

SISTEMA PARA CONSTRUÇÃO DE APLICAÇÕES EDUCACIONAIS

André Santanchè
UNIFACS

Rua Roque José Silva, 50 casa 4 Lot. Miragem, Lauro de Freitas - BA 42700-000

Resumo

O desenvolvimento de software para aplicações educacionais carece de ferramentas que sejam concebidas especificamente para tal processo, e que sejam voltadas não só para profissionais da área de informática, bem como para professores, educadores e alunos.

O presente trabalho aborda o processo de desenvolvimento de uma ferramenta para a construção de aplicações na educação. Tal ferramenta se propõe a reunir os recursos de uma ferramenta de produção gráfica e interativa (com características de um software de autoria) e uma linguagem de programação simples e poderosa, utilizando o novo paradigma da orientação ao objeto.

Introdução

O projeto “Casa Mágica” surgiu da necessidade concreta, defrontada em escolas de primeiro e segundo graus, e depois também em faculdades e universidades, de encontrar respostas a crescente necessidade de programas que pudessem ser utilizados no ambiente educacional.

No contexto do papel das novas tecnologias dentro das escolas, muitas têm sido as propostas, principalmente no que tange o uso do computador e, atualmente, da telemática, seja como ferramenta ou como nova perspectiva educacional. As propostas se alternam entre o computador como uma ferramenta CAI - Computer Aided Instruction - passando pela perspectiva construtivista onde o computador assume um novo papel que vai além de um simples transmissor de conhecimentos, como afirma Seymour Papert [PAP88]: "O computador não é usado somente como instrumento, mas essencialmente de maneira conceitual."

A explosão de tantas novas tecnologias, em especial o computador e a telemática, que passaram hoje, muito mais do que em outros tempos, a estar presentes dentro das residências, e a sua capacidade de transformar-nos cada vez mais numa “aldeia global”, tornou-se um agente que “modela progressivamente um outro comportamento intelectual e afetivo” [BAB89].

Muitas escolas passam então a utilizar recursos computacionais e de Internet para levar aos alunos uma nova perspectiva de comunicação, tornando-os capazes de utilizar os modernos recursos de multimídia, através de ferramentas de autoria, e das novas linguagens da Internet como novo meio de captação e expressão de conhecimentos e idéias. Algumas barreiras ainda devem ser transpostas neste panorama.

Dentro deste contexto o professor se encontra muitas vezes perdido. Ainda existem poucos e insípidos programas educacionais produzidos, e muitos deles repetem o antigo processo de transportar para o computador o quadro negro. Estes programas são, em geral, bastante fechados, impossibilitando que o professor o adeque a sua realidade ou até mesmo repense o seu uso.

Diante de ferramentas tão amplas e genéricas, como as de autoria e para produção de material para Internet, professores e profissionais de informática se debatem na tentativa de produzir algo que aproveite o fantástico potencial da computação, mas isto tem demandado muito tempo e investimento para se alcançar algum resultado.

O sistema “Casa Mágica” surge então com a proposta de ser uma ferramenta para que não só professores e profissionais de informática, mas também alunos, possam produzir aplicações educacionais. No entanto, ao invés de uma ferramenta genérica, o sistema tem sido projetado para atender especificamente a área educacional, trazendo benefícios como:

- Por ser uma ferramenta orientada, aumenta a produtividade de aplicações educacionais, bem como facilita a condução nas etapas de produção da mesma.
- Amplia os horizontes de produção de aplicações educacionais, pois traz recursos que colocam o usuário diante de uma nova forma de abordar o que se pretende produzir.
- Possibilita a criação de objetos que englobem elementos de estudo em aplicações. Estes objetos podem ser reutilizados e são projetados de forma a promover um intercâmbio entre produtores, o que possibilitará, no futuro, se criar um banco de objetos que serão como tijolos para a construção de

aplicações, facilitando por demais sua criação e incrementando progressivamente a produtividade.

