

Desenvolvimento de Ferramentas no iGeom: Utilizando a Geometria Dinâmica no Ensino presencial e a Distância

SEIJI ISOTANI †

Orientador:

LEÔNIDAS DE OLIVEIRA BRANDÃO

MAC 5701 - Tópicos em Ciência da Computação

Profa. Responsável:

YOSHIKO WAKABAYASHI

USP - São Paulo

Junho de 2004

† Aluno de Mestrado em Ciência da Computação do IME-USP

Sumário

1	Introdução	3
1.1	Geometria Dinâmica	3
1.2	Recursos para o Ensino Presencial e a Distância no iGeom	5
1.3	Correção Automática	7
1.4	Aprendizagem Cooperativa	7
1.5	Interatividade e Educação à Distância	8
2	Correção Automática em Geometria Dinâmica	11
2.1	O Problema	12
2.2	Correção Numérica	15
2.3	O Algoritmo	16
2.3.1	Transformação e Avaliação	16
2.3.2	Instanciação e Validação	19
2.4	Corretor automático no iGeom	21
2.4.1	Autoria	22
2.4.2	Correção	23
2.4.3	Comunicação	24
3	Colaboração e Cooperação	27
3.1	Aprendizagem Cooperativa	27
3.2	O iGeom Cooperativo	29
3.2.1	Diálogo, Mediação e Comunicação	30
3.2.2	Espaços Compartilhados, Coordenação e Cooperação	31
4	Considerações Finais	33

Bibliografia

Capítulo 1

Introdução

Desde a década de 70 diversos sistemas de auxílio à aprendizagem tem sido desenvolvidos, alguns deles possuem explicitas ligações entre as teorias de aprendizado e o desenvolvimento dos sistemas de ensino por computador mostrando como um modelo de ensino pode guiar o desenvolvimento de programas educacionais mais conhecidos como CAI - computer assisted Instruction [Hannafin and Peck, 1988].

Os primeiros sistemas CAI eram baseados em textos, rígidos (não adaptáveis) e pouco “amigáveis” ao usuário [Barr and Feigenbaum, 1989, Silveira, 1996], porém com a incorporação de recursos patrocinadores da interatividade e de comunicação (hipertexto, chats, fóruns, *applets* e outros) o uso destes sistemas se tornaram muito mais eficazes. Com a interatividade e a comunicação podemos criar sistemas onde o processo de descoberta e a ajuda mútua façam parte da aprendizagem, criando um ambiente rico em situações nas quais os alunos podem explorar, testar conjecturas e compartilhar idéias.

Neste capítulo, introduziremos alguns conceitos da geometria dinâmica, da correção automática de exercícios de geometria, da aprendizagem colaborativa e da interatividade na educação a distância. Em particular, apresentaremos o iGeom, um programa de geometria dinâmica utilizado para os desenvolvimentos citados acima.

1.1 Geometria Dinâmica

O nome "Geometria Dinâmica"(GD) hoje é largamente utilizado para especificar a Geometria implementada em computador, a qual permite que objetos sejam movidos mantendo-se todos os vínculos estabelecidos inicialmente na construção. Este nome pode ser

melhor entendido como oposição à geometria tradicional de régua e compasso, que é "estática". Nesta última, após o aluno realizar uma construção, se ele desejar analisá-la com alguns dos objetos em outra disposição terá que construir um novo desenho, enquanto que na GD basta mover os objetos para a nova posição que o programa redesenhará toda a construção em sua nova configuração.

Em função desta possibilidade de alterar objetos preservando-se a construção, podemos dizer que a GD é uma geometria do tipo 1-construção, N-testes, enquanto a tradicional de régua e compasso é do tipo 1-construção, 1-teste [Brandão, 2004]. Esta é, para nós, a grande vantagem da GD sobre a geometria tradicional, pois permite que o aluno teste conjecturas e procure descobrir propriedades.

Nesta situação de experimentação, o aspecto dinâmico desencadeia um processo desafiador e interessante de ensino e aprendizagem [Isotani and Brandao, 2001]. As explorações e estratégias que vão se delineando ao longo das tentativas de solucionar o problema são similares às que acontecem no ambiente de pesquisa de um matemático profissional. Esta postura investigativa contribui para a formação de uma concepção sobre matemática diferente daquela construída, usualmente, ao longo da vida escolar [Gravina, 1996].

O uso da GD no ensino da Geometria traz boas possibilidades de mudança em uma área que vem sendo relegada ao segundo plano no ensino fundamental e médio, quando ensinada. De um lado, notamos problemas na forma tradicional de se ensinar geometria, como nota Gravina [Gravina, 1996]: "Os livros escolares iniciam o ensino de Geometria com definições, nem sempre claras, acompanhadas de desenhos bem particulares, os ditos desenhos prototípicos. Por exemplo, quadrados com lados paralelos às bordas da folha de papel, retângulos sempre com dois lados diferentes, altura em triângulos sempre acutângulos, entre outros. Isto leva os alunos a não reconhecerem desenhos destes mesmos objetos quando em outra situação. E mais, os alunos passam a acreditar que a posição relativa do desenho ou seu traçado particular façam parte das características do objeto, o que os leva a estabelecer desequilíbrios na formação dos conceitos. O aspecto de construção de objetos geométricos raramente é abordado. Dificilmente encontramos no livro escolar a instrução "construa", e no entanto, esta é uma das atividades que leva o aluno ao domínio de conceitos geométricos."

Por outro lado, temos o potencial interativo e aberto de um programa de GD. Como nota Valente [Valente, 1999], discutindo a linguagem Logo [Papert, 1999], uma "característica relevante da visão do computador como ferramenta é o ambiente aberto, ou

seja, o aluno é livre para propor e resolver qualquer projeto que tenha interesse". Do ponto de vista do aprendizado, também podemos notar vantagem da GD sobre a geometria estática. Usando o modelo de aprendizado de Geometria proposto pelos van Hiele [Crowley, 1998], que classificam os níveis cognitivos de Geometria em cinco: visualização, análise, dedução informal, dedução formal e rigor. Notamos que a Geometria Dinâmica pode ser bem empregada nos três primeiros níveis. Nestes níveis iniciais o estudante está começando a abstrair conceitos matemáticos e deste modo a experimentação pode contribuir muito. Como observa [Gravina, 1996], a Geometria Dinâmica evidencia uma nova abordagem ao aprendizado geométrico, onde conjecturas são feitas a partir da experimentação e criação de objetos geométricos. Deste modo, podemos introduzir o conceito matemático dos objetos a partir do retorno gráfico oferecido pelo programa de GD, surgindo naturalmente daí o processo de argumentação e dedução.

No ensino tradicional, o aluno apenas "ouve", não sendo 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. Segundo [Melo et al., 2000], se o aluno agir ativamente, modificando características de vários objetos matemáticos, ele aprenderá pesquisando, relacionando as modificações feitas, analisando e verificando o que ocorre genericamente.

Neste contexto, a geometria dinâmica nos oferece uma nova proposta que visa explorar os mesmos conceitos da geometria clássica, porém, através de um programa interativo [Rodrigues, 2002]. Assim é possível disponibilizar representações gráficas de objetos geométricos que aproximam o objeto material da tela do computador (desenho) ao objeto teórico (figura), favorecendo o desenvolvimento de uma leitura geométrica dos desenhos por parte do aluno, contornando, assim, uma das dificuldades do ensino da Geometria [Braviano and Rodrigues, 2002].

1.2 Recursos para o Ensino Presencial e a Distância no iGeom

O **iGeom: Geometria Interativa na Internet**, é um programa de Geometria Dinâmica que proporciona recursos facilitadores para o ensino e aprendizagem de Geometria. Além disso, o iGeom oferece ferramentas que auxiliam o professor na produção de material

didático e no acompanhamento de alunos.

A interface do programa iGeom foi desenvolvida com o intuito que o mesmo pudesse ser utilizado nas formas aplicativo e *applet*. A menos das opções de gravação, que não são permitidas em *applet* (portanto via Internet) por razões de segurança, todas as demais estão acessíveis a partir dos botões.

A versão atual deste programa permite realizar todas as operações básicas de Geometria Dinâmica, como por exemplo: criar objetos geométricos (como pontos, retas, semi-retas, segmentos e circunferências); opções de edição (esconder/mostrar, remover ou desfazer remoção, rastrear objetos,...); e opções de gravação/recuperação de arquivos em diferentes formatos (incluindo gerar imagens em PostScript e GIF). Um recurso implementado no iGeom e que é raro nos outros programas de Geometria Dinâmica é a possibilidade de fazer macro/script com recorrência (figura 1.1).

