

WCC – WEB COURSE CREATOR UM FRAMEWORK PARA A CRIAÇÃO DE AMBIENTES DE APRENDIZADO PARA A WEB

Edson José Pacheco (*edsonpacheco@hotmail.com*)

Giovan Araújo de Marco (*giovandemarco@hotmail.com*)

Henri Frederico Eberspächer (*henri@rla01.pucpr.br*)

PUC PR - Pontifícia Universidade Católica do Paraná
LAMI - Laboratório de Mídias Interativas
Rua Imaculada Conceição, 1155
CEP: 80215-901 - Curitiba - PR - Brasil

Palavras Chave: Frames, Inteligência Artificial, Internet, Sistemas Multi-Agente, Sistemas Tutores Inteligentes, Java, Programação Orientada a Objetos.

Resumo

Este trabalho apresenta a arquitetura, o projeto e o desenvolvimento do WCC (*Web Course Creator*), que é um sistema para criação de ambientes de treinamento baseado no paradigma de Treinamento e no uso de hipertexto e multimídia. São apresentados os fundamentos teóricos básicos sobre as técnicas de Inteligência Artificial utilizadas para fazer o gerenciamento do conhecimento, a forma metodológica da interação dos assistentes com o aprendiz e a arquitetura de desenvolvimento do WCC e dos exercícios criados pelo mesmo.

1. Introdução

A velocidade e o dinamismo da informação torna inadequado os meios clássicos de publicação (livros, revistas, etc). Cada vez mais se faz necessário inserir recursos que diferenciem o conteúdo, tornando-o convidativo aos “consumidores de informação”.

Hoje é impossível imaginar um sistema educacional ou empresarial que desconsidere as vantagens da tecnologia para o aprendizado. Somente um ambiente rico em detalhes, envolvendo recursos multimídia bem dimensionados, podem motivar a curiosidade do aprendiz, levando-o a inferir informações, caracterizando o verdadeiro aprendizado.

Corroborando a tecnologia como ferramenta de ensino, cita-se um estudo desenvolvido pela Força Área dos EUA (*apud* [SCH94]), onde é relatado o grau de retenção de informações pelas pessoas:

- 25% do que ouvem;
- 45% daquilo que vêem e ouvem;
- 70% daquilo que vêem, ouvem e fazem.

Destá forma, o nível de envolvimento e interatividade da informática é de potencial aplicação para o aprendizado. No entanto, deve-se otimizar utilização do computador, tornando-o uma ferramenta ativa e motivadora para o processo de aprendizagem.

Muitos ambientes de aprendizado estão sendo criados, entre eles os CBTs (*Computer-Based Training*), os CAIs (*Computer-Assisted Instruction*), os CALs (*Computer-Assisted Learning*) e os CBLs (*Computer-Based Learning*), comprovando a eficácia do uso da tecnologia, em especial da multimídia, como ferramenta de ensino. No entanto, estas ferramentas apresentam alta complexidade de desenvolvimento, o que dificulta o seu desenvolvimento por profissionais que não dominem as técnicas envolvidas. Além disso, a maioria dos ambientes categorizados exigem o uso de tecnologias proprietárias, dificultando seu uso por um grande número de aprendizes.

De forma minimizar a dependência tecnológica e maximizar o uso dos ambientes de aprendizagem, bons projetos estão sendo desenvolvidos de forma a utilizar a arquitetura Web, entre estes pode-se citar os trabalhos de Cattani e

Otsuka. No entanto, existem conteúdos em hipertexto que apresentam deficiências em sua concepção, tornando-se uma mera transcrição de conteúdo, não levando em consideração os aspectos cognitivos do aprendiz.

O presente artigo traz informações sobre o projeto WCC (*Web Course Creator*) que implementa uma ferramenta de autoria, que permite ao tutor¹ o desenvolvimento de conteúdo em hipertexto para a Internet, conforme as correntes metodológicas construtivistas.

O WCC utiliza técnicas pertinentes à Inteligência Artificial na representação do conhecimento, possibilitando a geração de um ambiente conciso e extensivamente adaptável às características individuais do aprendiz. Em paralelo existe a geração de agentes, que monitoram o desempenho e as atividades dos aprendizes, possibilitando a assistência da exploração do conteúdo publicado.