O Ambiente Casa Mágica

Cada aplicação em Casa Mágica é denominada de **Mundo**. Um mundo consiste em uma reunião de **Locais** interligados entre si e um conjunto de **Entidades** que podem estar em alguns destes Locais, se deslocar por Locais ou simplesmente representar algo abstrato. Os Locais podem representar um espaço físico, por onde se pode mover e interagir com objetos ou apenas um estado ou parte de algo, tal como a página de um livro - ou uma aplicação hipermídia - ou um slide de uma apresentação.

Existe uma Entidade em especial que representa o *self*, ou seja, o próprio aluno ou usuário que navega pelo Mundo. Esta Entidade possui atributos e ações específicos, tal como a capacidade de se deslocar de um lugar para outro, atributos que permitem registrar pontuação (que pode ser utilizada para avaliação) e atributos que definem determinados objetos que o usuário carrega.

Nos locais são dispostos os **Objetos** que, por sua vez, também podem representar objetos concretos ou abstratos. O usuário modela o local dispondo no mesmo os objetos e dando-lhes propriedades, tal como cor, imagem, tamanho, etc. O sistema possui um **Editor Visual de Locais**, onde é disponibilizado um conjunto de objetos prontos a serem dispostos, que precisam somente da configuração de suas propriedades. Este editor possibilita a fácil interligação dos locais. Existe também a **Oficina de Classes** que permite que o usuário construa seus próprios objetos através da definição de **Classes** que descrevem genericamente os atributos que possui um objeto, bem como a forma que ele age e reage (métodos).

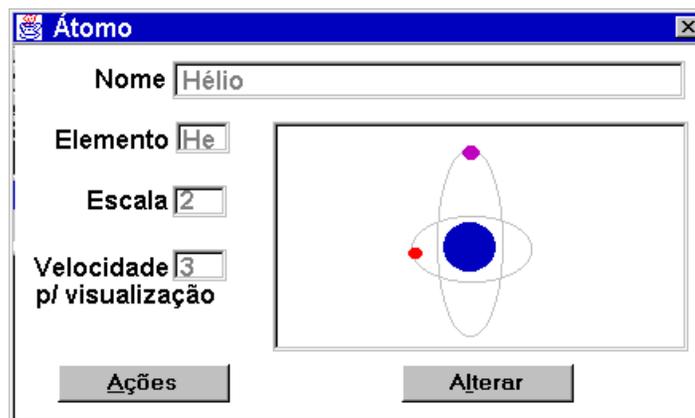


Figura 1: Janela de modificação de atributos de um objeto

A janela apresentada demonstra a forma como os atributos de um objeto são definidos. A idéia básica é que o criador da aplicação irá também criar um conjunto de classes e também a forma como atributos e ações desta classe poderão ser visualizados e modificados.

Tomemos o exemplo da figura 1, onde foi criada uma classe átomo. Esta classe pode ter sido criada para uma aplicação com o intuito de ensinar conceitos de química. O criador, no entanto, não se limita a construir uma aplicação, mas constrói classes e define como elas podem ser visualizadas e configuradas. Para isto, a linguagem do Casa Mágica possui um recurso denominado **Visor**, que permite que sejam definidos quais os atributos

que serão visualizados, com que títulos, em que posições da janela, e com que controle de edição (visor de imagens, caixa de textos, caixas de verificação, botões de rádio, etc.).

A perspectiva adotada pelo Sistema cria a possibilidade de que sejam definidos objetos que encapsulem determinados conhecimentos e que até mesmo simulem o funcionamento de determinado objeto. O Casa Mágica provê um módulo de **Laboratório** que permite que sejam instanciados objetos de classes isoladamente e que se estude seu comportamento. Assim estas classes podem se tornar objeto de estudo ativo por parte do aluno, ou seja, ultrapassa o conceito de conhecimento estático.

Além disto, é possível se criar facilmente bibliotecas destes objetos, incentivando o intercâmbio dos mesmos entre desenvolvedores. Um professor que pretenda desenvolver uma aplicação na área de química, poderá utilizar algumas classes que já foram desenvolvidas, isto incrementa e facilita o desenvolvimento da aplicação. Vale ressaltar que estão sendo estudados mecanismos que salvaguardem os direitos autorais.

