

Anima: Promovendo Integração de Componentes na Web

André Santanchè¹, Cesar Augusto Camillo Teixeira²

¹Núcleo de Pesquisa em Redes de Computadores / Núcleo de Pesquisa e Projetos em Educação a Distância – Universidade Salvador (UNIFACS)
Av. Cardeal da Silva, 747, Federação – 40220-141 – Salvador – BA.- Brasil

²Núcleo de Pesquisa em Redes de Computadores – Universidade Salvador (UNIFACS)
Av. Cardeal da Silva, 747, Federação – 40220-141 – Salvador – BA.- Brasil

{santanche,cesar}@unifacs.br

Abstract. *The combination of modern mobile code languages, especially object-oriented languages, with new open and extensible structured markup standards – such as XML and its derivations – have issued in a reformulation in the way applications are structured for the Internet.*

This work presents how emergent technologies related to the Web are combined to structure the Anima Project. The model of such project allows the integration of objects from different languages and implementations, especially those of mobile code, on an application. As a result, we can get compositions that explore the best of each world and combine them in unique and diversified productions, which are able to cross the Web independently, promoting, in this way, a broad integration and collaboration.

Resumo. *A combinação de modernas linguagens de código móvel, especialmente aquelas orientadas a objetos, com novos padrões de marcação estruturada, abertos e extensíveis – tal como XML e suas derivações – tem resultado numa reformulação na forma como aplicações são estruturadas para a Internet.*

Este trabalho demonstra como tecnologias emergentes na Web foram combinadas para estruturar o Projeto Anima, cujo modelo é capaz de integrar objetos de diferentes linguagens e implementações, principalmente aquelas de código móvel, em uma aplicação. Como resultado, obtemos composições que exploram os melhores recursos de cada sistema e os combinam em produções únicas e diversificadas, capazes de trafegar pela Web de forma independente, promovendo, desta maneira, uma ampla integração e colaboração.

1. Introdução

Objetos têm representado nos últimos anos um paradigma que cada vez mais amplifica sua atuação nos domínios da computação, não apenas em termos de adeptos aos seus princípios, como também no que diz respeito às múltiplas formas que assumem nas diversas áreas, encontrando, para cada uma, novas configurações e benefícios.

Como resultado, obtemos uma extensa malha de linguagens, metodologias, modelos, sistemas, etc. que se identificam sob o rótulo de objetos.

Por outro lado, o intercâmbio e integração dinâmica dos dados de máquinas distintas incentivaram a construção de objetos que se intercomunicam e trabalham cooperativamente através da rede. No entanto, o encontro entre a Internet e os objetos

processa-se gradativamente. Diversas linguagens e padrões para a integração de objetos, tal como CORBA, têm buscado caminhos que os tornam mais compatíveis e, conseqüentemente, mais integrados com a Internet, particularmente com a *Web*, que é sua principal aplicação.

Dentre as principais mudanças está a adaptação do formato utilizado para representação e transmissão dos dados, de modo que se utilizem aqueles padrões que têm se estabelecido na *Web*, tal como o XML [W3C98] e suas derivações (aplicações XML).

O projeto *Anima* combina estas duas fortes tendências - modelo de objetos para *Web* e linguagens abertas de marcação estruturadas extensíveis - com a codificação de objetos em linguagens de código móvel.

Ele integra técnicas e ferramentas de *software* e um esforço na concepção de modelos, que possibilitam a elaboração de projetos baseados em objetos, dentro de uma perspectiva de intercâmbio e colaboração. Para isto, *Anima* apresenta propostas para diminuir as barreiras entre diferentes padrões e linguagens, e também para o intercâmbio de produções através da *Web*.

Nos dois primeiros tópicos, a seguir, são analisados os fundamentos tecnológicos sobre os quais está estruturado *Anima*.

No tópico **Tecnologias para Web** apresentamos uma descrição básica de três linguagens, XML, RDF e XSL, que estruturam os dados sobre os quais está montado o projeto *Anima*. A partir da concepção aberta, genérica e extensível destas linguagens, tornou-se possível armazenar, transmitir, reconstruir e reaproveitar modelos e sistemas moldados sobre os objetos, independente da plataforma ou linguagem em que eles estejam implementados.

Em **Código Móvel e Objetos na Web** é analisada a influência das linguagens de código móvel na constituição de objetos que trafegam pela rede, além de uma abordagem introdutória a modelos de objetos adequados a *Web*. A partir desta revisão, é possível estabelecer os princípios que definem as principais características do projeto *Anima*.

Finalmente em *Anima* é apresentado o projeto, sua estrutura, implementação e experiências realizadas. São analisados os pontos positivos e propostas para o futuro.