O WCC permite a gerência do conhecimento, criando conteúdos em hipertexto que permitem a exploração com objetivo, onde o aprendiz é conduzido na aquisição do conhecimento.

2. Conceitos de IA utilizadas na gerência do conhecimento

É consenso que um sistema de IA não é capaz somente de armazenamento e manipulação de dados, mas também da aquisição, representação e manipulação de conhecimento. Esta manipulação inclui a capacidade de deduzir ou inferir novos conhecimentos – novas relações sobre fatos e conceitos – a partir do conhecimento existente.

Quando trata-se de representação do conhecimento, fora do âmbito de IA, uma das técnicas mais conhecidas é a de mapas conceituais. Um mapa conceitual parte de um conhecimento inicial, bem generalizado, a partir disso, constrói-se uma rede de conhecimento que, de uma maneira geral, se especializam ou se relacionam com o conhecimento inicial, como pode ser observado na **Figura 1**.

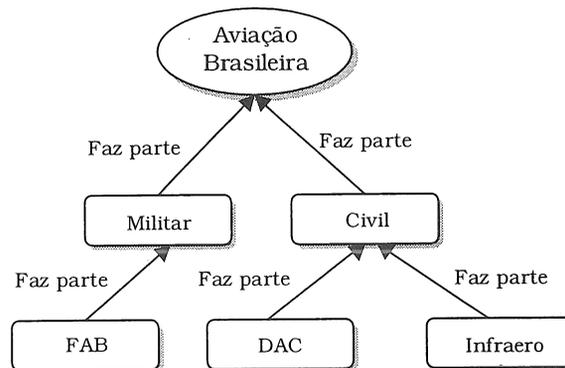


Figura 1 — Mapa Conceitual referenciando parte da estrutura da Aviação Brasileira

Formalizando o conceito de mapa conceitual surgem as Redes Semânticas, que, graficamente, são formados de nós e arcos entre estes nós. Os nós representam objetos, conceitos, fatos, princípios e situações de um domínio de conhecimento, os arcos representam as relações entre os nós. Se a **Figura 1** fosse instanciada como uma Rede Semântica, um nó poderia ser o DAC (Departamento de Aviação Civil) e um arco poderia ser a relação que o DAC faz parte do nó da aviação Civil.

2.1. Sistema de Frames

A estrutura dos Frames² foi originalmente proposta por Minsky (1975), como a base para o entendimento da percepção visual, diálogos de linguagem natural e outros comportamentos complexos [MES98].

¹ Criador do ambiente de aprendizado

² É encontrado na literatura, a designação classe, entidade ou objeto; para um frame

Frames possibilita a representação das estruturas internas dos objetos, mantendo a possibilidade de representar herança. O método de *frames* também está ligado a origem das idéias que levaram às linguagens de programação orientada a objetos (linguagens OOP).

Uma grande inovação diz respeito a introdução de nós com estrutura interna. Tal estrutura interna foi possível de ser idealizada graças ao mecanismo representacional proporcionado pelo atributo, que representa o local onde uma parcela do conhecimento é armazenada.

Sendo assim, um *frame* consiste em um conjunto de atributos que, através de seus valores, descrevem as características de um objeto que representa alguma entidade do mundo. Os atributos interligam-se a outros *frames* criando uma rede hierárquica dependente, atrelando a estrutura como um todo.

Como o *frame* possui o poder de herança de propriedades, pode-se especificar uma classe de objetos que é instanciada em uma subclasse, sendo que esta usufrui de todas as propriedades da sua super-classe. Pode-se observar na Figura 2 um exemplo de derivação entre objetos.

Frame: <i>Avião</i>		
Atributos	Default	Tipo
Número de Motores	1	Número
Formato da Asa	Retangular	Símbolo
Trem de Pouso	3	Número

↑ é

Frame: <i>Boeing</i>		Super Frame: <i>Avião</i>	
Atributos		Default	Tipo
Número de Passageiros		115	Número
Finalidade		Transporte de Passageiros	Letra

Figura 2 — Exemplo de frames

Outro conceito importante é o *raciocínio guiado por expectativas* [BIT97]. Um frame contém atributos, e estes podem ter valores típicos. Ao instanciar um frame para que corresponda a uma dada situação, o processo de raciocínio deve preencher os valores dos atributos com as informações disponíveis na descrição da situação. O fato de que o processo de raciocínio conheça as informações necessárias, e o caso destas informações estarem disponíveis é preponderante para a eficiência do reconhecimento de uma situação complexa.