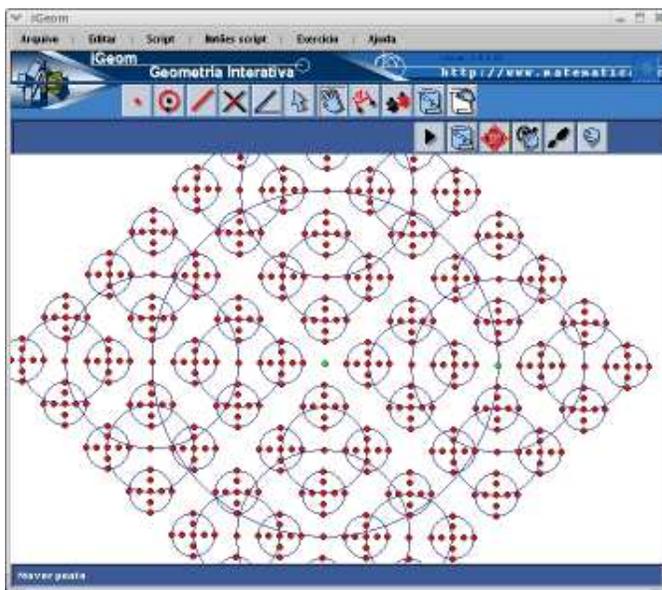


Figura 1.1: Figura construída utilizando a recorrência no programa iGeom.

Além das características anteriormente apresentadas, existem outros recursos que podem ser incorporados no iGeom para simplificar as tarefas do professor e ampliar as formas de interação com o aluno. Uma delas é a correção automática de exercícios e outra são os recursos de comunicação e cooperação.

Os desafios de permitir a comunicação também estão relacionados com o estudo autodidata, onde o aluno pode entrar a qualquer momento num sistema Web para estudar,

resolver exercícios e verificar seu desempenho. Mas para isso, são necessários mecanismos confiáveis para avaliação de respostas e meios de sugerir dicas ou contra-exemplos aos alunos.

Considerando as discussões apresentadas, os objetivos e desafios deste trabalho são criar e explorar recursos no iGeom que permitam a comunicação, a cooperação/colaboração, a correção automática de exercícios e a inclusão deste programa na Web. Alguns destes recursos como a correção automática e a inclusão do iGeom em sistemas Web já foram implementados e serão apresentados com mais detalhes no capítulo 2. Os recursos cooperativos no iGeom ainda estão em fase de estudo e serão abordados no capítulo 3.

1.3 Correção Automática

A correção automática em sistemas educacionais pela web existentes, em sua maioria, resumem-se na verificação das respostas de exercícios do tipo múltipla-escolha, preenchimento de lacunas e verdadeiro ou falso, como apresentado em [Gibson et al., 1995, Scapin, 1997, Abrão et al., 2004]. Porém este tipo de correção não permite a avaliação de uma questão dissertativa, inviabilizando o desenvolvimento de sistemas mais elaborados de ensino. O mesmo ocorre na utilização de exercícios dissertativos em geometria.

Em particular, quando buscamos corretores automáticos de exercícios dissertativos em geometria, precisamos capturar as múltiplas soluções e técnicas que podem ser utilizadas.

Para solucionar esta dificuldade será apresentado no capítulo 2 uma forma de corrigir problemas em geometria dinâmica utilizando um algoritmo heurístico (baseado na posição dos objetos geométricos) que faz comparações numéricas entre a solução dada pelo professor com a solução do aluno.

1.4 Aprendizagem Cooperativa

A aprendizagem cooperativa envolve metodologias pedagógicas que buscam promover a aprendizagem através da interação entre alunos que trabalham em uma determinada tarefa [Nitzke et al., 1999]. Através desta cooperação as pessoas poderão se comunicar visando a troca e o refinamento de idéias, permitindo uma possível melhora em seu aprendizado se comparada a sua atuação individual [Fuks et al., 2003].

Em um aprendizado em que o aluno constrói o seu conhecimento, o professor deve atuar como mediador e facilitador [Neto, 2002], incentivando as interações e a troca de idéias entre os alunos.

Existe atualmente uma área na fronteira entre a educação e a computação que tem como objetivo estudar sistemas educacionais baseados na cooperação conhecida como sistema de aprendizado colaborativo apoiado por computador (*CSCL - Computer Supported Collaborative Learning*). Esta área é considerada por muitos autores uma subdivisão da área de sistema de trabalho cooperativo apoiado por computador (*CSCW - Computer Supported Cooperative Work*), uma vez que, os sistemas CSCL também permitem atividades básicas de trabalho cooperativo só que associadas as atividades de ensino [Barros, 1994].

No capítulo 3 discutiremos propostas para o desenvolvimento de ferramentas de comunicação e interação que permitirão o uso do iGeom em atividades cooperativas dentro e fora da sala de aula.

1.5 Interatividade e Educação à Distância

Atualmente, o ensino baseado na Web é um dos campos em expansão na área da pesquisa e desenvolvimento em Computação [Brusilovsky, 1998], e seus benefícios atingem diversas áreas do conhecimento o que possibilita a disseminação de uma grande quantidade de informação, e muitas vezes, liberam os estudantes e professores dos limites de tempo e espaço que normalmente regem uma sala de aula.

Com o uso da internet e de suas tecnologias é possível desenvolver sistemas personalizados que tem a capacidade de se adaptar às características de cada estudante, guiando seus passos através do conteúdo hipermídia considerando o seu estilo de navegação, seu nível de conhecimento e seu ritmo de aprendizagem.

Devido a existência de *applets* Java (programas que podem ser embutidos em páginas da Internet) [Java,], o uso desta linguagem é uma das melhores alternativas para o desenvolvimento de programas interativos para Internet, pois além da portabilidade e da interatividade que os *applets* oferecem, podemos deixar parte do processamento no computador cliente. Sem esse recurso para distribuir o processamento, muito tempo seria gasto na comunicação cliente/servidor como ilustra as figuras 1.2 e 1.3.

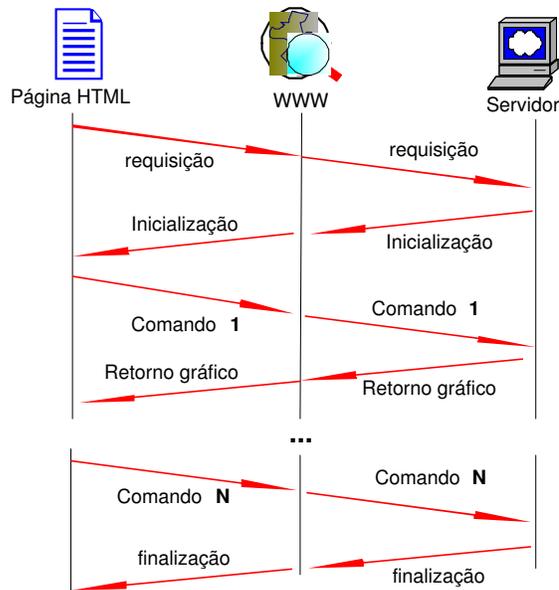


Figura 1.2: Execução de comandos via Web sem *Applets*

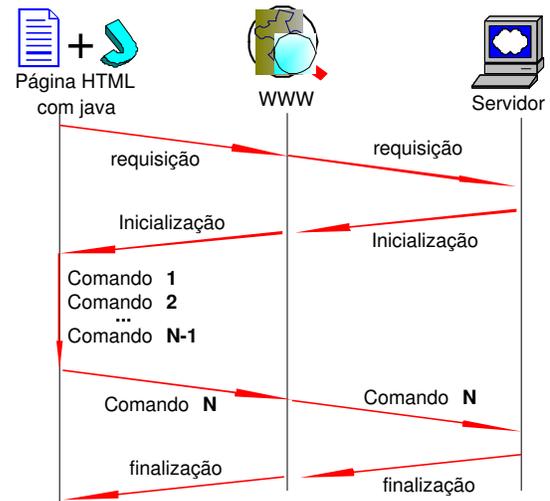


Figura 1.3: Execução de comandos via Web utilizando *Applets*

O desenvolvimento de sistemas utilizando *applets* pode melhorar o ambiente de aprendizado, pois pode: motivar o aluno; permitir múltiplas visões de objetos; e permitir que o aluno imprima seu próprio ritmo de aprendizado.

Devido as características de flexibilidade, interatividade e comunicação apresentadas pelo iGeom é possível incorpora-lo em sistemas Web, ampliando a interatividade, proporcionando ferramentas de autoria de exercícios e uma forma de avaliação automática de exercícios.

Capítulo 2

Correção Automática em Geometria Dinâmica

No processo de ensino-aprendizagem fazer exercícios é indispensável para desenvolver as habilidades e competências em um determinado assunto. Dessa forma, a resolução de exercícios pelos alunos e sua correção em tempo hábil por um professor é uma prática essencial para ajudá-los em seus estudos.

A correção de exercícios pode ser simplificada quando considerarmos questões do tipo múltipla-escolha, preenchimento de lacunas e verdadeiro ou falso. Além disso, o processo de correção e avaliação destas questões são agilizados devido à existência de corretores automáticos implementados nos sistemas computacionais de ensino-aprendizagem atuais [Gibson et al., 1995, Scapin, 1997, Abrão et al., 2004].

Em questões dissertativas (“abertas”) onde o aluno pode resolver o exercício de várias formas utilizando diversas estratégias, a correção manual torna-se uma tarefa trabalhosa e demorada. Da mesma forma, a correção de exercícios dissertativos de geometria (construções geométricas) não é uma tarefa simples, pois cada exercício pode ser resolvido de maneiras totalmente diferentes.

