out\}$
- PA é uma função de atribuição de portas
- AT é uma função de ativação, que representa a ativação da tela de um OA. Essa função é uma extensão da função  $C_i$  e é definida por  $AT: Lt \rightarrow \{e\}$ , em que  $Lt$  é o subconjunto de lugares que representa telas/janelas de um OA.

Pela Definição 2, algumas características próprias das redes descritas com o modelo LOCPN podem ser observadas. Inicialmente, alguns lugares representam telas/janelas de implementação de OAs. Essas telas podem ser tanto telas tradicionais como também telas de *pop-ups* ou caixas de diálogo. Esses lugares são modelados graficamente como um círculo de bordas grossas, como exemplificado na Figura 3. Outros tipos de lugares são representados normalmente com um círculo de bordas finas.



Figura 3: Representação de uma Tela Ativa

Quanto às fichas (cores), alguns valores possuem um significado importante nas representações de OAs em LOCPN. A ficha  $e$ , por exemplo, quando presente em lugares que representam telas, indicam que esse controle está em foco (ativo) nesse momento. A Figura 3, por exemplo, mostra a representação de uma tela, denominada Tela1, que atualmente está ativa, visto que a ficha  $e$  está nesse lugar atualmente.

Os demais tipos de fichas, caso estejam presentes em lugares representando telas de um OA, representam os controles (widgets) associados às telas, como por exemplo botões e caixas de texto. Na Figura 4, por exemplo, é apresentada uma tela, denominada Janela2, que possui 3 controles, btOK, btFechar e btHelp, representando 3 botões na interface do OA. Caso essas fichas estejam em outros tipos de lugares, não possuem representação específica.

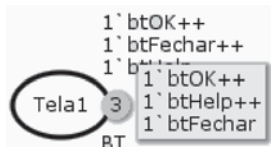


Figura 4: Representação de uma Tela com 3 Controles

Quanto às transições que atuam em lugares que representam telas, essas devem ou ser marcadas com pesos indicando as ações sobre os controles da tela, ou indicam uma mudança de foco, pela passagem da ficha *e*. Na Figura 5 é representada uma transição entre as telas Tela1 e Tela2 através da utilização do controle btOK de Tela1. Essa ação leva o foco para a Tela2 (passagem da ficha *e* para a Tela 2).

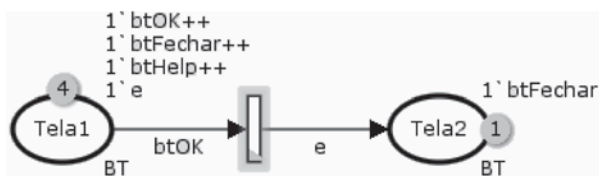


Figura 5: Representação de uma Transição entre Telas

## 5.2. MODELAGEM DE FLUXO

Para realizar a modelagem de fluxo em um OA, deve-se, inicialmente, ter compreendido precisamente as relações entre as telas que formam um OA. Essas telas são controles onde os conteúdos são mostrados e devem ser apresentadas em uma seqüência capaz de facilitar, em termos pedagógicos, a compreensão dos conteúdos relacionados ao OA. É válido ressaltar que nesse artigo não são tratadas questões relacionadas ao projeto pedagógico do OA, assim, deve ser aceito que o Roteiro de Atividades foi elaborado pela equipe pedagógica e que deve ser implementado pela equipe técnica sem a necessidade de contestar o projeto instrucional do OA.

Algumas situações são possíveis na modelagem de fluxo entre as telas de um OA. Ou seja, existe um conjunto de transições entre telas passíveis de serem modeladas em um OA. De um modo geral esse conjunto pode ser reduzido às seguintes situações:

- Fluxo Direto Interativo: Uma tela é apresentada a partir da realização de alguma atividade de interação do usuário, como por exemplo, um botão explicitamente invoca uma tela com informações adicionais sobre o conteúdo que está sendo apresentado no OA.

- Fluxo Direto Automático: Uma tela é apresentada espontaneamente sem a requisição direta do usuário. Nesse caso não existe uma invocação explícita do usuário. Um exemplo para essa situação é a apresentação de um *pop-up* com algum aviso sobre a utilização de um exercício no OA.
- Fluxo Direto Condicional: Nesse caso uma tela deve ser apresentada dependendo de alguma informação relacionada ao estado da tela que realizou a invocação. Por exemplo, uma tela pode ser apresentada caso o usuário do OA tenha errado três vezes consecutivas um determinado exercício.

De modo a permitir uma melhor visualização do fluxo de controle em um OA, é adotado nesse trabalho uma representação hierárquica na modelagem. Nessa representação, são modeladas em um primeiro momento, as principais telas (janelas) e suas relações, sendo que, posteriormente cada uma das telas devem ser modeladas separadamente. Essa representação hierárquica, exemplificada na Figura 6, além de facilitar a compreensão da rede, visto que esse formato permite que sejam agrupados apenas elementos que estão diretamente relacionados em cada nível hierárquico, também facilita a compreensão dos aspectos de fluxo de controle do OA.

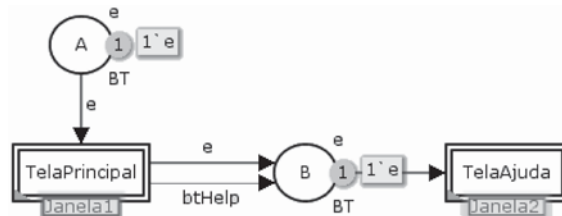


Figura 6: Representação Hierárquica da Estrutura de um OA

Na Figura 6 é representado um OA formado por duas telas, a primeira, denominada TelaPrincipal tem sua modelagem detalhada pela rede indicada no nome descrito na parte inferior da transição (Janela1). A segunda, denominada TelaAjuda, é detalhada na especificação Janela2. Existem duas conexões (denominadas lugares A e B) entre as duas especificações de telas. Esses lugares são pontos de entrada e saída de informações entre as modelagens individuais e servem de “cola” entre as especificações.

