

**Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação**

**Relatório de Estudos - MAC 5701
Prof. Yoshiko Wakabayashi**

Sistema de Aprendizagem pela Web baseado em plug-ins.

Aluna: Janine Gomes Moura
NºUSP:3163317
janine@ime.usp.br
Orientador: Prof. Dr. Leônidas de Oliveira Brandão

Índice

1. Introdução	3
1.1 Motivação	3
2. A Arquitetura do sistema	5
2.1 Camada de dados	6
2.2 Camada de Tarefas.....	8
2.2.1 Adaptação	9
2.2.2 Autoria	10
2.3 Camada de Comunicação	12
2.4 Camada de Interação	12
3 Os Plug-ins.....	13
4 iGeom: Um Software de Auxílio à Educação à Distância	14
5. MAC 118 ON-LINE	16
5.1 Criação de componentes	17
5.2 Consulta de relatórios	20
5.3 Classificação dos links	20
5.4 O Sistema para o Aluno	21
6. Trabalho futuro	23
7. Bibliografia.....	25

Sistema de Aprendizagem pela Web baseado em plug-ins

1. Introdução

Este trabalho descreve os estudos realizados no primeiro semestre de 2004 para a disciplina MAC5701 – Tópicos em Ciência da Computação. A área abordada neste trabalho é Informática na Educação e este trabalho fará parte da minha dissertação de mestrado.

1.1 Motivação

Um problema enfrentado por professores que ministram disciplinas em turmas grandes é conseguir oferecer um atendimento personalizado ao aluno ou simplesmente conseguir corrigir rapidamente os trabalhos dos alunos. Isso ocorre tanto no ensino presencial quando no ensino a distância.

Neste trabalho apresentaremos um sistema que estamos desenvolvendo e que objetiva ajudar a resolver este problema de atendimento personalizado ao aluno. Ele é baseado na Internet e pode ser empregado em cursos dados à distância ou em presenciais.

O desenvolvimento desse sistema torna possível oferecer uma grande quantidade de informação, com maior qualidade para o aprendiz, devido principalmente aos seguintes fatores: promove a motivação; permite múltiplas visões de objetos dentro do ambiente, tornando disponíveis melhores explicações e resoluções de problemas; permite que o aprendiz imprima seu próprio ritmo de aprendizado; e possibilita a obtenção de mais informação através de material *on-line*.

Já apareceram vários ambientes hipermídia com estas características, sendo denominados de Sistemas Adaptativos Hipermídia (SAH), porém a maioria destes sistemas não possibilita a expansão e/ou modificação de seus recursos de modo simples. E devido as suas diferentes características podemos dividi-los em 2 categorias: os genéricos e os dedicados. Os genéricos funcionam para qualquer domínio, porém permitem pouca interatividade do conteúdo com o usuário diminuindo as possibilidades de oferecer dica/ajuda de acordo com estado de conhecimento do usuário. Os dedicados conseguem oferecer maior interatividade do conteúdo e um auxílio adequado às dificuldades do usuário, porém são construídos

apenas para um determinado domínio do conhecimento e não possuem uma forma simples de modificar as partes que compõe sistema.

O objetivo deste trabalho é apresentar um SAH, que permita sua utilização em diferentes domínios trocando-se apenas módulos de aprendizagem (*plugins Java*). Esse sistema oferecerá maior flexibilidade, interatividade do aprendiz com o conteúdo e permitirá ajudar o aprendiz considerando suas preferências e necessidades. Dessa forma, podemos oferecer a cada aluno um ensino individualizado considerando o seu estilo de navegação, seu nível de conhecimento e seu ritmo de aprendizagem.

Esse sistema está em desenvolvimento atualmente com um módulo para aprendizagem de Geometria, e está sendo utilizado em uma disciplina obrigatória para a turma de licenciatura em matemática do IME-USP.

2. A Arquitetura do sistema

Muitos dos Sistemas Adaptativos Hipermedias utilizados pela internet são baseados na arquitetura cliente/servidor como, por exemplo: ELM-ART [7], InterBook [8], TANGOW [9], AHA! [10], RATH [11], AHM [12] e AdaptWeb [13]. A principal característica destes sistemas é à parte Cliente requisitar uma operação e o Servidor cuida de sua análise e execução.

Para tornar o sistema mais flexível e tornar o conteúdo mais interativo apresentaremos uma arquitetura cliente/servidor sendo que a parte Cliente será equipada com plug-ins, aplicativos Java (applets) que fará boa parte do processamento associado com o domínio de aplicação, reduzindo assim as restrições da base de conhecimento do domínio.

Este aplicativo será distribuído de três camadas globais, que consiste em uma interface com o usuário, uma lógica do sistema e o acesso a banco de dados. A interface com o usuário será criada utilizando HTML, PHP e applets Java. A HTML é o mecanismo preferido para representar a interface com o usuário em sistemas onde a portabilidade é uma questão importante. Como a HTML é suportada por todos os navegadores, projetar a interface com o usuário para ser acessada por um navegador da web garante portabilidade entre todas as plataformas que têm navegadores. Utilizando a rede fornecida automaticamente pelo navegador, a interface com o usuário pode comunicar-se com a lógica do sistema na camada intermediária. A camada intermediária então pode acessar o banco de dados para manipular os dados.

Na arquitetura proposta, partes das operações computacionais são executadas no computador cliente diminuindo as chances de sobrecarga do Servidor. Além disso, como cada plug-in tem seu próprio analisador de interação, os diferentes domínios podem dispor de diferentes formas de avaliação, isto é pode-se usar diversos plug-ins, tornando o sistema independente do domínio e utilizável em diversos contextos. A visão mais detalhada desta arquitetura pode ser vista na figura 1.