Ao considerarmos a necessidade de corrigir um grande número de exercícios dissertativos em turmas muito grandes, o esforço gasto pelo professor para avaliar, em tempo hábil, todos estes exercícios é consideravelmente alto.

Neste capítulo, apresentaremos as dificuldades encontradas na tentativa de automatizar a correção de exercícios (dissertativos) de geometria e através deste estudo mostraremos o desenvolvimento de um algoritmo que faz esta correção. Com o uso deste algoritmo no

programa iGeom conseguimos corrigir automaticamente os exercícios tanto localmente quanto via Internet, ampliando a utilidade do iGeom e viabilizando seu acoplamento em sistemas educacionais Web que atingem um grande número de usuários.

2.1 O Problema

Na aprendizagem de matemática, em geral, fazer exercícios¹ é indispensável para um bom aprendizado. Segundo o matemático Georgo Pólya [Pólya, 1975]: “*A Matemática é a arte de resolver problemas ... e para resolver problemas é preciso resolver problemas*”. Entretanto, em turmas muito grandes, esta tarefa indispensável requer um longo período de tempo do professor que será o responsável por fazer a correção destes exercícios. Neste contexto, sem o auxílio de um sistema que automatize esta tarefa ou a existência de uma equipe de monitores, é inviável para o professor cobrar (e corrigir) muitos exercícios. Essa situação é claramente observada na disciplina Cálculo Numérico oferecida para aproximadamente mil alunos dos cursos de Engenharia da Escola Politécnica da Universidade de São Paulo [Bevilacqua, 2002].

Para ilustrar a dificuldade de se corrigir um exercício de geometria, considere o problema 2.1:

Problema 2.1 *Dados dois pontos A e B, construir o ponto médio entre eles.*

Nas tabelas 2.1 e 2.2 são apresentadas duas construções diferentes que resolvem o problema 2.1. Além destas, uma infinidade de outras construções são possíveis, sendo elas minimais² ou não. Essa característica em exercícios de matemática/geometria dificulta ainda mais a correção automática, principalmente quando se está interessado em analisar as “idéias contidas” na solução do aluno.

Na geometria dinâmica (GD) podemos verificar visualmente a corretude de uma construção para “quaisquer” instâncias do problema, pois é possível movimentar os objetos da tela livremente. Por exemplo, para verificar se a construção do problema 2.1 é correta bastaria alterar as posições de A e B verificando se as propriedades esperadas se mantêm.

¹todos os exercícios que estamos apresentando são dissertativos

²construção minimal: qualquer construção na qual nenhum objeto pode ser removido sem comprometer o resultado.

Nosso objetivo ao fazer a correção automática é:

1. Detectar como corretas, soluções que utilizam diferentes técnicas, mesmo aquelas que o professor não tenha cogitado (veja o problema 2.2).
2. O resultado da correção deve ser imediato.
3. Se a solução é incorreta, mostrar um contra-exemplo.

Tabela 2.1: Algoritmo e visualização da solução padrão no iGeom

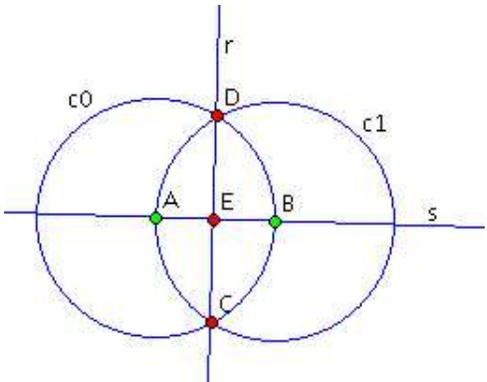
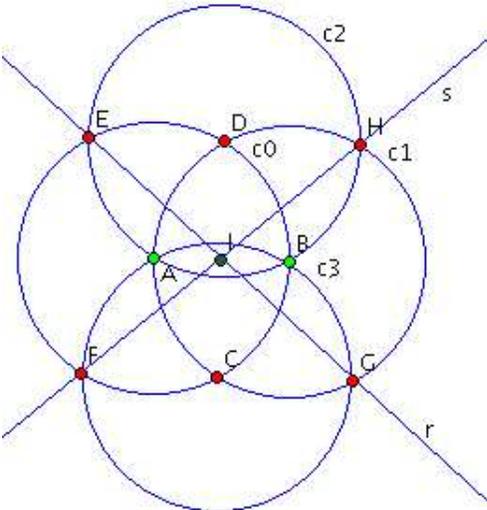
<pre> c0:= Circunferência (A,B) c1:= Circunferência (B,A) C:= Interseção (c0,c1) D:= Interseção (c1,c0) r:= Reta (D,C) s:= Reta (A,B) E:= Interseção (r, s) MarcaResposta(E) </pre>	
---	---

Tabela 2.2: Algoritmo e visualização da solução alternativa no iGeom

<pre> c0:= Circunferência (A,B) c1:= Circunferência (B,A) C:= Interseção (c0,c1) D:= Interseção (c1,c0) c2:= Circunferência (D,A) c3:= Circunferência (C,A) E:= Interseção (c0,c2) F:= Interseção (c0,c3) G:= Interseção (c1,c3) H:= Interseção (c1,c2) r:= Reta (E,G) s:= Reta (H,F) I:= Interseção (r,s) MarcaResposta (I) </pre>	
---	--

Problema 2.2 Dadas retas r e s paralelas e os pontos A e B (A “acima” da reta r e B “abaixo” da reta s), encontrar os pontos C e C' (sendo CC' um segmento ortogonal a r e s) que minimizam a soma das distâncias entre: A e C ; r e s ; e C' e B ($dist(A,C)+dist(r,s)+dist(C',B)$).

Existem ao menos duas soluções para o problema 2.2 que utilizam conceitos bastante diferentes. Numa delas, usa-se a translação do ponto B na direção GF (figura 2.1) e na outra utiliza-se semelhança entre triângulos (figura 2.2).

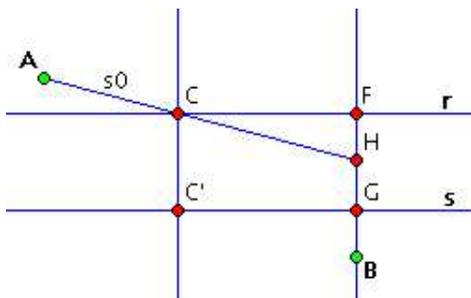


Figura 2.1: Solução para o problema 2.2 usando translação.

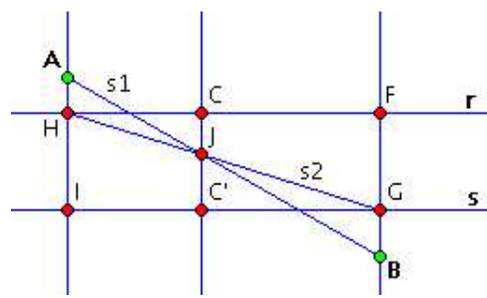


Figura 2.2: Solução para o problema 2.2 usando semelhança de triângulo.

A forma de correção que utilizamos é comparar se a solução dada é “equivalente” à uma solução gabarito. Na seção 2.2, apresentaremos a definição desta “equivalência” e também a idéia do algoritmo implementado.

Vale destacar que não estamos querendo **provar** que a solução é correta, mas apenas mostrar que a solução avaliada tem os mesmos resultados de uma outra solução previamente conhecida e, supostamente, correta (gabarito) para um problema.

Para provar que uma solução é correta teríamos que entrar em ramos da computação conhecidos como *Automated Reasoning* e *Theorem Proving*, como por exemplo, as técnicas utilizadas em [Gelernter et al., 1995, Gelernter, 1995, Matsuda and Vanlehn, 2004]. Estes trabalhos encontram-se dentro da área de *Inteligência Artificial* e apresentam métodos para provar teoremas em geometria.

2.2 Correção Numérica

Para se fazer a correção numérica é necessário definir o critério de **distância** entre pares de objetos geométricos de uma mesma família³, a partir de sua **descrição computacional**. A distância pode ser representada como uma função *dist* que recebe um par de objetos geométricos $(og1, og2) \in F_{og}$ e retorna um valor em \mathcal{R}_+ :

$$dist : (og1, og2) \longrightarrow \mathcal{R}_+ \quad (2.1)$$

A descrição computacional dos objetos, na maioria dos casos, é simples. Por exemplo: um ponto pode ser representado por um par (x, y) , onde x e y são as coordenadas do ponto; uma reta $(r(x) = ax + b)$, pode ser representada pelos seus coeficientes a e b ; e uma circunferência pode ser representada pelas coordenadas de seu centro e a medida do raio. Deste modo, representando as famílias dos pontos, retas e circunferências, respectivamente por, F_{og}^P , F_{og}^R e F_{og}^C , podemos definir *dist* conforme a equação 2.2.

$$dist(og1, og2) = \left\{ \begin{array}{ll} |x_{og1} - x_{og2}| + |y_{og1} - y_{og2}| & se (og1, og2) \in F_{og}^P \times F_{og}^P \\ |a_{og1} - a_{og2}| + |b_{og1} - b_{og2}| & se (og1, og2) \in F_{og}^R \times F_{og}^R \\ |x_{og1} - x_{og2}| + |y_{og1} - y_{og2}| + \\ + |r_{og1} - r_{og2}| & se (og1, og2) \in F_{og}^C \times F_{og}^C \end{array} \right\} \quad (2.2)$$

A partir da distância entre pares de objetos, podemos definir a distância entre pares de construções distintas, OG_p e OG_a , representadas por seus objetos, $(og_1^p, og_2^p \dots og_i^p)$ e $(og_1^a, og_2^a \dots og_j^a)$, conforme a equação 2.3.

$$dist(OG_p, OG_a) = \left\{ \sum_{i=1}^n dist(og_i^p - og_i^a) \right\} \quad (2.3)$$

Esta é apenas uma das possibilidades para medir distância entre construções. Uma generalização simples desta função é colocar coeficientes positivos para ponderar cada objeto.

