

Universidade Federal de Santa Catarina
Programa de Pós-graduação em
Engenharia de Produção

UMA AVALIAÇÃO COMPARATIVA
DE INTERFACES HOMEM-COMPUTADOR
EM PROGRAMAS DE GEOMETRIA DINÂMICA

Daniel Wyllie Lacerda Rodrigues

Dissertação apresentada ao
Programa de Pós-Graduação em
Engenharia de Produção da
Universidade Federal de Santa Catarina
como requisito parcial para a obtenção
do título de mestre em
Engenharia de Produção,
(Área de Concentração: Ergonomia)

Florianópolis
2002

Daniel Wyllie Lacerda Rodrigues

**UMA AVALIAÇÃO COMPARATIVA
DE INTERFACES HOMEM-COMPUTADOR
EM PROGRAMAS DE GEOMETRIA DINÂMICA**

Esta dissertação foi julgada e aprovada para a
obtenção do título de **Mestre em Engenharia de
Produção no Programa de Pós-Graduação em
Engenharia de Produção da
Universidade Federal de Santa Catarina**

Florianópolis, 8 de fevereiro de 2002.

Prof. Ricardo Miranda Barcia, Ph.D.
Coordenador do Curso de Pós-Graduação
Em Engenharia de Produção

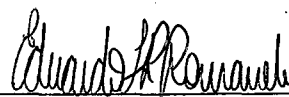
BANCA EXAMINADORA



Prof. Gilson Braviano, Dr.



Prof. Vania Ribas Ulbricht, Dr.
Orientadora



Prof. Eduardo Félix Romanelli, Dr.

Dedicatória

À minha mãe,
profa. Dra. Maria Helena Wyllie Lacerda Rodrigues
pela incondicional ajuda em várias etapas de minha vida.

Agradecimentos

A meus pais, pelo constante apoio intelectual, emocional e financeiro.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES.

A todos os professores e alunos que cooperaram respondendo ao instrumento de pesquisa utilizado.

À minha orientadora, profa. Dra. Vania Ribas Ulbricht, pelo inestimável exemplo de dedicação e sabedoria, que em muito contribuiu para a conclusão deste trabalho.

Ao Dr. Franck Bellemain, pela valiosa troca de informações sobre o programa Cabri Géomètre II durante o Graphica 2001.

Ao Dr. Ulrich Kortenkamp, pelas respostas fornecidas quanto ao programa Cinderella.

SUMÁRIO

SUMÁRIO	v
LISTA DE FIGURAS	viii
LISTA DE TABELAS	x
RESUMO	xii
ABSTRACT	xi
1 INTRODUÇÃO	1
1.1 Introdução	1
1.2 Origem do trabalho	2
1.3 Justificativa e relevância da pesquisa	2
1.4 Objetivos.....	4
1.4.1 Objetivo geral	4
1.4.2 Objetivos específicos.....	4
1.5 Questões de estudo.....	4
1.6 Procedimentos metodológicos.....	5
1.7 Limitações do trabalho.....	6
1.8 Descrição e organização dos capítulos	7
2 ERGONOMIA E DESIGN DE IHC	8
2.1 Interface e interação	8
2.2 Usabilidade e comunicabilidade.....	9
2.3 Engenharia cognitiva.....	12
2.4 Engenharia semiótica	15
2.5 Gestos e modos de operação	16
2.6 Estilos de Interação	17
2.7 Heurísticas fundamentais	20
2.8 Métodos para a avaliação de interfaces.....	23
2.8.1 Avaliação heurística	23
2.8.2 Avaliação por testagem.....	24
2.8.3 Práticas alternativas ou complementares.....	25
2.9 Conclusão do capítulo 2.....	26

3	ESTADO DA ARTE DA GEOMETRIA DINÂMICA.....	28
3.1	Geometria euclidiana.....	28
3.2	Geometria dinâmica.....	30
3.3	Programas conhecidos	31
3.4	Benefícios e aplicações.....	33
3.4.1	Precisão de construção	33
3.4.2	Visualização	34
3.4.3	Exploração e descoberta.....	35
3.4.4	Prova	37
3.4.5	Transformações.....	39
3.4.6	Loci.....	40
3.4.7	Simulação.....	42
3.4.8	Micromundo.....	44
3.5	Discussões e problemas sobre interfaces conhecidas	44
3.5.1	Comandos em ordem pré e pós-fixada	45
3.5.2	<i>Undo</i> ilimitado <i>versus</i> redefinição de lugares geométricos.....	50
3.5.3	Problemas comuns de interação	52
3.6	Conclusão do capítulo 3.....	55
4	METODOLOGIA E DESENVOLVIMENTO DA PESQUISA	56
4.1	Introdução	56
4.2	Ordem de comandos	59
4.3	Mudança de atributos	67
4.4	<i>Feedback</i>	73
4.5	Manipulação direta	76
4.6	Animação.....	90
4.6.1	Exemplo 1	95
4.6.2	Exemplo 2	97
4.6.3	Exemplo 3	99
4.7	Lugar Geométrico	102
4.7.1	Exemplo 1	104
4.7.2	Exemplo 2	107
4.8	Macros e Scripts	109

4.9	Conclusão do capítulo 4.....	113
5	CONCLUSÕES.....	114
5.1	Conclusões finais	114
5.2	Sugestões para trabalhos futuros	118
6	REFERÊNCIAS BIBLIOGRÁFICAS.....	119
7	ANEXOS	124
7.1	Questionário p/ avaliação das IHC dos programas de G.din.	124
7.2	Dados brutos obtidos pelo instrumento de pesquisa	134

LISTA DE FIGURAS

Figura 1: Processo de interação homem-computador.....	9
Figura 2: Fatores de aceitação de um sistema.....	10
Figura 3: “AIBO vai mostrar para você quando ele estiver feliz ou triste”.....	11
Figura 4: Comunicação visual através de ícones.....	12
Figura 5: Processo de desenvolvimento centrado no usuário.....	12
Figura 6: Comunicação entre usuário e designer na engenharia semiótica	16
Figura 7: Heurísticas para a avaliação de interfaces segundo.....	21
Figura 8: Tipo de laboratório de usabilidade utilizado pela Microsoft.....	25
Figura 9: Esboço de um triângulo ABC “qualquer”.....	29
Figura 10: Quando C gira, D também gira e conseqüentemente.....	31
Figura 11: Um desenho que vale por todos.....	31
Figura 12: Problemas de complexidade e imprecisão no papel são.....	34
Figura 13: Visualização do teorema de Pitágoras.....	35
Figura 14: Exploração e descoberta.....	36
Figura 15: Verificação interna de propriedades no Cabri Géomètre II.....	38
Figura 16: Checagem de teoremas randomizada no Cinderella.....	39
Figura 17: Possibilidade de homotetia no Cinderella.....	40
Figura 18: <i>Locus</i> na configuração de barras interligadas.....	42
Figura 19: <i>Locus</i> representa uma cicloide.....	42
Figura 20: Exemplo de simulação de equilíbrio em mecânica.....	43
Figura 21: Simulação do movimento para se construir um helicóide.....	43
Figura 22: Ícone para criação de círculo ativado no Cabri Géomètre II.....	47
Figura 23: Ângulo vira seu replemento.....	52
Figura 24: Ponto P salta de repente.....	53
Figura 25: Pontos sob a mesma reta convergem para aquele que a define.....	54
Figura 26: Barra de ferramentas do Cabri Géomètre II.....	60
Figura 27: Menu textual acessado a partir de determinada opção.....	60
Figura 28: Caixa de ferramentas com opções disponíveis.....	62
Figura 29: Ícones com pistas sobre os menus associados.....	62

Figura 30: Menu com opções possíveis e impossíveis.	63
Figura 31: Quatro pontos e uma reta selecionados.....	64
Figura 32: Quatro retas perpendiculares criadas de uma só vez.	64
Figura 33: Barra de ferramentas do Cinderella.	64
Figura 34: Últimos botões saltam para baixo quando a 65	65
Figura 35: Atributos associados à ferramenta (A) e ao objeto (B).....	68
Figura 36: Menu <i>pull down</i> para a mudança de atributos (cores, neste caso). 69	69
Figura 37: Atributos do Cinderella.	70
Figura 38: Interface para modificação das componentes RGB de uma cor.	70
Figura 39: Etapas para a reflexão de um círculo em relação a um ponto.	74
Figura 40: Barra de mensagens no canto inferior esquerdo da tela.....	74
Figura 41: Lista textual dos objetos na tela.	79
Figura 42: Reta definida pelo método A.	84
Figura 43: Reta definida pelo método B.	84
Figura 44: Círculo definido pelo método A.	88
Figura 45: Círculo definido pelo método B.	88
Figura 46: Controle de velocidade da animação no Cabri Géomètre II.....	92
Figura 47: Controle de velocidade da animação no Sketchpad.	92
Figura 48: Controle de velocidade da animação no Cinderella.	93
Figura 49: Primeiro exemplo de animação.....	95
Figura 50: Segundo exemplo de animação.....	97
Figura 51: Terceiro exemplo de animação.....	100
Figura 52: Possível aplicação apontada por um estudante.....	101
Figura 53: Primeiro exemplo de Lugar Geométrico.....	105
Figura 54: Segundo exemplo de Lugar Geométrico.....	107

LISTA DE TABELAS

Tabela 1: Conhecimento no mundo e conhecimento na cabeça.....	14
Tabela 2: Modos de operação diferentes.....	16
Tabela 3: Vantagens e desvantagens de cinco estilos.....	18
Tabela 4: Comparação de sete estilos.....	19
Tabela 5: Exemplos em sintaxe pré-fixada.....	46
Tabela 6: Exemplos em sintaxe pós-fixada.....	48
Tabela 7: Familiaridade dos respondentes com os programas avaliados.....	59
Tabela 8: Resultados da pergunta 5 do questionário.....	65
Tabela 9: Resultados das perguntas 6.1 e 6.2 do questionário.....	71
Tabela 10: Resultados da pergunta 6.3 do questionário.....	72
Tabela 11: Resultados da pergunta 7 do questionário.....	75
Tabela 12: Resultados da pergunta 8 do questionário.....	77
Tabela 13: Programas e abordagens para a seleção de um grupo de objetos.....	79
Tabela 14: Resultados da pergunta 9 do questionário.....	80
Tabela 15: Alternativa A para criação de reta.....	81
Tabela 16: Alternativa B para criação de reta.....	81
Tabela 17: Programas e respectivas abordagens para a criação de retas.....	81
Tabela 18: Resultados da pergunta 10.1 do questionário.....	82
Tabela 19: Programas e respectivas abordagens para a definição de retas.....	84
Tabela 20: Ações e reações em relação à manipulação direta de retas.....	84
Tabela 21: Resultados da pergunta 10.2 do questionário.....	85
Tabela 22: Alternativa A para criação de círculo.....	86
Tabela 23: Alternativa B para criação de círculo.....	86
Tabela 24: Programas e abordagens para a criação de círculos.....	86
Tabela 25: Resultados da pergunta 11.1 do questionário.....	86
Tabela 26: Programas e abordagens para a definição de círculos.....	87
Tabela 27: Ações e reações em relação à manipulação direta de círculos.....	88
Tabela 28: Resultados da pergunta 11.2 do questionário.....	89
Tabela 29: Programas e respectivos recursos para a animação de objetos.....	90
Tabela 30: Resultados da pergunta 12 do questionário.....	90

Tabela 31: Possibilidades de animação entre os programas analisados.....	94
Tabela 32: Resultados da pergunta 13 do questionário.....	96
Tabela 33: Resultados da pergunta 14 do questionário.....	98
Tabela 34: Resultados da pergunta 15 do questionário.....	100
Tabela 35: Resultados da pergunta 16 do questionário.....	104
Tabela 36: Resultados da pergunta 17 do questionário.....	106
Tabela 37: Resultados da pergunta 18 do questionário.....	108
Tabela 38: Programas e respectivas alternativas para a criação de macros.	110
Tabela 39: Resultados da pergunta 19 do questionário.....	110
Tabela 40: Programas e alternativas para a execução de macros.....	111
Tabela 41: Resultados da pergunta 20 do questionário.....	112

RESUMO

RODRIGUES, Daniel Wylie Lacerda. **UMA AVALIAÇÃO COMPARATIVA DE INTERFACES HOMEM-COMPUTADOR EM PROGRAMAS DE GEOMETRIA DINÂMICA**. Florianópolis, 2002. 161f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção, UFSC, 2002.