2. Tecnologias para Web

A linguagem e metalinguagem XML – *eXtensible Markup Language* [W3C98], pela sua característica aberta e extensível, tem dado origem a um universo de linguagens derivadas com os mais diversos propósitos. Dentre elas, iremos analisar neste trabalho, além de XML, as linguagens derivadas RDF e XSL. Como veremos mais adiante, juntas estas três linguagens constituem a base de representação na qual o Projeto *Anima* está fundamentado.

2.1. XML

XML é uma linguagem de marcação que possui raízes em SGML (*Standard Generalized Markup Language*). Ela tem a capacidade de, através do uso de marcadores, agregar informações adicionais a documentos com os mais diversos propósitos. Neste caso os marcadores são caracterizados pelos símbolos “<” e “>”. Deste modo a expressão

```
<indivíduo> <nome>Quincas</nome> tem <idade>15</idade> anos.  
</indivíduo>
```

permite diferenciar o conteúdo da frase “Quincas tem 15 anos.” das marcações (<indivíduo>, <nome> e <idade>) que agregam informações adicionais à frase. Estas informações são facilmente extraídas e isoladas por programas de computador.

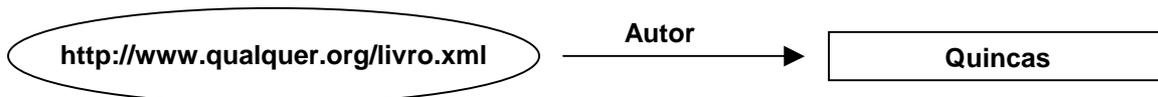
Além de linguagem, XML comporta-se também como meta linguagem. Isto significa que possui recursos, por exemplo, para que uma comunidade defina seus próprios marcadores. Através deles, podem definir ou estender linguagens, tornando-as mais apropriadas à representação dos documentos que manipulam.

2.2. RDF

RDF – *Resource Description Framework* – é uma linguagem baseada em XML destinada à codificação, troca e reutilização de metadados estruturados.

A idéia básica de RDF [LAS99] não é definir um conjunto universal de metadados e sim prover os mecanismos necessários para que as diversas comunidades codifiquem, troquem e reutilizem metadados estruturados.

Como pode ser observado no diagrama abaixo, RDF utiliza, para descrever um recurso, uma instrução composta de três partes: identificação do recurso, tipo de propriedade e valor da propriedade.



Neste caso, o **Autor** (tipo de propriedade) do livro definido no documento <http://www.qualquer.org/livro.xml> (recurso) é **Quincas** (valor).

2.3. XSL

O mecanismo de funcionamento da linguagem XSL – *eXtensible Stylesheet Language* [ADL00] se baseia na concepção de um documento XML como uma árvore. XSL cumpre dois papéis: transformar uma árvore de um documento XML em outra árvore (XSLT), e definir uma sintaxe para a árvore resultante, apropriada para a definição da formatação e apresentação do documento (*XSL Formatting Object*).

Para orientar a transformação, um documento XSLT estabelece a forma como subárvores do documento XML de origem devem ser transformadas em subárvores do documento resultante. Processadores XSL são capazes de realizar a conversão. É comum a transformação de um documento XML (ou documento escrito em linguagem descendente da metalinguagem XML) em um documento HTML.

3. Código Móvel e Objetos na Web

Segundo Alfonso Fuggeta, “*Mobilidade de código pode ser definida como a capacidade de modificar dinamicamente as ligações entre fragmentos de código e o local onde eles são executados*” [FUG98]. Esta capacidade permite que programas possam trafegar livremente pela Internet, apesar de sua grande diversidade de plataformas e sistemas.

A crescente possibilidade de se distribuir programas, além das mídias habituais da *Web*, conduz a uma nova ótica no desenvolvimento de aplicações para a rede. Uma

significativa parcela destas MCL – tal como Java – adota o modelo de objetos e distribui seu código sob a forma de “objetos móveis”.

Esta rota convergente entre objetos e *Web* tem exigido um padrão mais integrado.

Frank Manola adota o termo *Object Model*ⁱ em um estudo referente a modelos de objetos adequados para *Web* (*Web Object Model*) [MAN99]. Ele destaca três aspectos fundamentais para qualquer modelo de objetos: estruturas de dados que representam o *estado do objeto*, meios para associar comportamento (métodos do objeto) ao estado do objeto e meios para os métodos do objeto acessarem e operarem as variáveis que representam o seu estado.