Muitos dos aspectos analisados acima são inerentes ao próprio paradigma da programação orientada ao objeto. A novidade está no fato de utilizá-los ativamente na produção de aplicações educacionais, e de tirar proveito da possibilidade que este paradigma traz de reutilização de código, para se fazer reutilização de classes que encapsulem verdadeiras representações de algum conhecimento; além de explorar a concepção que traz o próprio paradigma de se poder configurar e modificar visualmente os atributos e ações de um objeto; o que significa, neste caso, a possibilidade de configurar e modificar um objeto que representa ou simula uma porção de conhecimento, por exemplo, poderia criar-se uma classe para simular o gráfico de uma equação de segundo grau, onde o aluno pode ativamente alterar os atributos do objeto, tais como os coeficientes da equação, e verificar quais os resultados alcançados imediatamente. Neste caso, se concretiza a maior vocação do próprio Sistema Casa Mágica que é o de trabalhar com simulações de determinadas realidades.

A Linguagem

A linguagem utilizada pelo Sistema para o desenvolvimento de aplicações foi criada com o intuito de ser simples e refletir de forma bastante clara e limpa o que se desenvolve com a ferramenta visual.

Além disto, a linguagem foi desenvolvida em português, com o intuito de aproxima-la dos seus usuários, principalmente professores, alunos e educadores em geral. Esta decisão obviamente torna a linguagem restrita a países de língua portuguesa, mas está previsto no projeto se criar um compilador que aceite comandos tanto em português como em outra língua, tal como inglês ou castelhano.

Toda aplicação consiste num conjunto de **Unidades** que podem ser: **Locais**, **Entidades** ou **Bibliotecas**. O conjunto das Unidades interligadas, cada uma cumprindo seu papel, corresponde ao que o Sistema denomina Mundo, que também corresponde a um arquivo onde ficam registradas todas as Unidades, qual o papel que cada uma delas cumpre e como elas se relacionam.

Cada uma destas Unidades pode conter Classes, Objetos e Visores, segundo a necessidade. O trecho de código a seguir ilustra, parcialmente, a definição de um Local onde está definida a classe Átomo e um objeto denominado Hélio desta classe.

Local Laboratório

Classe Átomo

```
.Visual      Imagem  
.Nome        Texto = ""  
.Elemento    Texto = ""  
.Escala      Inteiro= 1  
.Velocidade  Inteiro= 1
```

```
Movimento responde mensagem "CLICADO"
```

```
:  
:
```

```
Fim Movimento
```

```
Reação responde mensagem "REAGE"
```

```
:  
:
```

```
Fim Reação
```

```
:  
:
```

Fim Átomo

Hélio é um Átomo:

```
.Nome        =    "Hélio"  
.Elemento    =    "He"  
.Escala      =    2  
.Velocidade  =    3
```

Fim Hélio

Fim Laboratório

Observe-se que os atributos da classe vêm precedidos de um ponto, seguidos das ações, que correspondem aos métodos da classe (afim de simplificar o exemplo, não foi listado o código dentro das ações). O uso do ponto antes dos atributos é um recurso meramente visual que facilita sua distinções das ações, por este motivo, seu uso é opcional.

O objeto pode estabelecer valores para alguns ou todos os atributos da classe, aqueles que não são definidos assumem um valor padrão. Cada elemento do objeto reflete um item da janela apresentada na Figura 1 e, à medida que o usuário muda os atributos visualmente, eles são modificados no programa de forma bastante clara e limpa.

A cada ação pode ser associado um conjunto de mensagens às quais ela responde. Os Objetos da aplicação podem se intercomunicar entre si através da chamada direta de ações do Objeto ou através do envio de uma mensagem que será respondida por uma das ações do objeto.

Podem existir mensagens que o Objeto receba que não correspondam a nenhuma ação, isto não constitui um erro, uma vez que muitas das mensagens são enviadas apenas para notificar os Objetos de determinado acontecimento, não exigindo destes uma resposta. Por exemplo, ao entrar-se em um Local, todos os objetos deste Local recebem uma mensagem notificando tal fato, alguns deles podem reagir a esta mensagem (mostrando-se na tela, por exemplo), mas outros podem simplesmente ignorar a mensagem.