Para a realização da modelagem dos aspectos de fluxo de um OA deve-se considerar, inicialmente a seqüência de ativação das telas, ou seja, a partir de uma tela qual(is) outras telas podem ser alcançadas. Para atingir esse objetivo, será utilizado, como apresentado anteriormente, um tipo de ficha que possui um significado especial no modelo LOCPN, representado pela ficha *e*. Essa ficha, quando presente em um lugar que representa uma tela, indica que o foco está exatamente naquele elemento. Desse modo, o controle de fluxo em um OA deve ser realizada pelo gerenciamento da ordem com que essas fichas são passadas através das transições entre as telas.

Assim, observando as três situações listadas anteriormente com relação aos tipos de fluxos, será descrito, inicialmente, como realizar a representação formal de um fluxo direto interativo. Esse tipo de fluxo é realizado por ações de interação em controle nas telas de um OA.

A Figura 7, por exemplo, apresenta um modelo no qual uma tela, (Main), aciona uma outra tela (Help) através da ação do controle btHelp. Ou seja, o acionamento do botão, representado pela ficha btHelp em Main, e pela presença do peso com mesmo nome que liga esse lugar à transição, indica o direcionamento do fluxo para a tela Help.

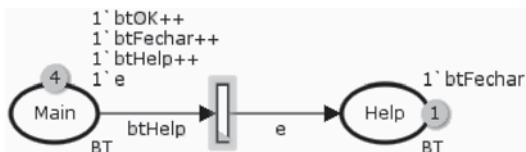


Figura 7: Fluxo Direto Interativo.

Na representação de um fluxo direto automático, não existe a necessidade da interação do usuário com os controles de uma tela do OA para que a transição seja ativada. Nesse caso, como apresentado na Figura 8, o acionamento da tela Help, se dá apenas pela inserção da ficha e no lugar Help.

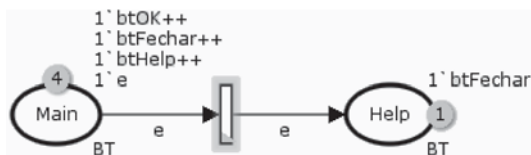


Figura 8: Fluxo Direto Automático

Já na representação de um fluxo direto condicional, deve ser considerada uma condição para que uma transição seja disparada. Como apresentado na Figura 9, a transição só será habilitada se o valor da sua expressão de guarda for verdadeira. Nesse exemplo, apenas se o valor de y for maior que 5, haverá a habilitação da transição.

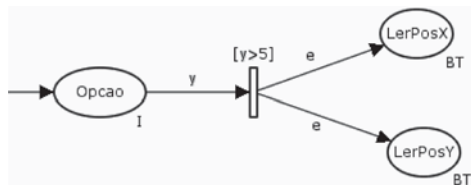


Figura 9: Fluxo Direto Condicional

### 5.3. MODELAGEM DE INTERAÇÃO

As interações do usuário devem ser modeladas de acordo com as funções dos controles. De fato, cada tipo de controle permite um conjunto diferente de interações, e, desse modo, uma forma diferente de modelagem formal. Para exemplificar a especificação desse aspecto de um OA, é representada na Figura 10, uma interação do usuário em um controle do tipo ComboBox. Esse tipo de controle pode

ser utilizado na realização de uma atividade de múltipla escolha por um estudante, por exemplo.

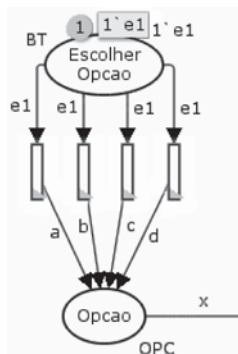


Figura 10: Interação em um ComboBox

Como apresentado na Figura 10, a presença de fichas do tipo e1 no lugar EscolherOpcao habilita as transições que representam as opções do ComboBox. Assim, o usuário fica livre para escolher entre as opções (a, b, c ou d). Quando uma das opções for selecionada, cuja ação é modelada pelo disparo da transição relativa à sua escolha, uma ficha com o valor correspondente ao peso do arco que liga a transição ao lugar Opcao é depositada em Opcao. Esse valor fica disponível na variável x, que liga o lugar Opcao a qualquer outra transição, para ser utilizado posteriormente.

### 5.4. MODELAGEM DE NAVEGAÇÃO

Na modelagem dos aspectos de navegação, devem ser consideradas as ações que controlam o fluxo entre as telas de um OA. Assim, esse aspecto trabalha regulando as formas de seqüenciamento de ações dentro de um OA. Por exemplo, na Figura 7, a tela Help é ativada a partir da tela Main, contudo o controle desse fluxo deve ser regulado para que possa ser definido se ambas as telas estarão ativas ou se o foco estará em apenas uma delas.

Para tratar esses tipos de informação na modelagem de um OA, o fluxo deve ser controlado. Na Figura 11, é apresentado um modelo no qual a tela Main, aciona a tela Help através da ação do controle btHelp. Como pode ser observado, no lugar que representa a tela Main estão presentes as fichas btHelp, e e. Somente se essas duas fichas estiverem presentes é que a transição para a tela Help estará habilitada. Esse fato é representado pela existência dos dois arcos ligando o lugar Main com a transição, onde estão apresentadas as expressões que validam essa transição em forma de pesos nos arcos. Isso significa que, além da tela estar ativa (representado pela ficha e em Main), a transição é acionada pela utilização do botão btHelp.