Para a inicialização do estado interno do frame, o mesmo deve possuir um *construtor*. O construtor é representado através de uma *função*³ de leitura, responsável pela captura de valores do frame, e uma *Via de escrita* responsável pela alteração do estado interno do frame.

2.1.1. Uma abordagem matemática de Frames

Matematicamente, os frames podem ser formalizados a partir da teoria de conjuntos. Deve-se considerar, que a teoria dos conjuntos não manipula apenas objetos simples, mas é uma ferramenta poderosa no trato de objetos complexos. Considera-se, na composição dos frames, que cada objeto pode possuir um estado interno (como variáveis internas).

Na teoria originária de frames é distinto o conceito de atributos⁴ e frames, ou seja, um frame pode apresentar um componente simples — como uma variável booleana — ou um outro frame; no entanto, é possível generalizar o conceito de atributo, incluindo-o como uma categoria de um frame, ou seja, um atributo passa a ser representado como um frame com poucos estados internos — no caso da variável booleana apenas dois (Verdadeiro ou Falso). Desta forma a composição dos objetos torna-se genérica o suficiente para ser reutilizada em diversos níveis de projetos.

³ O termo função é o que melhor se emprega na situação. Função é um termo muito utilizado na área de informática e designa um procedimento capaz de retornar e receber valores.

⁴ A diferença entre um atributo e um frame, diz respeito ao grau de complexidade entre ambos. Enquanto um atributo não pode ser composto (como é o caso de uma variável), um frame pode ser formado pela composição de atributos e outros frames.

Para formalizar as relações dos frames, é necessário definir as inter-relações entre os objetos. Os atributos apresentados na Tabela 1 são tradicionalmente encontrados na literatura, outros podem ser obtidos pela generalização ou especificação dos apresentados. Destaca-se que os três atributos apresentados são suficientes para a geração das relações entre os objetos a serem instanciados no WCC.

Atributo	Relação que representa
É um	Herança
Tem parte	Agregação
Não é	Disjunção

Tabela 1 — Relacionamento entre as Classes e Objetos

Para exemplificar as relações, propõe-se o exemplo representado na Figura 3.

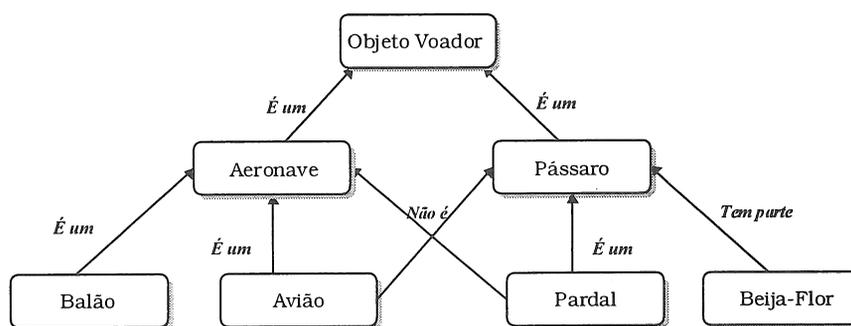


Figura 3 — Representação de relacionamento entre Frames⁵

Como é observado a classe Aeronave e a classe Pássaro são ambas derivadas de Objeto Voador (apesar do termo *objeto* não ser próprio na designação de animais). Desta forma, pode-se deduzir que o Objeto Voador contém informações genéricas sobre o ato de voar, como, por exemplo, as características aerodinâmicas necessárias.

Seguindo o diagrama, a classe Avião é derivada de Aeronave, desta forma **Avião** \subset **Aeronave** (a entidade Avião está contida em Aeronave). Por sua vez, **Avião** \subset **Objeto Voador** (a entidade Avião está contida em Objeto Voador). A Figura 4 representa em detalhes a composição dos objetos:

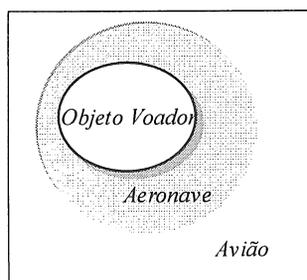


Figura 4 — Hierarquia da composição do Objeto Avião

Observa-se que o objeto Avião é formado por Aeronave e pelo Objeto Voador, bem como por suas características próprias, compondo o universo Avião.