O Sistema prevê um recurso de criação e edição de Objetos de forma visual. Este recurso recebe o nome de Visor.

Abaixo a definição de um Visor para a Classe Átomo:

Visor Átomo

Objeto Campo_Visual eh um Campo_Imagem_Objeto :

```
.x = 0  
.y = 0  
.largura = 100  
.altura = 50  
.atributo_objeto = "Visual"
```

Fim Campo_Visual

Objeto Campo_Nome eh um Campo_Texto_Objeto :

```
.x = 0  
.y = 60  
.numero_caracteres = 10  
.atributo_objeto = "Nome"  
.rotulo = "Nome"
```

Fim Campo_Nome

Objeto Campo_Elemento eh um Campo_Texto_Objeto :

```
.x = 30  
.y = 60  
.numero_caracteres = 3  
.atributo_objeto = "Elemento"  
.rotulo = "Elemento"
```

Fim Campo_Elemento

Objeto Campo_Escala eh um Campo_Inteiro_Objeto :

```
.x = 60  
.y = 60  
.numero_caracteres = 10  
.menor_valor = 0  
.maior_valor = 100  
.atributo_objeto = "Escala"  
.rotulo = "Escala"
```

Fim Campo_Escala

Objeto Campo_Velocidade eh um Campo_Inteiro_Objeto :

```
.x = 90  
.y = 60  
.numero_caracteres = 10  
.menor_valor = 0  
.maior_valor = 100  
.atributo_objeto = "Velocidade"  
.rotulo = "Velocidade p/ visualização"  
Fim Campo_Velocidade
```

Fim Visor

No Visor fica estabelecido que recurso será usado para editar e visualizar o campo (Campo_Imagem, Campo_Texto, Campo_Inteiro, Seletor, etc.), qual sua posição na tela e tamanho, bem como outras definições adicionais que se façam necessárias.

Como pode ser observado, as definições de como cada atributo será visualizado e editado é determinado por Classes nativas especiais próprias para edição de atributos em tempo de edição.

Quando os Visores são construídos visualmente, cada campo é posicionado, seu tipo é estabelecido, seu tamanho, etc. Tudo isso atualiza automaticamente o código. Uma Classe pode possuir mais de um Visor que possibilita perspectivas diferentes da mesma. Por exemplo, um professor pode ter uma visão de determinada Classe que lhe permita configurar determinados recursos que não serão acessíveis ao aluno, seja porque são detalhes que o professor decidiu didaticamente esconder, para que o aluno se concentre apenas no que é relevante em sua abordagem, seja porque são detalhes que só digam respeito ao professor, tais como atributos de avaliação.

É possível criar-se uma Classe que aproveite todas as características de uma outra Classe definida, acrescentando-lhe novos recursos e modificando alguns já existentes. Esta capacidade é conhecida como Herança. Por exemplo, o professor de química poderia criar uma Classe de átomos que representassem especificamente os átomos dos elementos denominados metais da seguinte forma:

Classe Metal da classe Átomo:

```
:
```

Fim Metal

A partir daí poderiam ser definidos novos atributos e ações, bem como modificados alguns já existentes.

Segundo os critérios estabelecidos por Bertrand Meyer [MEY88] para a caracterização de uma linguagem Orientada ao Objeto, a linguagem do Casa Mágica implementa praticamente todos os requisitos básicos, com exceção da herança múltipla

Sublinguagem de Médio Nível

Além da linguagem especificada anteriormente, existem duas outras sub-linguagens que dão suporte a aspectos específicos do desenvolvimento de aplicações.

A primeira é uma sublinguagem de médio nível, que será utilizada apenas por programadores experientes que desejam extrair o máximo em velocidade e flexibilidade da ferramenta.