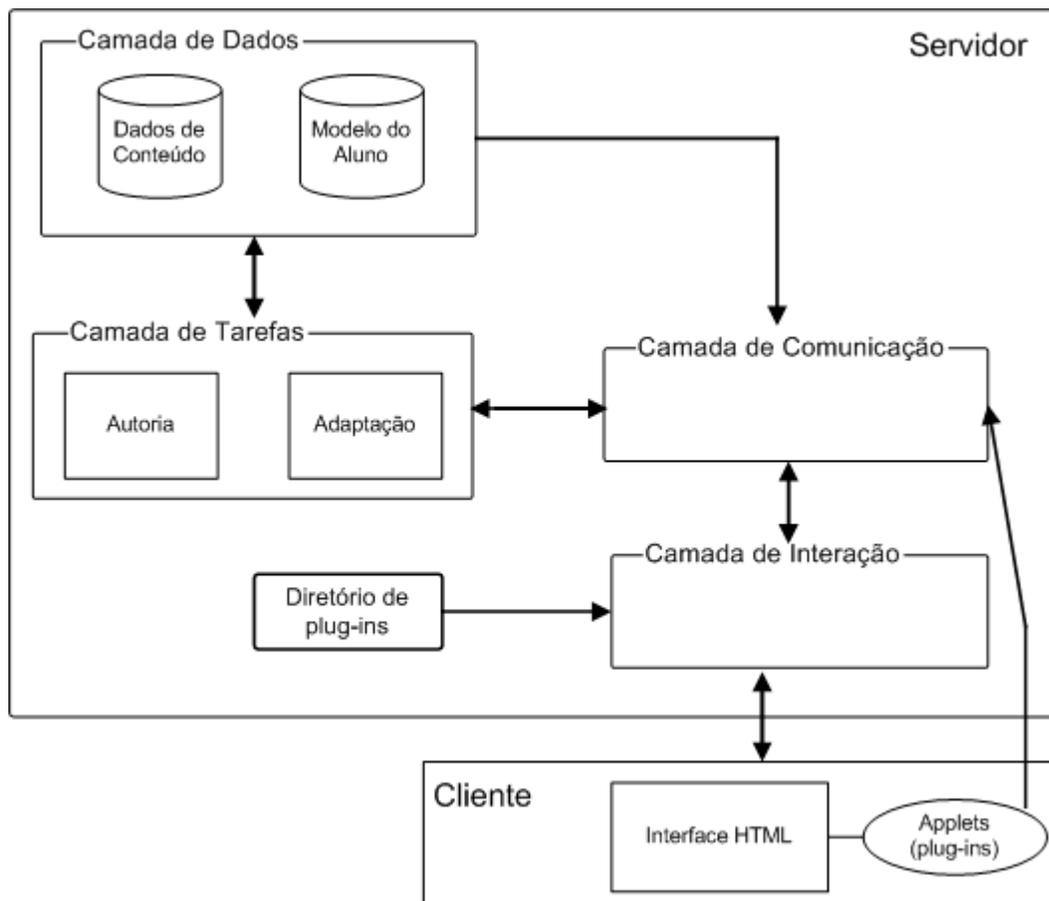


figura 1: Arquitetura do Sistema Adaptativo Hipermídia baseado em plug-ins.

2.1 Camada de dados

A camada de dados tem duas partes principais: os dados de conteúdo e o modelo do aluno.

2.1.1 Dados do conteúdo

Os dados de conteúdo são armazenados em **componentes** que podem ser cursos, aulas, tópicos, exercícios, textos e exemplos. E podem se associar a outros componentes de forma a prover o máximo de flexibilidade e reuso. Assim um curso pode possuir diversas aulas, cada aula diversos tópicos e cada tópico diversos exercícios, textos e exemplos (figura 2a e 2b). Além disso, um componente pode ser pré-requisito de outro(s) e assim por diante. Desta maneira, por exemplo, uma determinada aula cadastrada no sistema pode fazer parte de diversos curso.

Os dados de conteúdo por si só não formam a base de conhecimento do domínio, parte deste conhecimento pertence ao plug-in no qual o conteúdo está vinculado.

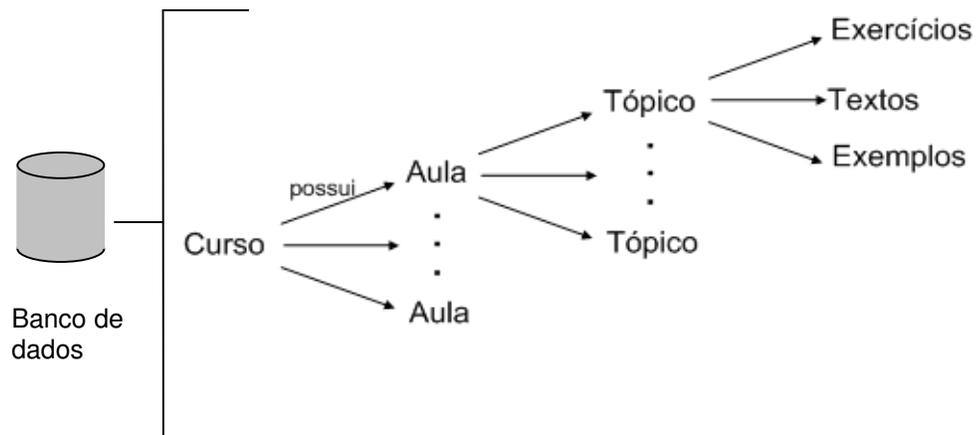


figura 2a: Relação entre os componentes.

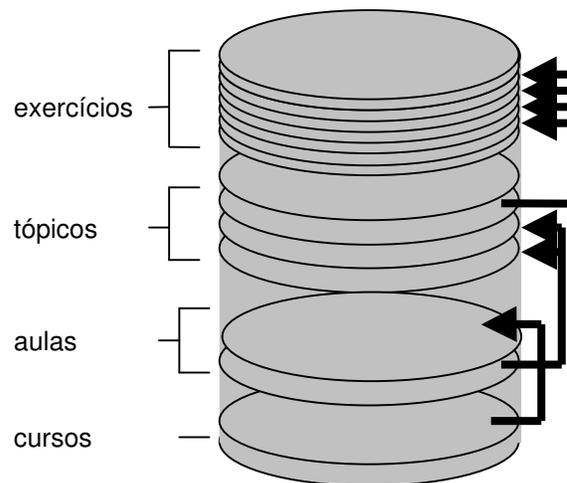


figura 2b: Relação entre os componentes.

2.1.2 Modelo do Aluno

O Modelo do Aluno representa o conhecimento e as características do aprendiz num determinado instante. Para isso o modelo deve armazenar informações que incluem os componentes que o aluno já visitou, dados sobre a identificação de cada aluno e armazenar seu caminho percorrido, ou seja, guardar as aulas, tópicos, exercícios, número de acessos, tempo que demorou em cada problema, trabalhos enviados, etc.. Dessa maneira é possível saber, a cada

conteúdo acessado, o que o aluno fez/acertou e atualizar o modelo do aluno no sistema para indicar qual o próximo caminho que o aluno deve/pode seguir.

Assim, o caminho a ser seguido pelo aluno dependerá do seu conhecimento prévio, do seu desempenho ao longo do curso. Desta forma o sistema apresenta a cada momento, o tópico mais adequado à aprendizagem, disponibilizando os link adequados à necessidade do aluno.