A Geometria Dinâmica é uma poderosa ferramenta para o aprendizado de uma série de disciplinas gráficas e matemáticas. A partir da construção de modelos geométricos precisos e interativos, os usuários podem usufruir de uma variedade de aplicações, das quais destacam-se: a visualização de lugares geométricos, a descoberta de relações entre elementos e a obtenção de provas experimentais de teoremas para cada modelo criado. Vários programas contendo essas características surgiram na década passada e vêm evoluindo desde então. A trinca Cabri Géomètre, The Geometer's Sketchpad e Cinderella revelou ser composta pelos mais significativos. Neste trabalho busca-se comparar as diferenças entre as abordagens de IHC (interface homem-computador) adotadas por cada um dos três programas citados. Para fundamentar este estudo, primeiramente são abordados os conhecimentos provenientes das áreas da ergonomia e do *design* de interfaces. Posteriormente, com base nos fundamentos e nas aplicações da geometria dinâmica, assim como nas discussões travadas pelos projetistas destes aplicativos, realiza-se uma avaliação envolvendo alunos de graduação, de pós-graduação e professores de matemática e de expressão gráfica. Como resultado, verifica-se que mesmo em suas versões mais recentes, estes programas trazem uma série de características que dificilmente podem ser julgadas e determinadas sob um mesmo ponto de vista.

Palavras-Chave: *Design* de Interfaces, Ergonomia de Interfaces, Geometria Dinâmica, Interfaces Homem-Computador, Usabilidade.

ABSTRACT

RODRIGUES, Daniel Wyllie Lacerda. **UMA AVALIAÇÃO COMPARATIVA DE INTERFACES HOMEM-COMPUTADOR EM PROGRAMAS DE GEOMETRIA DINÂMICA**. Florianópolis, 2002. 161f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção, UFSC, 2002.

Dynamic Geometry is a powerful tool for the learning of many graphic and mathematical disciplines. Through the use of precise and interactive computer models, users are able to enjoy a variety of applications such as: the visualization of loci, identification of relationships between elements and the acquisition of experimental proof for theorems of each model created. Several programs sharing these characteristics have been developed in the last few years, the most powerful of which are Cabri Géomètre, the Geometer's Sketchpad and Cinderella. In this work, the author has sought to compare the differences between the human computer interfaces (HCI) used by each one of these programs. To support this study, we first explore the knowledge obtained from the areas dealing with the ergonomics and design of interfaces. Then, based on these foundations and on the applications of dynamic geometry plus the result of discussions between the creators of these applications, an evaluation is performed with undergraduate and graduate students as well as with mathematics and graphical expression professors. Results show that even the updated version of these software embody characteristics which can hardly be evaluated and defined under a single view point.

Key words: Interface Design, Interface Ergonomics, Dynamic Geometry, Human Computer Interaction, Usability.

1 INTRODUÇÃO

1.1 Introdução

Durante séculos têm sido utilizados instrumentos tradicionais para a criação de modelos e problemas na geometria clássica, tais como lápis, papel, régua, esquadros e compasso. No entanto, no início da década de 80, alguns pesquisadores matemáticos começaram a imaginar a possibilidade de se criar programas de computador capazes de substituir o processo tradicional de traçado de figuras geométricas por uma alternativa mais prática, e que pudesse agregar outros benefícios além dos previamente existentes. A princípio, um programa deste tipo deveria agilizar o trabalho manual, garantir uma precisão superior de traçado e permitir a reprodução exata e facilitada dos desenhos.

Hoje em dia estes programas não somente existem mas também estão sendo utilizados em algumas escolas e universidades do Brasil e do mundo. Inclusive, os mais difundidos já se encontram em sua segunda ou terceira versão. Além de atingir os objetivos iniciais propostos, apresentando uma nova alternativa para o desenho geométrico, estes produtos alcançaram uma meta visivelmente superior, e provavelmente sequer imaginada pelos seus criadores nos momentos iniciais de suas concepções: o ensino e a aprendizagem através de uma suposta “nova” geometria, a geometria dinâmica.

Basicamente, este trabalho busca uma comparação entre o estado da arte da geometria dinâmica e os benefícios e contribuições do *design* e da ergonomia de interfaces homem-computador (IHC). Procura-se reconhecer e comparar os diferentes aspectos ergonômicos de IHC encontrados em 3 programas conhecidos: Cabri Géomètre II, The Geometer's Sketchpad 3 e Cinderella.

1.2 Origem do trabalho

A idéia de se pesquisar as características de IHC em programas de geometria dinâmica partiu de indagações do próprio autor quanto aos motivos que levam os projetistas a adotar diferentes estratégias de uso para recursos aparentemente semelhantes.

A experiência conjunta na elaboração de uma seqüência didática (ambiente Transpontuais), para o estudo das transformações pontuais na geometria plana e no desenho projetivo (Rodrigues & Rodrigues, 2000), foi um fator determinante para se perceber e trabalhar com as principais características do Cabri Géomètre II, o primeiro dos três programas utilizados. O conhecimento posterior do Cinderella e do The Geometer's Sketchpad 3 colaborou para se intensificar uma série de questionamentos a respeito dos diferentes recursos e formas de uso implementados. Essa situação representou o ponto de partida para se investigar a relevância e a possibilidade de um estudo aprofundado sobre o tema.

1.3 Justificativa e relevância da pesquisa

“A integração da geometria dinâmica no ensino é um processo recente. Apesar da existência de várias experiências e pesquisas bem sucedidas sobre essa integração, são muitos os questionamentos sobre as formas de usar esse novo sistema de representação dos objetos geométricos”.

(BELLEMAIN, 2001, p.1327)

Para se transformar instrumentos físicos manipuláveis tais como lápis, borracha, régua e compasso em ferramentas computacionais é necessário não somente compreender os estilos de interação que os sistemas informatizados oferecem, mas também reconhecer a maneira pela qual o ser humano age e raciocina numa possível situação de uso. Na tentativa de se aplicar o conceito da geometria dinâmica na criação de um programa de computador, vale refletir

sobre a importância das ferramentas. Além disso, no processo de ensino e aprendizagem, é fundamental que o sistema de representação de objetos seja completo e adequado em função do modelo matemático proposto (Bellemain, 2001). Não por acaso, pode-se afirmar que a criação de recursos associados à interface de um programa de geometria dinâmica representa um grande desafio.

Nos últimos anos, uma série de programas desta categoria tem surgido e ocupado lugar nas escolas e universidades do mundo inteiro. Com o rápido avanço tecnológico e o crescimento exponencial da internet, parece natural que cada vez mais pessoas façam uso da informática não somente no âmbito profissional, mas também educacional. Neste sentido, no Brasil também é crescente o interesse da comunidade acadêmica (de expressão gráfica e da matemática) em conhecer e aplicar os fundamentos da geometria dinâmica em sala de aula. Alguns indícios deste fato são:

- o desenvolvimento nacional de programas de geometria dinâmica;
- a criação nacional de ambientes de estudo aplicando os recursos da geometria dinâmica pela internet;
- o relevante número de trabalhos sobre o tema publicados em periódicos e congressos nacionais;
- o espaço cada vez mais amplo destinado à geometria dinâmica através de palestras, mini-cursos e *workshops* em congressos e eventos.

Ao se levar em consideração o atual aspecto de tal cenário, julga-se que o estudo comparativo das características de IHC em programas desta categoria pode contribuir significativamente em dois sentidos:

1. atuando como um ponto de referência para o discernimento entre os usuários dos possíveis aspectos positivos e negativos das soluções de usabilidade observadas.

2. servindo de base para a correção e o aprimoramento de versões anteriores, assim como para a criação de novos produtos que melhor se adequem às expectativas e às tarefas dos usuários.

1.4 Objetivos

1.4.1 Objetivo geral

- Reconhecer e comparar os diferentes aspectos ergonômicos de IHC encontrados em programas de geometria dinâmica.

1.4.2 Objetivos específicos

- apontar e discutir as principais falhas de usabilidade provenientes dos programas Cabri Géomètre II, The Geometer's Sketchpad 3 e Cinderella;
- estabelecer distinções entre estes programas com base na diversidade de suas características de uso para um certo número de funcionalidades em comum;

1.5 Questões de estudo

Com base nos objetivos traçados, procurou-se resposta para as seguintes questões:

1. Quais são os principais problemas de usabilidade que podem ser encontrados nos programas Cabri Géomètre II, The Geometer's Sketchpad 3 e Cinderella através de uma análise heurística?

2. Para um conjunto básico de funcionalidades e recursos, comum às diversas implementações da geometria dinâmica, quais são as características de IHC que melhor se adequam às expectativas de uso e tarefas executadas pelos usuários?

1.6 Procedimentos metodológicos

Este trabalho busca, através de uma avaliação heurística e da aplicação de um questionário entre usuários de programas de geometria dinâmica, levantar informações que possam ser analisadas e comparadas entre si. Para fundamentar esta pesquisa foram estudados os conceitos e aspectos de ergonomia de IHC, assim como os recursos, aplicações e benefícios da geometria dinâmica. A aplicação do instrumento de pesquisa contou com a participação de 14 indivíduos, que colaboraram preenchendo os formulários durante o período do dia 26 de novembro até o dia 10 de dezembro de 2001. Divididos por categoria, os participantes foram:

- 4 alunos de graduação do curso de licenciatura em Educação Artística (habilitação em desenho) da UFRJ – EBA;
- 5 alunos de pós-graduação (lato-sensu) do curso de Especialização em Técnicas de Representação Gráfica da UFRJ – EBA;
- 5 professores, sendo 3 deles da UFRJ, um da Universidade Estadual de Londrina e outro do Colégio Pedro II.

O pesquisador também entrevistou o aplicador do instrumento aos alunos, de modo a obter um maior número de informações ao seu alcance.