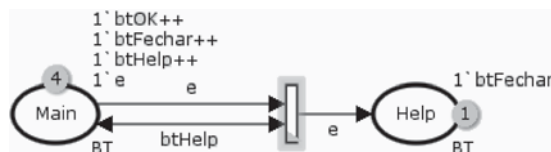


Figura 11: Controle de Navegação



Também pela Figura 11, pode ser observado que o acionamento da transição levará a ficha *e* para o lugar Help, o que indica que o foco será passado para essa tela. É importante notar que o arco com a inscrição btHelp é bidirecional. Isso se dá pela necessidade de permitir o retorno da ficha btHelp para o lugar Main, para que, eventualmente, esse controle possa ser reutilizado.

## 6. ESTUDO DE CASO

Para mostrar a expressividade do modelo LOCPN, são modeladas nessa seção algumas funcionalidade do OA Pontos em Batalha, apresentado na seção 4. Contudo, antes de apresentar essas modelagens, serão mostrados alguns fragmentos do Roteiro de Atividades desse OA. Esses fragmentos servirão de base para as modelagens e ajudarão na compreensão das vantagens da adoção do modelo LOCPN no processo de desenvolvimento de um OA.

O OA Pontos em Batalha (PemB) foi implementado, em sua primeira versão, utilizando o modelo de processo do RIVED, já com os devidos ajustes propostos em [1]. Desse modo, essa primeira versão seguiu fielmente o Roteiro de Atividades como única forma de comunicação entre a equipe pedagógica e a equipe técnica. Para facilitar a citação dessa versão nesse trabalho será usado o termo V1 para identificar essa implementação.

A versão 2 do PemB, que será denominada daqui para frente de V2, foi desenvolvida a partir de ajustes na V1 e utilizando especificações LOCPN. Essas especificações foram construídas pela equipe técnica, em parceria com a equipe pedagógica, que validou os modelos apresentados em uma ferramenta denominada CPNTools [10]. Essa ferramenta, que permite a animação da execução de redes de Petri coloridas, possibilitou que diversos ajustes no comportamento do OA fossem observados e discutidos por ambas as equipes antes de iniciar o processo de codificação do OA. Isso retirou do desenvolvedor a responsabilidade de decidir pequenas situações que, em muitos casos, acabam tendo valor desprezado a princípio, mas que podem afetar a qualidade final do objeto.

### 6.1. IMPLEMENTAÇÃO BASEADA NO ROTEIRO DE ATIVIDADES

O Roteiro de Atividades (ou simplesmente Roteiro), como apresentado na seção 3, é uma ferramenta importante em termos de documentação e, principalmente, de comunicação entre a equipe técnica e a equipe pedagógica. Como discutido no início desse trabalho a multidisciplinaridade inerente às equipe de desenvolvimento de OAs, podem inserir nos produtos, defeitos difíceis de serem detectados antecipadamente.

Para ilustrar mais precisamente esse fato, seguem alguns tópicos relacionados no Roteiro do PemB, gerados pela equipe pedagógica, e que foram implementados pela equipe técnica.

- 1) As coordenadas da caravela são fornecidas, cabendo ao usuário descobrir as coordenadas da embarcação pirata;
- 2) Descoberta as coordenadas da embarcação

pirata, o usuário deverá informa-la na caixa “localização dos barcos no mar”;

- 3) Inserido o valor de X e de Y da embarcação pirata na caixa “localização dos barcos no mar” o usuário deverá efetuar o cálculo da distância entre os dois pontos;
- 4) Caso o usuário não saiba como efetuar esse cálculo, o objeto disponibiliza um ícone “ajuda” que fica no campo superior a direita da tela e é simbolizado por um farol;
- 5) Calculada a distância, o usuário deverá inserir o resultado do seu cálculo na caixa “distância da trajetória da bala”;
- 6) Imediatamente, após inserir o valor na caixa especificada no passo 5, será fornecido o botão “atirar” em que o usuário irá verificar se seu cálculo está correto.
- 7) O botão “estatísticas” pode ser utilizado para mostrar uma tela com estatísticas de erros e acertos nos tiros.

Esses itens foram implementados na V1 do PemB. Nessa implementação, algumas suposições foram realizadas a partir da especificação do Roteiro. Desse modo, alguns comportamentos não desejáveis foram observados na implementação. Dentre eles os seguintes foram os mais importantes:

- 1) A caixa para entrar com o valor da distância a ser percorrida pela bala fica sempre ativa, independente de se ter preenchido os valores das coordenadas X e Y das embarcações;
- 2) Quando a tela “estatísticas” está ativa ainda se pode ativar a ajuda;
- 3) O controle relativo às coordenadas da caravela são fixos, mas mesmo assim eles receberam o foco.
- 4) A distância pode ser preenchida antes das coordenadas serem inseridas.

Embora esses pequenos problemas possam, em um primeiro momento, parecer pouco relevantes, eles podem gerar algumas situações indesejáveis no comportamento do OA. Além disso, esses erros de implementação mostram a fragilidade do processo de desenvolvimento. De fato, mesmo com um conjunto bastante simples de requisitos identificados no Roteiro, vários pequenos problemas de implementação foram introduzidos no OA. Isso significa que em roteiros mais extensos e complexos, as chances de erros mais severos devem aumentar consideravelmente.