As principais características particulares estabelecidas para esta linguagem são:

- É a linguagem que será interpretada diretamente pelo sistema, portanto, deve ser o mais eficiente possível;
- A ênfase é dada principalmente à eficiência, deixando a facilidade de uso em segundo plano;

Pela inerente necessidade de eficiência e de simplicidade, a construção desta linguagem foi baseada em estudos de uma linguagem de médio nível denominada FORTH¹. O FORTH se destaca por sua extrema eficiência, por apresentar uma estrutura de fácil implementação e cujo código pode ser interpretado de forma simples. Devido a algumas de suas características, tal como a escrita de expressões em notação polonesa, o interpretador fica dispensado da maior parte de seu trabalho, detendo-se exclusivamente na conversão de instruções.

Apesar de tomar como base de estudos o FORTH, a linguagem de médio nível em questão é uma linguagem bastante diferente por vários aspectos: o principal consiste no fato de que ela é Orientada ao Objeto. Além disso, ela consegue trabalhar com mais tipos de dados que o FORTH, e se difere bastante nas instruções.

Sublinguagem Portinari

Para permitir um conjunto amplo de recursos gráficos, foi acrescentada uma sub-linguagem específica denominada Portinari. Ela possui um conjunto de primitivas gráficas que possibilitam facilmente a construção de modelos geométricos, bem como a inserção de imagens, em formato de mapa de *bits*, tais como fotografias.

Para a definição da linguagem foi feito um estudo de quatro formas de representação de gráficos a fim de se estabelecer qual a melhor forma de fazê-lo. Foram analisadas as seguintes especificações gráficas:

- GKS (*Graphics Kernel System*) - núcleo gráfico desenvolvido com o intuito de padronizar o desenvolvimento de aplicações gráficas [BON88];
- Classe *Graphics* do Java - responsável pela apresentação de gráficos na tela [RIT96];
- GDI (*Graphics Device Interface* - Interface de Dispositivo Gráfico) do Microsoft Windows 3.1;
- Comandos Gráficos do TURBO PASCAL da Borland;

Junto com a linguagem foi elaborado um **Editor Gráfico** no qual são produzidas todas as imagens da aplicação. As imagens são registradas como uma descrição na

¹ **FORTH** - linguagem de médio nível criada no início da década de 70 por Charles Moore e Elizabeth Rather, no "National Radio Astronomy Observatory" [KNE86].

linguagem dos elementos geométricos e das imagens mapa de *bits*, isso permite uma interação ativa entre a aplicação e os objetos gráficos no que diz respeito à reconfiguração dinâmica de elementos que estão na tela, além disto, todos os elementos geométricos podem ser redimensionados e rotacionados sem perda de qualidade pois a imagem é redesenhada a partir de seu modelo geométrico cada vez que é feita uma modificação.

A tabela abaixo ilustra as primitivas da linguagem:

Ponto	CoordX CoordY Cor
Linha	CoordXinicial CoordYinicial CoordXfinal CoordYfinal
Retângulo	CoordX CoordY Largura Altura
RetânguloP	CoordX CoordY Largura Altura
RetânguloArred	CoordX CoordY Largura Altura LargElipCanto AltElipCanto
RetânguloArredP	CoordX CoordY Largura Altura LargElipCanto AltElipCanto
Oval	CoordX CoordY Largura Altura
OvalP	CoordX CoordY Largura Altura
Arco	CoordX CoordY Largura Altura anguloInicial anguloFinal
ArcoP	CoordX CoordY Largura Altura anguloInicial anguloFinal
Polígono	NumPontos PontosX[] PontosY[]
PolígonoP	NumPontos PontosX[] PontosY[]
Preenche	CoordX CoordY CorLimite
Texto	Texto
Imagem	ArquivoImagem

Além das primitivas existe um conjunto de comandos que configuram cores e modos de apresentação. Exemplo de uma descrição em Portinari:

linha	85 121 261 38
retângulo	80 53 118 89
oval	80 53 118 89

Ferramenta de Desenvolvimento e Execução

O Sistema Casa Mágica completo reúne:

- **Conjunto de Ferramentas Visuais** - para a construção da aplicação como um todo. Deste grupo fazem parte:
 - **Gerenciador geral da Aplicação** - este é o módulo principal, a partir dele se organiza toda a aplicação e se dispara a construção de Unidades. Este

módulo possui um *browser* que possibilita a organização visual dos Locais e sua interligação, bem como a disposição das Entidades. Nele, cada Unidade é representada por um ícone e o usuário pode dispor estes ícones da forma que lhe convier, por exemplo, se cada Local representa um lugar físico, cada ícone representado o Local que pode ser disposto no *browser*, de acordo com o lugar no espaço físico que ele ocupa em relação aos demais (semelhante a um mapa da região).

