

SUMÁRIO

Lista de Figuras	2
1 Introdução	3
2 A ferramenta Amaya	4
3 SVG	5
4 Workflow Process Model	6
5 Workflow DTD	8
5.1. Objetivos e Estrutura	8
5.2. Elementos	9
5.2.1. Elemento wbegin	9
5.2.2. Elemento wconnector	9
5.2.3. Elemento wend	10
5.2.4. Elemento wtask	10
5.2.5. Elemento wstask	10
5.2.6. Elemento wmtask	11
5.2.7. Elemento wfjtotal	11
5.2.8. Elemento wfjparcial	11
5.2.9. Elemento wjinteractive	12
5.2.10. Elemento wfconditional	12
5.2.11. Elemento wfconditionale	12
5.3. Exemplo	13
5.4. Código da Workflow DTD	15
6 Implementação	18
7 Trabalhos Futuros	21
8 Conclusões	22
9 Referências Bibliográficas	23

LISTA DE FIGURAS

FIGURA 1 – Símbolos gráficos do modelo	7
FIGURA 2 – Workflow Construir Software	13
FIGURA 3 – Workflow Reunião	13
FIGURA 4 – Tela do Amaya Workflow	18
FIGURA 5 – Determinação de atributos no AW	19

1. Introdução

A importância do uso de tecnologias de workflow em sistemas complexos como a autoria colaborativa de cursos a distância via WWW destaca-se pela sua adaptabilidade[PAL01], para viabilizar o uso de tais tecnologias foram desenvolvidas soluções que utilizam a modelagem de workflow, dentre estas soluções destaca-se o modelo proposto por Casati/Ceri, o "*Workflow Process Model*"[WID97]. Através deste modelo pode-se expressar na forma de um esquema gráfico um fluxo de atividades, porém, a implementação destes modelos é uma das principais dificuldades encontradas no uso de tecnologias de workflow. Visando facilitar e tornar esta implementação mais ágil e precisa o projeto "*Conception of Cooperative Environment for Editing Multimedia Documents with Workflow Technology*" (CEMT) propôs a criação de um editor gráfico de workflow com base no software Amaya, o Amaya Workflow (AW).

O projeto CEMT é resultado da parceria entre a CNPq-UFRGS-Universidade Federal do Rio Grande do Sul (Brasil) e o INRIA-Institut National de Recherche en Informatique et en Automatique (França), esta parceria visa o desenvolvimento de um ambiente de edição cooperativa de documentos multimídia com tecnologia de workflow utilizando-se de padrões reconhecidos pela comunidade científica. Este projeto encontra-se em fase de implementação.

O Amaya é o Browser/Editor oficial do W3C, atualmente esta sendo desenvolvido no INRIA, este software possui entre outros recursos um suporte as tecnologias XML e SVG. O SVG é um padrão de representação de gráficos oficial do W3C, o Amaya possui um editor de gráficos que segue este padrão e tendo como base este editor o projeto CEMT implementou no Amaya o AW, que é um editor gráfico da técnica de modelagem de workflow "*Workflow Process Model*".

Com este editor um usuário pode definir graficamente um workflow segundo o modelo Casati/Ceri. Paralelamente a geração do gráfico SVG o AW gera um arquivo XML que contém todos os dados referentes ao workflow que esta sendo modelado, para tanto, o projeto CEMT precisou definir uma DTD de Workflow, os arquivos XML que seguem esta DTD são capazes de armazenar informações que podem recompor o fluxo do modelo e fornecer as informações necessárias para a sua execução.

O presente relatório descreve a implementação, o funcionamento, as tecnologias e padrões utilizados no editor de workflow. A seção 2 descreve o software no qual o editor foi adicionado como um módulo, o software Amaya, a seção 3 descreve o SVG, o padrão visual que foi utilizado no editor, a seção 4 descreve mostra o "*Workflow Process Model*" que é o sistema de modelagem de workflow implementado no editor a implementação e funcionamento deste módulo adicionado ao Amaya, a seção 5 descreve a DTD de Workflow criada pelo projeto CEMT para armazenar informações de um workflow. Após expor as tecnologias utilizadas o relatório descreve na seção 6 a implementação e funcionalidades do editor, por fim as seções 7 e 8 relatam os futuros trabalhos e conclusões obtidas nesta implementação.

2. A ferramenta Amaya

O Amaya[VAT01] é um browser e editor de documentos, desenvolvido pelo W3C que permite a publicação de documentos na Web, estes documentos podem ser do tipo texto, HTML, XHTML, folhas do estilo do CSS, expressões de MathML, e desenhos de SVG. O Amaya além de poder exibir documentos nestes formatos, pode também editá-los, em especial destaca-se o editor SVG que através de uma paleta de elementos gráficos permite que um usuário edite figuras geométricas que seguem o padrão SVG. Sua interface é do tipo "WYSIWYG"(a aparência do documento durante a edição é semelhante à aparência do documento final).

Seu funcionamento se baseia sobre a fragmentação do documento em diversas partes ou elementos, a fragmentação utilizada pelo Amaya é resultado do trabalho da API Thot. O Thot é utilizado para o desenvolvimento de aplicações que manipulam documentos estruturados. Ele fornece um conjunto de funções e linguagens de alto nível que manipulam a estrutura lógica e o conteúdo dos documentos.

Um documento no Thot, e conseqüentemente no Amaya e no AW é representado por sua estrutura lógica, essencialmente hierárquica, que se chama "árvore abstrata". Essa estrutura segue algumas regras, uma espécie de formato de documento, que ajuda (ou obriga) seus usuários a produzirem documentos daquele tipo. Para documentos html o Amaya segue as regras da DTD do HTML implementadas através das linguagens de alto nível do Thot, para documentos SVG o Amaya segue a DTD do SVG implementada e para documentos de Workflow o projeto CEMT implementou a sua DTD para guiar e formatar corretamente a estrutura de documentos deste tipo.