A qualidade de um OA, como um produto de software com finalidade educativa, deve auxiliar na compreensão e fixação de conceitos formais. Desse modo, problemas na execução de um OA podem afetar também o aprendizado. Nesse sentido, o propósito pedagógico de um OA se minimiza se os alunos não estão satisfeitos com sua implementação.

## 6.2. MODELAGEM EM LOCPN

A descrição do PemB usando LOCPN foi realizada em duas etapas. Na primeira, a estrutura geral da aplicação foi descrita em termos de telas e de suas

relações. Em seguida, observando o caráter hierárquico de LOCPN, foram realizadas as especificações detalhadas de cada um dos componentes da descrição geral.

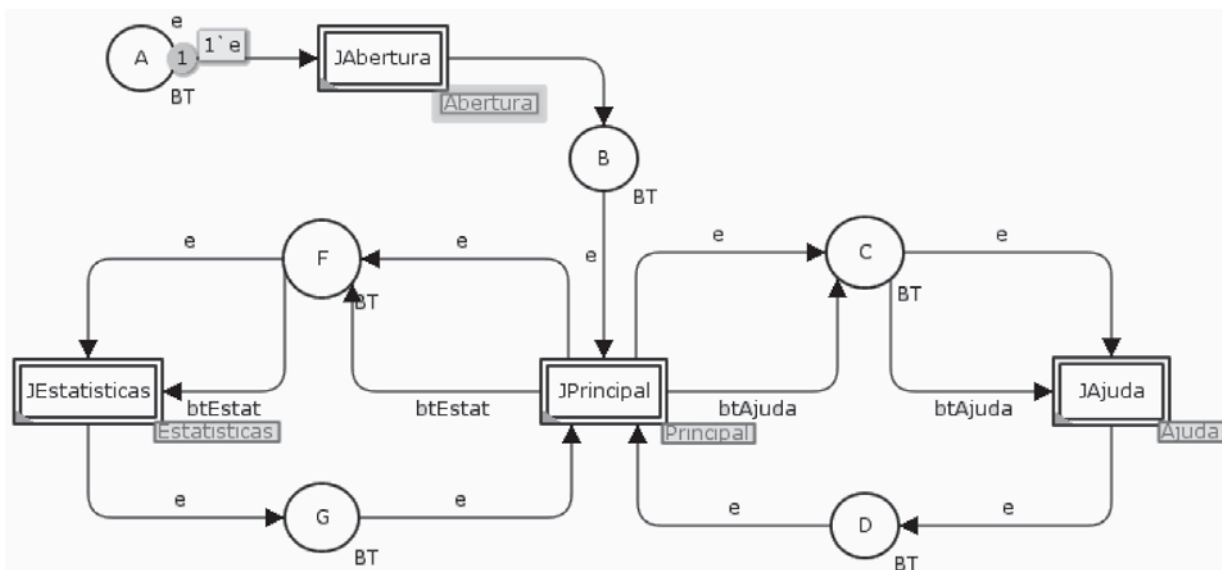


Figura 12: Modelagem Geral do OA Pontos em Batalha (PemB)

Na Figura 12 é apresentada a especificação geral do OA PemB. Nessa figura as telas do OA PemB são representadas pelas transições JAbertura, que representa a tela de abertura do OA; JPrincipal, representando a tela principal de trabalho do OA; JAjuda, que representa a tela de ajuda e JEstadísticas representando a tela de estatísticas. Essas transições que substituem a especificações complexas são denominadas no CPNTools de transições de substituição [10], e representam subredes que devem ser expandidas. Assim, como apresentado na seção 5, em cada uma das transições há a definição do nome da rede que essa transição está representando. Adicionalmente, também são representadas na especificação, os lugares ou portas de conexão entre essas transições. Dessa forma, a porta C, por exemplo, é uma porta de saída para a tela JPrincipal e uma porta de entrada para a tela JAjuda.

Observando os principais aspectos relacionados a especificação de um OA, descrito na seção 5. E considerando os pontos do Roteiro da descrição do OA PemB, indicado no início dessa seção. Serão apresentados a seguir um conjunto de especificações que conseguem tratar esses pontos do Roteiro. Como a intenção principal dessa seção é comprovar a relevância do modelo LOCPN como mecanismo proposto para aumentar a precisão na interpretação do comportamento dinâmico de um OA, serão apresentados apenas as especificações que são importantes para esse fim. Assim, para não prejudicar a clareza dessas especificações, serão suprimidos alguns detalhes que não são relevantes.

Inicialmente é apresentada na Figura 13 a especificação das telas de abertura do PemB. Nessa especificação pode ser vista duas telas, a primeira, denominada JAbertura, em que são apresentadas informações gerais sobre o grupo de desenvolvimento do OA. Essa tela, ao ser fechada através de um botão (ação representada pelo arco com inscrição btFechar, ligando essa tela à transição), abre automaticamente a tela JInstrucoes, onde são apresentadas instruções gerais sobre o funcionamento do OA. Essa última tela, por sua vez, ao ser fechada, com ação similar à tela anterior, envia a ficha  $e$  através da porta de saída B, repassando o foco da aplicação para uma outra tela.