1.7 Limitações do trabalho

Devido a uma série de características relacionadas à concepção, à implementação e ao uso dos programas analisados, este trabalho conta com algumas limitações:

- em virtude do grande número de funcionalidades e de recursos oferecidos, foi necessário o estabelecimento de um mínimo denominador comum entre os mesmos; várias das funcionalidades específicas de cada programa não foram consideradas;
- não foi possível a aplicação de um método formal para a avaliação de software, pois, entre diversos fatores, o tempo foi escasso e houve a dificuldade de se encontrar um público alvo significativo e próximo que pudesse contribuir para as avaliações dos três programas em laboratório;
- o método de avaliação heurística utilizado só contou com a presença de um especialista, que neste caso foi representado pelo próprio pesquisador;
- o método de avaliação por questionário contou com a presença de apenas 14 voluntários, representativos de um grupo bastante heterogêneo;
- a falta de um maior embasamento matemático e pedagógico acabou por impedir uma abordagem mais ampla sobre o tema, restringindo a pesquisa ao escopo dos conceitos fundamentais de usabilidade.

1.8 Descrição e organização dos capítulos

Da forma como é apresentada, esta dissertação se divide em 5 capítulos:

1. Introdução: este capítulo contém a apresentação resumida do tema, a especificação das questões e dos objetivos de estudo levantados, a justificativa e a relevância da pesquisa, assim como uma breve notícia da metodologia adotada.
2. Ergonomia e *design* de interfaces: consistindo na primeira parte da revisão de literatura, este capítulo trata dos principais conceitos e aspectos da ergonomia e *design* de IHC, tais como as teorias da engenharia cognitiva e semiótica, o processo iterativo de desenvolvimento de *software* e os métodos de avaliação da usabilidade.
3. Estado da arte da geometria dinâmica: consistindo na segunda parte da revisão de literatura, este capítulo trata das definições, programas, recursos, aplicações e benefícios da geometria dinâmica na atualidade.
4. Metodologia e desenvolvimento da pesquisa: este capítulo apresenta os procedimentos metodológicos adotados, descreve os passos seguidos e a decorrente análise dos dados obtidos.
5. Resultados e conclusões: o tema deste capítulo se refere aos resultados e conclusões provenientes dos métodos de avaliação utilizados.

2 ERGONOMIA E DESIGN DE IHC

2.1 Interface e interação

“Usuários não se importam com o que está dentro da caixa, pelo menos enquanto ela faz o que eles precisam. Qual processador é utilizado? A linguagem de programação é orientada a objetos? Ela trabalha com transações simultâneas? Possui algum outro jargão popular recente? Nada disso é levado em consideração. Os usuários querem conveniência e resultados, mas o que eles enxergam é a interface. Até onde o consumidor está envolvido, a interface é o produto.”

(RASKIN, 2000, p.5)

Cada usuário tem a sua própria concepção de Interfaces Homem-Computador (IHC), sendo a experiência particular de interação com sistemas o principal fator que caracteriza o termo. Portanto, um determinado indivíduo pode acreditar que a interface é “o que ele vê na tela” (Mandel, 1997). Este é o motivo pelo qual muitas pessoas associam, de forma equivocada, o significado de IHC com os recursos das interfaces gráficas modernas (Raskin, 2000). Em contrapartida, Moran e Baecker apresentam definições mais consistentes:

“Restritamente definida, a interface homem-computador compreende os dispositivos de entrada e saída e o software utilizado; amplamente definida, ela inclui tudo que se adapta às experiências dos usuários com computadores, incluindo a documentação, o treinamento e o suporte.”

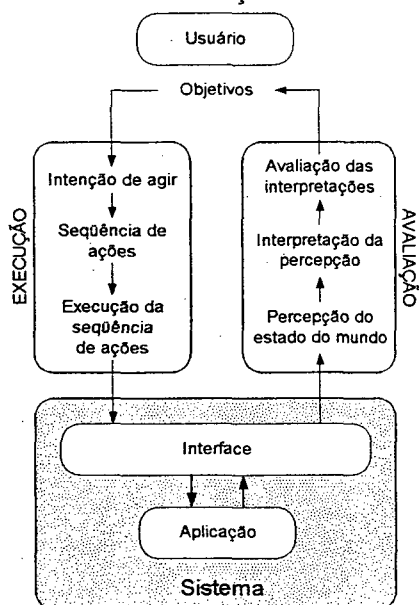
(BAECKER et al. apud MANDEL, 1995, p.14)

“A interface de usuário deve ser entendida como sendo a parte de um sistema computacional com a qual uma pessoa entra em contato física, perceptiva e conceitualmente.”

(MORAN apud DE SOUZA ET.AL., 1999, p.3)

O contato físico, perceptivo e conceitual expresso por Moran representa a interação do usuário com o sistema. Um modelo simplificado deste processo é apresentado por Norman (1990) como uma seqüência de sete estágios de ação (figura 1), onde estão presentes tanto as intenções de uso sobre a interface quanto as interpretações das respostas apresentadas pela mesma.

Figura 1: Processo de interação homem-computador.

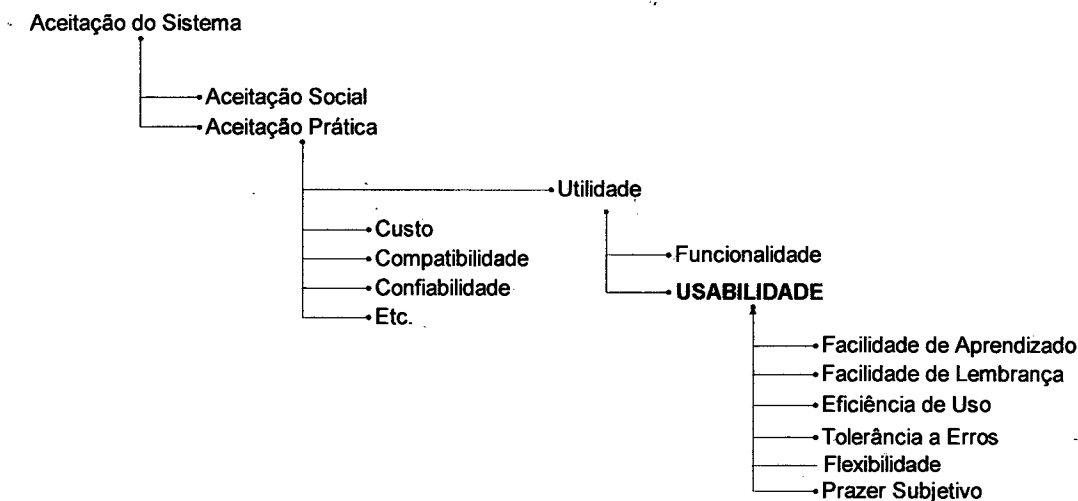


2.2 Usabilidade e comunicabilidade

A usabilidade é um conceito que se refere à qualidade de interação entre os usuários e sistemas, sendo apenas um entre vários fatores que influenciam a aceitação de um produto pelo usuário final (Nielsen, 1993). Observe-se a figura 2. A aprovação integral das características de um sistema depende fundamentalmente de dois aspectos principais: o social e o prático. Um jogo de computador pode ser socialmente inaceitável se apresentar níveis elevados de

violência para usuários infanto-juvenis. Um *site* fornecedor de serviços gratuitos na internet (ex: *web-mail*, páginas pessoais, listas de discussão) também terá pouca aceitação se, no instante de cadastro, um elevado número de informações desnecessárias for exigido. O receio de uso destas informações por terceiros (mala direta, pesquisas de mercado) diminui o interesse dos usuários.

Figura 2: Fatores de aceitação de um sistema.



Fonte: Adaptado de Nielsen (1993).

Além da aceitação social, existem também as questões correspondentes à aceitação prática. O custo, a compatibilidade, confiabilidade e estabilidade de um sistema são fatores essenciais neste quesito de aquiescência. A utilidade, por sua vez, determina o quanto um sistema está capacitado para que os usuários alcancem seus objetivos iniciais, dividindo-se entre (1) funcionalidade e (2) usabilidade¹ (Nielsen, 1993). A primeira trata da escolha das funções necessárias para a realização das tarefas desejadas, e a segunda trata do método de acesso das mesmas pelos usuários.

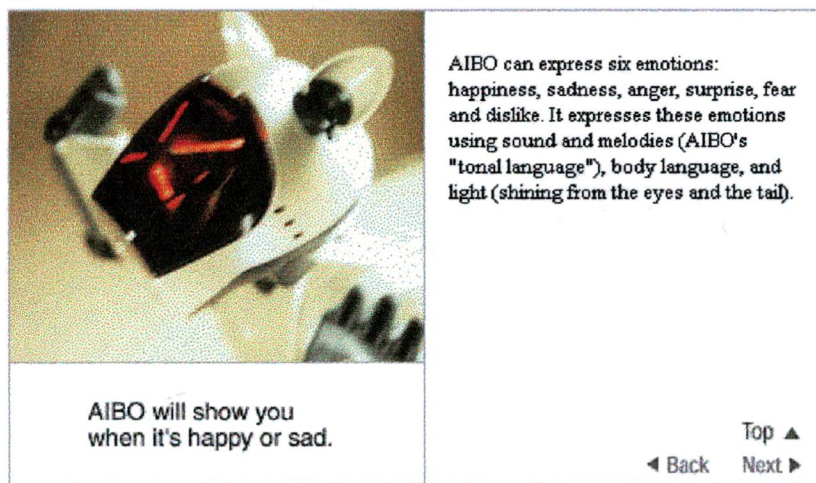
De forma abrangente, para um produto ser útil, ele não precisa estar associado ao conceito de “trabalho” (emprego) do usuário. A utilidade de um *software* educacional se refere à sua capacidade de auxílio no processo de

¹ Os termos interpretados do inglês são: *usefulness* - utilidade, *utility* - funcionalidade, e *usability* - usabilidade.

ensino e aprendizagem dos alunos, assim como a utilidade de jogos de computador se refere à garantia de entretenimento para quem os adquire.

Seja qual for a aplicação, o grande desafio dos profissionais da usabilidade corresponde à criação de novas tecnologias, capazes de explorar ao máximo as capacidades dos usuários, gerando produtos e ambientes mais eficazes e produtivos (Adler, Winograd, 1992). Nesse sentido, em oposição ao marketing apresentado por grandes empresas de informática (ex: Microsoft, Apple), o desenvolvimento de “interfaces amigáveis” é um requisito controverso. À exceção de alguns brinquedos eletrônicos modernos, como os pequenos *tamagochi* ou os robôs AIBO da Sony (figura 3), um usuário não procura sinais de amizade em uma máquina. O que lhe importa é ela auxiliá-lo em suas tentativas de completar tarefas (Nielsen, 1993). Além de o termo antropomórfico ser desnecessário, a facilidade de uso que ele denota nem sempre deve ser considerada como o principal aspecto da usabilidade. Sistemas *idiot-proof* (“à prova de idiotas”) minimizam o problema da ocorrência de erros, mas acabam por limitar as opções de ação ou decisão dos que se habilitam a operá-los (Adler, Winograd, 1992).