3. SVG

A especificação Scalable Vector Graphics (SVG) é um vocabulário XML que descreve gráficos bidimensionais, esta especificação tornou-se recentemente uma recomendação oficial do World Wide Web Consortium (W3C) sendo portanto um padrão desta instituição.

Diferentemente de formatos como JPEG e PNG que representam um gráfico descrevendo-o pixel a pixel o SVG representa um gráfico através de vetores, vetores gráficos representam figuras geométricas através de parâmetros e atributos. O exemplo abaixo mostra uma linha de um arquivo SVG que representa um retângulo preenchido internamente de vermelho, contornado de preto e com dimensões de 118 e 89 pixels.

```
<rect stroke="black" fill="red" y="62px" x="142px" width="118px" height="89px"/>
```

Para desenhar este retângulo um aplicativo que interpreta códigos SVG coloca estes dados em um vetor que é enviado a uma função que então o desenha.

Junto com suas figuras geométricas, a representação vetorial do SVG permite que figuras (JPEG por exemplo) e textos sejam adicionados a um gráfico, os textos adicionados podem ser tanto copiados e manipulados como figuras quanto editados como textos comuns.

Devido ao seu modo de representação, um gráfico SVG é escalonável, ou seja, ele permite que uma área em particular de um gráfico seja aproximada sem que haja uma degradação da imagem, tal característica também permite que um gráfico seja reproduzido em diferentes resoluções sem que haja uma perda de qualidade da imagem.

Outra característica importante do SVG é o fato deste ser um vocabulário XML compatível com a recomendação XML 1.0 da W3C, tendo portanto todas as características e vantagens de outros arquivos XML. Por também ser compatível com a recomendação "Namespaces in XML" do W3C, o vocabulário SVG pode ser usado em um mesmo arquivo com outros vocabulários XML tais como MathML, XHTML e HTML4. Os gráficos SVG podem ser estilizados tanto pela tecnologia Cascading Style Sheets (CSS) quanto pela Tecnologia XSL.

O SVG também é compatível ou pode ser usado com várias outras tecnologias que estão sendo desenvolvidas no W3C tais como SMIL, DOM, XLink e Xpointer. Esta compatibilidade lhe fornece outros recursos (animações, manipulação por strings entre outras) e o torna uma tecnologia em constante evolução.

4. Workflow Process Model

Este capítulo descreve o método de modelagem de Workflow definido no projeto Wide [WID01] por Casati/Ceri, o "*Workflow Process Model*", bem como uma apresentação sucinta dos elementos que o compõe e o caracterizam.

Workflow é “um conjunto coordenado de atividades (seqüenciais ou paralelas) que são interligadas com o objetivo de alcançar uma meta comum”, sendo atividade conceituada como “uma descrição de um fragmento de trabalho que contribui para o cumprimento de um processo”[WMC 96]. Assim, pode-se assumir que workflow é a divisão de um grande trabalho em várias tarefas menores, com pré-requisitos entre elas, que devem ser respeitados para o avanço da atividade.

O modelo proposto por Casati/Ceri no projeto Wide, *Workflow Process Model*, é considerado como um dos modelos mais eficientes para a modelagem de workflows[GLA]. Através deste modelo pode-se projetar o fluxo de execução de várias tarefas por vários agentes diferentes, chamados aqui de atores. Para isso se vale de gráficos, símbolos e textos, não apenas para descrever as tarefas envolvidas, mas também para especificar os mecanismos de disparo e término das ações previstas.

Para que a idéia proposta acima fosse possível, alguns elementos foram definidos pelo modelo. Tem-se assim como elementos principais o início e o fim das execuções, as tarefas em si, as conexões entre os elementos, modularização das tarefas, construções em paralelo, elementos de junção e elementos de separação. Adiante, trataremos com um pouco mais de detalhe cada um destes elementos.

A tarefa é a unidade fundamental de trabalho do workflow, é uma atividade, identificada por um nome que pode ser executada por um dos atores definidos.

As conexões entre as tarefas definem os possíveis caminhos a serem tomados a partir das tarefas que foram interligadas. Existem três tipos possíveis de conexões : causalidades, os forks e os joins. Na conexão tipo causalidade, o término de uma determinada tarefa habilita automaticamente a tarefa posterior.

Nas conexões definidas como *forks*, uma determinada tarefa não é seguida por apenas uma tarefa, mas sim por múltiplas tarefas sucessoras. Assim, existem diferentes tipos de *forks*, a saber : total, não- determinístico, condicional e condicional com exclusão mútua.

No *fork* tipo total, todas as atividades que seguem a tarefa são habilitadas com o seu término. No *fork* tipo não- determinístico, um número qualquer de atividades (mas não todas) são habilitadas a prosseguirem. No *fork* tipo condicional, após o término da atividade, somente as atividades sucessoras que cumprirem certos requisitos irão prosseguir. Finalmente, no *fork* condicional com exclusão mútua, as atividades sucessoras são avaliadas e somente uma delas é escolhida para ser habilitada.

Nas conexões definidas como *join*, uma determinada tarefa não é, como no item anterior, sucedida, mas sim precedida por múltiplas tarefas diferentes. Do mesmo modo, existem alguns tipos de *joins*, a saber : *join* total, *join* parcial e *join* interativo. O *join* total defini-se assim pois a tarefa em questão somente é habilitada assim que todas as tarefas predecessoras forem completadas. O *join* tipo parcial define que a tarefa pode ser executada assim que um certo número de tarefas antecessoras forem completadas (não todas). Finalmente, o *join* tipo interativo define que a tarefa será executada a cada

término de um número definido de tarefas. Assim, a tarefa pode ser executada várias vezes, cada vez que uma tarefa anterior for completada.

Existem marcas que delimitam o começo e o término de uma instância de workflow. Com o símbolo de *início*, habilita-se a atividade sucessora a este, e com a marca de *fim* termina-se a instância, interrompendo qualquer atividade que porventura ainda esteja em execução.

Consegue-se modularizar o workflow através do elemento *supertarefa*, que reúne várias tarefas num só elemento, relacionando-se com outros elementos através das conexões. Seria como uma *função* em uma linguagem de programação estruturada.