Como será apresentado adiante, o projeto *Anima* desenvolve um modelo de objetos móvel, adequado para *Web* e independente de plataforma. Para alcançar estes objetivos, o modelo adota as linguagens padronizadas XML e RDF, e contempla os três aspectos mencionados por Manola [MAN99]. Sua estrutura inclui o uso de MCL na constituição de objetos móveis, capazes de se deslocarem através da rede e de se integrarem dinamicamente.

4. Anima

O Projeto Anima foi impulsionado por um conjunto de necessidades constatadas durante a elaboração de ferramentas para o desenvolvimento de atividades de ensino-aprendizagem fazendo uso do computador, em especial o Sistema Casa Mágica [SAN00].

O sistema Casa Mágica constitui-se em um ambiente para construção de aplicações educacionais que combina os recursos de uma ferramenta de autoria com recursos que permitem a construção e exploração de modelos de estudo.

Este sistema organiza suas composições sob a forma de espaços de trabalho (Locais), nos quais aprendizes ou educadores dispõem de objetos, principalmente componentes, para a construção de projetos. Os objetos são interligados através da troca de mensagens e o conjunto final configura-se em uma rede intercomunicante.

Toda a produção é codificada em linguagem própria, compilada em código intermediário (móvel) e pode ser executada em qualquer máquina que suporte Java, pois o sistema Casa Mágica possui um módulo de run-time escrito nesta linguagem.

Tal modelo incentiva uma participação ativa dos alunos na construção dos projetos e permite o intercâmbio de produções (e a conseqüente colaboração entre eles), seja sob a forma de componentes individuais ou composições completas.

Neste contexto, *Anima* se insere no sistema Casa Mágica, fornecendo-lhe recursos para intercâmbio de produções através da *Web* e a possibilidade de enriquecer sua biblioteca, com objetos provenientes de outros domínios ou produzidos em outras ferramentas e linguagens.

Apesar de surgir a partir do sistema Casa Mágica, o propósito do projeto Anima é justamente ampliar os horizontes para outros sistemas e linguagens. Como será apresentado a seguir, isto foi feito através da aplicação de um conjunto de tecnologias emergentes, tal como XML, XSL e RDF.

4.1. Estrutura do Projeto

O projeto *Anima* abrange:

- **Modelo genérico**, independente de plataforma e linguagem para a representação de objetos e composições, de modo que possam ser distribuídos através da *Web*. Ele se subdivide em:
 - **Representação da composição e objetos** que a constituem.
 - **Representação de classe, interface e metadados** associados ao objeto.
- **Mecanismos de conversão** padronizados que realizam a transposição dos objetos e composições da representação genérica para implementações específicas.
- **Protocolo padronizado de comunicação** entre componentes dentro da composição.

As características do modelo e dos mecanismos implementados são genéricas o suficiente para serem aplicadas em outros contextos, além do educacional. Por este motivo, eles podem servir de contribuição para um estudo mais abrangente em áreas que se aplicam à orientação a objetos, tal como uma representação genérica de objetos na *Web*.

4.2. Representação da composição e objetos

O Modelo Genérico para Representação da Composição do *Anima* foi concebido de modo a poder ser utilizado por diversos sistemas baseados em objetos, independente de plataforma ou linguagem.

Este modelo é representado em XMLⁱⁱ e abrange os seguintes aspectos: caracterização do espaço de trabalho onde será montada a composição, serialização dos objetos da composição e interligação entre objetos que realizarão troca de mensagens.

No nível hierárquico mais alto do modelo, está o espaço de trabalho (*workspace*), que estabelece uma área física dentro da qual os objetos irão atuar. Dentro do espaço de trabalho maior podem ser colocados espaços de trabalho menores, conforme a necessidade de divisão física. Isto será especialmente importante quando a composição for mista e não for possível o compartilhamento da mesma área física por dois objetos com diferentes implementações.

O espaço de trabalho de mais alto nível determina quais os ambientes em que se pode executar a composição (ex.: navegador *Web*, aplicação Java, etc.), e qual deles é o preferencial. Cada espaço subordinado sempre tem que ser compatível com o espaço superior.

A serialização dos objetos só necessita trabalhar alguns dados específicos de sua instância, tal como, seus valores de estado, classe a que pertence e, no caso de *Anima*, o seu identificador dentro da composição. A conexão entre objetos é representada por três informações: evento de origem, mensagem enviada e destino.

4.3. Representação da classe, interface e metadados (CIM)

Enquanto a representação anterior diz respeito aos aspectos dinâmicos de objetos em uma composição, existem outros, tal como informações sobre a classe a que o objeto pertence, sua interface e metadados a ele agregados, que são estáticos e, portanto, são os mesmos para um conjunto de objetos que tem estas características em comum.