Figura 3: “AIBO vai mostrar para você quando ele estiver feliz ou triste”.



Fonte: [on-line] Disponível na World Wide Web: <<http://www.aibo.com/>>

A comunicabilidade representa outro importante aspecto no desenvolvimento de IHC. O projetista precisa buscar uma linguagem adequada

para transmitir as suas premissas e decisões de *design* para os usuários do sistema. A dependência de um contexto compartilhado entre o mensageiro e o receptor torna-se fundamental, devendo ser cuidadosamente estudada a pragmática dos sinais em uso (figura 4) (Mullet, Sano, 1995).

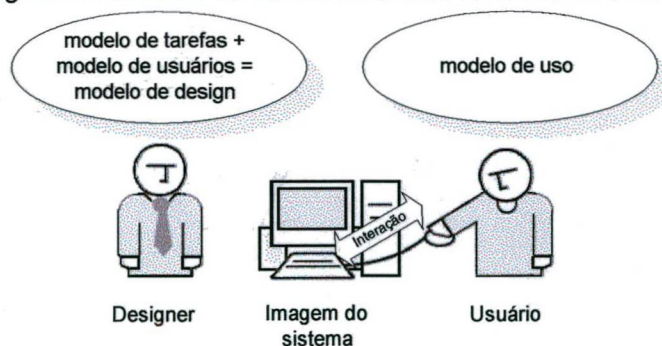
Figura 4: Comunicação visual através de ícones.



2.3 Engenharia cognitiva

Quando um *designer* pretende criar um produto, ele deve ter em mente qual é o tipo de usuário que irá utilizá-lo e quais são as tarefas que serão executadas. Esta visão particular do sistema pelo designer é denominada modelo de *design*. Uma vez implementado, o sistema apresenta uma imagem física própria, percebida pelo usuário através de interações. Se a imagem do sistema não for compatível com o modelo de uso gerado, o usuário terá problemas, sendo incapaz de prever os efeitos de suas ações. Este processo corresponde ao *design* centrado no usuário (*UCD*²) (figura 5).

Figura 5: Processo de desenvolvimento centrado no usuário.



Fonte: Adaptado de de Souza (1999).

Assim como a ergonomia, o processo de design centrado no usuário também considera a natureza estatística da variabilidade humana. Apesar de

cada ser humano ter características singulares, exigindo diferentes abordagens para a realização de tarefas, a população como um todo compartilha uma série de atributos mentais equivalentes. Portanto, antes de se preocupar com os requisitos individuais de cada usuário, o *designer* de um sistema precisa conhecer os aspectos fundamentais da cognição humana comuns a todos (Raskin, 2000). Como a mente funciona? Qual o processo pelo qual se adquire conhecimento? Como pensamos, lembramos e aprendemos? A aplicação das respostas para estas questões, no âmbito da criação de interfaces centradas no usuário, constitui o campo de estudo da engenharia cognitiva (Norman, 1990; Raskin, 2000).

“Nós precisamos dominar a ergonomia da mente se quisermos desenvolver interfaces que provavelmente funcionem bem. Por incrível que pareça, freqüentemente estamos cegos para nossos próprios limites mentais; devemos contar com a cuidadosa experimentação e observação para descobrir as fronteiras de nossas próprias habilidades mentais.”

(RASKIN, 2000, p.10)

O ser humano possui dois tipos de memória: a memória de curto termo (MCT) e a memória de longo termo (MLT). Para as informações serem assimiladas, elas precisam ser captadas pelos sentidos. Mesmo quando não se presta atenção em algo específico, estes estão sempre prontos para receber diferentes tipos de mensagens. Se uma informação é transmitida dos sentidos para o cérebro, ela se armazena primeiramente na MCT, a qual possui pequena capacidade de armazenamento, pois guarda aproximadamente sete itens elementares de informação (Miller apud Shneiderman, 1998). Além de limitada, a MCT é rapidamente volátil, dissipando seu conteúdo em 30 segundos ou menos, por causa de lembranças da MLT ou distrações do mundo físico. Para prolongar o tempo de armazenamento de informações neste tipo de memória, são utilizadas as estratégias de repetição e agrupamento (Mandel,

² O acrônimo da língua inglesa é conhecido por *UCD = Used Centered Design*.

1997; Shneiderman, 1998; Raskin, 2000). Nem sempre o conteúdo da MCT passa para a MLT. Esta passagem também depende das estratégias citadas acima.

A MLT possui grande capacidade de armazenamento, sendo praticamente ilimitada, mas o processo de aquisição e recuperação de informações não é confiável. Ou seja, o conhecimento pode ser retido por um longo período, mas o seu retorno da MLT para a MCT não é certo.

A tabela 1 mostra dois cenários possíveis para a construção da interface de um sistema. O primeiro não considera o conhecimento do usuário sobre a aplicação, pois as informações necessárias para a realização das tarefas estão disponíveis em sua própria interface (conhecimento no mundo). O uso da MCT é suficiente neste caso, caracterizando o perfil de usuários iniciantes. No segundo cenário a situação se inverte, exigindo que o usuário tenha um conhecimento prévio de uso residente em sua MLT (conhecimento na cabeça). Somente usuários experientes são capazes de lidar com interfaces pouco intuitivas.

Tabela 1: Conhecimento no mundo e conhecimento na cabeça.

Propriedade	Conhecimento no mundo	Conhecimento na cabeça
Habilidade de recuperação	<ul style="list-style-type: none"> Recuperável sempre que é visível ou audível 	<ul style="list-style-type: none"> Não é facilmente recuperável. Requer uma lembrança ou busca pela memória.
Aprendizado	<ul style="list-style-type: none"> O aprendizado é desnecessário. A interpretação da informação depende de como o produto explora seus mapeamentos e restrições naturais. 	<ul style="list-style-type: none"> Requer aprendizado, o qual pode ser considerável. O aprendizado torna-se fácil se houver um significado na estrutura do material (ou se existir um bom modelo mental).
Eficiência de uso	<ul style="list-style-type: none"> Tende a ser diminuída pela necessidade do usuário ter de encontrar e interpretar a informação externa. 	<ul style="list-style-type: none"> Pode ser bastante eficiente.
Facilidade de uso ao primeiro contato	<ul style="list-style-type: none"> Alta. 	<ul style="list-style-type: none"> Baixa.
Estética	<ul style="list-style-type: none"> Pode ser desagradável ou deselegante, especialmente se existir a necessidade de manutenção de várias informações. No final, o apelo estético depende do talento do designer. 	<ul style="list-style-type: none"> Nada precisa estar visível. Este fato traz maior liberdade para o designer, que pode dar mais atenção para o tratamento estético.

Fonte: Norman(1990).

Outro importante aspecto mental do ser humano se refere às suas atividades conscientes e inconscientes. Embora uma pessoa seja capaz de realizar um certo número de tarefas ao mesmo tempo, somente uma delas pode ser executada de forma consciente. Mas com a repetição e a conseqüente formação de hábitos, os atos conscientes tornam-se inconscientes, permitindo o processamento paralelo. Entretanto, ao contrário do que se possa esperar, nem sempre este processo resulta em benefícios. Inúmeras falhas de interfaces estão associadas à dificuldade de se reverter hábitos adquiridos.

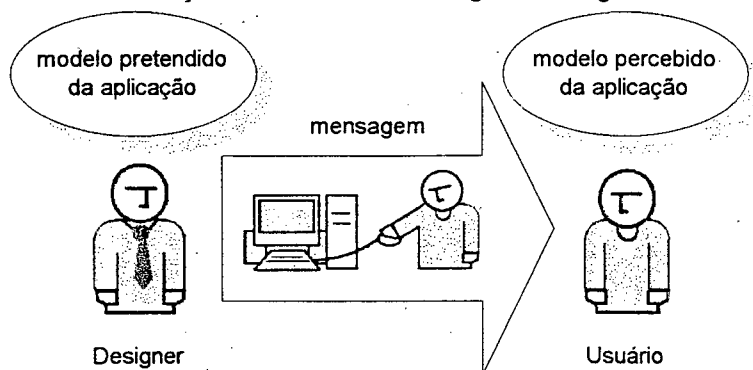
2.4 Engenharia semiótica

A engenharia semiótica considera a interface de um sistema como uma mensagem unilateral enviada pelo designer ao usuário (de Souza, 1993). Além disso, a própria interface é o elemento de interação com o usuário, caracterizando o seu aspecto de meta-comunicação. Ou seja, a própria mensagem que o designer envia ao usuário é capaz de trocar mensagens com o mesmo (figura 6).

“Ao trazer o designer para dentro do foco, a Engenharia Semiótica evidencia a sua presença e permite ao usuário entender que todo o sistema é uma solução potencial de um designer (ou de uma equipe de *design*). Assim, o usuário, ao ter problemas de interação com a aplicação, pode tentar entender o que o designer pretendia, e acertar o seu modelo mental da aplicação, aproximando-o cada vez mais daquele do designer. Fazendo isto, o usuário é capaz de alcançar um melhor entendimento das motivações e decisões tomadas pelo designer, e assim usar a aplicação de forma mais eficiente.”

(DE SOUZA, 1999, p.20)

Figura 6: Comunicação entre usuário e designer na engenharia semiótica.



Fonte: Adaptado de de Souza (1999).

2.5 Gestos e modos de operação

Segundo Raskin (2000), uma definição simples para modo depende do conceito de gesto, que por sua vez significa uma seqüência ininterrupta de ações humanas para a execução de uma tarefa. Por exemplo, um usuário experiente estará realizando um único gesto quando:

1. guiar o cursor em cima de um ícone;
2. selecioná-lo (pressionado o botão esquerdo do *mouse*);
3. arrastá-lo de uma pasta para outra;
4. deixá-lo (soltando o botão esquerdo do *mouse*).

Simultaneamente, para um usuário iniciante, cada uma das 4 ações isoladas também representará um gesto. A maneira pela qual a interface reage a um determinado gesto é o que caracteriza seu modo. Por exemplo, se um gesto G resulta em uma ação A_x , então o modo M_x está selecionado. Se em outro instante o mesmo gesto G resulta numa ação A_y diferente, é porque M_x não está mais ativo. Conforme a tabela 2 indica, outro modo terá sido selecionado (M_y).

Tabela 2: Modos de operação diferentes.

Modo	Gesto \rightarrow Ação
M_x	$G \rightarrow A_x$
M_y	$G \rightarrow A_y$

Se a interpretação de um gesto depende da visão do usuário em relação à interface, então é possível afirmar que um determinado recurso pode ser modal para um usuário e não modal para outro. Ao enunciar uma definição mais completa sobre o conceito de modos, torna-se fundamental a descrição deste aspecto de dependência. Raskin (2000) explica que uma interface homem-máquina é modal em relação a um gesto dado quando:

1. o estado corrente da interface não é o foco de atenção do usuário;
2. a interface irá executar uma entre várias possíveis respostas diferentes para esse gesto, dependendo do estado corrente do sistema.