Por último, tem-se o elemento multitarefa, que reúne uma série de tarefas que podem ser executadas em paralelo. Deve-se definir quantas instâncias dos elementos serão geradas, bem como o número de atividades que devem ser concluídas para que a multitarefa como um todo esteja concluída.

Abaixo estão os símbolos gráficos que no "*Workflow Process Model*" representam os elementos citados acima.

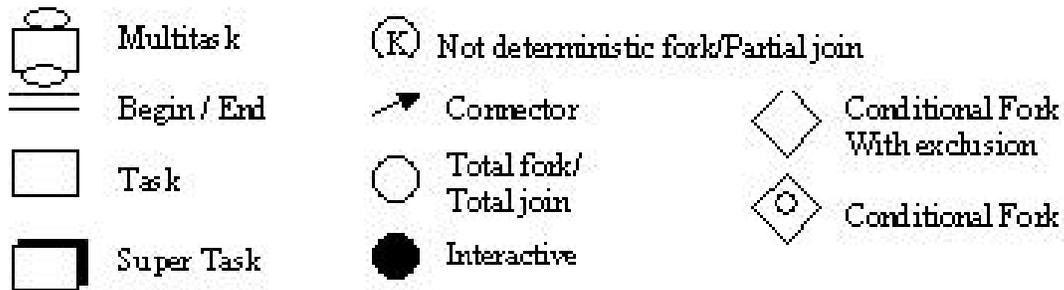


FIGURA 1 – Símbolos gráficos do modelo

5. Workflow DTD

5.1 Objetivos e Estrutura

Esta DTD foi criada para tornar-se uma especificação de um vocabulário XML capaz de descrever um Workflow que foi projetado com o sistema de modelagem "Workflow Process Model". Os arquivos XML gerados com base nesta DTD devem conter as informações de um Workflow que foi projetado com o sistema de modelagem "Workflow Process Model" de forma que uma máquina de Workflow possa ler este arquivo e posteriormente gerar o Workflow inicialmente projetado.

A Workflow DTD possui o elemento raiz "workflow" que é uma seqüência dos demais elementos sendo que todos estes elementos são irmãos. Cada um dos elementos filhos do elemento workflow corresponde a uma figura gráfica do "Workflow Process Model" que por sua vez representam um ou mais elementos lógicos do Workflow conforme mostrou a figura 1. Os elementos da DTD são os seguintes:

- o elemento wbegin corresponde ao símbolo que representa o begin.
- o elemento wend corresponde ao símbolo que representa o end.
- o elemento wtask corresponde ao símbolo que representa a task.
- o elemento wconnector corresponde ao símbolo que representa o connector.
- o elemento wmtask corresponde ao símbolo que representa o multi task.
- o elemento wstask corresponde ao símbolo que representa a super task.
- o elemento wfjtotal corresponde ao símbolo que representa o fork e o join total.
- o elemento wfjparcial corresponde ao símbolo que representa o join parcial e o fork não-determinístico
- o elemento wjinteractive corresponde ao símbolo que representa o join interativo.
- o elemento wfconditional corresponde ao símbolo que representa o fork condicional.
- o elemento wfconditionale corresponde ao símbolo que representa o fork condicional com exclusão mutua.

Cada elemento desta DTD possui atributos onde são armazenadas as informações do workflow que esta sendo projetado, como por exemplo os atributos fromID e PostID do elemento wconnector que informam respectivamente o elemento antecessor e o posterior ao conector na lógica do workflow.

Os elementos filhos do elemento workflow são distribuídos dentro deste em qualquer ordem, para ler e gerar um workflow que foi projetado uma máquina de workflow deve proceder da seguinte forma:

- localizar o elemento wbegin para se inicializar o workflow;
- localizar o conector cujo atributo formID aponta para o elemento wbegin;
- localizar o elemento que é apontado pelo atributo ToID do conector encontrado no passo anterior;
- se o próximo elemento for um wtask, com a ajuda dos atributos nele setados deve-se ler, interpretar e/ou executar a atividade definida. Se o elemento for uma stask ou um mtask deve-se procurar o arquivo apontado por

estes elementos, executar o workflow definido neste arquivo e voltar a execução do workflow original. Se o elemento for um *wfjtotal*, *wfjparcial*, *wjinteractive*, *wfconditional* ou *wfconditionale* deve-se com ajuda dos correspondentes atributos executar os forks ou joins possíveis(maiores detalhes são explicados adiante);

- localizar o conector que vem depois deste elemento e em seguida o elemento que vem depois do conector encontrado;
- Repetir os dois últimos passos até que o elemento *wend* seja encontrado. Ao ser encontrado, o elemento *wend* termina a execução do workflow.

5.2 Elementos

Esta seção descreve os elementos utilizados no editor de workflow que está sendo desenvolvido pelo Projeto CEMT. Os elementos *wbegin*, *wconnector* e *wend* devem necessariamente aparecer no desenvolvimento do projeto de workflow, para que este seja considerado válido. Os outros elementos devem aparecer conforme a necessidade de uso, seguindo as regras e definições que seguem abaixo.

Os atributos, caso ocorram em mais de um elemento, são explicados no primeiro elemento em que aparecerem, segundo a lista abaixo. O atributo ID aparece em todos os elementos, e é atribuído de forma implícita; serve para identificar o elemento na representação interna, utilizada pelo programa.

5.2.1 Elemento *wbegin*

O primeiro elemento que deve ser incluído na edição de um modelo, ele define o começo de uma instância do workflow.

```
<!ELEMENT wbegin EMPTY>
<!ATTLIST wbegin ID CDATA #REQUIRED
             wStatus CDATA #IMPLIED >
```

O atributo *wstatus* define o estado atual do elemento durante a execução do workflow.