Ao entrar no sistema, o aluno terá acesso aos cursos em que está cadastrado (um mesmo aluno pode estar cadastrado em vários cursos), ao entrar em algum curso estarão acessíveis as aulas que foram disponíveis (para aquele aluno, naquele dia), além do aluno poder verificar notas, trabalhos entregues, lista de presença, entre outros. Após escolher uma aula o aluno terá acesso aos tópicos disponíveis para esta aula, cada tópico poderá conter um conjunto de exercícios e/ou textos explicativos. Depois de escolher um tópico o aluno deverá realizar o número mínimo de exercícios (testes, ou textos) pré-determinados (pelo professor responsável pelo curso) de acordo com seu desempenho.

No modelo no aluno, cada componente pode assumir um dos seguintes estados: completo, liberado, incompleto, pré-requisito, como relacionados abaixo.

Estado	
completo	Foram realizadas todas as tarefas determinadas para o componente.
liberado	Foram realizadas o número mínimo de tarefas determinadas para o componente, assim o aluno está apto para passar para o próximo componente.
incompleto	Ainda faltam realizar tarefas para o componente.
Pré-requisito	O componente não está liberado, possui pré-requisito

2.2 Camada de Tarefas

A camada de tarefas é responsável por criar e adaptar o conteúdo a ser repassado para o usuário. É ela que faz a conexão com a camada de dados e reflete a organização e apresentação do conteúdo levando em conta as diversas preferências do sistema e o modelo do aluno. Esta camada define o caminho do usuário dentro do conteúdo oferecido pelo sistema apresentando aulas, tópicos, textos, exemplo e exercícios dependendo do estado de conhecimento de cada

usuário e de suas últimas interações com o sistema, determinado conteúdo pode ou não estar disponível para o aluno. Esta camada está dividida em dois módulos: adaptação e autoria.

2.2.1 Adaptação

Os métodos de adaptação conforme a classificação dada por [1], [2] e [14], são a adaptação do conteúdo e a adaptação de navegação. A adaptação do conteúdo diz respeito à forma como o conteúdo será apresentado, enquanto a adaptação de navegação considera os caminhos possíveis dentro do conteúdo do sistema. O enfoque desta arquitetura é na adaptação de navegação, pois esta permite direcionar o usuário dentro do espaço hipermídia.

Como mostra [1], existem diversas formas de fazer uso da adaptação de navegação: guia direto, ordenação adaptativa, ocultação e anotação. O guia direto oferece uma ligação (link) para a próxima página cujo sistema considera mais adequada para o usuário de acordo com as preferências e outras informações presente no modelo do aluno. A ordenação adaptativa faz um arranjo de links colocando os mais relevantes no topo de acordo com um critério de avaliação baseado no modelo do aluno. A ocultação esconde o link para uma página tornando-a inacessível ao usuário quando o sistema acreditar que o conteúdo desta página é inapropriado considerando as informações do modelo do aluno e características do sistema. E a anotação modifica a forma de apresentar os links para algumas páginas, por exemplo, modificar a cor dos links de páginas já visitadas ou colocar um ícone ao lado de links para páginas onde as lições estão incompletas.

Segundo [14], existem benefícios e problemas na utilização de algumas destas formas de adaptação de navegação. Por exemplo, utilizando a ordenação adaptativa podemos apresentar links ao aprendiz cuja informação seja inadequada. Ou usando a ocultação podemos restringir demais o espaço de navegação do usuário, dando pouca liberdade para que este explore o conteúdo. É responsabilidade dos autores do conteúdo do sistema identificarem as melhores formas de adaptação que beneficiarão o conjunto usuário-objetivo-conteúdo.

Para oferecer as adaptações apresentadas nesta seção cada componente pode possuir pré-requisitos, ou seja, um exercício, tópico ou aula será disponibilizado ao usuário caso o sistema acredite que o mesmo possua os conhecimentos mínimos necessários ou a configuração do sistema permita a

visibilidade deste componente. Um componente pode ter como pré-requisito, nenhum ou vários outros componentes, sendo que quem definirá essas ligações será o criador deste componente.

2.2.2 Autoria

O sistema deve suportar pelo menos 4 tipos de usuários: administrador, professor, monitor e aluno.

Administrador

É de responsabilidade do administrador do sistema:

- Cadastrar os alunos e professores do ambiente em uma base de dados para somente os autorizados possa entrar no sistema.
- Cadastrar o nome de curso e definir quais professores terão acesso;
- Define as possibilidades de navegação de acordo com o curso e/ou professor;
- É o único que possui permissão para excluir algum componente da base de dados.

Professor

O professor pode interagir com o ambiente a seguinte maneira:

- Criar ou modificar os componentes de conteúdo (cursos, aulas, tópicos, exercícios, textos e exemplos). A proposta é que os componentes disponíveis no ambiente fiquem abertas para os professores além de utilizar, possam adicionar e/ou modificar componentes.
- Disponibilizar, no momento adequado, o componente que se encaixa para determinado conteúdo;
- Escolher algumas características do sistema, como: *layout*, posição dos links, formato dos relatórios, etc.
- Definir e agendar tarefas para os alunos, onde tarefas podem ser trabalhos ou exercícios;
- Acessando os dados de cada aluno para avaliar o grau de aprendizagem alcançada e se o objetivo programado está sendo atingido;

Este usuário tem papel fundamental na geração do conteúdo, pois os sucessos das atividades propostos e do sistema também dependem de suas

habilidades em propor exercícios e montar as aulas, sendo eles os responsáveis pelas escolhas de adaptação, criação e apresentação dos componentes para o aluno.

Como a arquitetura do sistema é baseada em plug-ins, a autoria de alguns componentes (exercícios, textos, exemplos) necessita que parte deste processo seja realizada pelos plug-ins. Cada componente pode ser editado e reeditado pelo professor de forma on-line, e caso o plug-in permita, também é possível a criação off-line de componentes (como é atualmente possível no protótipo existente com o iGeom, seção 5).

Os componentes criados são enviados para o servidor onde ficarão armazenados.

Monitor

O monitor pode interagir com o ambiente a seguinte maneira:

- Têm a permissão de testar os componentes de conteúdo (cursos, aulas, tópicos, exercícios, textos e exemplos);
- Corrigir tarefas realizadas pelos alunos;
- Visualizar os conteúdos resolvidos pelos alunos;
- Participar de forma cooperativa, na elaboração de conteúdos.