No próximo capítulo, no item sobre discussões e problemas de interfaces conhecidas, o conceito de modos servirá como base para a explicação de algumas características de interface dos programas de geometria dinâmica. Em geral, modos levam a erros e devem ser evitados sempre que possível (Lewis & Rieman, 1994; Nielsen, 1993; Norman, 1990; Raskin, 2000).

2.6 Estilos de Interação

Estilos de interação são formas de comunicação ou interação entre usuários e sistemas computacionais (Shneiderman, 1998). Alguns exemplos são:

- WYSIWYG (*what you see is what you get*);
- ícones;
- manipulação direta;
- seleção de menus;
- linguagem de comandos;
- preenchimento de formulários;
- linguagem natural e diálogo de perguntas e respostas.

Entre os estilos apresentados acima, Foley et al. (1996) considera os três primeiros essencialmente gráficos, enquanto Shneiderman (1998) classifica apenas cinco como sendo os principais (tabela 3). Geralmente, os desenvolvedores de sistemas fazem uso de vários estilos em conjunto, de maneira a adequá-los para diferentes usuários e tarefas (tabela 4).

Tabela 3: Vantagens e desvantagens de cinco estilos.

Vantagens	Desvantagens
Manipulação direta	
<ul style="list-style-type: none"> • apresenta visualmente os conceitos das tarefas; • permite fácil aprendizado; • permite fácil retenção (memorização); • permite que erros sejam evitados; • encoraja a exploração; • causa alta taxa de satisfação subjetiva. 	<ul style="list-style-type: none"> • pode ser de difícil programação; • pode requerer dispositivos específicos para visualização.
Seleção de menus	
<ul style="list-style-type: none"> • encurta o aprendizado. • reduz o uso do teclado; • estrutura a tomada de decisão; • permite fácil tratamento de erros; • permite o uso de ferramentas para gerenciamento de diálogo. 	<ul style="list-style-type: none"> • apresenta o perigo de muitos menus; • pode retardar os usuários experientes; • consome espaço na tela; • requer uma rápida taxa de atualização do <i>display</i>.
Preenchimento de formulários	
<ul style="list-style-type: none"> • simplifica a entrada de dados; • requer pouco treinamento; • provê assistência adequada; • permite o uso de ferramentas para gerenciamento de formulários. 	<ul style="list-style-type: none"> • consome espaço na tela.
Linguagem de comandos	
<ul style="list-style-type: none"> • é flexível; • agrada aos usuários experientes; • suporta as iniciativas do usuário; • permite a criação de macros. 	<ul style="list-style-type: none"> • possui tratamento de erro precário; • requer treinamento substancial e memorização.
Linguagem natural	
<ul style="list-style-type: none"> • alivia a carga de aprendizado de uma nova sintaxe. 	<ul style="list-style-type: none"> • requer um diálogo claro e específico. • pode exigir um uso intensivo do teclado. • não mostra o contexto (o usuário pode não ter noção das capacidades e limitações do sistema). • é imprevisível.

Fonte: Shneiderman (1998)

³ O que você vê é o que você obtém.

Tabela 4: Comparação de sete estilos.

	Tempo de aprendizagem	Velocidade de uso	Predisposição a erros	Extensibilidade ⁴	Habilidade de digitação requerida
WYSIWYG ⁵	Baixo	-	Baixa	Baixa	-
Manipulação direta	Baixo	Média	Baixa	Baixa	Nenhuma
Seleção de menus	Médio	Média	Baixa	Média	Nenhuma
Preenchimento de formulários	Baixo	Alta	Baixa	Média	Alta
Linguagem de comandos	Alto	Alta	Alta	Alta	Alta
Linguagem natural	Baixo	Média	Alta	Alta	Alta ⁶
Diálogo de perguntas e respostas	Baixo	Baixa	Baixa	Alta	Alta

Fonte: Foley et. al, (1996).

Os programas de geometria dinâmica, abordados em detalhes no próximo capítulo, se utilizam dos estilos:

- WYSIWYG - a representação do documento ou modelo com o qual o usuário interage na tela é basicamente a mesma gerada pela aplicação; os códigos que representam as características de cada objeto no formato de arquivo são mascarados;
- manipulação direta - poderoso e fácil de aprender, este estilo apresenta relações, objetos e atributos que podem ser selecionados e operados com o auxílio de um *mouse* ou qualquer outro tipo de dispositivo de entrada gráfico (*GID*); as ações realizadas sobre os elementos visuais denotam comandos que são ativados implicitamente;
- ícones - um ícone significa um objeto em função de sua própria semelhança com ele; não somente objetos mas também ações ou até mesmo propriedades também podem ser representadas visualmente em forma de ícones; Mullet & Sano (1995) apresentam cinco princípios básicos que devem ser considerados para alcançar o resultado de uma analogia adequada: imediação, generalidade, coerência, caracterização e comunicabilidade;

⁴ Se refere ao fato de o desenvolvedor acrescentar novos comandos, novas operações.

⁵ WYSIWYG tem vários campos em branco pois não é um estilo completo de interface, ele precisa estar acompanhado de alguma forma de entrada de comandos.

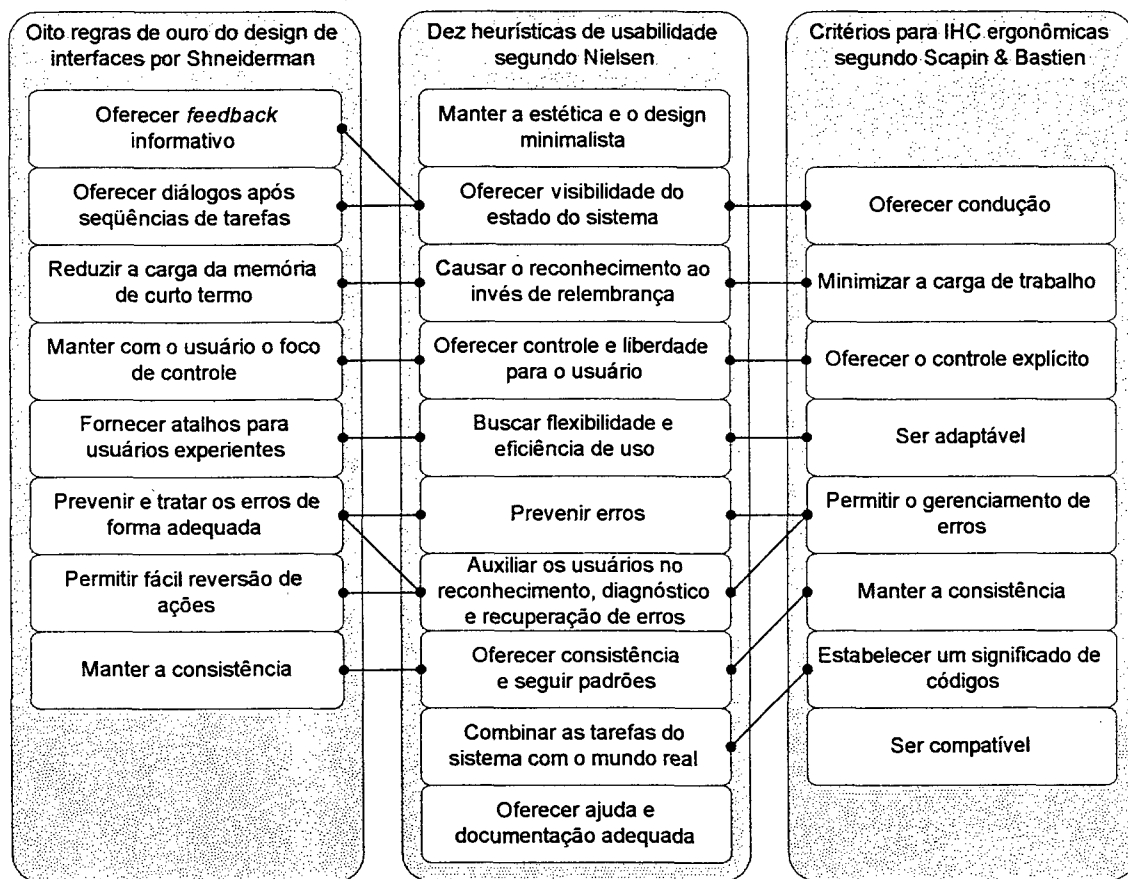
⁶ Assumindo a entrada de dados pelo teclado, não pelo reconhecimento de voz.

- seleção de menus - em sistemas que usam seleção de menus, os usuários visualizam uma lista de itens, escolhem o mais apropriado para a tarefa desejada, e observam o efeito resultante (Shneiderman, 1998); uma das vantagens do uso de menus se refere à possibilidade que os usuários têm de trabalhar com a sua memória de reconhecimento, onde as imagens visuais são associadas a palavras e significados já conhecidos (Foley et al., 1996);
- preenchimento de formulários - quando é necessária a entrada de dados exatos em formato alfanumérico, o preenchimento de formulários representa a solução mais adequada; campos de texto são preenchidos pela movimentação de um cursor e com o auxílio de um teclado;
- linguagem de comandos – anteriormente à era dos sistemas gráficos, este estilo constituía o caminho mais comum de interagir com computadores; para um usuário efetuar uma ação, devem ser digitados o comando que a representa e também os parâmetros de controle que determinam o seu comportamento.

2.7 Heurísticas fundamentais

Seja qual for a natureza de uma interface, ela deve ser elaborada através de um processo iterativo de *design* que se baseia em sólidos princípios ergonômicos. Estes critérios, derivados heurísticamente de experimentos, servem para a validação e o refinamento recorrente dos estilos de diálogo, interação e navegação dos sistemas. Vários autores sugerem heurísticas, como por exemplo: Hansen (1971), Rubenstein & Hersch (1984), Mayhew (1992), Bastien & Scapin (1993), Nielsen (1993) e Shneiderman (1998). A figura 7 mostra os itens abordados pelos três últimos autores, e também as suas associações mais representativas.

Figura 7: Heurísticas para a avaliação de interfaces segundo diferentes autores e suas relações diretas.