5.2.2 Elemento *wconnector*

O elemento *wconnector* tem por função conectar dois outros elementos quaisquer na edição do workflow. Ele possui informações de identificação sobre ambos os elementos que são interligados por ele.

```
<!ELEMENT wconnector EMPTY>
<!ATTLIST wconnector ID CDATA #REQUIRED
             wStatus CDATA #IMPLIED
```

```
wAnswer CDATA #IMPLIED
wToID CDATA #REQUIRED
wFromID CDATA #REQUIRED >
```

O atributo *wanswer* é utilizado no conector quando este é definido na edição após um elemento do tipo *fork*, e contém a resposta para a pergunta do mesmo.

O atributo *wFromID* identifica o elemento anterior ao conector, e o atributo *wToID* identifica o elemento posterior ao conector.

5.2.3 Elemento wend

O elemento *wend* deve estar obrigatoriamente na edição do workflow, definindo o término da execução de uma instância de workflow.

```
<!ELEMENT wend EMPTY>
<!ATTLIST wend ID CDATA #REQUIRED
           wStatus CDATA #IMPLIED>
```

5.2.4 Elemento wtask

O elemento *wtask* define o elemento fundamental do workflow, a tarefa. Define o que realmente deve ser feito por aquela instância de workflow.

```
<!ELEMENT wtask EMPTY>
<!ATTLIST wtask ID CDATA #REQUIRED
           wName CDATA #REQUIRED
           wRole CDATA #IMPLIED
           wDescPreCondition CDATA #IMPLIED
           wDescPosCondition CDATA #IMPLIED
           wDescAction CDATA #IMPLIED
           wStatus CDATA #IMPLIED
           wType #IMPLIED >
```

O atributo *wName* é utilizado como identificador do elemento. O atributo *wRole* descreve quem é o ator que pode executar a tarefa. O atributo *wDescPreCondition* descreve quais são os pré-requisitos para que a tarefa possa ser desempenhada. O atributo *wDescPostCondition* descreve quando a tarefa é considerada completada. O atributo *wDescAction* é utilizado para descrever as ações que devem ser realizadas pela tarefa. O atributo *wType* define o tipo da tarefa.

5.2.5 Elemento wstask

O elemento *wstask* é utilizado para representar a supertarefa. A supertarefa é importante por modularizar a edição do workflow, evitando que o mesmo fique por demasiado complexo.

```
<!ELEMENT wstask EMPTY>
<!ATTLIST wstask ID CDATA #REQUIRED
  wName CDATA #REQUIRED
  wTaskLink CDATA #REQUIRED
  wRole CDATA #IMPLIED
  wDescPreCondition CDATA #IMPLIED
  wDescPosCondition CDATA #IMPLIED
  wDescAction CDATA #IMPLIED
  wStatus CDATA #IMPLIED
  wType #IMPLIED >
```

O atributo *wTaskLink* contém a URL para o arquivo que contém o workflow desta supertarefa.

5.2.6 Elemento *wmtask*

O elemento *wmtask* define, na edição do workflow, uma multitarefa, isto é, tarefas que podem ser executadas em paralelo.

```
<!ELEMENT wmtask EMPTY>
<!ATTLIST wmtask ID CDATA #REQUIRED
  wName CDATA #REQUIRED
  wTaskLink CDATA #REQUIRED
  wRole CDATA #IMPLIED
  wDescPreCondition CDATA #IMPLIED
  wDescPosCondition CDATA #IMPLIED
  wDescAction CDATA #IMPLIED
  wStatus CDATA #IMPLIED
  wType #IMPLIED >
```

5.2.7 Elemento *wfjtotal*

O elemento *wfjtotal* representa o *join* e o *fork* total. No *join* total, é habilitada a tarefa posterior a ele somente após todas as anteriores terem sido concluídas. No *fork* total, ele habilitará todas as próximas tarefas após a tarefa anterior.

```
<!ELEMENT wfjtotal EMPTY>
<!ATTLIST wfjtotal ID CDATA #REQUIRED
  wStatus CDATA #IMPLIED >
```

5.2.8 Elemento *wfjparcial*

O elemento *wfjparcial* representa o *join* parcial e o *fork* não-determinístico. Quando representa o *join*, permite que a tarefa posterior seja executada somente após que um certo número de tarefas antecessoras tenha sido completada. Quando representa o *fork*, ele habilita um certo número de tarefas para serem executadas.

```
<!ELEMENT wfjparcial EMPTY>
<!ATTLIST wfjparcial ID CDATA #REQUIRED
           wStatus CDATA #IMPLIED
           wKf CDATA #IMPLIED
           wKj CDATA #IMPLIED >
```

O atributo *wKf* representa quantos conectores saem do *fork*; já o atributo *wKj*, quantos conectores chegam até o *join*.

5.2.9 Elemento *wjinteractive*

O elemento *wjinteractive* define o *join* interativo, habilitando uma tarefa posterior após cada término de um certo número de tarefas anteriores.

```
<!ELEMENT wjinteractive EMPTY>
<!ATTLIST wjinteractive ID CDATA #REQUIRED
           wStatus CDATA #IMPLIED >
```

5.2.10 Elemento *wfconditional*

O elemento *wfconditional* define o *fork* condicional, que faz um teste sobre alguma condição, e somente as tarefas posteriores que atenderem à condição necessária serão habilitadas.

```
<!ELEMENT wfconditional EMPTY>
<!ATTLIST wfconditional ID CDATA #REQUIRED
           wAsk CDATA #REQUIRED
           wStatus CDATA #IMPLIED >
```

O atributo *wAsk* contém a pergunta que será feita as tarefas posteriores, a fim de executar a avaliação da condição.

5.2.11 Elemento *wfconditionale*

O elemento *wfconditionale* define o *fork* condicional com exclusão mútua, que avalia alguma condição e habilita somente uma das tarefas posteriores.

```
<!ELEMENT wconditionale EMPTY>
```

```
<!ATTLIST wfexclude ID CDATA #REQUIRED
          wAsk CDATA #REQUIRED
          wStatus CDATA #IMPLIED >
```

5.3 Exemplo

Nesta seção será apresentado um exemplo da construção de um workflow, que demonstra o modelo gráfico desenvolvido bem como o código XML gerado.