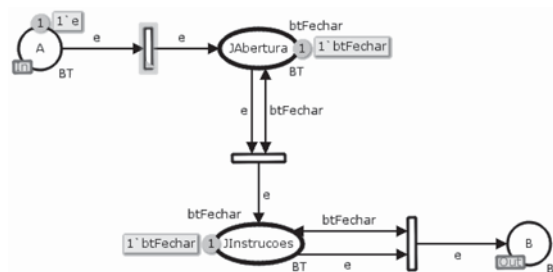
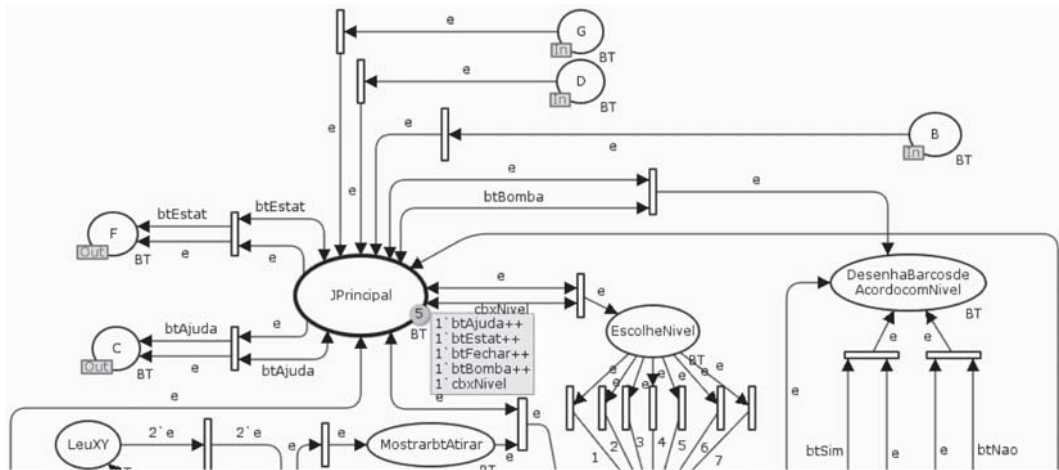


Figura 13: Especificação das Telas de Abertura

Pode ser observado claramente nessa especificação as relações de fluxo, identificadas pela passagem da ficha  $e$  entre as telas, fornecendo a seqüência de ativação dessas telas, e também as questões relacionadas à navegação, na qual as inscrições nos arcos controlam o fluxo de navegação entre as telas.

De acordo com a especificação da Figura 12, o fluxo entre as telas indica que há um relacionamento dessa especificação com a tela JPrincipal, dessa forma, a porta de

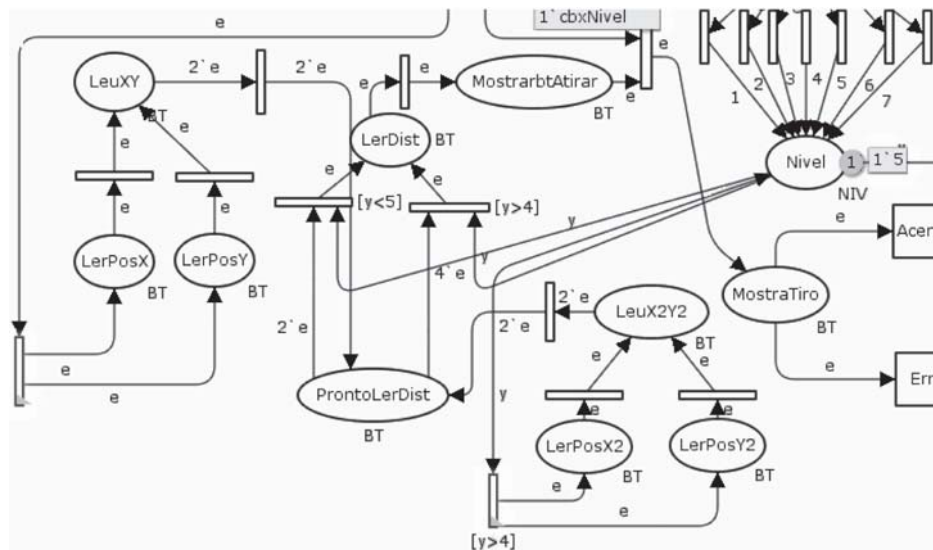
saída B, lança sua ficha para uma porta de entrada da especificação da tela JPrincipal, como apresentada no trecho de especificação dessa tela pela Figura 14.



**Figura 14:** Trecho de Especificação da Tela JPrincipal

Com relação a interatividade do PemB, alguns problemas podem ser tratados em nível de especificação LOCPN. Na relação de problemas identificados na seção 6.1, tem-se, no item 4, que a entrada da distância a ser percorrida pela bala pode ser preenchida antes de se ter inserido os valores

para as coordenadas do barco pirata. Essa seqüência de ações sutis, acabou sendo negligenciada na implementação da V1 do PemB. Contudo, no decorrer da especificação desse OA usando o modelo LOCPN, a especificação da Figura 15 não deixa margem a interpretação para essas ações.



**Figura 15:** Modelagem de Interatividade no preenchimento de informações no PemB

Nessa especificação (Figura 15), pode ser observado que, dependendo do valor dos níveis, há a definição de quais campos devem ser preenchidos e em que ordem esse preenchimento deve ocorrer. De fato, entre os níveis 1 e 4, apenas o valor das coordenadas X e Y do barco pirata deve ser informado, e em seguida a distância a ser percorrida pela bala para atingir esse barco. Já entre os níveis 5 e 7, há a necessidade de se informar, adicionalmente, as coordenadas de outra caravela (Figura 2).