O conceito de **Tem Parte**, demonstra que a classe não herda todas as características da super-classe, mas apenas algumas. Em programação orientada a objetos, pode-se verificar o mesmo efeito com o uso dos mnemônicos *private* e *protected* em uma classe. Desta forma, representa-se, **(Beija-Flor \cap Pássaro $\neq \emptyset$) \wedge (Beija-Flor \cap Pássaro \neq Pássaro)**

⁵ A diferença maior do Beija-Flor para as aves conhecidas, está na sua forma de vôo. Seria a grosso modo, comparar o vôo dos aviões (aves em geral) com o vôo dos helicópteros (Beija-Flor).

(a interseção entre Beija-Flor e Pássaro não representa um conjunto vazio e, também, não é correta a afirmação de que $\text{Pássaro} \subset \text{Beija-Flor}$).

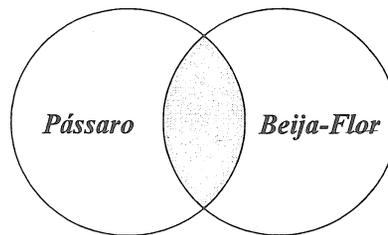


Figura 5 — Exemplo sobre o atributo *Tem Parte*

Ou seja, existem elementos que a classe Beija-Flor herda de pássaro, mas nem todas.

A relação **Não é**, é utilizada conforme o modelo a ser adotado, alguns modelos não permitem tal relação. No caso do WCC o aspecto didático do conteúdo é extremamente enriquecido com tais relações, permitindo a geração de exercícios e interações do aprendiz junto ao sistema. A expressão $\text{Pardal} \cap \text{Avião} = \emptyset$, afirma que entidades Pardal e Avião não apresentam nenhum elemento em comum. A Figura 6 ilustra tal situação.

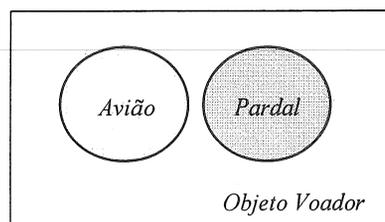


Figura 6 — Relacionamento entre as entidades Avião e Pardal

Para a generalização, considera-se A como sendo a super-classe e B como sendo a classe derivada, conforme Tabela 2.

Atributo	Expressão
É um	$B \subset A$
Tem parte	$(A \cap B \neq \emptyset) \wedge (A \cap B \neq B)$
Não é	$A \cap B = \emptyset$

Tabela 2 — Relacionamentos entre duas entidades A e B

2.2. Sistemas Multi-Agentes

Tem se observado o surgimento de problemas de alta complexidade e de maior dimensão, tornando necessário, então, uma troca de informações entre conhecimentos de domínios distintos. Conforme Jaques a Inteligência Artificial Distribuída “assume a metáfora baseada no comportamento social”, onde entidades distribuídas buscam uma solução colaborativa de problemas globais, o que a difere da IA clássica, que é baseada no comportamento individual humano [JAQ98]. Tais entidades são denominadas de agentes e um conjunto dos mesmos compõe um Sistema Multi-agente (*Multi-agent System – MAS*).

Os agentes em um Sistema de Multi-agentes podem ser projetados para a resolução dos mais variados tipos de problemas. Decorrente do fato de que tais agentes representem entidades autônomas, que tem conhecimento da sua existência e de outros agentes, trabalhando de forma colaborativa na solução do problema.

Um agente é definido como uma entidade, que está apta para perceber e representar parcialmente seu ambiente. Como é necessário a troca de informações e a decomposição de tarefas, um agente deve possuir capacidades específicas para a interação no ambiente.

Os agentes devem possuir conhecimento da existência dos outros agentes. A comunicação entre os agentes deve ser realizada num protocolo conhecido. O particionamento das tarefas permite a cooperação e a solução de problemas de monta maior.

2.2.1. Agentes Cognitivos

Os agentes cognitivos possuem estados mentais de informação e manipulam o conhecimento. Os estados mentais são representados internamente nos agentes, e representam as características: crenças, conhecimento, desejos, intenções, obrigações etc. [JAQ98].