Aluno

A relação do aluno com o ambiente é:

- Antes de liberar um determinado conceito, o sistema verifica quais os conceitos que são de conhecimento do aluno se o aluno já conhece os conceitos pré-requisitos para o conteúdo que será apresentado;
- Os *links* (de aulas, tópicos, exercícios) são classificados de acordo com sua relevância para o aluno (pré-requisito, conhecimento atual);
- O aluno tem acesso somente às aulas programadas para o curso;
- Para cada aula o aluno tem acesso aos tópicos e exercícios programados ;
- Resolução dos exercícios programados pelo tópico;
- Refazer os exercícios até serem aceitos pelo sistema;
- Verificar tempo de solução nos exercícios efetuados;

Um aluno conectado ao Servidor pode, por exemplo, pegar um exercício, resolve-lo e solicitar sua correção. Ao fazer esta requisição o plug-in irá avaliar a solução do aluno e enviar o resultado ao Servidor para que este responda de forma apropriada (figura 3). Na seção 4 apresentaremos como o plug-in faz isso no domínio de Geometria.

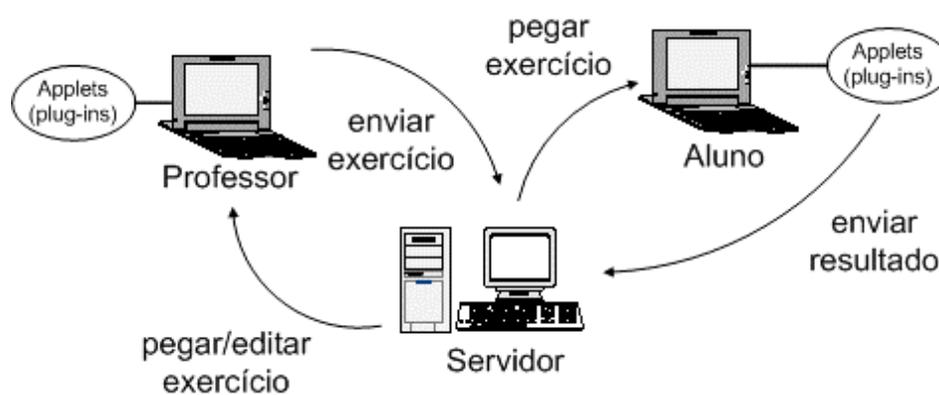


Figura 3: Criação e uso de um exercício.

2.3 Camada de Comunicação

A camada de comunicação é responsável por fazer a conexão entre as camadas do sistema, sendo que suas principais funções são:

- Enviar os dados da camada de tarefas para a camada de interação. Estas informações podem ser dados sobre o usuário, preferências do sistema, adaptações de interface, dentre outras.
- Receber os dados da camada de interação e atualizar as informações presentes na camada de dados. Assim, a cada interação do usuário será possível atualizar o modelo do aluno ou inserir um componente nos dados de conteúdo.
- Notificar a camada de tarefas sobre o recebimento de dados da camada de interação e a modificação da informação presente na camada de dados.

2.4 Camada de Interação

A camada de interação tem como principal função acoplar os plug-ins e gerar o código HTML que deverá ser interpretado pelo navegador do usuário. O código HTML é gerado a partir das informações recebidas da camada de tarefas (intermediada pela camada de comunicação) que indicam quais os plug-ins pertencentes a está página HTML, quais são os componentes e links a serem apresentados e quais as preferências do sistema.

Ao utilizar a adaptação em conjunto com os plug-ins podemos explorar dados em diferentes formas de mídia (sons, vídeos, textos e animações) aumentando a capacidade de percepção e memorização, pois trabalham com a informação dinâmica, a associação multissensorial, a interação com o usuário, a flexibilidade e a experimentação. Isto resulta numa forma lógica e interativa de apresentar o conteúdo o que facilita o usuário a assimilar a informação desejada de modo motivador e eficiente [15].

Um exemplo da interação do usuário com os plug-ins que estamos particularmente interessados são as resoluções de exercícios. O plug-in deve dispor de avaliadores automáticos que quando executados analisam a resposta do aluno e devolvem o resultado para o Servidor modificando as informações pertencentes à camada de dados e criando o efeito de adaptação.

3 Os Plug-ins

Os plug-ins estão diretamente relacionados com o domínio de aplicação, sendo responsáveis pela avaliação das interações do usuário e pela maior interatividade com o sistema. Cada plug-in é um programa completo e independente do sistema, podendo ser construído e acoplado a qualquer momento pelo administrador do sistema.

Como parte do processamento das interações com o usuário é efetuada na própria máquina Cliente pelos plug-ins, estes devem se comportar da mesma maneira independentemente da plataforma onde o usuário se encontra. Essa é a razão pela qual os plug-ins são desenvolvidos utilizando a linguagem Java (applet).

O desenvolvimento de um plug-in deve seguir a estrutura formada pelos seguintes módulos:

- **Interface:** Para desenvolver a interface dos plug-ins devemos oferecer um design “amigável” que interaja com a interface do sistema em execução.
- **Adaptação:** Os plug-ins também precisam se adaptar ao receberem informações do Servidor. Esta adaptação pode modificar a forma de correção de um exercício, atualizar a interface mostrando/escondendo algumas das funcionalidades do plug-in ou apresentar o conteúdo de forma diferenciada.
- **Núcleo:** O núcleo deve ser o responsável por gerenciar o plug-in. Será ele que tomará as decisões pertinentes a cada tipo de interação com o usuário,

incluindo a apresentação do conteúdo, interação do sistema com o usuário, animações, avaliações, comunicação entre módulos, entre outras.

- **Avaliador:** Ao final da interação do usuário com o plug-in o avaliador irá analisar o resultado desta interação e devolverá o resultado para o sistema em execução. Por exemplo, o avaliador poderia corrigir o exercício que um aluno fez e retornar o resultado para o sistema (correto, incompleto, incorreto).

- **Comunicação:** Todo plug-in deve ter um módulo de comunicação que: receberá as informações do sistema repassando-a para os outros módulos e enviará para o sistema o resultado da avaliação da interação do usuário.

- **Autoria:** O uso dos plug-ins é viável graças ao seu sistema de autoria que permite criar/editar exercícios, textos e outras formas de apresentação do conteúdo. O sistema de autoria deve permitir a criação/edição on-line, mas também pode possuir uma forma off-line que permita a um professor criar um componente em sua máquina local para depois envia-lo ao Servidor.

4 iGeom: Um Software de Auxílio à Educação à Distância

Com a finalidade de oferecer um programa gratuito no apoio ao ensino de Geometria e de Matemática está em desenvolvimento a plataforma iGeom apresentada em [15] e disponível no endereço: <http://www.matematica.br/igeom>. Esta é uma plataforma de Geometria Dinâmica, cujo principal mérito didático é propiciar um aprendizado geométrico, onde conjecturas são feitas a partir de experimentações e criações de objetos geométricos e a partir do retorno gráfico podemos introduzir o conceito matemático dos objetos, resultando no desenvolvimento natural do processo de argumentação e dedução.