Para modelar essa estrutura, no lugar Nível (Figura 15), está uma ficha com o valor do nível a ser utilizado na interação (selecionado em um ComboBox). Essa ficha, cujo valor capturado pela variável y presente nos arcos de saída desse lugar, indica a forma com que os campos devem ser inseridos pelo usuário. Inicialmente, as posições X e Y da caravela estão aptas a receber valor, fato sinalizado no modelo pela habilitação da transição de entrada dos lugares LerPosX e LerPosY. Quando disparada essa transição indica que é possível ler os valores de X e Y em qualquer ordem.

Com a presença de uma ficha com valor 5 no lugar Nivel, a transição de entrada para a leitura das coordenadas da segunda caravela também ficam ativas (transição de entrada para os lugares LerPosX2 e LerPosY2). Ao término das leituras das coordenadas das duas embarcações, indicada pela presença de duas fichas do tipo *e* nos lugares LeuXY e LeuX2Y2, a leitura da distância deve ser realizada.

O fato de se estar apto a realizar a leitura da distância é modelada pela presença de fichas no lugar ProntoLerDist. Nesse lugar, podem haver duas ou quatro fichas, dependendo do nível do jogo. Essas duas possibilidades são avaliadas nas transições que ligam esse lugar ao lugar LerDist. A necessidade de duas transições é explicada pela necessidade de se interpretar a entrada das coordenadas. De fato, se o jogo estiver entre as fase 1 e 4, duas entradas devem ser realizadas (coordenadas do barco pirata), mas caso o jogo esteja entre os níveis 5 e 7, o usuário estar livre para entrar, em qualquer ordem, as coordenadas de dois barcos, assim, o fato de se ter duas fichas no lugar ProntoLerDist, deve ser avaliada pelas transições, pois a presença de duas fichas pode ter dupla interpretação (estamos no nível entre 1 e 4, ou estamos nos níveis entre 5 e 7 e entramos apenas com os dados de uma embarcação)

Para solucionar esse problema, na modelagem da Figura 15, na primeira transição de entrada mais a esquerda que liga os arcos de saída de ProntoLerDist, existe uma expressão de guarda ( $[y < 5]$ ), indicando que essa transição só é válida para os níveis inferiores a 5, contudo essa transição só é habilitada se o valor do arco for verdadeiro, no caso deve haver a presença de duas fichas do tipo *e* no lugar ProntoLerDist. Como essas fichas chegam a esse lugar aos pares (X,Y) ou (X2,Y2) em decorrência dos disparos das transições que ligam os lugares LeuXY e LeuX2Y2 respectivamente, ao lugar ProntoLerDist, isso indica que o OA está apto a ler a distância. Caso contrário, caso modelado pela transição a direita, na qual existe uma expressão de guarda ( $[y > 4]$ ), indicando que o jogo está entre o nível 5 e 7, deve haver quatro fichas no lugar ProntoLerDist, indicado pelo peso do valor do arco de entrada para esse transição, o que sugere que os valores de todas as coordenadas foram fornecidos.

### 6.3. DISCUSSÃO

O modelo LOCPN vem sendo utilizado pelo Grupo de Pesquisa e Produção de Ambientes Interativos e Objetos de Aprendizagem - PROATIVA [12] da Universidade Federal do Ceará, desde o início de 2007. Essa modelo tem sido utilizado como ferramenta de apoio ao processo de desenvolvimento de OAs e os resultados de sua aplicação vem demonstrando a importância da abordagem formal para a produção de software de qualidade. Entretanto, a introdução de uma ferramenta formal em um grupo heterogêneo e, principalmente, que tem por objetivo desenvolver software para fins educativos, vem também sendo um grande desafio.

Como resultados principais da utilização prática de LOCPN pelo PROATIVA, algumas vantagens se tornaram claras em um curto espaço de tempo. A vantagem mais

clara está na qualidade dos resultado de implementação de OAs. Essa qualidade pode ser verificada pela diminuição drástica no número de erros identificados pela equipe de testes de OAs.

Um outro fator importante é a melhoria da documentação. De fato, a própria modelagem em LOCPN é um modelo completo de funcionamento dos OAs. Essa documentação é gerada pela equipe técnica, em um trabalho sincronizado com a equipe pedagógica antes do início do processo de codificação dos OAs. E como as dúvidas relacionadas à implementação já são resolvidas no momento da construção da modelagem formal do OA, a documentação não sofre alterações significativas posteriormente enquanto o OA está sendo implementado. Anteriormente, com a utilização somente do Roteiro de Atividades, diversas modificações eram realizadas no momento da implementação, pelo próprio programador, que interpretava as informações de forma livre. Visto que essas interpretações não eram reproduzidas no Roteiro, essa documentação acabava ficando desatualizada e sem utilidade.

Um outro fator bastante relevante na adoção de LOCPN foi a introdução da participação da equipe pedagógico no início do processo de desenvolvimento. De fato, com o modelo antigo, a equipe pedagógica apenas produzia o Roteiro e repassava para a equipe técnica. Como alguns aspectos relacionados à implementação só ficam claros com a produção de algum protótipo que possa mostrar a execução do OA. Diversas melhorias em termos de funcionalidades que eram sugeridas pela equipe pedagógica de modo a aperfeiçoar os aspectos de aprendizagem através dos OAs, foram antecipados no processo. Assim, ao participar da elaboração das especificações em LOCPN dos OAs, as equipes pedagógicas já inseriam melhorias no projeto. Essas melhorias podem ser observadas pela antecipação da visualização da dinâmica dos OAs fornecidas pelos modelos formais e pelas simulações nas ferramentas. Esse fator também contribui para uma produção mais rápida de resultados pela equipe.