O exemplo aqui construído corresponderá ao seguinte estudo de caso:

De um modo extremamente simplificado, será demonstrado o workflow de como será construído um software. Primeiramente, o analista responsável pelo desenvolvimento do software reúne-se com o usuário do futuro sistema, nesta reunião alguns detalhes importantes no desenvolvimento do aplicativo são definidos. Após este passo, passa-se então para a fase de programação propriamente dita.

Um Workflow para estas atividades segundo o método WIDE corresponde as figuras 2 e 3.

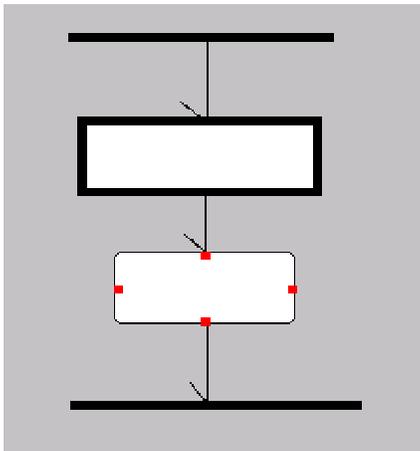


Figura 2 -Workflow Construir Software

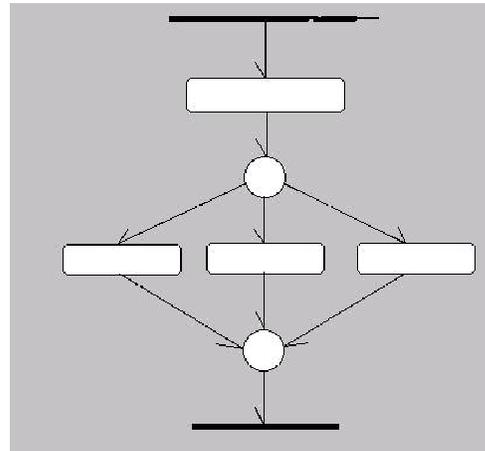


Figura 3 - Worflow Reunião

Os códigos de um arquivo XML que segue a DTD de Workflow que correspondem a estas figuras são os seguintes:

Código da Figura 2

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE workflow PUBLIC "-//UFRGS//DTD Workflow//EN"/>
<workflow xmlns="http://www.inf.ufrgs.br/">
  <wbegin id="element"/>
  <wconnector id="connector" wFromID="element"/>
  <wstask id="element1" xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:type="simple" xlink:href="reuniao.xml"/>
  <wconnector id="connector1" wFromID=""/>
  <wtask id="element2"/>
```

```

<wconnector id="connector2" wFromID=""/>
<wend id="element3"/>
</workflow>

```

Código da Figura 3

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE workflow PUBLIC "-//UFRGS//DTD Workflow//EN">
<workflow xmlns="http://www.inf.ufrgs.br/workflow/">
  <wbegin id="element"/>
    <wconnector id="connector" wFromID="element" wToID="element1"/>
    <wtask id="element1" wName="Reuniao com Usuário"
      wDescription="Deve-se marcar uma reuniao com o Usuário, para definir alguns
aspectos importantes do aplicativo"/>
    <wconnector id="connector1" wFromID="element1" wToID="element2"/>
    <wfjtotal id="element2"/>
    <wconnector id="connector2" wFromID="element2" wToID="element3"/>
    <wconnector id="connector3" wFromID="element2" wToID="element4"/>
    <wconnector id="connector4" wFromID="element2" wToID="element5"/>
    <wtask id="element3" wName="Definicao da Interface" wRole="Programador"
      wDescPreCondition="Reuniao com o Usuário realizada"
      wDescPostCondition="Definicao concluida"
      wDescription="O objetivo e definir a interface do aplicativo solicitado pelo
Usuário"
      wDescAction="Definir posicao das janelas, caixas de texto, etc"/>
    <wtask id="element4" wName="Definir Linguagem"/>
    <wtask id="element5" wName="Definir Requisitos"/>
    <wconnector id="connector5" wFromID="element3" wToID="element6"/>
    <wconnector id="connector6" wFromID="element4" wToID="element6"/>
    <wconnector id="connector7" wFromID="element5" wToID="element6"/>
    <wconnector id="connector8" wFromID="element6" wToID="element7"/>
  <wend id="element4"/>
</workflow>

```

A figura 2 representa o workflow principal, seguindo a lógica do workflow tem-se primeiramente o símbolo de início do workflow depois seguindo o fluxo determinado pelos conectores aparece a supertarefa "Reunião" que possui um atributo com um link para o arquivo que contém o workflow desta super tarefa. Depois tem-se a tarefa "Programação" e por último o símbolo que representa o fim do workflow.

A figura 3 representa a supertarefa "Reunião" referenciada no arquivo anterior. Todo este workflow faz parte do workflow principal porém por motivos de legibilidade ele apenas é referenciado na figura 2.

A primeira tarefa a ser realizada no workflow da figura 3 é marcar uma reunião com o usuário. Isto pode ser visto através da primeira tarefa (a finalidade de cada tarefa é averiguada pelo atributo wName de cada uma). Uma melhor explicação sobre a tarefa é encontrada no atributo wDescription, que traz uma pequena descrição sobre o que deve ser feito.

Após a reunião com o usuário, passa-se para a fase de definições. O elemento "wfjtotal" define que a reunião com o usuário habilita todas as tarefas consequentes. Cada uma das tarefas que seguem trata da definição de um item em particular. Quando todas as tarefas forem concluídas, o segundo "wfjtotal" permite que o fluxo do workflow continue.

Para não tornar o exemplo complexo, apenas a tarefa "Definição da Interface" teve seus atributos detalhados. Começando, temos o atributo "wName", explicado anteriormente. O atributo "wRole" define quem é o responsável por desempenhar aquela tarefa. O atributo "wDescPreCondition" detalha que condições devem ser satisfeitas para que aquela tarefa possa ser desempenhada. O atributo "wDescPostCondition" define o que pode acontecer depois que a tarefa foi concluída. O atributo "wDescription" foi explicado anteriormente. Finalmente, o atributo "wDescAction" explica que ações devem ser tomadas naquela tarefa.