No ensino tradicional (no qual o aluno apenas “ouve” e algumas vezes “vê”), o aprendiz não é incentivado a ter uma postura investigativa (ativa), não sendo desafiado a construir seu próprio conhecimento. Em uma aula de matemática tradicional o professor enuncia conceitos, definições e propriedades que, muitas vezes, são apenas memorizados e futuramente reproduzidos pelo aluno sem sua devida compreensão.

Em contrapartida, a geometria dinâmica nos oferece uma nova proposta que visa explorar os mesmos conceitos da geometria clássica, porém, através de um programa interativo disponibilizando representações gráficas de objetos geométricos que aproximam o objeto material na tela do computador (desenho) ao objeto teórico

(figura). Isso favorece o desenvolvimento de uma leitura geométrica dos desenhos por parte do aprendiz, contornando, uma das dificuldades do ensino da Geometria [16] (figura 4).

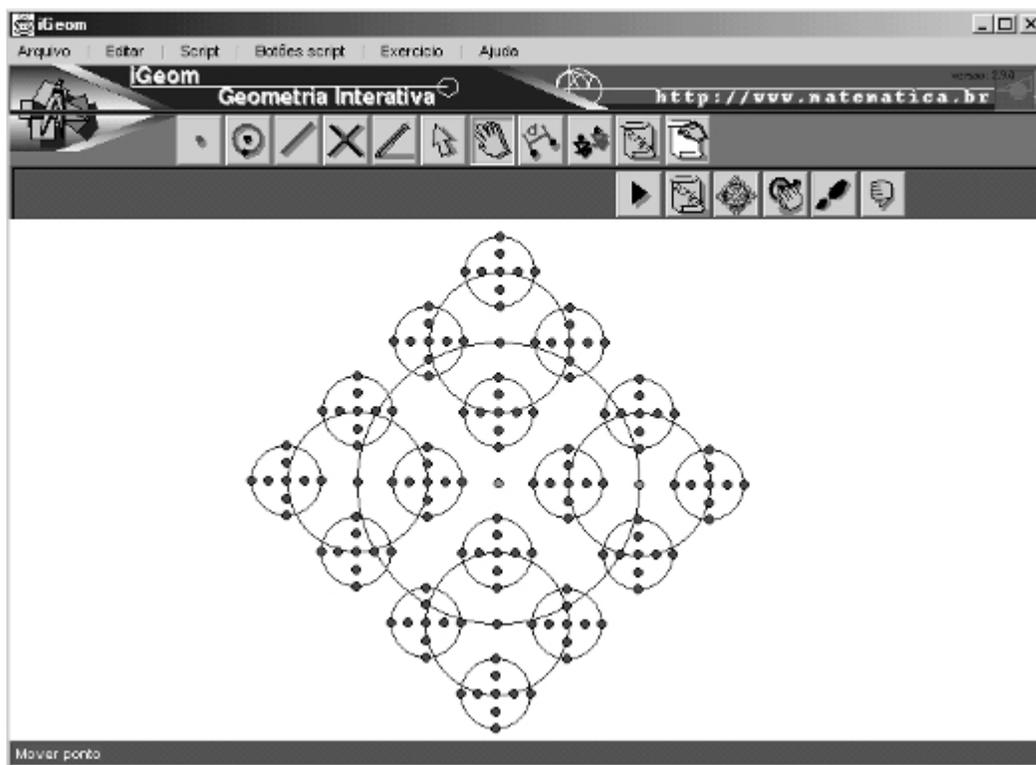


Figura 4: Fractais no programa de Geometria Dinâmica, iGeom.

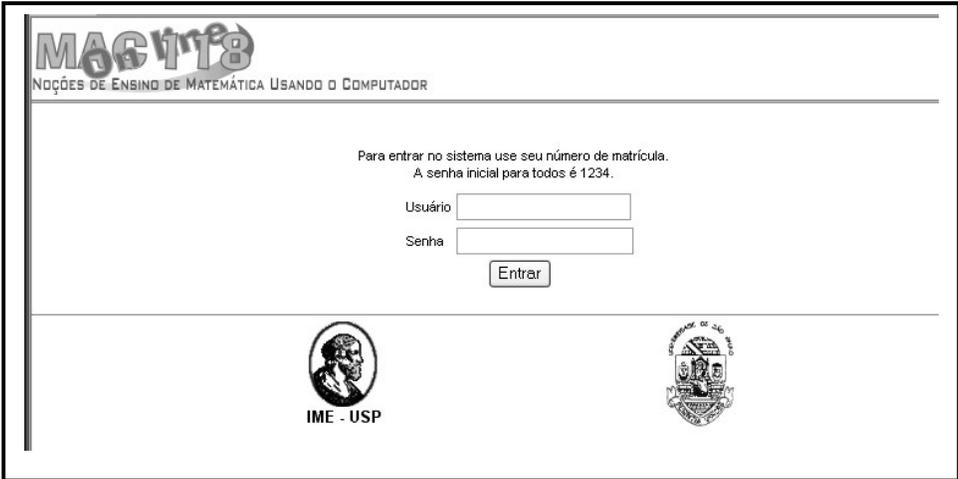
Os principais benefícios e aplicações de um sistema computacional de Geometria Dinâmica segundo apresentado em [17] são: a prova de teoremas, a precisão e visualização, a explorações e descobertas, as transformações e lugares geométricos e, por fim, a simulação de micro-mundos.

Atualmente existem vários programas de Geometria Dinâmica como o Cabri [18], o C.A.R. [19], o Cinderella [20], o Geometer's Sketchpad [21] e o Tabulae [22]. Apesar da maioria dos programas citados permitir criar applets para inserir em páginas Web, nenhum deles tem todas as características que o iGeom pode oferecer, como por exemplo, a comunicação com o servidor, à flexibilização das ferramentas e da interface (ativar/desativar funcionalidades), retornar a avaliação da interação com o usuário, correção automática de exercícios, criação de macros e a execução do programa em qualquer sistema operacional que possua pelo menos o Java 1.1.