Todos os fatores apresentados anteriormente têm, por consequência, uma redução no tempo de desenvolvimento. Embora pareça estranho se inserir uma etapa a mais no modelo de processos e mesmo assim ter uma diminuição no tempo de desenvolvimento esse fator é facilmente explicado pela qualidade das especificações geradas utilizando essa abordagem. De fato, com a adoção de uma modelagem formal para os OAs houve um aumento significativo na precisão das informações repassadas para a fase de implementação. Assim, com uma implementação baseada em uma especificação clara e detalhada, e portanto, com menos erros a corrigir e menos dúvidas para e implementação, os OAs acabam sendo produzidos mais rapidamente.

Entretanto, a introdução dessa abordagem no processo de desenvolvimento de OAs do PROATIVA não está sendo uma tarefa fácil, visto que houve, particularmente no início da utilização de LOCPN, uma certa resistência das equipes em adotar esse modelo. Como toda equipe de

desenvolvimento de software, há sempre uma certa impaciência em iniciar a fase de implementação. Como a introdução da modelagem dos OAs em LOCPN significa mais uma etapa que retarda o início da atividade de desenvolvimento, o grupo teve que ser convencido da importância dessa atividade. Portanto, a estratégia adotada pelo grupo foi a de utilizar LOCPN para o desenvolvimento de apenas alguns OAs, de modo que o grupo pudesse se familiarizar com a nova linguagem e perceber as vantagens em adotá-la. No entanto, como os resultados em termos de qualidade e redução de tempo na produção dos OAs trazidos pelo uso de LOCPN já estão se tornando evidente, atualmente a maioria dos OAs do grupo já estão sendo modelados em LOCPN.

## 7. TRABALHOS RELACIONADOS

A utilização de Redes de Petri como mecanismo de validação de interfaces de aplicações pode ser encontrada em alguns trabalhos. Em [13], por exemplo, é apresentado o ICO (*Interactive Cooperative Objects*), que é um formalismo para projeto de sistemas interativos baseado em objetos, no qual o comportamento dos mesmos é modelado com Redes de Petri Hierárquica. Nesse trabalho, a estrutura utilizada para representar as interações torna a representação do sistema bastante complexa e, portanto, inviabiliza a utilização dessa representação como mecanismo de comunicação em equipes de desenvolvimento de OAs.

Em [14] é apresentado o sistema OSU (*Oregon Speedcode Universe*), que utiliza Redes de Petri ordinárias na descrição de interfaces de aplicações. Nesse sistema, cada ação do usuário é mapeada em uma transição do sistema. Por conta disso, as especificações se tornam bastante extensas e de difícil análise.

Alguns trabalhos propõem a utilização de Redes de Petri em OAs. Em [15] é desenvolvido um modelo de Rede de Petri Colorida, denominada DCPN (*Distance Learning Color Petri Net*) que permite a modelagem da seqüência de atividades relacionadas a um OA. Esse modelo possibilita a descrição do controle de acesso aos conteúdos de OAs compartilhados a partir do modelo de seqüenciamento do padrão SCORM (*Sharable Content Object Reference Model*) [16]. Esse modelo gerou uma ferramenta de autoria, denominada MIME [17], para o suporte ao projeto de cursos através de OAs.

Também em [15] é descrito um modelo, baseado em Redes de Petri Hierárquicas, para o projeto formal de seqüência de atividades pedagógicas em um OA. Esse modelo, denominado OOCM (*Object Oriented Course Modeling*), permite a modelagem de componentes básicos de seqüência, denominados *Object-Oriented Activity Tree* (OOAT). Cada OOAT representa o comportamento de uma seqüência de atividade de aprendizagem juntamente com um conjunto de regras baseadas no SCORM.

Todos os trabalhos apresentados nessa seção, diferem da nossa proposta por apresentarem um maior grau

de complexidade no desenvolvimento de seu produto. No LOCPN, o objetivo é facilitar e clarificar a comunicação entre as equipes pedagógica e técnica de modo que elas façam uso exclusivo apenas dessa notação para a produção do objeto, sem se preocupar com outros tipos de regras para formação do OA.

## 8. CONCLUSÃO E TRABALHOS FUTUROS

A utilização de OAs como ferramenta de aprendizado apoiado pelo uso do computador tem-se mostrado uma estratégia bastante relevante já há algum tempo. Atualmente, diversos grupos de pesquisa têm produzido no Brasil um grande número de resultados que comprovam essa idéia [12,18]. Esses grupos, muitas vezes, são financiados pelo próprio governo brasileiro, através de editais de projetos de pesquisa específicos para esse tema. Um exemplo melhor do interesse governamental está apresentado em [9], que é uma ação da Secretaria de Educação a Distância (SEED) que tem como objetivo a produção de conteúdos pedagógicos digitais na forma de OAs.

A produção de OAs, como em todo produto de software, deve seguir um modelo de processo que possa otimizar o desenvolvimento desses produtos. Entretanto, esses produtos de software são diferenciados pelos estritos requisitos pedagógicos que esses devem implementar. De fato, para tratar esses requisitos, as equipes envolvidas no processo de desenvolvimento de OAs são multidisciplinares, dentro das quais estão envolvidas pessoas com os mais diversos tipos de conhecimentos específicos. Esses grupos, por sua vez, freqüentemente são divididos em equipes pedagógicas, as quais são responsáveis pelo projeto pedagógico dos OAs, e em equipes técnicas, responsáveis pelo desenvolvimento dos mesmos. Experiências concretas têm mostrado que a heterogeneidade dessas equipes tem dificultado a comunicação de diversos detalhes importantes para a implementação desses produtos.