Como todos expõem os requisitos da usabilidade e tratam basicamente dos mesmos temas, apenas as heurísticas propostas por Nielsen (1993) são apresentadas abaixo:

1. visibilidade do estado do sistema – o usuário precisa estar ciente da situação do sistema; cada gesto seu deve receber um sinal de retorno, com o propósito de assegurá-lo do processamento efetivo de suas ações;
2. combinação entre o sistema e o mundo real – as informações devem ser apresentadas num formato intuitivo e de fácil compreensão; através de metáforas e mapeamentos adequados, auxilia-se o usuário na

- percepção dos controles e de seus resultados, das ações e de seus efeitos;
3. controle e liberdade do usuário – ao escolher uma função por engano, o usuário pode ficar preso em um modo de operação; para evitar este tipo de problema, é necessário que a interface ofereça uma “saída de emergência” (responsável pela transição do estado atual para o estado anterior);
 4. consistência e padrões – a formação de hábitos é uma característica inerente do ser humano; usuários acostumados com o uso de programas similares apresentam maior satisfação e produtividade quando suas convenções são herdadas pela interface em questão;
 5. prevenção de erros – antes de se preocupar com as mensagens relacionadas a erros, o projetista precisa buscar alternativas para minimizá-los; a ausência de modos de operação e o uso coerente dos estilos de interação podem auxiliar neste caso;
 6. reconhecimento ao invés de relembração – as opções, objetos e ações devem estar visíveis; o conhecimento das características das MCT e das MLT deve ser corretamente aplicado; um usuário, que deixar de utilizar o sistema durante um determinado período de tempo, precisa retornar ao seu uso sem a necessidade de um novo aprendizado;
 7. flexibilidade e eficiência de uso – refere-se à possibilidade de o usuário acrescentar ou modificar as funcionalidades do sistema, personalizando-o e utilizando-o de maneira inteligente e criativa; procura-se também permitir o uso de atalhos para que os usuários “experientes” possam obter resultados de uma forma mais rápida;

8. estética e *design* minimalista – os diálogos devem ser de natureza clara e objetiva; mensagens desnecessárias devem ser evitadas; quanto maior for a quantidade de informações irrelevantes em um objeto, maiores serão as chances de um usuário interpretá-lo equivocadamente, desviando seu foco de atenção e causando erros;
9. ajuda para reconhecimento, diagnóstico e recuperação de erros – qualquer mensagem de erro precisa estar numa linguagem clara e objetiva, apontando a raiz do problema e sugerindo construtivamente a sua solução; a recuperação de informações em caso de acidentes é igualmente necessária;
10. ajuda e documentação – apesar do desejável uso da interface sem o apoio de documentação adicional, julga-se importante que a mesma apresente algum tipo de auxílio (interno e/ou externo); as informações devem ser facilmente descobertas, sendo objetivas e diretamente relacionadas às tarefas em questão.

2.8 Métodos para a avaliação de interfaces

Existem várias maneiras de se avaliar a usabilidade em interfaces centradas no usuário. Qualquer que seja o método utilizado, ele deve se enquadrar em alguma destas categorias: (1) avaliação heurística, (2) avaliação por testagem e (3) práticas alternativas ou complementares.

2.8.1 Avaliação heurística

Embora este tipo de avaliação seja efetuado como parte do processo iterativo de *design* centrado no usuário, ele não exige a participação de usuários, mas de um conjunto de avaliadores, que examinam a interface e a

julgam conforme os princípios de usabilidade denominados “heurísticas⁷” (Nielsen, 1993).

Por mais experiente que seja, um avaliador isolado não é capaz de descobrir todos os problemas de usabilidade existentes numa interface. Além disso, diferentes pessoas podem encontrar diferentes problemas. Assim, justifica-se o envolvimento de múltiplos avaliadores como uma forma de se aumentar a eficiência deste método. Primeiro, cada um deles examina a interface isoladamente. Após todas as avaliações pessoais terem sido concluídas, eles se comunicam entre si. Trabalha-se desta forma para se garantir a imparcialidade e a independência dos resultados isolados.

2.8.2 Avaliação por testagem

Segundo Rubin (1994), a avaliação por testagem é uma ferramenta de pesquisa com raízes na metodologia experimental clássica. Alguns usuários são escolhidos para participar de tarefas, interagindo com a interface enquanto são observados por avaliadores em um laboratório de usabilidade. Idealmente, um laboratório desta categoria deve ser equipado com câmeras de vídeo, visando capturar as ações e reações dos usuários no processo de interação com o sistema, assim como os diferentes estados da interface. Espelhos falsos também são usados para se garantir a observação dos usuários de forma a minimizar a presença “intrusiva” dos avaliadores (figura 8). Entretanto, para testes menos rigorosos ou de custo reduzido, não há necessidade de laboratórios tão sofisticados. Câmeras filmadoras convencionais, gravadores ou até mesmo papel e lápis podem ser utilizados para o registro dos acontecimentos.

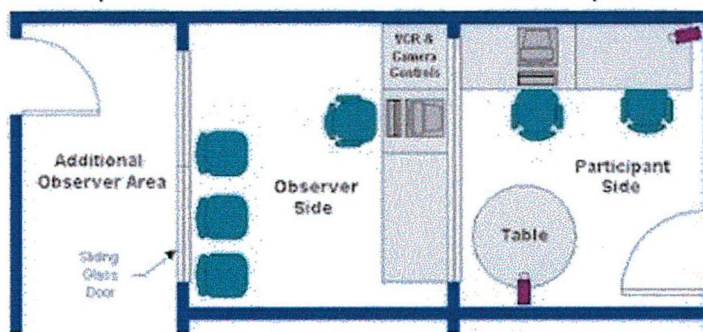
Por mais rigorosa que seja a forma de conduzir este tipo de avaliação, não se pode afirmar que um produto estará totalmente livre de falhas de usabilidade

⁷ Como as apresentadas no item anterior (2.5).

em virtude da ocorrência de resultados positivos. Rubin (1994) categoriza algumas das razões para este fato:

- uma testagem é sempre uma situação artificial;
- os resultados dos testes não provam que um produto funciona;
- os participantes raramente representam com exatidão o público-alvo;
- nem sempre a testagem é a melhor técnica a ser aplicada.

Figura 8: Tipo de laboratório de usabilidade utilizado pela Microsoft.



Fonte: [on-line] Disponível na World Wide Web: <<http://www.microsoft.com/usability/tour.htm>>.

Apesar do alto custo, das possíveis falhas e das armadilhas de uso, os resultados da avaliação por testagem são de grande valia para a descoberta de erros de usabilidade. Tal método não pretende substituir a avaliação heurística, mas complementá-la de alguma forma. Os melhores resultados surgem quando estes dois métodos são executados em diferentes etapas do ciclo iterativo de *design*, e as informações obtidas são compiladas posteriormente. Além destes, vários outros métodos também podem ser aplicados. Estas alternativas estão categorizadas adiante.

2.8.3 Práticas alternativas ou complementares

Além dos métodos apresentados anteriormente, existem várias outras alternativas para a avaliação da usabilidade, como por exemplo:

- o uso de questionários é aplicado quando se deseja identificar o perfil de usuários e verificar o grau de satisfação subjetiva dos mesmos em relação à interface;
- as entrevistas, quando realizadas adequadamente, derivam mais informações que os questionários pois são observadas as reações físicas e emocionais dos usuários correspondentes às perguntas efetuadas;
- os relatos de usuários sobre a ocorrência de incidentes críticos também devem ser incentivados para se descobrir a frequência com que eles ocorrem, como eles podem ser evitados e solucionados (Castilho, Hartson, Hix, 1998);
- os arquivos de *logs* gerados contêm as características sobre as interações durante o uso de sistemas, apesar da complexa filtragem dos dados que representam as informações efetivamente úteis;

2.9 Conclusão do capítulo 2

O conceito de interface não se caracteriza somente pelo aspecto visual de um produto, pois o recurso de interação também faz parte do contexto. Além disso, para se garantir a qualidade de interação entre o indivíduo e o sistema, é necessário que a usabilidade e a comunicabilidade sejam fatores de destaque. Para tanto, o projetista deve recorrer ao estudo e à aplicação das teorias da engenharia cognitiva e da engenharia semiótica. Conduzir o desenvolvimento centrado no usuário é uma tentativa aproximada de se obter sucesso neste empreendimento. Idealmente, a mensagem enviada pelo projetista (constituída pela própria interface) deve ser recebida e interpretada pelo usuário sem interferências ou perdas de informação.

Neste capítulo, um conjunto de recomendações para se elaborar interfaces (heurísticas) foi apresentado, servindo de base para a avaliação posterior dos programas em uso. Entretanto, a criação da interface de um *software* de geometria dinâmica engloba diversos aspectos, usualmente conflitantes, referentes às áreas da matemática, da computação e da pedagogia. O próximo capítulo pretende (em parte) levantar a importância de se observar estes fatores, definindo e mostrando alguns dos recursos, aplicações e benefícios da geometria dinâmica na atualidade.

3 ESTADO DA ARTE DA GEOMETRIA DINÂMICA

3.1 Geometria euclidiana

A geometria euclidiana se refere ao matemático grego Euclides, que viveu entre os anos de 330 a 277 a.C aproximadamente. Em sua obra intitulada *Elementos*, composta de postulados, axiomas e teoremas, Euclides procurou sistematizar o saber geométrico, baseando-se em trabalhos de outros matemáticos famosos, como por exemplo: Hipócrates, Platão e Pitágoras.

Considere-se o método tradicional de criar desenhos geométricos utilizado há pelo menos 2000 anos e com o qual se trabalha até hoje: que propriedades se observam nas figuras geradas sobre o papel, com o auxílio de ferramentas tais como lápis, régua, esquadros e compasso? Segundo Scher (2000), as construções resultantes destes instrumentos apresentam duas características principais:

- são estáticas, pois não podem ser alteradas sem o auxílio da borracha e de um novo traçado;
- são particulares, pois nenhuma delas garante o significado genérico de sua definição (o desenho de um círculo, por exemplo, possui um centro e raio fixos, mas o conceito de círculo não depende de valores arbitrários).

Conseqüentemente, estes fatores tornam-se condições desfavoráveis para a prova de teoremas e a descrição de conceitos. Alguns possíveis efeitos colaterais podem ser observados:

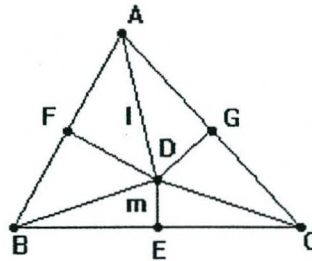
- a sugestão de hipóteses além das necessárias;
- a incapacidade de representar outras soluções válidas;

- a indução a resultados absurdos por um esboço inapropriado, pela complexidade do desenho ou pela imprecisão das ferramentas.

Como confiar na visualização de representações estáticas? Os matemáticos alertam para os perigos. A prova falaciosa de que “todos os triângulos são isósceles” mostrada a seguir é um bom exemplo. Partindo-se do esboço de um triângulo ABC “qualquer” (figura 9) são executados os passos:

- traçado de l como a bissetriz do ângulo A ;
- traçado de m , mediatriz do segmento BC com E como o ponto médio deste;
- construção de D como o ponto de interseção de l e m ;
- traçado das perpendiculares aos lados AB e AC (por D), cortando-os em F e G ;
- traçado dos segmentos DB e DC .

Figura 9: Esboço de um triângulo ABC “qualquer”.



$\triangle ADF \cong \triangle ADG$ (ângulo, ângulo, segmento), então $AF \cong AG$ e $DF \cong DG$.

$BD \cong CD$ por construção pois m é mediatriz de BC . Isto implica em

$\triangle BDF \cong \triangle CDG$ (hipotenusa, cateto), então $FB \cong GC$. Assim,

$AB \cong AF + FB = AG + GC \cong AC$, e portanto $\triangle ABC$ é isósceles!

Na realidade, a interseção de l com m nunca determina um ponto D interno ao triângulo ABC . Apesar de a dedução estar correta, ela só é válida para uma configuração específica, que por sua vez não está aparente nesta figura de

análise. Como evitar este tipo de “armadilha”? Uma alternativa será mostrada na próxima seção, após ter sido definido o conceito de geometria dinâmica.