Frank Manola [MAN98] sugere que a ligação entre uma classe em um objeto pode ser tomada como uma ligação entre o metadado e o dado, por este motivo, a representação CIM é definida como um conjunto de metadados dos objetos serializados na

representação da composição. O padrão RDF foi escolhido para tal tarefa, dada sua especialidade neste tipo de descriçãoⁱⁱⁱ.

A Representação CIM abrange os seguintes aspectos: descrição de características da classe relevantes para o projeto de composições (identificação, herança, atributos, interface, etc.), declaração dos tipos de mensagens que as classes podem receber ou enviar e metadados segundo o padrão LOM - *Learning Object Metadata* [HOD01].

Utilizando a própria estrutura de descrição de recursos RDF, é possível se estabelecer uma conexão das descrições de CIM e as implementações das classes codificadas em linguagens específicas, tal como o Java, que não ficam armazenadas no mesmo arquivo.

Para a elaboração da representação CIM foi implementada uma ferramenta denominada *Anima Workshop*, que permite a modelagem visual das características de uma classe e dos aspectos de interface, bem como a descrição de metadados educacionais. Além disto, a ferramenta possui recursos que permitem a extração automática dos dados de uma classe da linguagem Java (propriedades, métodos, eventos a que responde, etc.), através de um mecanismo denominado Introspecção (também disponível em outras linguagens).

A Representação CIM se torna, segundo este modelo, independente de linguagem e plataforma. A partir dela, outra ferramenta de produção é utilizada na construção da composição. A ferramenta pode ser operada em qualquer ambiente de produção, em qualquer linguagem ou plataforma capacitada a ler um arquivo RDF.

Durante o processo de produção da composição, no momento em que o usuário seleciona um determinado componente, a ferramenta decide se está habilitada a instanciar um objeto na linguagem em que a respectiva classe foi escrita. Isto depende da compatibilidade entre a plataforma da ferramenta e a da classe. Ambas podem estar na mesma plataforma, em plataformas compatíveis ou em plataformas que possuam interfaces que lhes permita trabalhar em conjunto.

Se for possível gerar uma instância do objeto, o usuário poderá, durante o processo de produção, manipular diretamente o componente, configurá-lo e experimentar seu comportamento. Caso contrário, a ferramenta ainda tem acesso, a partir da representação CIM, a um conjunto suficiente de informações sobre a interface e apresentação do objeto, que possibilitam sua inserção e integração na composição, mesmo que ele não possa ser experimentado interativamente.

O resultado da produção é serializado em XML, conforme descrito no tópico 4.2.

Anima dispõe de uma ferramenta de modelagem de composições implantada dentro do sistema Casa Mágica.

4.4. Mecanismos de conversão

Os mecanismos de conversão são a ponte entre o modelo representado genericamente e sua versão equivalente, implementada em uma linguagem (ou conjunto de linguagens), pronta para ser executada no computador.

Para realizar esta conversão, o projeto *Anima* faz uso de ferramentas, também independentes de plataforma, mantendo a coerência da estrutura como um todo. Ao invés de se utilizar programas de conversão em linguagens de programação, a linguagem XSL foi adotada nesta tarefa.

As folhas XSL são associadas aos arquivos de composição em XML, e realizam sua conversão para um novo arquivo resultante. Cada linguagem destino, para conversão, possui uma folha XSL diferente. Atualmente já foram implementadas folhas para Casa Mágica e Java.

De acordo com a necessidade, após a conversão pode ser disparado algum programa adicional que realiza a compilação de código. Isto ocorre quando a composição destino está em uma única linguagem e o produto resultante for código fonte na mesma.

Composições mistas, por outro lado, são projetadas para serem executadas em algum ambiente que suporte a instanciação e apresentação de todos os objetos nela envolvidos, por mais diversas que sejam as linguagens em que eles estão implementados.

O ambiente que mostra maior poder de integração, utilizado neste projeto, é o navegador *Web*, no qual pode-se executar programas de mais de uma linguagem simultaneamente, utilizando-se máquinas virtuais ou *plugins*. Existem, no entanto, restrições às áreas de apresentação que não podem se sobrepor – isto é administrado através dos *workspaces*.

4.5. Protocolo padronizado de comunicação

Realizar a comunicação entre os objetos, foi o desafio final deste projeto.

O problema central consiste em promover a comunicação entre objetos que possuem formatos diferentes de comunicação de dados, seja porque estejam codificados em linguagens diferentes, ou porque, ainda que estejam na mesma linguagem, trabalhem com formatos incompatíveis.