Nesse trabalho foi discutido a introdução de uma nova etapa formal no processo de produção de OAs. Para dar suporte a essa nova etapa, foi proposto um formalismo denominado LOCPN (*Learning Objects production with Colored Petri Nets*). Esse modelo, que é baseado em redes de Petri coloridas, permite a produção de especificações formais de OAs capazes de tornar mais clara as especificações antes de iniciar o processo de implementação. Dessa forma, há uma melhoria significativa nos mecanismos de comunicação entre as equipes pedagógicas e técnicas, o que pode ser comprovado pelos resultados em termos de qualidade e de redução do tempo de produção de OAs quando da aplicação dessa abordagem. Essa melhoria é refletida também na diminuição dos erros que levam a uma re-implementação dos OAs.

Por ser um modelo baseado em uma estrutura gráfica, LOCPN fornece as informações necessárias para se inferir sobre o comportamento de um OA antes de realizar sua

implementação. Com a utilização de ferramentas de simulação de modelos é possível também verificar o comportamento dinâmico de OAs complexos e realizar os devidos ajustes.

Atualmente, está sendo iniciado um estudo para o desenvolvimento de um possível simulador de LOCPN que é capaz de produzir telas de OAs a partir de modelos LOCPN. Esse simulador permitirá a criação automática de telas com controles de interação e possibilitará uma melhor compreensão da dinâmica de OAs em termos de fluxo de atividades, navegação e relações de interação com usuários.

## REFERÊNCIAS

- [1] L. L. do Amaral, T. de A. Gomes, M. de F. C. de Souza, J. A. de Castro Filho, M. C. Pequeno. Um Aprimoramento do Modelo de Processo de Criação de Objetos de Aprendizagem do Projeto RIVED. WIE 2006 - Workshop de Informática na Educação. Campo Grande, MS, Brasil, 14-20 Julho, 2006.
- [2] D. A. Wiley. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy in D. A. Wiley (Ed.), *The Instructional Use of Learning Objects*. <http://reusability.org/read/chapters/wiley.doc>, 15 Fev 2007.
- [3] IEEE. Learning Technology Standardization Committee (LTSC). <http://wiki.unifacs.br/gnufacs/twiki/bin/view/ObjetosDeAprendizagem/WebHome>: 19 Mar 2007.
- [4] R. W. de Bettio, A. Martins *Objetos de aprendizado: um novo modelo direcionado ao ensino a distância*. In: 9o. Congresso Internacional de Educação a Distância, 2002, São Paulo - SP. 9o. Congresso Internacional de Educação a Distância, 2002. <http://www.universiabrasil.net/materia/materia.jsp?id=5938>. 24 de Mai de 2007.
- [5] N.L.R. Fernandes. Professores e computadores: navegar é preciso, Porto Alegre: Mediação, pp. 36-41. 2004
- [6] Labvirt. <http://www.labvirt.futuro.usp.br>, 30 Jun. 2007.
- [7] W. Longmire. *A Primer On Learning Objects*. American Society for Training & Development. Virginia. USA. 2001.
- [8] T. Murata. Petri Nets: Properties, Analysis and applications. *Proceedings of the IEEE*. Vol.77, nº 4, 541-580, 1989
- [9] RIVED. <http://www.rived.mec.gov.br>. 30 Jun. 2007.
- [10] A. V. Ratzer, L. Wells, H. M. Lassen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. West-ergaard, S. Christensen, K. Jensen. CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In W.M.P. van der Aalst and E. Best, editors, *Applications and Theory of Petri Nets 2003: 24th International Conference, ICATPN 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 450-462. Springer Verlag, Jan 2003.
- [11] A. Nascimento, E. Morgado. Um projeto de colaboração Internacional na América Latina. [http://www.rived.mec.gov.br/site\\_objeto\\_lis.php](http://www.rived.mec.gov.br/site_objeto_lis.php), 29 Mai. 2007.
- [12] PROATIVA. <http://www.proativa.vdl.ufc.br>, 30 Mai 2007.
- [13] P. Palanque, R. Bastide. Modeling clients and servers in the Web using Interactive Cooperative Objects, *Formals Methods in HCI*, Springer, 1997, pp.175-194
- [14] H.C. Keh, T.G. Lewis. Direct-manipulation user interface modeling with high-level Petri nets. In: *Proceedings of the 19th annual conference on Computer Science*, ACM Press (1991) 487-495
- [15] H.W. Lin, L.K. Shih, Wen-Chih Chang; Chao-Hsun Yang; Chun-Chia Wang. "A Petri nets-based approach to modeling SCORM sequence", *Proceedings of 2002 IEEE International Conference Multimedia and Expo*, 2004.
- [16] ADL Technical Team. *Sharable Content Object Reference Model (SCORM) 2004, 3rd Edition*, Advanced Distributed Learning (ADL), November 16, 2006
- [17] T. K. Shih, W.-C. Chang and W.-C. Ko, "SCORM Sequence and Template Authoring System", *Proceeding of Resources Management Association Annual Conference*, New Orleans, Louisiana, USA, May 23-26, 2004.
- [18] NOA. <http://www.fisica.ufpb.br/~romero>. 30 Jun 2007.
- [19] C. M. Holloway. "Why Engineers Should Consider Formal Methods". 16th Digital Avionics Systems Conference. October 1997.
- [20] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Volume 1. EATCS monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1997.
- [21] Alberta Learning. <http://www.learnalberta.ca>, 26 Abr. 2006.
- [22] G. Booch, J. Rumbaugh, Ivar Jacobson. *The Unified Modeling Language User Guide (2nd Edition)*. Addison-Wesley Professional. 2005.