Os agentes cognitivos são ditos sociais pois, além de manipular o conhecimento, os mesmos dominam as características do meio. Para o trabalho num sistema multi-agente é necessário definir o tipo de organização, a comunicação, a delegação de tarefas e a cooperação entre os agentes.

3. Arquitetura do sistema WCC

O WCC é composto por cinco assistentes:

- Assistente de Formatação;
- Assistente de Edição;
- Assistente de Geração de Hiperlinks;
- Assistente de Aprendizagem;
- Assistente de Estatísticas.

O *Assistente de Formatação*, é responsável pela aquisição das informações do tutor e a representação deste conteúdo em estruturas de Frames. Na criação do frame, o usuário deve informar palavras chaves, por intermédio destas é possível inferir relacionamentos não criados pelo tutor.

Na literatura, encontram-se trabalhos que sugerem a utilização de softwares de *workflow* para a criação dos frames e de seus inter-relacionamentos. Buscando uma metodologia de maior consistência e familiaridade com o tutor, o WCC utiliza a estrutura hierárquica (padrão na maioria dos sistemas operacionais existentes no tratamento de arquivos, o que facilita a ambientação do tutor com a ferramenta), como representado na **Figura 7**.

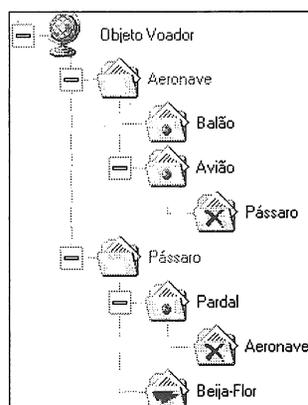


Figura 7 — Hierarquia da Estrutura de Frames representada na Figura 3

A grande vantagem desta abordagem é no tratamento de sistemas com níveis reduzidos de interação entre os objetos, em sistemas de alta complexidade, como os que envolvem derivação múltipla e relações de agregação, a estrutura como proposta mostra-se ineficiente. Desta forma, o WCC apresenta uma estrutura mista, permitindo ao usuário a interação no sistema hierárquico e a visualização das estruturas desejadas através de métodos de workflow. Cada objeto criado e suas relações irão compor uma estrutura definida no ambiente de edição de conteúdo Web.

O *Assistente de Edição* e o *Assistente de Geração de Hipermissão*, possibilitam ao tutor a criação do conteúdo Hipermissão, com a inserção de texto, hiperlinks e outras mídias (como som, imagens, animações, etc), bem como a criação de exercícios. Durante a edição, o tutor é assistido inteligentemente na criação do material, o WCC sugere a criação de links, metodologias de passagem e a inserção de exercícios em pontos estratégicos.

Os *Agentes*, são responsáveis pela análise do desempenho dos aprendizes e pela interação com os mesmos. Definidos os sistemas lógicos, pode-se definir a arquitetura do sistema, representada na **Figura 8**.

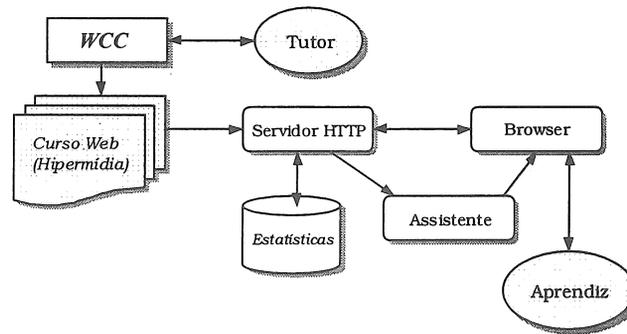


Figura 8 — Arquitetura do sistema WCC

Observa-se que o Aprendiz não interage diretamente com o WCC, mas sim com o conteúdo Hipermissão gerado e os assistentes do ambiente, a interação é realizada exclusivamente com o tutor. As estatísticas são geradas pela movimentação e ações do aluno no browser e atualizadas no banco de dados de Estatísticas, localizado no servidor. O Assistente, atua diretamente no browser, conforme as regras especificadas pelo tutor (e armazenadas no servidor) e as ações do aprendiz.