- **Editor visual de Locais** - como já foi descrito anteriormente, permite a modelagem dos Locais, dispondo visualmente Objetos concretos e abstratos no mesmo, bem como definindo como as Entidades se comportarão no Local.
 - **Editor de Entidades e Bibliotecas** - permite estabelecer quais as Classes, Objetos e Visores que irão compor as Entidades e Bibliotecas, bem como construí-las.
 - **Oficina de Classes** - módulo disparado tanto pelo Editor visual de Locais como pelo Editor de Entidades e Bibliotecas; possui um conjunto de ferramentas para a construção de Classes e montagem visual de seus Visores.
 - **Laboratório** - esta ferramenta trabalha interligada com a Oficina de Classes. Ela permite que sejam instanciados Objetos de determinada Classe e interativamente seus atributos sejam modificados, bem como sejam enviadas mensagens para os Objetos, para se observar sua reação.
 - **Editor Gráfico** - permite a edição de gráficos que serão transformados em uma descrição em Portinari.
-
- **Compilador** - traduz todo o código gerado para um formato de código intermediário (semi - compilado) , que será utilizado no momento de execução. Este compilador realiza todos os processos de Análise Léxica e Sintática inerentes de um compilador, notificando os erros encontrados.
 - **Módulo de Execução** - carrega e executa o código semi - compilado.

Todo o Sistema foi escrito na linguagem de programação Java. Tal ferramenta foi escolhida por uma série de vantagens verificadas, sendo dentre elas as principais:

- Possui uma linguagem Orientada ao Objeto, que é importante não só pelos benefícios que traz este paradigma bem como pelo fato da linguagem do Sistema Casa Mágica também ser Orientada ao Objeto, o que produz um casamento perfeito;
- O código produzido pelo Java pode rodar num variado conjunto de plataformas sem uma única modificação, tanto no código fonte quanto no compilado;
- A linguagem possui suporte completo a Internet, o que abre amplas perspectivas para o projeto de criar aplicações que possam rodar via Internet, interligar usuários e Locais.

Um sistema em Java pode ser classificado com uma *Applet* ou uma *Application*. Uma *Applet* consiste em um pequeno programa que pode ser executado em páginas HTML na Internet, portanto necessita um navegador de Internet para que possa ser executado; já uma *Application* consiste em um programa independente, que pode ser executado diretamente a partir do Sistema Operacional, através de um módulo

interpretador denominado Virtual Machine, uma vez que o Java não compila seu código diretamente para linguagem de máquina, mas para uma linguagem intermediária (*bytecodes*) que consiste em um código genérico de máquina que pode ser facilmente convertido para a linguagem nativa da máquina, onde o programa está sendo executado.

Todo o Sistema Casa Mágica foi produzido sob a forma de *Application*, seja pela sua independência do navegador de Internet, seja pelo seu porte que o tornaria excessivamente pesado para ser executado como *Applet*.

Estão sendo feitos estudos, no entanto, no sentido de se transformar o Módulo de Execução do Sistema em uma *Applet*, o que trará ganhos significativos para o projeto, pois o usuário através de seu navegador na Internet poderá entrar diretamente em um Mundo produzido em Casa Mágica. Além disto, está previsto neste projeto a possibilidade de se espalhar em Locais por diversas máquinas conectadas à Internet, bem como permitir que os diversos usuários interajam nos Locais, a partir de suas Entidades, ou seja, se um usuário X entra no mesmo Local e no mesmo Mundo que um usuário Y, no momento em que ambos estão conectados, via Internet, à aplicação, o Sistema cuidará de criar uma imagem do usuário X para o usuário Y e vice-versa. Para isto serão utilizados os riquíssimos recursos de suporte à multitarefa do Java.