Uma solução (construção geométrica) pode ser representada como uma função (equação 2.4) que recebe uma lista de objetos geométricos (entrada) e retorna uma outra lista de objetos geométricos (saída).

$$S : OG_i \longrightarrow OG_f \quad (2.4)$$

³Estas famílias podem ser pontos, retas, circunferências, dentre outras

Uma vez determinada a distância entre pares de listas de objetos e a representação de uma solução, podemos definir a noção de equivalência entre duas soluções como mostra a definição 2.3.

Definição 2.3 (Equivalência) *Sejam S_p e S_a duas construções sobre a mesma lista OG de objetos geométricos. Então S_a será considerada equivalente à S_p se, e somente se, para qualquer configuração OG_0 da lista OG , tivermos $dist(S_p(OG_0), S_a(OG_0)) = 0$*

Utilizando as definições desta seção, dizemos que **correção numérica** é o teste de equivalência entre duas soluções. Caso ambas sejam equivalentes então a correção retorna correta, caso contrário retorna como incorreta.

Este tipo de correção consegue identificar como correto as soluções apresentadas para o problema 2.2 (figura 2.1 e 2.2).

Observação 2.4 *Para simplificar a notação, nas próximas seções denotaremos a aplicação de uma solução para uma lista de objetos geométricos qualquer por S e não $S(OG)$.*

2.3 O Algoritmo

Baseado na definição de correção numérica apresentada na seção 2.2 desenvolvemos um algoritmo capaz de verificar a corretude de uma solução através de uma análise numérica dos objetos selecionados como resposta. Este algoritmo possui quatro passos principais: instanciação, transformação numérica, avaliação e validação.

As listas de objetos-resposta do professor e do aluno serão os dados de entrada do corretor e o resultado será um natural de 1 a 3 (1-correto, 2-correto, porém com alguma instância onde as soluções ficam diferentes, 3-incorreto). A execução deste algoritmo será apresentada nas duas sub-seções seguintes.

2.3.1 Transformação e Avaliação

A transformação numérica recebe um objeto geométrico e retorna uma lista de escalares que representam este objeto. Na maioria dos casos esta transformação é simples como apresentado na seção 2.2, porém alguns objetos como o polígono necessitam além da

transformação numérica, a ordenação dos pontos que os representam. Com esta lista de escalares podemos fazer a comparação entre objetos.

Assim, para analisar duas construções, o gabarito S_p do professor e a solução S_a do aluno, transformamos os objetos marcados como resposta em listas de escalares para então fazer a avaliação (figura 2.3). (figura 2.3).

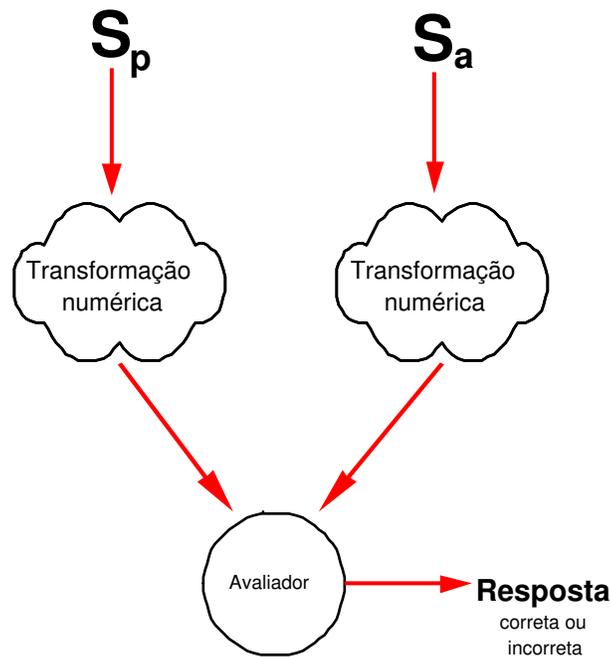


Figura 2.3: Correção - transformação numérica e avaliação.

A avaliação é feita em duas etapas. A primeira etapa consiste no mapeamento entre as listas e a segunda na comparação delas. A razão da primeira etapa é permitir que o aluno tenha a liberdade de fazer a marcação dos objetos-resposta em qualquer ordem. Por exemplo, caso a solução do professor seja representada pelos pontos A e B e a do aluno pelos pontos C e D, a etapa de mapeamento identificará se o ponto C equivale ao ponto A ou ao ponto B.

O mapeamento⁴ dos objetos de S_a para S_p é realizado comparando cada elemento de S_a com todos os elementos de S_p que pertençam a mesma família de objetos geométricos (F_{og}) e minimizando a distância entre eles.

Como cada objeto (og^x) é transformado em uma lista de escalares (l^x), podemos então

⁴Neste caso é uma função bijetora, mas por simplicidade escreveremos apenas “mapeamento”

generalizar a função $dist$ introduzida na equação 2.2, da seguinte forma (equação 2.5):

$$dist(og^p, og^a) = \left\{ \begin{array}{ll} \sum_{i=1}^n \|l_i^p - l_i^a\| & \text{se } og^p \text{ e } og^a \text{ pertencem a mesma família} \\ \infty & \text{caso contrário} \end{array} \right\} \quad (2.5)$$

Assim, um objeto de S_a será mapeado para um objeto de S_p se ambos pertencerem à mesma família de objetos geométricos e a distância entre eles for a menor possível em relação aos outros objetos de S_a .

A segunda etapa da avaliação consiste na comparação entre os pares de objetos geométricos mapeados (og_i^a, og_i^p). Nesta comparação, utilizamos uma versão relaxada da definição 2.3 (apresentada na seção 2.2), para levar em consideração as imprecisões numéricas ao se empregar diferentes soluções. Assim adotamos uma margem de erro ε que permite desconsiderar os erros numéricos gerados por imprecisões (definição 2.5).

Definição 2.5 (Quase Equivalência) *Seja S_p uma construção para um problema P , S_a uma outra construção para este problema e OG a lista de objetos geométricos de entrada. Então S_a será considerada quase equivalente à S_p se, e somente se, para qualquer configuração OG_0 da lista OG , tivermos $dist(S_p(OG_0), S_a(OG_0)) < \varepsilon$.*

A avaliação retornará verdadeiro se as soluções S_a e S_p forem “quase equivalentes” (definição 2.5), caso contrário retorna falso.

O pseudo-código da avaliação é apresentada na tabela 2.3. Note que nele é tratado apenas uma **instância** do exercício, considerando uma posição fixa para cada objeto de entrada. O problema em analisar apenas uma instância, aquela fornecida inicialmente, é a possibilidade do aluno apresentar uma solução próxima a do professor (sob o critério definido na equação 2.5) que, entretanto, seja “invariante” a mudanças na coordenada dos objetos de entrada.

Por exemplo, se no problema 2.1 o aluno tentar achar o ponto médio apenas colocando um ponto entre A e B de modo a, visualmente, aproximá-lo do ponto médio, apesar de difícil, é possível que o aluno consiga colocar o ponto na exata posição do ponto médio para a instância inicial. Esse problema poderia ser resolvido procurando na solução do aluno construções que conectam os objetos-resposta com os objetos de entrada do exercício⁵.

⁵A ligação entre os objetos-resposta com os objetos de entrada pode ser feita examinando o grafo de conexões entre objetos

Tabela 2.3: Pseudo-código do avaliador implementado no iGeom

1 .	Recebe duas listas S_a e S_p de objetos geométricos
2 .	Crie uma nova lista de objetos geométricos $S_t \leftarrow \emptyset$
3 .	Para cada elemento og_i^p da lista S_p
4 .	Para cada elemento og_j^a da lista S_a
5 .	Se og_i^p e og_j^a pertencem a mesma famílias de objetos geométricos então
6 .	Se $\text{dist}(og_i^p, og_j^a) < \text{MenorDistanciaEncontrada}$
7 .	ObjetoCorrespondente $\leftarrow og_j^a$
8 .	MenorDistanciaEncontrada $\leftarrow \text{dist}(og_i^p, og_j^a)$
9 .	Se Não existe ObjetoCorrespondente
10.	Retorne Falso
11.	Senão
12.	$S_t \leftarrow S_t \cup \text{ObjetoCorrespondente}$
13.	Para cada elemento og_i^p da lista S_p
14.	Seja og_j^t o primeiro elemento S_t
15.	Se Se $\text{dist}(og_i^p, og_j^t) < \varepsilon$ então
16.	Remova og_j^t de S_t
17.	Senão
18.	Retorne Falso
29.	Retorne Verdadeiro

Outro problema que surge quando avaliamos apenas uma instância pode ser identificado ao propor soluções ao problema 2.6.