Quando o Workflow "Reunião" termina deve-se voltar ao workflow principal e ir para o elemento seguinte, a super tarefa "Reunião".

5.4 Código da Workflow DTD

```
<!DOCTYPE workflow [
<!-- ===== XML ELEMENTS FROM Workflow DTD ===== -->
<!-- ===== -->
<!-- start and finish the connection -->

<!ELEMENT workflow
(wtask|wbegin|wend|wstask|wmtask|wconnector|wfjtotal|wfjpartial|wjinteractive|wfxclu
de|wfconditional)*>
<!ELEMENT wtask EMPTY>
<!ATTLIST wtask ID CDATA #REQUIRED
wName CDATA #REQUIRED
wRole CDATA #IMPLIED
wDescPreCondition CDATA #IMPLIED
wDescPosCondition CDATA #IMPLIED
wDescAction CDATA #IMPLIED
wStatus CDATA #IMPLIED
wType #IMPLIED >
```

```
<!ELEMENT wbegin EMPTY>
<!ATTLIST wbegin ID CDATA #REQUIRED
      wStatus CDATA #IMPLIED >
```

```
<!ELEMENT wend EMPTY>
<!ATTLIST wend ID CDATA #REQUIRED
      wStatus CDATA #IMPLIED>
```

```
<!ELEMENT wtask EMPTY>
<!ATTLIST wtask ID CDATA #REQUIRED
      wName CDATA #REQUIRED
      wTaskLink CDATA #REQUIRED
      wRole CDATA #IMPLIED
      wDescPreCondition CDATA #IMPLIED
      wDescPosCondition CDATA #IMPLIED
      wDescAction CDATA #IMPLIED
      wStatus CDATA #IMPLIED
      wType #IMPLIED >
```

```
<!ELEMENT wmtask EMPTY>
<!ATTLIST wmtask ID CDATA #REQUIRED
      wName CDATA #REQUIRED
      wTaskLink CDATA #REQUIRED
      wRole CDATA #IMPLIED
      wDescPreCondition CDATA #IMPLIED
      wDescPosCondition CDATA #IMPLIED
      wDescAction CDATA #IMPLIED
      wStatus CDATA #IMPLIED
      wType #IMPLIED >
```

```
<!ELEMENT wconnector EMPTY>
<!ATTLIST wconnector ID CDATA #REQUIRED
      wStatus CDATA #IMPLIED
      wAnswer CDATA #IMPLIED
      wToID CDATA #REQUIRED
      wFromID CDATA #REQUIRED >
```

```
<!ELEMENT wfjtotal EMPTY>
<!ATTLIST wfjtotal ID CDATA #REQUIRED
      wStatus CDATA #IMPLIED >
```

```
<!ELEMENT wfjpartial EMPTY>
<!ATTLIST wfjpartial ID CDATA #REQUIRED
      wStatus CDATA #IMPLIED
```

wKf CDATA #IMPLIED
wKj CDATA #IMPLIED >

<!ELEMENT wjinteractive EMPTY>
<!ATTLIST wjinteractive ID CDATA #REQUIRED
wStatus CDATA #IMPLIED >

<!ELEMENT wconditionale EMPTY>
<!ATTLIST wfexclude ID CDATA #REQUIRED
wAsk CDATA #REQUIRED
wStatus CDATA #IMPLIED >

<!ELEMENT wfconditional EMPTY>
<!ATTLIST wfconditional ID CDATA #REQUIRED
wAsk CDATA #REQUIRED
wStatus CDATA #IMPLIED >

]>

6. Implementação

O AW (Amaya Workflow) é um editor/browser que foi acrescentado de algumas funcionalidades que possibilitam a edição gráfica de documentos que expressam um workflow conforme a simbologia e semântica da técnica de modelagem *Workflow Process Model* do projeto WIDE.

O usuário poderá editar o documento utilizando uma ferramenta gráfica adicionada ao software Amaya, esta ferramenta gráfica consiste de uma paleta com símbolos da técnica de modelagem *Workflow Process Model* conforme a figura 4.

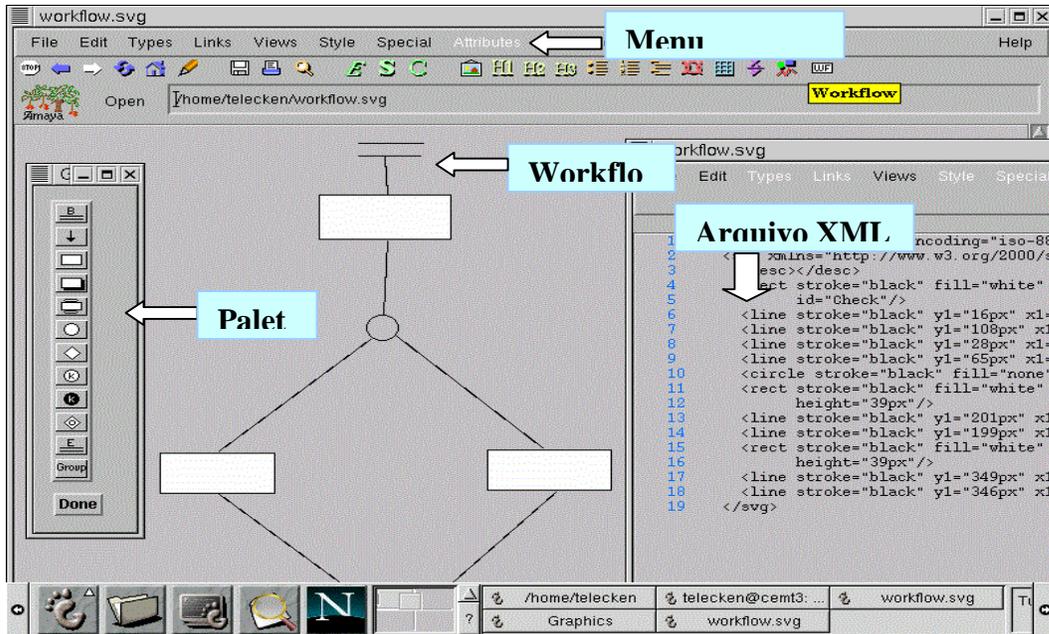


Figura 4- Tela do Amaya Workflow

O usuário utiliza-se da paleta de símbolos para adicioná-los ao documento, conforme os símbolos são adicionados o workflow vai sendo estruturado com suas tarefas, super tarefas, forks, joins e ligações, gerando desta maneira os arquivos que descrevem graficamente o fluxo lógico das atividades e no formato xml a estrutura lógica do workflow.