A **Figura 9** demonstra a interface entre o aprendiz e o curso, pode-se observar a representação a conotação do conteúdo hipermissão fora do ambiente Internet. Tal representação ressalta que o conteúdo não está armazenado na máquina do usuário, permitindo a geração de estatísticas sobre o comportamento do mesmo. O módulo assistente, obtém as regras definidas pelo tutor diretamente do servidor, e as ações do usuário são reportadas ao banco de dados de Estatísticas.

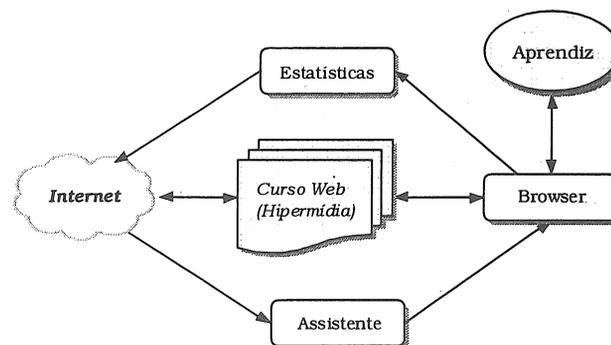


Figura 9 — Representação da Interface entre o Aprendiz e o curso na Web

3.1. Geração de Exercícios no WCC

Os exercícios podem ser criados pelo tutor em tempo de compilação, definindo pontos de passagem e situações desafio. Os exercícios são apresentados ao aprendiz na forma de um applet Java, com a questão e as opções disponíveis. O tutor pode criar exercícios de resposta única ou de múltipla escolha.

Internamente, o applet recebe as informações sobre a apresentação do exercício no formato CSV(campos separados por vírgula), organizados conforme a **Tabela 3**.

	<i>Título</i>	<i>Questão</i>	<i>Itens de Resposta</i>	<i>Índice(s) da(s) Resposta(s) Correta(s)</i>
<i>Tipo</i>	String	String	Número inteiro	Número real

Tabela 3 — Estrutura do BD de exercícios

Desta forma, a estrutura de geração de exercícios é adaptável às necessidades do tutor e independe do número de questões a ser inserida no ambiente. Na **Figura 10**, pode-se ver uma configuração típica de exercício implementada no WCC.

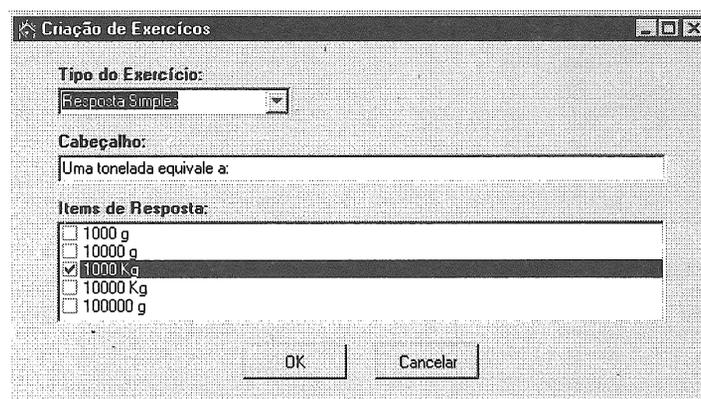


Figura 10 — Configuração de exercícios no WCC

O tutor seleciona o tipo do exercício (Resposta Simples ou múltipla escolha), o cabeçalho, os itens que comporão a resposta e a resposta correta, no caso de exercício de Resposta Simples, ou as Respostas corretas, para exercícios de múltipla escolha.

Na **Figura 11**, observa-se o resultado dos exercícios (resposta simples e múltipla escolha) no browser do aprendiz.

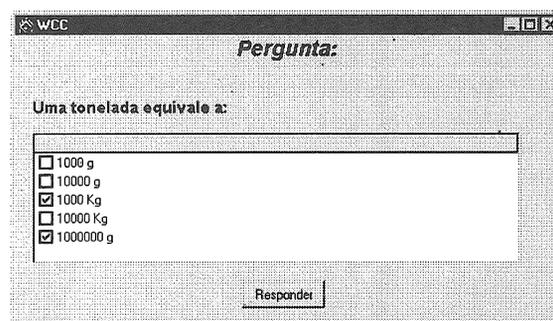
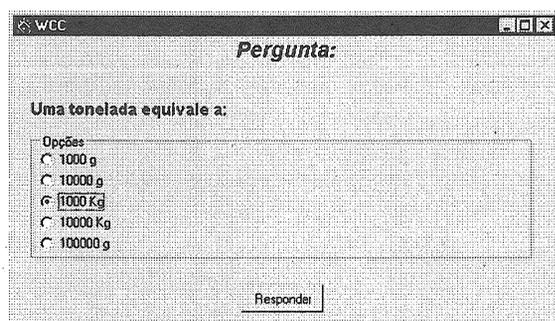