Problema 2.6 *Dado dois pontos A e B, construir um triângulo equilátero ABC.*

Para resolver este problema são apresentadas duas soluções (figura 2.4 e 2.5). A primeira representa uma construção correta para o problema. A segunda é a construção de um triângulo isósceles, que coincidentemente tem um ponto C na posição tal que $\overline{AB} = \overline{AC} = \overline{BC}$. Neste caso, a construção do triângulo isósceles foi erroneamente utilizada para se construir um triângulo equilátero, porém mesmo utilizando o critério de “quase equivalência” e verificando as conexões entre os objetos de resposta e de entrada, não é possível diferenciá-las.

2.3.2 Instanciação e Validação

Uma maneira simples de detectar os erros apontados no final da seção 2.3.1 é criar um mecanismo que analise o exercício em várias instâncias (instanciação) e somente após um

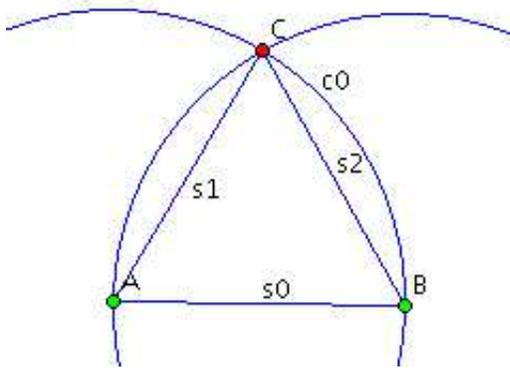


Figura 2.4: Construção do triângulo equilátero.

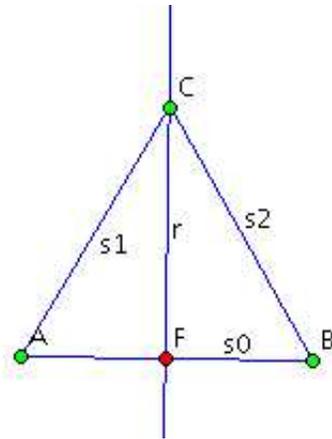


Figura 2.5: Construção do triângulo isósceles.

número considerável de avaliações o sistema retorna o resultado da correção (validação). A simulação deste procedimento pode ser visualizada na figura 2.6.

A cada iteração do algoritmo de correção automática (figura 2.6) os objetos geométricos da entrada do exercício devem ter sua posição alterada. Essa alteração é feita para cada objeto de entrada movimentando-os com paços “aleatórios” para tornar o processo de correção mais confiável.

Ao executarmos o exercício para suas diversas instâncias, verificamos a quantidade de resultados considerados corretos e incorretos. Quando o número de resultados corretos forem maiores que os incorretos então retornaremos que a solução do aluno é correta, caso contrário é considerada incorreta sendo armazenada uma instância na qual a solução foi avaliada como tal.

Esse algoritmo de correção automática foi implementado no iGeom (seção 2.4). Outros programas de geometria dinâmica como o C.a.R [Grothman, 1999] e o Cinderella [Richter-Gebert and Kortenkamp, 1999] possuem algoritmos semelhantes, sendo que a implementação no Cinderella é mais sofisticada, permitindo, por exemplo, que o usuário criador do exercício (professor) insira dicas que serão apresentadas na tela durante a resolução do mesmo.

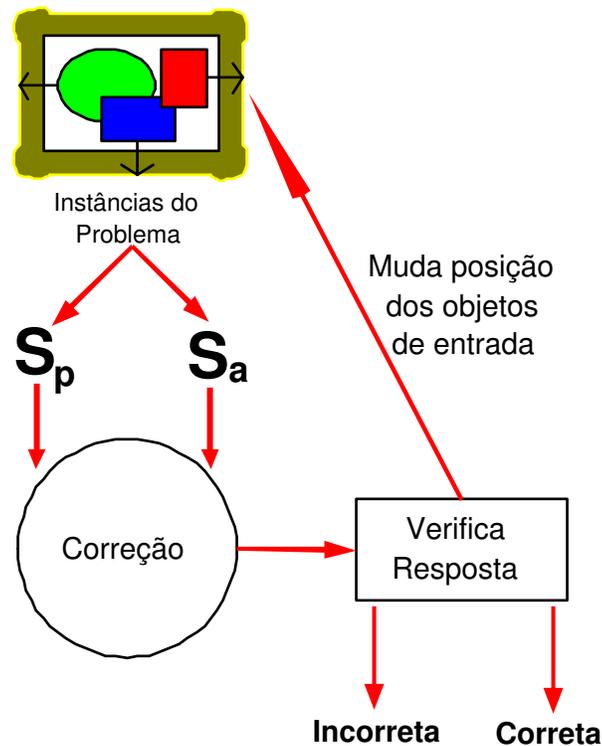


Figura 2.6: Correção Automática

2.4 Corretor automático no iGeom

A ferramenta de correção automática desenvolvida no iGeom possui três funcionalidades principais: autoria, correção e comunicação. E sua execução é possível em dois modos: localmente (via aplicativo) ou remotamente pela Internet (via *applet*).

Autoria: Neste modo, o professor pode criar exercícios determinando os objetos de entrada, enunciados, funcionalidades do sistema que estarão disponíveis (por exemplo, retornar a avaliação de um exercício e botões visíveis ao usuário), e um gabarito que será utilizado para posterior correção.

Correção: A correção utiliza o gabarito dado pelo professor fazendo a comparação dos objetos marcados como resposta no gabarito e na solução do aluno. O resultado da correção no modo aplicativo é apresentado para o usuário. No modo *applet* temos duas possibilidades: a primeira é enviar para um servidor previamente especificado o resultado da avaliação do exercício feito pelo aluno; a segunda é apresentar esta avaliação para o aluno da mesma maneira que o aplicativo o faz.

Comunicação: Caso o iGeom seja executado em modo *applet*, a cada tentativa de resolver o exercício será enviado uma mensagem para o servidor contendo alguns dados, tais como os objetos construídos, objetos marcados como solução e resultado da correção.

2.4.1 Autoria

Uma das características mais importantes na construção de sistemas de apoio ao ensino está no desenvolvimento de conteúdo, pois a dificuldade de se criar o conteúdo pode ser o fator determinante na utilização ou não do sistema, devido ao tempo necessário para a aprendizagem do sistema e produção do material.

O processo de criação de um exercício no iGeom possui quatro etapas: construção da solução, seleção dos objetos resposta, seleção dos objetos de entrada e salvar exercício. A interface de autoria do exercícios é bastante simples, contando com uma janela para separar os objetos de entrada e de saída, dentre aqueles construídos na área de desenho do iGeom (figura 2.4.1).

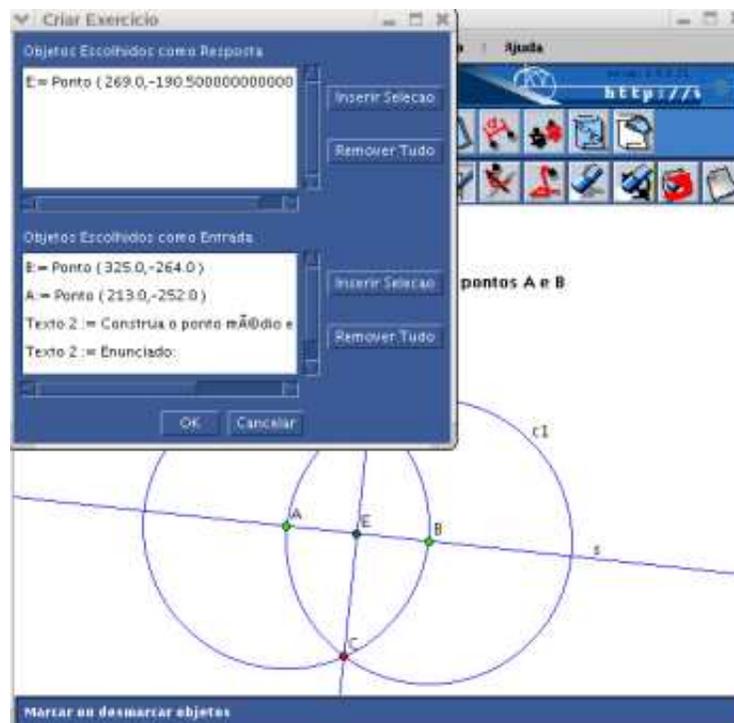


Figura 2.7: Janela para criação de exercícios no iGeom.

correção automática será iniciado. Se o aluno selecionar corretamente o número e tipos de objetos o algoritmos avaliador será disparado retornando correto, correto com restrições (correto na maioria das instâncias), incorreto com restrições (incorreto na maioria das instâncias) ou incorreto.