O processo de elaboração do compilador toma como base os estudos de Alfred V. Aho e outros [AHO86] e de Valdemar Setzer e Inês Melo [SET86] referentes à construção de compiladores.

Para a elaboração do Analisador Léxico e do Analisador Sintático, foram utilizados programas de apoio que, a partir de gramáticas, geram os programas para realizar as respectivas análises. Os analisadores foram gerados por programas diferentes, mas que geram módulos que trabalham integrados; desta forma, o Analisador Sintático gerado por um programa utiliza o Analisador Léxico produzido pelo outro.

Considerações Finais

O desenvolvimento do Sistema Casa Mágica, fruto de um trabalho que se iniciou em 1994, quando foi produzida sua primeira versão, tem evoluído muito suas idéias e, apesar de manter sempre a sua característica básica de uma ferramenta voltada para o ambiente educacional, abre-se para tantas novas possibilidades sempre criadas com a evolução da computação e com a Internet.

Como fruto deste trabalho, surge a base para um trabalho ainda maior, uma vez que todos os esforços têm-se concentrado em tornar concreta uma ferramenta que possa alavancar a criação de outros recursos, potencializando o educador em sua tarefa de criação.

O Sistema Casa Mágica pretende ser a base para a criação de outros recursos, tais como Assistentes Especialistas que orientem o professor na criação de determinadas aplicações. Por exemplo, se o professor pretende criar um conjunto de textos e imagens interligados através de *links*, no estilo hipertexto ou hipermídia, haverá um assistente especializado nisto. Se ele pretende ir além e criar uma simulação, haverá outros assistentes que auxiliem na construção de determinados aspectos da mesma, tais como a montagem e interligação dos ambientes básicos. Podem existir assistentes para elaborar a avaliação dos resultados, e assim por diante.

Está prevista também a formação de uma grande equipe interdisciplinar para a produção de uma grande biblioteca de objetos básicos em principais áreas do

conhecimento, afim de que o professor tenha uma rica base de elementos dos quais partir para montar a aplicação.

No que diz respeito ao uso do Casa Mágica na Internet, além do que já foi explanado, está prevista a criação de um conjunto de Classes que produzam uma interface em HTML. Desta forma poderá haver uma aplicação sendo executada em um servidor enquanto, através de uma interface HTML o usuário envia comandos para a aplicação, que por sua vez produz novas páginas HTML com os resultados. Enfim, a aplicação rodará em um servidor, que não é a máquina do usuário propriamente dita, e através do protocolo HTTP - *Hyper Text Transfer Protocol* - ela interage com o navegador deste usuário. Assim, através deste novo recurso, será possível a criação de Escolas Virtuais rodando elaboradas aplicações e capacitando o Sistema a ser utilizado no ensino à distância.

REFERÊNCIAS

- [AHO86] AHO, A. V., SETHI, R., ULLMAN, J. D. Compilers, Principles, Techniques and Tools. USA: Addison - Wesley Publishing Company, 1986.
- [MEY88] MEYER, B. Object - Oriented Software Construction. USA: Prentice Hall International Ltd., 1988.
- [BAB89] BABIN, P., KOULOUMDJIAN M. Os novos modos de compreender. São Paulo: Edições Paulinas, 1989.
- [BON88] BONO, P., ENCARNAÇÃO, J. L. HERZNER, W. R. - PC - Graphics with GKS. USA: Prentice Hall International Ltd., 1988.
- [KNE86] KNECHT, K. FORTH. Rio de Janeiro: Editora Campus, 1986.
- [PAP88] PAPERT, S. LOGO: Computadores e Educação. São Paulo: Brasiliense, 1985.
- [RIT96] RITCHEY, T. Programando em Java & JavaScript para Netscape 2.0. São Paulo: Editora Quark do Brasil, 1996.
- [SET86] SETZER, W. W.; MELO, I. S. H. A construção de um compilador. Rio de Janeiro: Editora Campus, 1986.