3.2 Geometria dinâmica

A geometria dinâmica é uma nova proposta que visa explorar os mesmos conceitos da geometria clássica, porém, através de um programa interativo (King, Schattschneider, 1997; Winroth, 1999). As principais características de um software de geometria dinâmica são apresentadas abaixo.

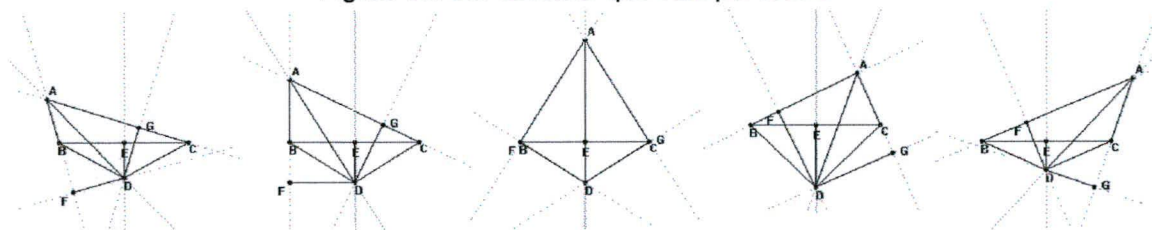
- A interface é baseada em *WIMP*⁸, com ênfase no estilo de interação em manipulação direta. Os elementos geométricos podem ser transformados de forma interativa, isto é, ao controle do *mouse*, pelo ato de ‘clique’ e arrastar, os objetos criados podem ser reescalados, transladados e rotacionados.
- É um *sistema baseado em restrições* (Foley et al., 1996; Winroth, 1999). Uma instância isolada de um objeto geométrico na tela representa uma classe completa de objetos com a mesma definição. Um quadrado na tela é estático, mas se um de seus vértices for movido, ele também mudará de aparência. Mesmo assim, as propriedades da definição de um quadrado serão mantidas, ou seja, todos os lados terão comprimentos iguais e os ângulos medirão 90 graus. Como no mundo físico real, muitos objetos se movem de forma dependente das condições impostas por outros objetos. Conceitos como paralelismo, perpendicularidade e pertinência a lugares geométricos, entre outros, permitem a construção de elementos que dependem de regras preestabelecidas. Um exemplo é apresentado na figura 10, onde aparece a seqüência da animação de um sistema articulado com duas barras.

Figura 10: Quando C gira, D também gira e conseqüentemente o ponto E se desloca.



Em relação ao caso da prova dos triângulos isósceles (ver seção 3.1), conclui-se que, com o auxílio de um programa de geometria dinâmica, pode-se obter não somente um desenho com maior precisão, mas também “um que vale por todos” (figura 11). Através da manipulação direta, várias configurações são alcançadas. O usuário acaba verificando, de forma experimental⁹, que o único caso válido para sua resposta é aquele onde $B=F$ e $C=G$, coincidindo m com l .

Figura 11: Um desenho que vale por todos.



3.3 Programas conhecidos

De acordo com Winroth (1999), desenvolvedor do *software* PDB (*Projective Drawing Board*), os programas construtivistas¹⁰ de geometria dinâmica que mais se destacam na atualidade são os seguintes:

⁸ WIMP é um acrônimo da língua inglesa que significa: janelas, ícones, menus e apontador (GID).

⁹ Com a ferramenta de manipulação selecionada, o usuário é capaz de movimentar o ponto A, observando-o em várias posições diferentes. Como conseqüência, todos os outros elementos se modificam em função deste ponto.

¹⁰ Nos moldes de Seymour Papert, baseado em Jean Piaget e Lem Vigotsky (Soloway, Guzdial, Hay, 1994; Maddux, Johnson, Willis, 1997).

- The Geometer's Sketchpad – este programa surgiu do Projeto de Geometria Visual (VGP), adotado pela universidade Swarthmore em meados de 1980 e coordenado por Eugene Klotz e Doris Schattschneider. A proposta inicial tinha como objetivo a geometria tridimensional, mas por causa das limitações dos processadores disponíveis na época, acabou se restringindo à geometria plana. Nicholas Jackiw, o estudante de graduação selecionado para programá-lo, tornou-se o principal desenvolvedor da aplicação. Atualmente, o Geometer's Sketchpad se encontra em sua quarta versão¹¹ e é comercializado pela empresa Key Curriculum Press;
- Cabri Géomètre – a criação do Cabri foi proposta em 1985 por Jean Marie Laborde como um livro de exercícios de geometria euclidiana. Seu desenvolvimento original foi realizado por um grupo de pesquisadores do laboratório Leibnitz do Instituto de Informática e de Matemática Aplicada em Grenoble (IMAG), na França. Embora o desenvolvimento continue sob responsabilidade do IMAG, ele é atualmente comercializado pela Texas Instruments, que disponibiliza a sua versão mais recente - Cabri Géomètre II, tanto para o uso em computadores pessoais quanto para um modelo específico de calculadoras;
- Cinderella – originalmente concebido por Henry Crapo e Jürgen Richter-Gebert, este programa é o resultado da continuação de três projetos realizados entre 1993 e 1998. Comercializado pela Springer-Verlag e embora não tão popular quanto os seus antecessores, o Cinderella apresenta características singulares, tais como: a precisão matemática, o suporte para múltiplas geometrias, a independência de plataforma e a verificação automática de teoremas. Não por acaso, Burgiel (1999) afirma tratar-se de um programa voltado para matemáticos, feito por matemáticos.

¹¹ Este trabalho é baseado na terceira versão do programa The Geometer's Sketchpad. A menos de um mês do término desta pesquisa foi lançada a sua quarta versão no mercado.

Outros programas de geometria dinâmica têm surgido nesta última década. Alguns exemplos são: Dr.Genius (sucessor do Dr.Geo), Euklid, Géo Specif, Geometric Inventor, Geometric Supposer, Juno 2, PDB, Tabulae e Uni-Géom.

3.4 Benefícios e aplicações

King e Schattschneider (1997) classificam os benefícios e as aplicações da geometria dinâmica em oito tópicos, relacionados abaixo:

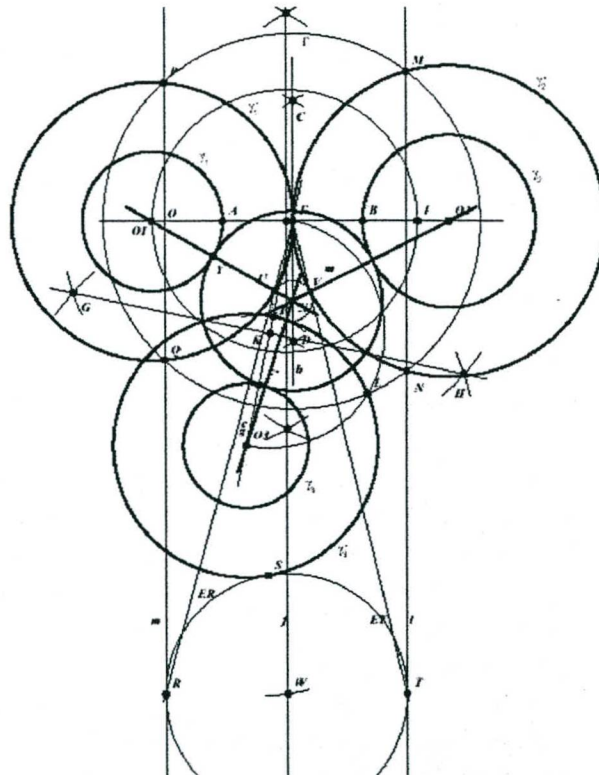
- precisão de construção;
- visualização;
- exploração e descoberta;
- prova;
- transformações;
- *loci*;
- simulação;
- micromundos.

3.4.1 Precisão de construção

Quando pesquisadores introduziram o conceito da geometria dinâmica, o principal objetivo era a criação de um programa onde os instrumentos tradicionais de traçado pudessem ser substituídos. Desenhos precisos seriam criados no computador, mas retornariam para o papel por meio de impressão. A determinação exata de pontos de interseção entre retas e outros elementos geométricos não dependeria das habilidades manuais do usuário, e sim das características do sistema em uso.

Atualmente, a geometria dinâmica amplia seus horizontes, porém permanece o fator inicial de precisão. No papel, desenhos que demandam inúmeros traçados, tornando-se cansativos, são de fácil criação no computador (figura 12). Entretanto, algumas limitações persistem. As mais simples se relacionam com a capacidade de processamento da máquina e à resolução do monitor ou da impressora. As complexas se referem às ambigüidades de soluções não triviais, exigindo uma fundamentação matemática superior e abrangente (Kortenkamp, 1999).

Figura 12: Problemas de complexidade e imprecisão no papel são contornados através da geometria dinâmica.



Fonte: Kortenkamp (1999)

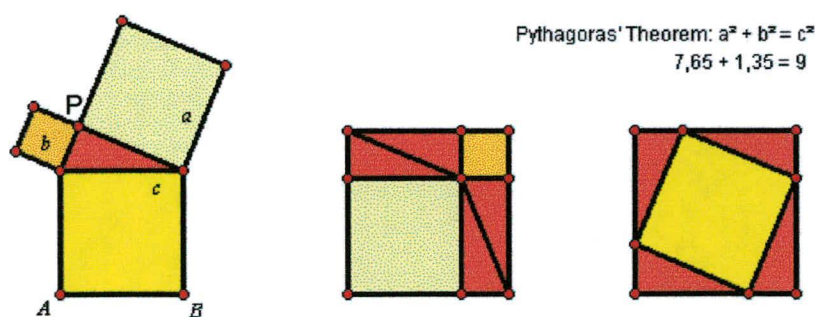
3.4.2 Visualização

Há oito anos, Soloway, Guzdial e Hay (1994) apresentavam um conceito inovador: o *design* de interfaces voltado para o aprendiz. Era o momento de os pesquisadores em interfaces homem-computador buscarem soluções para novas interfaces, capazes de tornar os usuários mais habilitados pelo próprio

processo de interação. Os recursos de visualização presentes na geometria dinâmica constituem, assim, um bom produto desta abordagem.

A visualização auxilia o raciocínio intuitivo, representando uma primeira etapa para a compreensão de conceitos e a prova de teoremas. Na figura 13 é apresentado um exemplo interativo, em que se demonstra o teorema de Pitágoras através de três figuras que são modificadas simultaneamente: a primeira expõe três quadrados com as respectivas áreas a^2 , b^2 e c^2 , a segunda exhibe um quadrado composto cuja área é igual a $a^2 + b^2 + 2(ab)$ e a terceira mostra outro, de área visivelmente equivalente ao anterior e cujo valor corresponde a $c^2 + 2(ab)$. Assim sendo, $a^2 + b^2 + 2(ab) = c^2 + 2(ab)$, ou seja, $a^2 + b^2 = c^2$. A movimentação do ponto P sobre o arco capaz de 90° escondido evidencia que o teorema de Pitágoras é válido para quaisquer valores de a e de b .

Figura 13: Visualização do teorema de Pitágoras.