Estes resultados poderão ser apresentados ao usuário para notificá-lo do resultado da avaliação de seu exercício (figura 2.10). Isso viabiliza o uso desta ferramenta de forma auto-didata, pois o aluno pode fazer e refazer exercícios, obtendo a avaliação deste processo. Uma aplicação desse recurso são *sites* de conteúdo aberto ⁶ para ensino-aprendizagem de geometria.

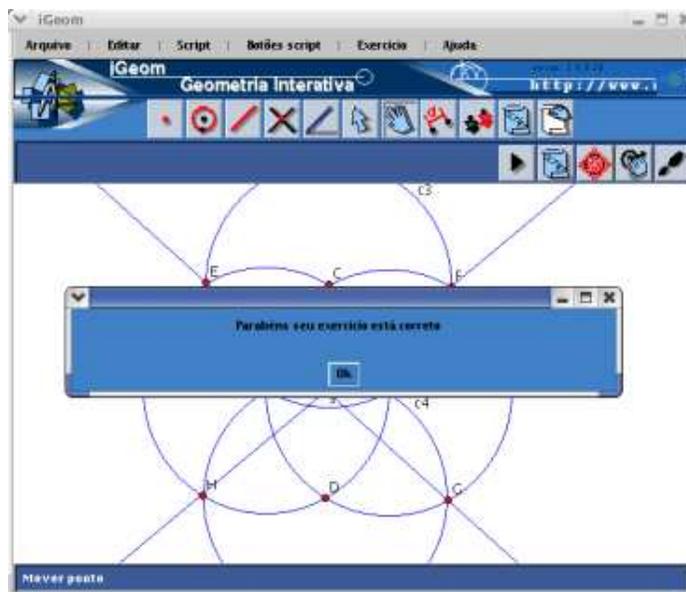


Figura 2.10: Janela que notifica o usuário se sua construção é correta ou não.

2.4.3 Comunicação

Para permitir a flexibilidade e a interatividade, o programa iGeom executado como *applet* permite a comunicação com servidor. A comunicação é criada através de uma conexão HTTP[W3C, 2004] direta entre o *applet* e o servidor e a troca de mensagens é feita utilizando o método POST[W3C, 2004]. Devido aos protocolos HTTP e POST serem muito populares a comunicação do iGeom pode ser facilmente incorporado em diferentes sistemas Web.

⁶sites sem a necessidade de senha ou cadastro prévio.

Este recurso em conjunto com os apresentados anteriormente, proporcionam a um aluno conectado ao servidor pegar um exercício criado pelo professor, resolve-lo e solicitar sua correção. Ao fazer esta requisição o iGeom irá corrigir o exercício (localmente) e enviar o resultado ao servidor (figura 2.11). Dentre as vantagens dessa implementação podemos listar:

1. um professor criar/editar um exercício em sua própria máquina e depois enviá-lo para o servidor ou fazê-lo pela Internet através de uma interface HTML;
2. fazer a adaptação da interface (ativar/desativar botões);
3. corrigir automaticamente os exercícios, oferecendo contra-exemplos quando possível;
4. Enviar ao servidor a avaliação da solução de um exercício.

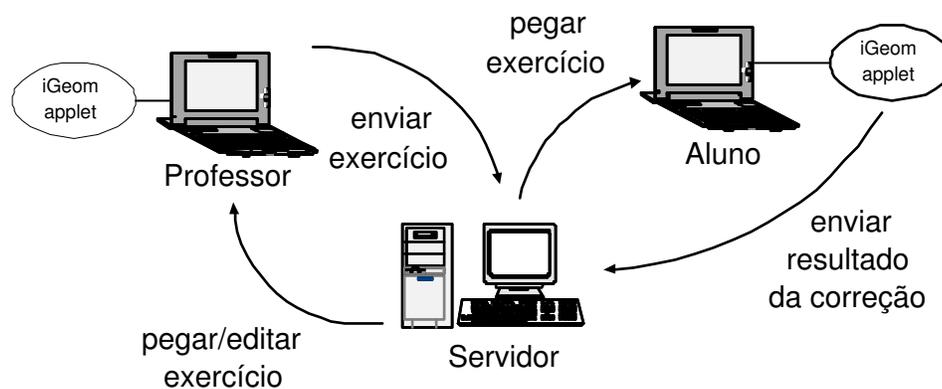


Figura 2.11: Comunicação entre um servidor e o iGeom em sistemas Web.

Capítulo 3

Colaboração e Cooperação

Atualmente existem duas áreas próximas que estudam aspectos colaborativos/cooperativos¹ em sistemas computacionais. A primeira, é a CSCW (Computer Supported Cooperative Work) que considera os sistemas que dão suporte as atividades em grupo visando proporcionar a cooperação na resolução de tarefas. A segunda, a CSCL (Computer Supported Collaborative Learning), designa a área que estuda os ambientes de ensino-aprendizagem que realizam tarefas em grupo sobre uma base cooperativa [Silveira and Ferreira, 2002]. Devido a estas características diversos autores consideram que a CSCL é uma subdivisão da CSCW [Barros, 1994, Silveira and Ferreira, 2002, Macedo, 1999].

Neste capítulo apresentamos nosso projeto de tornar o programa iGeom numa ferramenta de CSCL, parte deste projeto ainda será melhor embasado em futuras pesquisas bibliográficas e testes em laboratório.

3.1 Aprendizagem Cooperativa

O aprendizado cooperativo é uma técnica muito utilizada no ensino sendo atualmente explorada no contexto tecnológico. Seu objetivo é permitir o desenvolvimento de atividades

¹Existem divergências quanto uso dos nomes colaborar e cooperar. As definições segundo [Michaelis, 1998], **colaborar** é: 1. Trabalhar em comum com outrem na mesma obra; 2. Concorrer, cooperar para a realização de qualquer coisa. E **cooperar** é 1. Agir ou trabalhar junto com outro ou outros para um fim comum; 2. Agir conjuntamente para produzir um efeito.

Segundo [Houaiss, 2001], **colaborar** é: 1. Trabalhar com uma ou mais pessoas numa obra, cooperar, participar; 2. Concorrer ou contribuir para; 3. efetuar trabalho em cooperação. E **cooperar** é: 1. Atuar, juntamente, com outras, para um mesmo fim; 2. contribuir com trabalhos, esforços, auxiliar, colaborar.

No resto do texto adotaremos cooperação para representar o processo de ensino através da trabalho conjunto para colaborar/cooperar.

em grupo e proporcionar a cooperação mútua entre seus membros [Okamoto et al., 2001].

Segundo [Fuks et al., 2003], para colaborar os indivíduos devem trocar informações (se comunicar), organizar-se (se coordenar) e trabalhar em conjunto num espaço compartilhado (cooperar). Dessa forma, o aprendizado colaborativo apoiado por computador (CSCL) deve prover recursos que suportam estas três características oferecendo ferramentas, que conforme [Okamoto et al., 2001], são divididas em quatro categorias essenciais para se realizar o trabalho cooperativo. As categorias são:

- Formas de diálogo.
- Espaços compartilhados.
- Recursos para mediação da comunicação.
- Espaço pessoal.

Para desenvolver as ferramentas de comunicação precisamos identificar quando e onde os membros do grupo se inter-relacionarão. Essa ferramentas podem ser divididas nas dimensões espaço-tempo conforme mostra a figura 3.1 [Fuks et al., 2003]: Em grupos

		Síncrono	Tempo	Assíncrono
Espaço	Mesmo local	Interações face-a face		Interações assíncronas locais
	distribuído	Interações síncronas		Interações assíncronas distribuídas

Figura 3.1: Esquema de classificação espaço-tempo

onde os membros estão geograficamente distribuídos (distantes), a comunicação através de alguma tecnologia torna-se indispensável. Esta comunicação pode ser classificada como síncrona e assíncrona. Na comunicação síncrona as mensagens entre membros são enviadas e recebidas instantaneamente. Algumas ferramentas que permitem este tipo de comunicação são: chats, ferramentas de mensagens instantâneas (ICQ, Messenger), entre outras. Quando a comunicação é assíncrona significa que o “diálogo” não é feito

simultaneamente e os membros não precisam estar conectados ao mesmo tempo. Ou seja, as mensagens podem ser enviadas, mas só serão recebidas posteriormente. Exemplos de comunicação assíncrona são: fóruns, murais, correio eletrônico e listas de discussão.

A evolução do aprendizado em sistemas cooperativos está intimamente ligada ao modo como os membros do grupo se comunicam e se inter-relacionam. Muitas vezes, este processo não ocorre de forma natural, e portanto, é necessário que o professor aja como um facilitador e mediador, incentivando a cooperação.