Devido às características apresentadas acima, a plataforma iGeom pode ser facilmente incorporada em um SAH baseados na arquitetura apresentada na seção 2, tornando-se uma ferramenta poderosa de auxílio à educação à distância. Com o uso do iGeom como plug-in é possível que: (a) um professor possa criar/editar o conteúdo em sua própria máquina e depois envia-la para o Servidor; (b) fazer a adaptação da interface conforme as preferências e características do sistema e do usuário; (c) corrigir automaticamente os exercícios, oferecendo contra-exemplos ou correções quando possível; (d) comunicar ao SAH as avaliações da interação com o usuário; (e) se realize o trabalho colaborativo (em desenvolvimento), de tal forma que um usuário possa ver as construções de outros usuários em tempo real.

5. MAC 118 ON-LINE

Está em testes o protótipo de um sistema de aprendizagem pela web baseado na arquitetura apresentada na seção 2. O sistema tem sido desenvolvido utilizando a linguagem PHP [23], o gerenciador de banco de dados Mysql [24] e a linguagem Java [6] para a construção dos plug-ins e esta hospedado na rede IME. Este protótipo gera dinamicamente páginas HTML adaptadas para cada curso e usuário considerando as preferências sistema e o modelo do aluno. Este sistema está sendo utilizado por estudantes e professores na disciplina obrigatória "MAC0118- Noções de Ensino de Matemática usando o Computador" oferecida para o curso de licenciatura em matemática do Instituto de Matemática e Estatística da Universidade de São Paulo e o plug-in utilizado é o iGeom. (figura 5)



A screenshot of the login page for the 'MAC118 ON-LINE' system. The page has a header with the text 'MAC118' in a stylized font and 'NOÇÕES DE ENSINO DE MATEMÁTICA USANDO O COMPUTADOR' below it. The main content area contains the following text: 'Para entrar no sistema use seu número de matrícula. A senha inicial para todos é 1234.' Below this text are two input fields: 'Usuário' and 'Senha'. A button labeled 'Entrar' is positioned below the 'Senha' field. At the bottom of the page, there are two logos: on the left, the logo for 'IME - USP' featuring a portrait of a man; on the right, the official seal of the 'UNIVERSIDADE DE SÃO PAULO'.

figura 5:Tela inicial do “Sistema de mac118 on-line”.

Nesta disciplina estuda-se um método de ensino de Matemática que propõe um ensino baseado em *descobertas* e a dinâmica das aulas é fazer com que o licenciando vivencie este método. Um programa adequado permite que o aluno efetue testes e simulações com a exploração de programas computacionais visando compreender melhor o objeto estudado e também a verificar ou elaborar *conjecturas*.

O sistema desenvolvido para esta disciplina possui 4 tipos de usuário: administrador, professor, monitor e aluno e permite o desenvolvimento de componentes, isto é, aulas, tópicos, exercícios, exemplos e textos de forma on-line, e também permite a criação de exemplos, exercícios e textos de forma off-line utilizando o plug-in iGeom.

Os professores são responsáveis pela criação/edição dos componentes, escolha dos métodos de adaptação e ao acompanhamento dos alunos ativo no através de relatórios.

5.1 Criação de componentes

Para montar uma aula no sistema o professor deve clicar no menu "cadastrar aula" e:

- Digitar o título da aula;
- Tem a opção de escolher as aulas (entre as já cadastradas) que serão pré-requisitos para a aula ser inserida;
- Deve digitar o número mínimo de tópicos que necessariamente o aluno deverá realizar, ou seja, o número mínimo necessário para que o sistema acredite que o aluno sabe determinado conceito (figura 6);
- Por último deve escolher quais os tópicos (dos que já estão cadastrados) que farão parte da aula. A idéia é que os tópicos escolhidos estejam relacionados (mesmo conceito/assunto).

The screenshot shows the 'MAC 118' logo at the top left, with the subtitle 'NOÇÕES DE ENSINO DE MATEMÁTICA USANDO O COMPUTADOR'. Below the logo is a navigation bar with '[Home]' and '[logout]'. The main content area is titled 'Cadastramento de aula' and shows the user 'Administrador(a): Janine Gomes Mc'. The form includes a 'Título' text input field, a 'Pré requisito' dropdown menu with options '-', 'Aula Teste', 'Exercícios sobre retângulos e semelhanças', and 'Extra 1', a 'Número mínimo de tópicos para ir para outra aula:' text input field, and a 'Tópicos:' dropdown menu with options 'Básico', 'Estudo da função Seno', 'Exercícios complementares', and 'Exercícios para sala de aula'.

Figura 6. Cadastramento de aula.

Para montar um tópico no sistema o professor deve:

- Digitar o título do tópico;
- Tem a opção de escolher os tópicos (entre os já cadastrados) que serão pré-requisitos para o tópico ser liberado;
- Deve-se digitar o número mínimo de exercícios que necessariamente o aluno deverá realizar;
- Por último deve-se escolher quais os exercícios (dos que já estão cadastrados) que farão parte do tópico.

Atualmente só é possível cadastrar exercícios no sistema de forma off-line.

Para inserir um novo exercício no sistema, utilizando o plug-in IGeom, o professor deve:

- Digitar o título do exercício;
- Digitar o enunciado;
- Carregar o gabarito da máquina local, feito pelo professor no iGeom aplicativo;
- Escolher as ferramentas que devem estar disponíveis para o aluno. Com o iGeom é possível restringir as ferramentas que estarão disponíveis para cada exercício (figura 7).
- Digitar quais objetos o aluno deve dar como resposta;

Horário local: - 16h:34min:42s
 Atualização de aula - Data formato: dd/mm/aaaa hh:mm Administrador(a): Janine Gomes Moura

Aula	Ativar/Desativar	Início		Término	
Introdução ao Igeom	<input checked="" type="checkbox"/>				
Exercícios sobre retângulos e semelhanças	<input type="checkbox"/>				
Online 1	<input type="checkbox"/>				
Extra 1	<input type="checkbox"/>				
On line 3	<input type="checkbox"/>				
Online 4	<input type="checkbox"/>				
Extra 2	<input type="checkbox"/>				

Atualizar

Figura 8: Liberar ou travar aula pelo sistema.

5.2 Consulta de relatórios

É responsável pelo armazenamento e recuperação dos dados referentes aos exercícios e trabalhos definidas pelo professor ao aluno. Tem como objetivo a comunicação entre professor e aluno.

Espera-se que através destes relatórios possa-se estabelecer os pontos onde cada aluno necessitaria de maior aprofundamento no conteúdo estudado.

5.3 Classificação dos links

Para auxiliar a navegação do usuário, o sistema fornece classificação adaptativa de *links*. Nessa navegação adaptativa, o sistema faz uso de recursos visuais, tais como ícones, fontes e cores, para mostrar o tipo e o estado de cada *link*. Estes estados são definidos como:

- conhecido: o conteúdo do link já é de conhecimento do aluno;
- disponível: pronto para ver aprendido;
- indisponível: o aluno não pode acessar o conteúdo, necessita de pré-requisito.