Os elementos são adicionados com tamanho fixo, porém se o usuário achar necessário ele pode mudar o tamanho e forma dos elementos. Os elementos também podem ser movidos pela tela mantendo as suas formas e proporções.

Cada elemento possui uma lista de atributos onde são armazenadas informações sobre o workflow que esta sendo modelado. Muitos atributos são valorados automaticamente pelo sistema conforme é feita a distribuição gráfica dos elementos como os atributos ToID e FromID do elemento connector que fazem a ligação entre os elementos determinando o fluxo e a distribuição destes elementos no workflow.

Já outros atributos, devem ser valorados pelos usuários, para preencher e editar estes atributos o usuário precisa selecionar o elemento em que precisa adicionar atributos e em seguida deve selecionar o item de menu "Atributo" que esta na parte superior da tela

do Amaya (Ver figura 5), ao selecionar este item, um menu será mostrado ao usuário com todos os atributos do elemento, para alterar algum atributo basta o usuário selecionar o atributo desejado e em seguida preencher ou alterar um formulário onde devem ser inseridos os dados referentes a estes atributos. Abaixo estão alguns atributos do elemento que simboliza o elemento task no "Workflow Process Model" e que precisam ser preenchidos pelo usuário:

Role: Papel do autor que poderá executar a tarefa.

Type: Tipo da atividade que sera executada.

Action: Atividade que será executada.

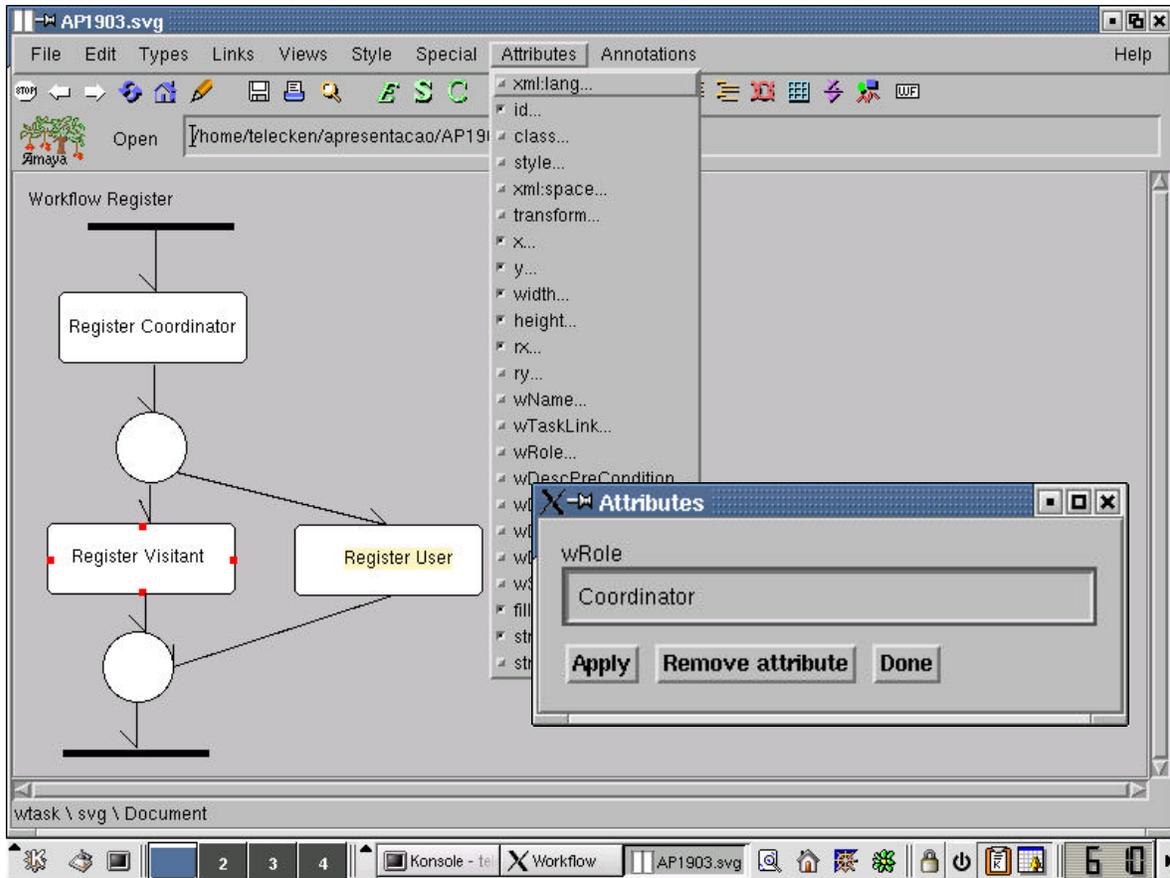


Figura 5 - Determinação de atributos no AW

Os elementos Multi-Task e Super-Task são links para documentos que contém o workflow destas tarefas, esta característica permite a modularização de um workflow, assim ao invés de escrever todo o workflow de uma determinada aplicação em um único documento, escreve-se um documento principal que poderá ter algumas multi tarefas ou super tarefas, uma super tarefa é representada por um único símbolo no workflow principal porém aponta, através de um link, para um documento onde esta o workflow desta super-tarefa. Recursivamente um documento apontado pelo documento principal pode ter supertarefas que apontam para outros documentos, portanto o workflow completo é formado por todos os documentos citados a partir do workflow principal. Ainda esta em fase de implementação a adição de um link nos documentos apontados por

super tarefas que levariam o usuário de volta para estas super tarefas facilitando a navegação do usuário através de todos os documentos de um workflow.