A abordagem cooperativa pode, potencialmente, produzir melhores resultados se comparado ao estudo individual. Num ambiente cooperativo, alunos podem compartilhar idéias, identificar problemas em seu raciocínio, tomar decisões fundamentadas nas argumentações de um grupo, entre outras [Fuks et al., 2003, Macedo, 1999]. Porém, nem sempre a cooperação torna a tarefa de ensino-aprendizagem mais simples. Uma metáfora interessante para a cooperação é apresentada por [Baker et al., 1997], que compara a atividade cooperativa com uma corrida onde a perna direita de um corredor é amarrada a perna esquerda de outro corredor.

Certamente seria muito mais fácil correr individualmente, mas correndo juntos eles irão se comunicar para compartilhar seus intenções, aprenderão a se coordenar para não cair e finalmente poderão correr juntos em direção ao mesmo objeto.

3.2 O iGeom Cooperativo

Estamos interessados em dotar o iGeom de recursos de cooperação e com isso ampliar seu potencial educacional. Nesse projeto pretendemos responder a algumas questões propostas por [Baker et al., 1997]:

- Os usuários preferem resolver os problemas através do método cooperativo ou de forma tradicional (individual) ? Eles trabalham melhor em conjunto?
- Quais aspectos da cooperação são mais interessantes para o usuário?
- Quais informações sobre a interação do usuário são úteis para avaliação?
- Como os alunos dividem suas tarefas (cooperam)?
- Os alunos realmente cooperam uns com os outros?

Para tornar o iGeom uma ferramenta cooperativa para uso em sala de aula e pela Internet apresentaremos na próxima seção alguns recursos que pretendemos desenvolver. Estes recursos serão criados para permitir a comunicação síncrona, viabilizando a interação direta e em tempo real entre os usuários.

3.2.1 Diálogo, Mediação e Comunicação

Para proporcionar uma forma de comunicação que possa ser utilizada tanto no ensino presencial quanto via Internet, está em desenvolvimento uma ferramenta do tipo bate-papo (*chat*) que permite o diálogo entre todos os usuários que pertencem a um mesmo grupo (previamente definido).

Existem várias possibilidades de arquitetura para a construção de grupos. Uma delas é embutir no programa *chat* ambos os serviços: tanto a parte cliente, quanto a parte servidor. Mesmo fixando esta opção o grau de liberdade de definição ainda é grande.

Por exemplo, ao viabilizar o usuário a criar grupos (neste caso ele seria o servidor) e entrar em seu grupo (como clientes), podemos identificar duas possíveis arquiteturas que seriam representadas pelas topologias de grafo (com circuito) ou de árvore.

Na topologia mais geral (grafo) um usuário poderia pertencer a diversos grupos, seja como cliente ou como servidor (figura 3.2). Na estrutura de árvore o usuário poderia criar apenas um grupo e pertencer a um outro como cliente (figura 3.3).

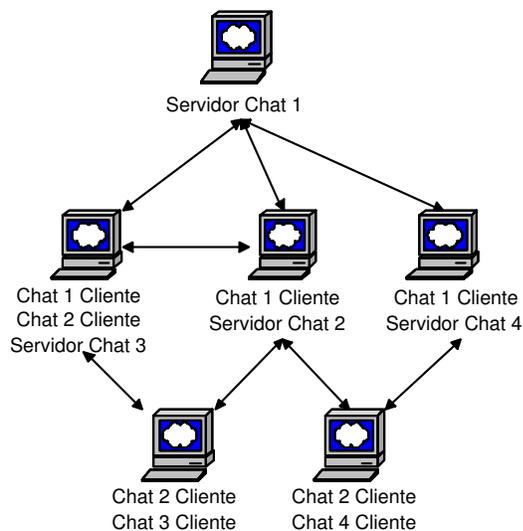


Figura 3.2: Estrutura em grafo

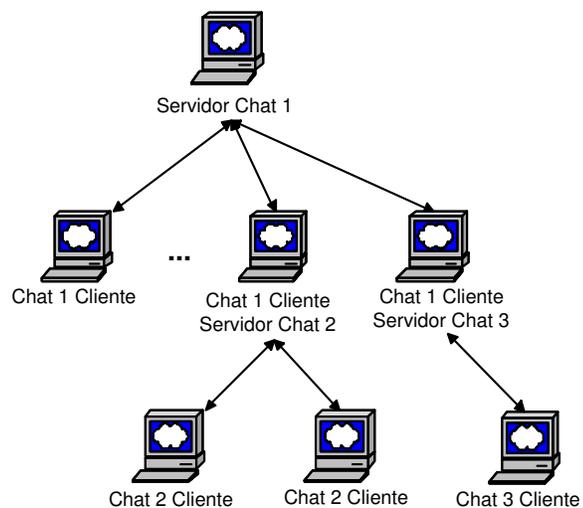


Figura 3.3: Estrutura em árvore

O uso do *chat* em conjunto com o iGeom permitirá que os usuário troquem informações a respeito de problemas, comentem suas idéias e compartilhem detalhes de suas construções. Além disso, esta ferramenta será de grande utilidade no ensino a distância onde os usuários se encontram distribuídos geograficamente, facilitando a comunicação e a interação entre eles. Além disso, professores e monitores poderão mediar a comunicação entre os usuários auxiliando-os em seus dificuldades, incentivando e guiando as discussões para que as soluções sejam encontradas através da experimentação e da argumentação.

Existem diversas aplicações e benefícios com o uso do *chat* [Kreutz et al., 2000], porém, às vezes, é muito complicado colocar na forma textual os problemas e soluções encontradas. Assim, será necessário criar um mecanismo que permita a cooperação usando o próprio iGeom que oferece suporte à recursos que possam ser compartilhados.

3.2.2 Espaços Compartilhados, Coordenação e Cooperação

Os espaços compartilhados vão estender os recursos do iGeom para permitir que grupos de alunos compartilhem construções de modo síncrono. Essa extensão implica no desenvolvimento e definição de um protocolo de permissões para definir, entre outras coisas, a maneira de se criar e remover objetos.

O uso do iGeom de forma cooperativa ainda está em estudo, contudo podemos identificar algumas possibilidades no controle dos objetos. Por exemplo, podemos imaginar o controle onde apenas um usuário pode criar e manipular os objetos, ou seja os objetos possuem dono. Outra forma de controle, poderia liberar qualquer usuário para criar e manipular os objetos, porém apenas um único usuário poderia fazê-lo por vez.

Para desenvolver o compartilhamento no iGeom a área de desenho será dividida em três classes (figura 3.4):

Área Compartilhada. Os objetos nesta área são visíveis para todos os membros do grupo. Nesta situação, podemos permitir que os membros do grupo possam manipular a construção ou apenas visualizá-la.

Área Individual. Esta área é exclusiva para um usuário que queira fazer suas próprias construções sem apresentá-las ao grupo. Essas construções poderão ser transferidas de uma área para outra a qualquer instante.

Área de Recepção. Uma construção enviada de um usuário para outro é apresentada nesta área. A construção enviada pode ser uma cópia da original ou dependente dela. No caso de dependência, a modificação da construção original modificará a construção apresentada na área de recepção.

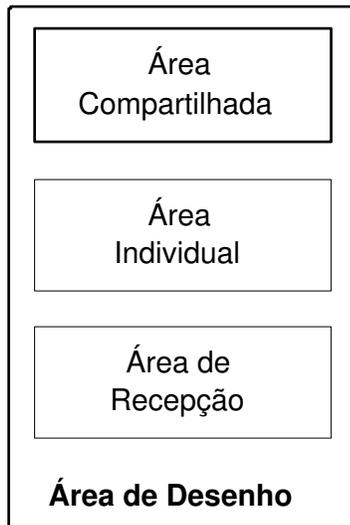


Figura 3.4: Classes de Área de Desenho

Capítulo 4

Considerações Finais

O propósito deste texto foi apresentar uma proposta, já em andamento, para a correção automática de exercícios de geometria e à aprendizagem colaborativa apoiada por computador (CSCL). Nele, mostramos os recursos já desenvolvidos e discutimos outros ainda não implementados que demandarão mais estudos e testes.

Outros trabalhos tem explorado o ambiente colaborativo/cooperativo como suporte ao aprendizado em geometria utilizando programas de geometria dinâmica, um deles é o Tabulae [Guimarães et al., 2002, Barbastefano et al., 2004]. Porém, as ferramentas de cooperação neste programa estão em desenvolvimento, não possuindo ainda análises de seu uso em sala de aula que demonstrem os benefícios e dificuldades de seu uso em atividades colaborativas.

Com o desenvolvimento do corretor automática no iGeom e sua utilização em sistemas Web, proporcionamos um ambiente que viabiliza o oferecimento de cursos onde o número de alunos é muito grande, pois suas funcionalidades auxiliam tanto o professor, reduzindo seu trabalho no desenvolvimento do material didático, quanto o aluno que recebe o retorno da avaliação de seus exercícios quase que instantaneamente. Esta solução já está sendo aplicada em uma disciplina do curso de licenciatura em Matemática oferecida no IME-USP (<http://www.ime.usp.br/~leo/mac118/04>). Esta aplicação tem sido muito útil no aprimoramento do sistema, pois os alunos tem participado ativamente, sugerindo aperfeiçoamentos e detectando algumas falhas.