Como o sistema possui conhecimento sobre quais são os conceitos pré-requisitos para cada componente, quando o aluno acessa uma página contendo conceitos que ele depende de outros que ainda não foram efetuados, o sistema avisa o aluno que a página contém pré-requisitos que ainda não foram completados e mostra *links* indicando onde esses conceitos são apresentados. Assim, o caminho

a ser seguido pelo aluno depende do seu conhecimento prévio, do seu desempenho ao longo do curso.

5.4 O Sistema para o aluno

Aos alunos, através de nome de acesso e senha, é possível interagir com o conteúdo de cada curso sendo que o sistema irá adaptá-lo de acordo as preferências do professor e com as ações e características do estudante.

Quando o aluno vai resolver o exercício o applet (iGeom) se abre com o enunciado do problema. A partir daí, o aluno (utilizando somente o mouse) tenta resolver o exercício na tela. Ao terminar os passos da resolução, o aluno deve selecionar os elementos gráficos que acredita pertencer à resposta e enviar os dados ao sistema.

A resposta enviada pelo aluno é comparada à solução armazenada para aquele exercício, ou seja, é verificado se a resposta do aluno coincide com a resposta do professor que cadastrou o exercício. Assim, é possível verificar se o aluno acertou ou errou, mesmo que a construção do exercício pelo aluno seja diferente da construção do professor (figura 9).

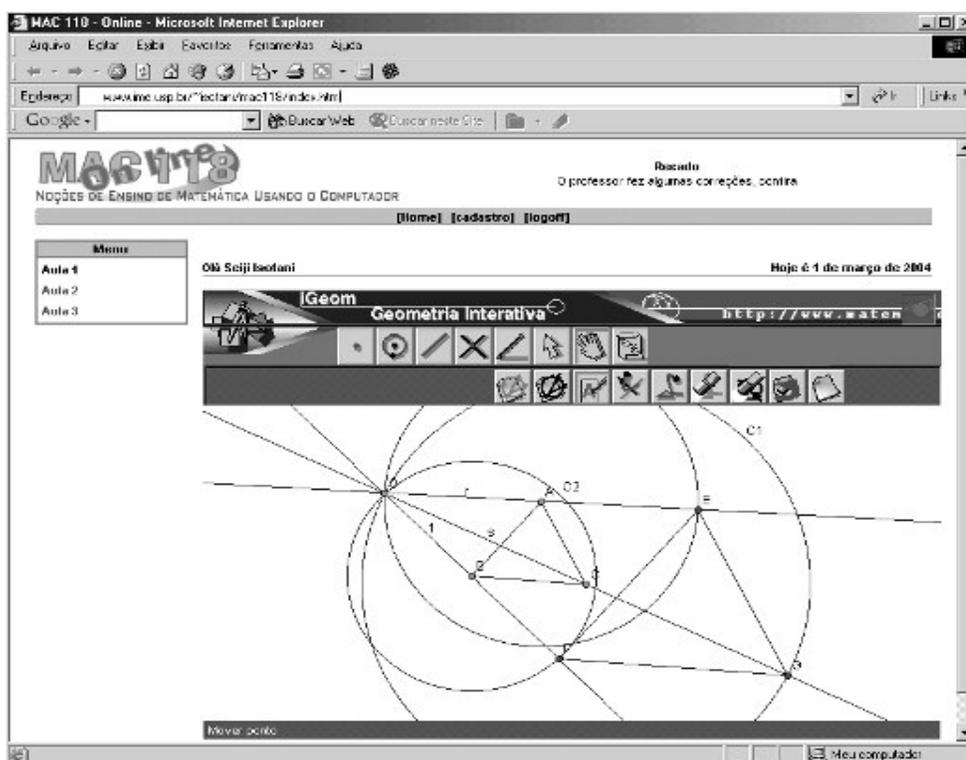


Figura 9: Resolução de um exercício no protótipo utilizando o plug-in iGeom.

Na correção, não importa qual tenha sido o desenvolvimento do exercício pelo aluno, nem se ele é similar ao gerado pelo docente, apenas se os elementos da resposta coincidem.

Quando o aluno faz um exercício e submete a resposta, ele pode obter de volta os seguintes resultados:

Resultado	
correto	A resposta está correta
funciona para alguns casos	Não funciona para todos os casos
incorreto	A resposta do aluno não bate com a resposta do professor
chute	A resposta bate, mas a construção não está correta
Pré-requisito	O exercício não está liberado, possui pré-requisito

Quando o aluno solicitar a correção do exercício, o plug-in irá verificar a resposta do aluno e se for julgada como incorreta o sistema indicará quais serão as páginas (que podem conter texto ou exercícios) mais apropriadas para ajudar o aluno a obter o conhecimento apropriado para resolução deste exercício.

Usando este sistema também é possível:

- Entrega de trabalhos individual e em grupo;

Cada professor cadastra o título do trabalho, o número de componentes e a data de entrega e o sistema cria um diretório onde serão guardados os trabalho submetidos pelo aluno. Cada aluno/grupo pode enviar o trabalho quantas vezes achar necessário (dentro do prazo), mas o sistema só armazena o último trabalho enviado.

- Correção on-line dos mesmos;

O professor/monitor pode carregar o trabalho na máquina local e também inserir nota e comentário on-line.

- Notícias;

O professor/monitor tem a opção de cadastrar datas de provas, exercícios e lembretes em geral.

- Lista de presença (que é cobrada pelos professores responsáveis pela disciplina).

6. Trabalho futuro

O objetivo deste trabalho é criar e implementar um ambiente aprendizagem pela web que pudesse servir de apoio ao aprendizado. Assim, foi necessário estruturar uma arquitetura que suportasse atividades de avaliação e que pudesse enriquecer as experiências de aprendizagem de cada aluno, individualmente ou em grupo.

Foi apresentado uma arquitetura que suporta o uso de plug-ins conferindo-lhe maior flexibilidade e interatividade tanto para os componentes quanto para o sistema. Além disso, os plug-ins podem ser facilmente inseridos e possibilitam a troca de domínio sem a necessidade de mudanças internas do sistema.

Foi desenvolvido um protótipo deste ambiente e um plug-in de Geometria Dinâmica. Com esta união foi possível simplificar a geração do conteúdo, viabilizar a criação dos componentes de forma On-line (via Web) ou off-line (criado na máquina do usuário e depois enviado para o servidor) e ofereceu aos alunos um conteúdo mais interativo e dinâmico no qual o aluno faz as construções diretamente nas páginas da Internet.