Para cada documento editado no AW são gerados três tipos de arquivos:

- Um arquivo que segue os padrões SVG que poderá mostrar o desenho gráfico do workflow em qualquer aplicativo que interprete o padrão SVG.
- Um arquivo XML que segue a DTD de Workflow relatada no capítulo 2 deste artigo, este arquivo terá somente informações relacionadas ao Workflow, nenhuma informação gráfica estará contida nele apesar destas informações serem retiradas da distribuição gráfica dos elementos e dos seus atributos. Como este arquivo possui todas as informações de um workflow, uma máquina de workflow poderá ler, interpretar e montar este workflow.
- Um arquivo que conterà tanto as informações sobre o workflow quanto informações sobre a aparência gráfica do documento, é através deste arquivo que o usuário poderá editar tanto os aspectos visuais do documento quanto as informações sobre o workflow modelado no documento. Os outros dois tipos de arquivos são gerados a partir deste, para o arquivo SVG não são mostradas as informações pertinentes a modelagem do workflow e para o arquivo que segue a DTD de workflow não são mostradas as informações relacionadas a aparência gráfica do documento.

Outra característica importante desta implementação e do projeto como um todo é a de usar exclusivamente ferramentas e padrões de códigos abertos e de livre distribuição, o que torna os sistemas gerados no projeto também sistemas de código aberto, livre distribuição, independentes de licenças de uso e de baixo custo. O Amaya roda e foi desenvolvido no sistema operacional Linux, este software e conseqüentemente o AW foram escritos na linguagem de programação C utilizam a API do THOT também escrita em C.

7. Trabalhos Futuros

O AW é apenas uma parte do projeto CEMT, e o seu uso, avaliação e adaptação ao restante do projeto são alguns dos futuros trabalhos relevantes relacionados a este editor. Resumidamente o projeto CEMT prevê que: Um especialista em workflow deverá montar no AW um workflow para a autoria de um curso a distância, os arquivos XML gerados pelo editor AW serão então enviados a uma máquina de workflow que interpretará os arquivos e com base neles construirá, executará e coordenará um ambiente cooperativo para a autoria de um curso na Web, sendo que o conteúdo deste curso poderá conter recursos de multimídia e hipermídia tais como SMIL. Estando pronto o ambiente para a construção de um curso basta os professores que irão fornecer o conteúdo do curso seguirem o workflow planejado para que seja gerado pelo ambiente um curso a distância. Neste contexto o AW serviria como uma ferramenta que de maneira ágil iria gerar um arquivo XML com instruções para uma máquina de workflow, para tanto o AW ainda precisa ser corretamente adaptado ao CEMT.

8. Conclusão

Além de sua essencial importância ao projeto CEMT pode-se citar como principais resultados obtidos com o desenvolvimento do AW a determinação de uma DTD de Workflow e a criação de um editor gráfico de Workflow de código aberto.

A DTD de Workflow é um vocabulário XML cujos arquivos XML são capazes de armazenar informações sobre qualquer workflow construído segundo o sistema de modelagem proposto por Casati/Ceri. Além de servir aos propósitos deste projeto, armazenando informações sobre um workflow que será interpretado e gerado por uma máquina de Workflow, esta DTD pode ser utilizada por qualquer outro aplicativo que necessite de uma DTD com tais características visto que esta foi feita de modo que representasse um workflow da maneira mais genérica possível não ficando presa a especificidades da máquina de workflow do projeto CEMT.

Embora algumas facilidades gráficas ainda não estejam implementadas o editor gráfico AW mostrou-se como uma maneira ágil e eficiente de se construir e visualizar workflows e também de gerar automaticamente arquivos SVG com os dados gráficos do workflow e XML com informações sobre a lógica do workflow. O AW também foi feito de maneira que funcionasse independentemente do seu uso no projeto CEMT, tal característica o habilita a ser usado por qualquer outra aplicação que necessite de seus recursos.

Dentro do projeto CEMT o AW será utilizado como uma ferramenta que agilizará a geração de workflows e a transformação dos mesmos em uma representação legível a uma máquina de workflow, o que será de fundamental importância para o projeto CEMT atingir o seu objetivo de facilitar a geração de conteúdos e apresentações de cursos a distância.

9. Bibliografia

[CAS 95] CASATI, F. et al. Conceptual Modeling of Workflows. In: The Object-Oriented and Entity-Relationship Conference, 1995, Gold Coast/Austrália. Proceedings...Austrália: LNCS,1995.

[CAS 96] CASATI,F.,PERNICI,B. Amethodology for the Design of WWW Sites and its Application to Distance Education. Proceedings of SEBD`96,Pisa, Italy, 1996

[PAL 01] PALAZZO, J ; KUNDE, G; SOUTO, M. Evolução Dinâmica de um Curso a Distância Modelado por Workflow. Revista Brasileira de Informática na Educação, nr. 9, Setembro/2001. ISSN 1414-5685.

[SIZ 00] SIZILIO, G.R.M.A.; EDELWEISS, N. Workflow for the Authoring of Long Distance Learning Courses. In:ICECE`2000 - International Conference on Engineering and Computer Education. 2000, São Paulo, Brasil. Proceedings... Aug. 27-30, 2000.

[SVG 01] . Scalable Vector Graphics (SVG) 1.0 Specification. Avaliable in <<http://www.w3.org/TR/2001/REC-SVG-20010904/>> . Access: Dez, 2001

[VAT 01] VATTON, Irène. Welcome to Amaya. Available in <<http://www.w3.org/Amaya/>> . Access: Dez, 2001.

[WID 97] The WIDE Workflow Modeling Language. Available at: <<http://dis.sema.es/projects/WIDE/Documents/4080-2.pdf>>. Access: Dez, 2001

[XML 01] Extensible Markup Language (XML) 1.0 (Second Edition). Avaliable in <<http://www.w3.org/TR/REC-xml>>. Access: Dez, 2001.