Figura 11 — Exercícios exibidos no micro do aprendiz

4. Agentes gerados pelo WCC

O WCC gerará junto com o material HiperMida agentes — implementados em Java —, responsáveis pelo acompanhamento do Aprendiz e a geração de análises estatísticas ao tutor.

O sistema multi-agente é formado por seis agentes, citados abaixo:

1. O **agente servidor** é responsável pela leitura e escrita de informações na base de dados do servidor.
2. O **agente estatística** recebe informações sobre o comportamento do aprendiz, efetua a análise e envia as informações para o agente servidor.

O **agente de aprendizagem** recebe informações sobre o comportamento do aprendiz e as compara com as informações da rede de frames, sugerindo exercícios e melhores caminhos de exploração ao aprendiz.

3. O **agente coletor** captura as ações do aprendiz e as envia para os outros assistentes.
4. O **agente de comunicação** fornece o endereço dos agentes, viabilizando a comunicação entre os mesmos.

5. O agente de análise é responsável pela geração de informações ao professor, formatando os dados do agente estatística, exibindo-os na forma gráfica.

A comunicação entre os agentes será realizada através do protocolo TCP/IP, e somente o agente servidor terá acesso a Base de dados armazenadas no servidor Web (rede de frames e base de dados estatística).

Desta forma, a arquitetura do sistema Multi-Agente fica representada na Figura 12.

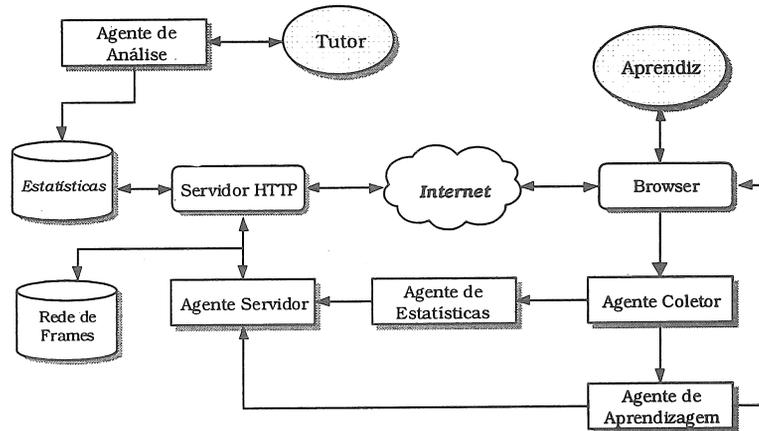


Figura 12 — Estrutura multi-agente gerada através do WCC

Os agentes apresentam dois objetivos básicos: alimentar o tutor com informações estatísticas sobre as ações dos aprendizes e o acompanhamento do processo de exploração por parte do aprendiz. Durante a exploração o Agente de Aprendizagem intervêm no micro do usuário, sugerindo desafios, sob a forma de exercícios, ou rotas alternativas de exploração do conteúdo. Caso o aprendiz encontre dificuldades na resposta de dado exercício, o agente sugere ao aprendiz a retomada de conceitos fundamentais.

5. Implementação de Frames segundo Programação Orientada a Objetos (OOP)

A teoria de frames exige que o inter-relacionamento dos objetos seja extremamente robusta e dinâmica. Para tanto é necessário definir uma estrutura de objetos que seja extensível e que possibilite o dinamismo característico. Escolheu-se como metodologia, a programação orientada a objetos, o que possibilita a implementação tenha as mesmas características da ferramenta que representa o conhecimento (Frames).

Para tanto, será utilizado o modelo de arrays para o armazenar os frames. As relações entre os frames serão representadas através de algoritmos de árvores de ponteiros, o que permite a rápida incursão pela estrutura e evita a repetição de nódulos na composição da estrutura. A Figura 13 apresenta o diagrama de classes para a estrutura de frames.