As duas técnicas de adaptação de conteúdo apresentadas na seção 2.2.1 foram utilizadas enfatizando-se a adaptação da navegação para guiar o estudante dentro do espaço hipermídia de acordo o modelo do aluno armazenado no sistema, além de levar em conta as preferências de adaptação que o professor sugerir.

Com o uso do ambiente protótipo criado, uma avaliação de sua usabilidade poderá ser feita, inclusive da parte de adaptatividade do aluno com o sistema.

Novos plug-ins para o ensino de matemática estão em fase de desenvolvimento. E uma nova extensão para o plug-in iGeom permitirá o trabalho colaborativo. Assim, um professor poderá movimentar em tempo real os objetos em sua tela de computador e todos os alunos habilitados acompanhem a movimentação, e caso o professor permita, poderão manipular estes objetos, aumentando ainda mais a interação entre usuários e sistema.

Em versões futuras deste protótipo a forma de armazenar os dados será modificada para que o sistema suporte o uso de XML. O armazenamento em banco de dados de todas as informações pertencente ao sistema acaba “misturando” acabam dificultando as suas atualizações ou modificações. Como proposto por [27], é possível utilizar o XML para armazenar os dados viabilizando a separação das

preferências do sistema (armazenadas em XML) e dos dados referentes aos usuários (armazenados em banco de dados).

7. Bibliografia

1. Brusilovsky, P.: Adaptive educational systems on the world-wide-web: a review of available technologies. *In: Proceedings of Workshop WWW-Based Tutoring at 4th International Conference on Intelligent Tutoring Systems (ITS'98)* (1998) 16-19.
2. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6 (2-3), (1996) 87-129.
3. Raad, H. and Causse, B.: Modeling of an adaptive hypermedia system based on active rules. *In: Proceedings of the 6th International Conference on Intelligent Tutoring Systems (ITS'02)* (2002) 149-157.
4. Isotani, S. and Brandão, L. O.: iMática, Interative environment for support of the teaching of mathematics through Internet. *In: VII Workshop on Informatics in the School, XXI Congress of the Brazilian Society of Computation* (2001) 533-543.
5. Brusilovsky, P.: Adaptive hypermedia: From intelligent tutoring systems to Web-based education. *In: Proceedings of the 5th International Conference on Intelligent Tutoring Systems (ITS'02)* (2000) 1-7.
6. See for JAVA the site <http://www.java.sun.com/>.
7. Brusilovsky, P., Schwarz, E. and Weber, G.: ELM-ART: An intelligent tutoring system on World Wide Web. *In: Proceedings of the 3th International Conference on Intelligent Tutoring Systems (ITS'96)* (1996) 261-269.
8. Brusilovsky, P., Eklund, J., and Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems* 30, 1-7 (1998) 291-300.
9. Carro, R. M., Pulido, E., and Rodrigues, P.: TANGOW: Task-based adaptive learner Guidance on the WWW. *Computer Science Report*, Eindhoven University of Technology, Eindhoven (1999) 49-57
10. De Bra, P. and Calvi, L.: AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia* 4 (1998) 115-139
11. Hockemeyer, C., Held, T., and Albert, D.: RATH - A relational adaptive tutoring hypertext WWW-environment based on knowledge space theory. *In: Alvegård, C. (ed.) Proceedings of CALISCE'98, 4th International Conference on Computer Aided Learning and Instruction in Science and Engineering*, Göteborg, Sweden (1998) 417-423

12. Silva, D. P., Durm, R. V., Duval, E. and Olivié, H.: Concepts and documents for adaptive educational hypermedia: a model and a prototype. *In: Computing Science Reports, Eindhoven University of Technology, Eindhoven (1998)* 35-43
13. Freitas, V., Marçal, V. P., Gasparini, I., Amaral, M. A., Proença Jr., M. L., Brunetto, M. A. C., Pimenta, M. S., Ribeiro, C. H. F., Pinto, J. V. L. and Palazzo, J. M. O.: AdaptWeb: an adaptive Web-based courseware. *In: Proceedings of International Conference on Information and Communication Technologies in Education (2002)* 131-134.
14. Hijikata, Y., Yoshida, T. and Nishida, S.: Adaptive hypermedia system for supporting information providers to direct users through hyperspace. *Transaction of IEEJ, 120C, No. 11, (2000)* 1720-1731,
15. Brandão, L. O. and Isotani, S.: A tool for teaching dynamic geometry on Internet : iGeom. *In: IX Workshop on Informatics in the School, XXIII Congress of the Brazilian Society of Computation (2003)* 1476-1487.
16. Braviano, G., Rodrigues, M. H. W. L.: Geometria Dinâmica: uma nova geometria?. *Revista do Professor de Matemática, Sociedade Bras. de Matemática, 49, (2002)* 22-26.
17. King, J. and Shattschneider, D.: Geometry turned on-dynamic software in learning, teaching and research, Washington: Mathematical Association of América (1997).
18. See for CABRI the site <http://www-cabri.imag.fr/>.
19. See for C.a.R. the site <http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/zirkel/index.html>.
20. Richter-Gebert, J. and Kortenkamp, U. H. The interactive geometry Software Cinderella, Book & CD Edition, Springer-Verlag, Heidelberg (1999) Springer, see also <http://www.cinderella.de/>.
21. See for Geometer's Sketchpad the site <http://www.keypress.com/sketchpad/>.
22. Barbastefano, R., Belfort, E. and Guimarães L. C.: A Toolkit for Synchronous Collaboration for Distance Teaching in Mathematics. *In: Proceeding of the 12th International Conference on College Teaching and Learning, (2001)* .
23. See for PHP the site <http://www.php.net/>.

24. See for the open source database MySQL the site <http://www.mysql.com/>.

25. Kellar, M., MacKay, B., Zhang, R., Watters, C., Kaufman, D., & Borwein, J.: Dynamic Composition of Math Lessons. *Educational Technology & Society*, 6 (4), (2003), 100-111.

26. Mavrikis, M., Maciocia, A., Abela, D. and Lee, J.: User-centred design and development of an applied Web-based ITS. *IEEE International Conference on Advanced Learning Technologies ICALT 2003* (2003) 364-365.

27. De Bra, P., Aerts, A., Berden, B., Lange, B., Rousseau, B., Santic, T. Smits, D. and Stash, N.: AHA! The adaptive hypermedia architecture. *In: Proceedings of the 14th ACM Conference on Hypertext and Hypermedia* (2003) 81-84.