O uso do iGeom em atividades colaborativas é o desafio atualmente. Esperamos com este novo recurso favorecer o trabalho em grupo e melhorar a aprendizagem através da troca de conhecimento entre alunos.

O cronograma do trabalho resume-se em:

1. pesquisa bibliográfica.
2. definição de protocolos para o compartilhamento de espaços no iGeom.
3. implementação e testes.
4. redação da dissertação.

Referências Bibliográficas

- [Abrão et al., 2004] Abrão, I. C., Rayel, F., and Abrão, M. A. V. L. (2004). Questcomp: Ferramenta para avaliação de aprendizado à distância. In *Proceedings World Congress on Engineering Technology Education*, http://www.inf.pucpcaldas.br/rayel/download/papers/published/WCETE2004_Questcomp.pdf.
- [Baker et al., 1997] Baker, M. J., Bricker, L. J., and Tanimoto, S. L. (1997). Cooperative interaction techniques for graphical objects in a collaborative activity. Technical report, Department of Computer Science and Engineering of University of Washington.
- [Barbastefano et al., 2004] Barbastefano, R., Mattos, F., and Guimarães, T. (2004). Tabuale, uma ferramenta de colaboração síncrona em geometria. In *Anais do Segundo Colóquio de História e Tecnologia no Ensino de Matemática*.
- [Barr and Feigenbaum, 1989] Barr, A. and Feigenbaum, E. A. (1989). *The Handbook of Artificial Intelligence*, volume 4. Addison-Wesley, Massachusetts.
- [Barros, 1994] Barros, L. A. (1994). *Suporte a Ambientes Distribuídos para Aprendizagem Cooperativa*. Tese de doutorado em engenharia, COPPE - Universidade Federal do Rio de Janeiro.
- [Bevilacqua, 2002] Bevilacqua, J. S. (2002). Introdução da disciplina cálculo numérico no ambiente webct. In *Proceedings of VII International Conference on Engineering and Technology Education*, <http://www.asee.org/international/INTERTECH2002/566.pdf>.
- [Brandão, 2004] Brandão, L. O. (2004). Programação geométrica: Uso da geometria dinâmica para programação. In *Anais do Segundo Colóquio de História e Tecnologia no Ensino de Matemática*, pages 191–202.

- [Braviano and Rodrigues, 2002] Braviano, G. and Rodrigues, M. H. W. L. (2002). Geometria dinâmica: uma nova geometria? *Revista do Professor de Matemática*, (49):22–26.
- [Brusilovsky, 1998] Brusilovsky, P. (1998). 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*, pages 16–19.
- [Crowley, 1998] Crowley, M. L. (1998). *Aprendendo e Ensinando Geometria*, chapter O modelo Van Hiele de desenvolvimento do pensamento geométrico. Editora Atual.
- [Fuks et al., 2003] Fuks, H., Gerosa, M. A., and Pimentel, M. G. (2003). Projeto de comunicação em groupware: Desenvolvimento, interface e utilização. In *XXII Jornada de Atualização em Informática*, volume 2, chapter 7, pages 295–338. Sociedade Brasileira de Computação.
- [Gelernter, 1995] Gelernter, H. (1995). Realization of a geometry-theorem proving machine. pages 134–152.
- [Gelernter et al., 1995] Gelernter, H., Hansen, J. R., and Loveland, D. W. (1995). Empirical explorations of the geometry-theorem proving machine. pages 153–163.
- [Gibson et al., 1995] Gibson, E. J., Brewer, P. W., Dholakia, A., Vouk, M. A., and Bitzer, D. L. (1995). A comparative analysis of web-based testing and evaluation systems. In *Proceedings of Fourth International World Wide Web Conference*, http://renoir.csc.ncsu.edu/Faculty/Vouk/Papers/Gibson/Gibson_WWW4_1995.pdf.
- [Gravina, 1996] Gravina, M. A. (1996). Geometria dinamica - uma nova abordagem para o aprendizado da geometria. In *Anais do VII Simposio Brasileiro de Informática na Educação*, pages 1–13.
- [Grothman, 1999] Grothman, R. (1999). *C.A.R - Compass And Rules*. <http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/zirkel/>.
- [Guimarães et al., 2002] Guimarães, L. C., Barbastefano, R., and Belfort, E. (2002). Tools for synchronous distance teaching in geometry. In *Proceedings of Second International Conference on the Teaching of Mathematics*, <http://www.math.uoc.gr/ictm2/Proceedings/pap385.pdf>.

- [Hannafin and Peck, 1988] Hannafin, M. J. and Peck, K. L. (1988). *The Design, Development and Evaluation of Instructional Software*. MacMillan Publishing Company, New York.
- [Houaiss, 2001] Houaiss, I. A. (2001). *Dicionário eletrônico Houaiss da língua portuguesa*. Objetiva, São Paulo.
- [Isotani and Brandao, 2001] Isotani, S. and Brandao, L. O. (2001). Informática: Ambiente interativo de apoio ao ensino de matemática via internet. In *Anais do Workshop sobre Informática na Escola*, pages 533–543. XXI Congresso da Sociedade Brasileira de Computação.
- [Java,] Java. *Linguagem de Programação Java*. Sun Microsystems, <http://java.sun.com/>.
- [Kreutz et al., 2000] Kreutz, R., Kiesow, S., and Spitzer, K. (2000). Netchat: Communication and collaboration via www. available at http://ifets.ieee.org/periodical/vol_3_2000/a04.pdf.
- [Macedo, 1999] Macedo, A. A. (1999). Explorando tecnologia hipermídia e de trabalho cooperativo em um ambiente de apoio ao ensino. Dissertação de mestrado em ciência da computação, Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo.
- [Matsuda and Vanlehn, 2004] Matsuda, N. and Vanlehn, K. (2004). Gramy: A geometry theorem prover capable of construction. *Journal of Automated Reasoning*, 32(1):3–33.
- [Melo et al., 2000] Melo, L. B., Ferreira, J. M., and Pontes, J. D. A. (2000). Um software educacional para o descobrimento de prioridades matemáticas. In *Anais do XX Congresso da Sociedade Brasileira de Computação*. Sociedade Brasileira de Computação.
- [Michaelis, 1998] Michaelis (1998). *Michaelis Moderno Dicionário da Língua Portuguesa*. Melhoramentos, São Paulo.
- [Neto, 2002] Neto, A. S. (2002). Comunicação e interação em ambientes de aprendizagem presenciais e virtuais. In *Anais do VI Congresso de Educação a Distância*, <http://fad.uta.cl/dfad/docum/cedm/2-br-Antonio%20Sim%20E3o%20Neto.pdf>.

- [Nitzke et al., 1999] Nitzke, J. A., Carneiro, M. L. F., Geller, M., and Santarosa, L. C. (1999). Criação de ambientes de aprendizagem colaborativa. In *Anais do X Simpósio Brasileiro de Informatica na Educação*, <http://penta.ufrgs.br/pgie/sbie99/acac.html>.
- [Okamoto et al., 2001] Okamoto, T., Kayama, M., and Cristea, A. I. (2001). Considerations for building a common platform of collaborative learning environment. In *Proceedings of International Conference on Computer in Education*, pages 800–807, <http://citeseer.ist.psu.edu/521884.html>.
- [Papert, 1999] Papert, S. (1999). *Mindstorms: children, computer and powerful ideas*. Basic Books, New York, second edition edition.
- [Pólya, 1975] Pólya, G. (1975). *Arte de resolver problemas : um novo aspecto do metodo matematico*. Editora Interciência, Rio de Janeiro.
- [Richter-Gebert and Kortenkamp, 1999] Richter-Gebert, J. and Kortenkamp, U. (1999). *The Interactive Geometry Software Cinderella*. Springer-Verlag, Heidelberg.
- [Rodrigues, 2002] Rodrigues, D. W. L. (2002). Uma avaliação comparativa de interfaces homem-computador em programas de geometria dinamica. Dissertação de mestrado em engenharia de produção, Universidade Federal de Santa Catarina.
- [Scapin, 1997] Scapin, R. H. (1997). Desenvolvimento de uma ferramenta para criação e correção automática de provas na world-wide-web. Dissertação de mestrado em física, Instituto de Física de São Carlos, Universidade de São Paulo.
- [Silveira and Ferreira, 2002] Silveira, I. F. and Ferreira, M. A. G. V. (2002). Implementando vygotsky com piaget: Autoria híbrida de conteúdo didático em um ambiente virtual distribuído de apoio à aprendizagem colaborativa a distância. In *Anais do XIII Simpósio Brasileiro de Informatica na Educação*.
- [Silveira, 1996] Silveira, R. A. (1996). Diagnóstico e modelagem cognitiva em ambientes de ensino inteligentes. Technical report, Instituto de Informática - Universidade Federal do Rio Grande do Sul.
- [Valente, 1999] Valente, J. A. (1999). *O computador na Sociedade do Conhecimento*. <http://www.nied.unicamp.br/oea/pub/livro1/>. nied.

[W3C, 2004] W3C, W. W. W. C. (2004). *HyperText Markup Language (HTML)*.
<http://www.w3.org/MarkUp/>.