Por este motivo, a linguagem XML foi escolhida, por resolver questões de compatibilidade entre diferentes formatos de dados, permitir a representação de estruturas complexas, como objetos, e também por manter coerência com o restante do modelo.

A comunicação é feita, deste modo, através de mensagens XML, conforme os exemplos:

<MS id="ativa"/>	Mensagem simples, sem parâmetros.
<MP id="NumeroLados">5</MP>	Mensagem que aciona a modificação de uma propriedade.
<MO id="mensagem" class="classe"> {objeto}</MO>	Mensagem acompanhada de um único objeto.

Cada aplicação *Anima* exige um módulo que coordena o envio e recepção de mensagens entre os objetos. Por trabalhar com mensagens em formato texto, este módulo se adapta facilmente a qualquer mecanismo de comunicação já existente.

Quando os objetos são da mesma linguagem, utilizam-se mecanismos nativos da mesma, caso contrário é utilizado um canal TCP/IP.

5. Considerações Finais

Experiências práticas utilizando *Anima*, desenvolvidas com o sistema Casa Mágica, demonstraram a viabilidade do modelo, além de alguns benefícios da integração entre Casa Mágica e Java. Ficou claro, no entanto, que grande parte destes benefícios tornar-se-ão mais significativos no momento em que tivermos uma variedade maior de sistemas integrados.

A estrutura inicial apresentada neste texto presume o desenvolvimento de subsídios para que outras linguagens e sistemas possam se integrar a *Anima*, além de Java e Casa Mágica (já implementados). Entre as plataformas e linguagens identificadas como adequadas este propósito, destacamos Alice, Squeak, Flash e Javascript.

Algumas questões ainda estão em aberto para melhores soluções. Entre elas, percebemos a necessidade de uma caracterização mais refinada de categorias compatíveis de mensagens entre emissor e receptor, e a capacidade de permitir que objetos de diferentes linguagens compartilhem a mesma área de trabalho.

Apesar de possuímos um modelo que abrange o ciclo completo, partindo da modelagem, desenvolvimento e a execução de composições, não é propósito deste trabalho apresentar uma solução definitiva. Pelo contrário, sua estrutura absolutamente aberta, pautada em padrões da *Web* que além de abertos são extensíveis, viabiliza sua expansão e refinamento.

Referências Bibliográficas

Adler, Sharon et al. “Extensible Stylesheet Language (XSL) Version 1.0”, W3C Candidate Recommendation 21 November 2000, <http://www.w3.org/TR/xsl/>.

Fuggeta, Alfonso & Picco, Gian Pietro & Vigna, Giovanni. “Understanding Code Mobility. IEEE Transactions on Software Engineering”, vol. 24, 1998. <http://www.elet.polimi.it/Users/DEI/Sections/Compeng/GianPietro.Picco/papers/tse98.ps.gz>

Hodgins, Wayne (Chair). “Draft Standard for Learning Object Metadata”. IEEE Learning Technology Standards Committee (LTSC), February 8, 2001, http://ltsc.ieee.org/wg12/LOM_WD6_without_tracking.pdf.

Lassila, O., Swick, R. R. “Resource Description Framework (RDF) Model and Syntax Specification - W3C Recommendation 22 February 1999”, W3C - World Wide Web Consortium, <http://www.w3.org/TR/REC-rdf-syntax/>.

Manola, Frank. “Towards a Web Object Model”. Object Services and Consulting, Inc. (OBJS), 10 February 1998, <http://www.objs.com/OSA/wom.htm>.

Manola, Frank. “Technologies for a Web Object Model”. IEEE Internet Computing, January - February 1999.

Santanchè, André; Teixeira, Cesar Augusto Camillo. “Construindo e explorando o conhecimento através de Componentes Educacionais embutidos em hiperdocumentos”. VI Workshop de Informática na Escola, 17 de julho de 2000, <http://www.brasil.terravista.pt/claridade/1622/publicado/WIE2000.pdf>.

W3C XML Working Group. “Extensible Markup Language (XML) 1.0 - W3C Recommendation 10-February-1998”. W3C - World Wide Web Consortium, <http://www.w3.org/TR/1998/REC-xml-19980210>.

ⁱ *Object Model* – termo adotado pelo Comitê H7 (*Object Information Management*) - <http://www.objs.com/x3h7/h7home.htm>.

ⁱⁱ DTD do modelo disponível no endereço: <http://www.geocities.com/santanche/pesquisa/anima-framework.dtd>

ⁱⁱⁱ *Schema* RDF disponível no endereço: <http://www.geocities.com/santanche/pesquisa/anima-cim-schema.rdf>