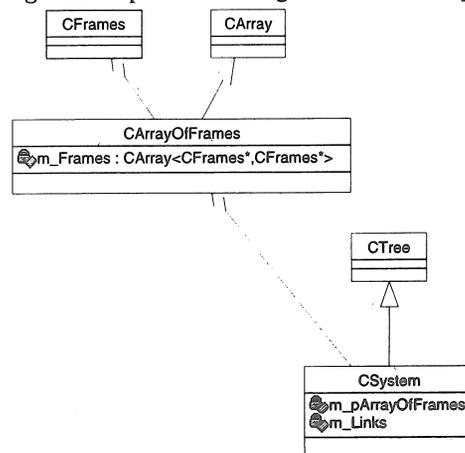


Figura 13 — Exemplo de Modelagem OOP para Frames

A classe CFrames representa os atributos necessários para identificar um frames, como as palavras chaves relacionadas e a intenção do frame. A classe CArray contém os métodos necessários para a representação das estruturas de array em memória. A classe CArrayOfFrames contém uma instância da classe CFrames e é derivada de CArray, possibilitando a criação das estruturas de array de frames em memória.

A classe CTree contém os métodos necessários para a representação de uma estrutura em árvore. A classe CSystem é derivada de CTree e implementa uma árvore de Frames e suas relações. Destaca-se que CSystem monta uma árvore de ponteiros para os elementos armazenados em CArrayOfFrames.

6. Considerações Finais

É válido mencionar que a qualidade do material gerado é um reflexo imediato da estrutura de Frames e, também, do conteúdo didático gerado pelo tutor. O projeto de Hiperdocumentos é uma atividade complexa e envolve muitas características próprias [MAR92]. Desta forma, o grau de envolvimento do tutor com a ferramenta e a preocupação do mesmo com a qualidade do material hipermédia é primordial para o desenvolvimento de um conteúdo de qualidade.

A possibilidade de geração de exercícios e a conseqüente criação de pontos de passagem, cria um ambiente de constantes desafios para o aprendiz. Desta forma, o aprendiz estará atuando no seu aprendizado, buscando novos conteúdos ou aprofundando-se em outros, conforme seus interesses individuais. A implementação de exercícios, possibilita que o tutor tenha uma ferramenta eficaz para avaliar seu conteúdo. Possibilitando assim um aprimoramento contínuo na qualidade do material hipermédia.

Cabe ao tutor fornecer caminhos alternativos para que o aprendiz possa explorar com maior profundidade temas de seu interesse. O tutor pode identificar pontos falhos ou faltantes no ambiente através do assistente de estatísticas. Somente através de um alto nível de interação do tutor com o ambiente de aprendizado, gerando melhorias contínuas no conteúdo, torna possível a motivação ininterrupta dos aprendizes.

O WCC possui, além das ferramentas e assistentes apresentados, um editor HTML visual, permitindo a edição do conteúdo didático de forma similar a um editor de texto. Além disso, existe um publicador de conteúdo na Web, diminuindo sensivelmente a necessidade de conhecimentos técnicos pertinentes a servidores HTTP por parte do tutor.

7. Referências Bibliográficas

- [BIT97] Bittencourt, Guilherme. *Representação de Conhecimento: da metafísica aos programas*. Departamento de Automação e Sistemas. Universidade Federal de Santa Catarina: 1997.
- [JAQ98] Jaques, Patrícia A. ; Oliveira, Flávio M. *Agentes de Software para Análise das Interações em um ambiente de Ensino à Distância*. Simpósio Brasileiro de Informática na Educação, 8., anais..., 1998.
- [MAR92] Martin, J. *Hiperdocumentos e como Criá-los*. Rio de Janeiro, Campus: 1992.
- [MES98] Mesquita, Lellis M.; Silva, Wagner Teixeira da . *Fundamentos de Software Educacional*. Brasília : UNB, 1998.
- [SCH94] Schank, Roger C. *Active learning through multimedia*. Multimedia IEEE, vº 1: 1994.
- [VIG84] Vigotsky, L. S. *A Formação Social da Mente*. Livraria Martins Fontes Editora LTDA, 1ª Edição brasileira: 1984.