Fonte: [on-line] Disponível na World Wide Web: <<http://www.cinderella.de/en/demo/gallery/pythagoras.html>>.

3.4.3 Exploração e descoberta

“A natureza interativa e as qualidades dinâmicas do *software* levam os estudantes a proporem suas próprias conjecturas e testarem-nas eficientemente. O retorno que os alunos podem obter por utilizarem o *software* de geometria dinâmica é eficiente e empolgante. Estudantes podem adquirir uma melhor percepção e compreensão visual da matemática que eles estão investigando.”

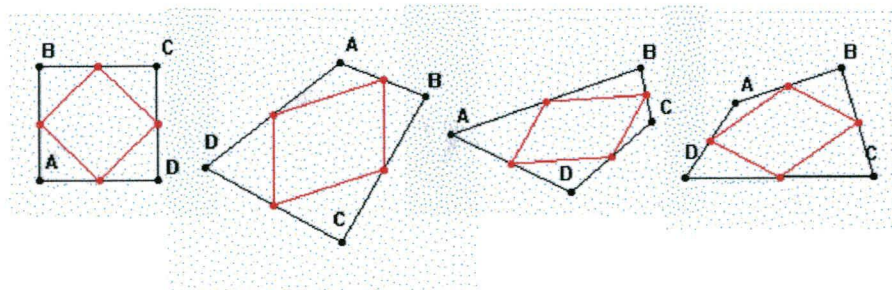
(GARRY, 1997, p.60)

Na geometria dinâmica, o aprendizado não se baseia num processo de cópia. Nem as definições de teoremas, nem os resultados de problemas e provas devem ser assimilados pelo método comportamentalista tradicional, onde o professor explica, o aprendiz anota, e o conhecimento vai sendo supostamente depositado na massa encefálica. Justifica-se o uso de uma abordagem construtivista pela natureza da informação que é assimilada.

Tome-se um exemplo clássico apresentado por Foley et al. (1996) e Scher (2000). Desenha-se um quadrilátero $ABCD$ qualquer e também os pontos médios de AB , BC , CD e AD . Quando os pontos médios dos lados vizinhos são ligados, obtém-se outro quadrilátero, inscrito no primeiro. Tal construção pode levar ao seguinte questionamento: independente das posições ocupadas pelos pontos A , B , C e D , o quadrilátero interno possui alguma propriedade especial?

A figura 14 apresenta uma série de configurações diferentes dos pontos A , B , C e D . Com o auxílio de um *software* de geometria dinâmica, a solução deste problema parece “saltar aos olhos” do usuário. Cria-se uma única vez o diagrama proposto para depois, por exploração, descobrir intuitivamente que o quadrilátero inscrito mantém o formato de paralelogramo.

Figura 14: Exploração e descoberta.



3.4.4 Prova

“Proponentes da visualização matemática auxiliada por computador argumentam que a visualização pode ajudar a construir a intuição necessária tanto para a proposição de teoremas, quanto para a compreensão e a criação de provas. Críticos acreditam que a tradicional construção imaculada da prova de teoremas matemáticos corre o risco de ser corroída pela perigosa falta de rigor. A aceitação, ou falta dela, da visualização como uma parte legítima da investigação matemática, tem implicações não apenas para os matemáticos, mas para a comunidade inteira de visualização.”

(MUNZER, 1996, p.451)

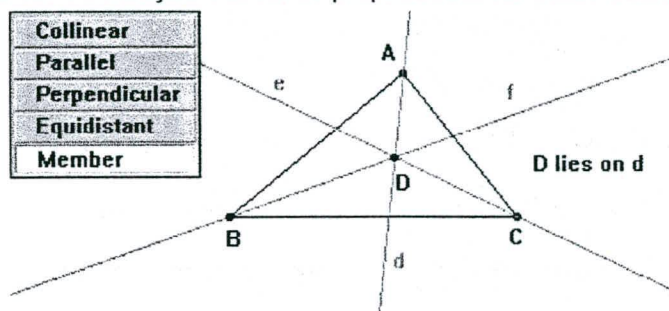
Realmente, a representação visual pode nos iludir. A prova de que todos os triângulos são isósceles, apresentado nas seções 3.1 e 3.2, é um exemplo convincente deste fato. Entretanto, na geometria dinâmica esta situação se inverte. A possibilidade de enxergar a mesma configuração, de diversas maneiras, facilita a compreensão do comportamento geométrico dos elementos envolvidos. Relações geométricas sutis, não óbvias numa representação estática, são reveladas pela exploração e pela visualização da “nova geometria” (de Villiers, 1997).

A verificação interna de propriedades e a checagem automática de teoremas também são características de algumas implementações da geometria dinâmica. O *software* Cabri Géomètre II oferece respostas para cinco tipos de questionamento, listados a seguir.

- Estes três pontos são colineares?
- Estas duas retas são paralelas?
- Estas duas retas são perpendiculares?
- Este ponto é equidistante em relação aos outros dois?
- Este ponto pertence a este objeto?

Observe-se a descoberta do incentro do triângulo ABC, mostrado na figura 15. Primeiramente são criados os três lados do polígono (AB, BC e AC); depois são traçadas as suas bissetrizes. No Cabri, o incentro, ponto D, pode ser encontrado pela ferramenta de interseção entre dois objetos, escolhidos como as bissetrizes e e f. Para o usuário verificar se o ponto D também pertence à bissetriz d, ele deve usar a ferramenta *member* (membro), que na figura retorna a seguinte resposta: *D lies on d* (D pertence a d).

Figura 15: Verificação interna de propriedades no Cabri Géomètre II.



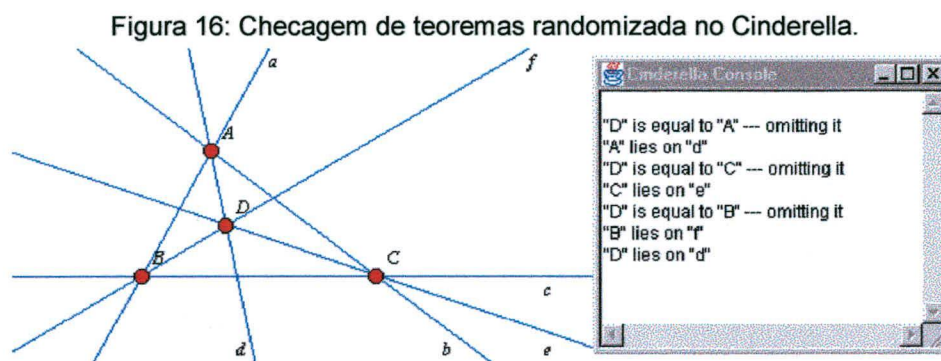
Outro *software*, o Cinderella, é capaz de averiguar e informar, automaticamente, novas relações em tempo de execução (figura 16). Ele também não utiliza métodos simbólicos para a criação de provas formais, mas uma técnica conhecida por *checagem de teoremas randomizada* (Kortenkamp, 1999).

“Primeiro é gerada a hipótese: ‘Parece que a linha k sempre passa através do ponto K’. Então a configuração é movida em várias posições diferentes e para cada uma delas é verificada se a hipótese ainda se mantém. Pode parecer ridículo, mas a geração de exemplos randômicos ‘suficientes’ nos quais o teorema mantém sua validade é ao menos convincente como uma prova simbólica gerada por computador. O Cinderella usa este método repetidas vezes para manter suas próprias estruturas de dados limpas e consistentes.”

(RITCHER-GEBERT, KORTENKAMP, 1999, p.18)

“Em outras palavras, se experimentos sucessivos com pontos selecionados aleatoriamente reafirmam o mesmo resultado, a probabilidade do resultado ser falso aproxima-se de zero.”

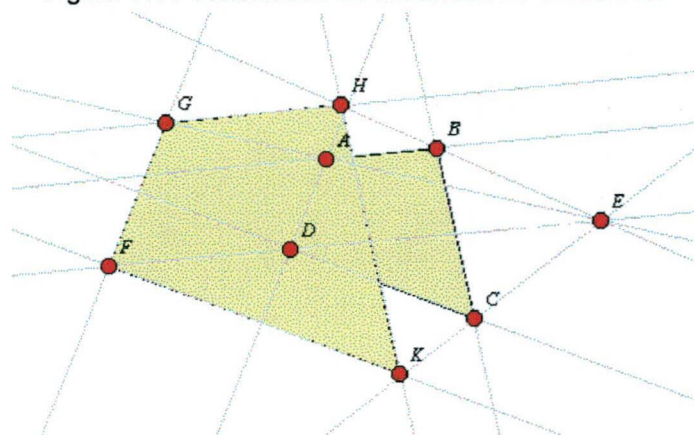
(DE VILLIERS, 1997, p.22)



3.4.5 Transformações

Muitos programas de geometria dinâmica permitem o trabalho facilitado de várias transformações pontuais, tais como translação, rotação, escala, reflexão e homotetia. Por exemplo, o Cabri Géomètre II e o The Geometer's Sketchpad 3 possuem comandos próprios, cada um associado a uma transformação geométrica. Mesmo quando um programa não possui uma ferramenta específica para realizar uma determinada transformação, em geral é possível a obtenção do mesmo resultado através de conhecimentos básicos de geometria plana, fazendo-se o uso combinado de outros recursos. Por exemplo, na figura 17 observa-se um quadrilátero homotético do outro, construído no programa Cinderella. Para se criar este modelo não foi necessário o uso de uma ferramenta própria chamada “homotetia”, até mesmo porque ela não existe neste software. Porém, bastou que o programa fornecesse meios para se construir retas, passando por um ou dois pontos, e principalmente paralelas, para que o usuário pudesse obter o mesmo resultado.

Figura 17: Possibilidade de homotetia no Cinderella.



3.4.6 Loci

Loci é o plural de *locus*, um termo em latim que significa lugar. Quando um conjunto de pontos satisfaz a uma dada condição (e nenhum outro, não pertencente a este mesmo conjunto obedece à mesma condição), constitui-se um lugar geométrico¹². Uma circunferência de círculo é um exemplo de lugar geométrico, pois todos os seus pontos distam um comprimento constante em relação ao centro. No método tradicional de desenho, em geometria clássica, utiliza-se um compasso para determinar o traçado desta curva. Neste caso existe uma ferramenta específica para tal finalidade. Cônicas também são lugares geométricos, assim como uma infinidade de outras classes de curvas, mas ao contrário do círculo, não existem instrumentos tradicionais específicos para construí-las. O traçado final de um *locus* pode exigir procedimentos trabalhosos, cansativos e imprecisos, tornando-se até mesmo inviáveis ao se usar papel e lápis.

“É virtualmente impossível para a maioria das pessoas imaginar um ponto se movendo numa configuração (na qual várias outras partes podem estar igualmente se movendo) e ser capaz de descrever o *locus* da trajetória do ponto enquanto ele transita. Um software de geometria dinâmica, com o seu recurso de traçar o *locus* de qualquer objeto

¹² “No plano α , lugar geométrico (LG) é todo conjunto l de pontos P_i ($i = 1, 2, 3, \dots$) para os quais vale bicondicionalmente uma relação R .” (PINHEIRO, 1974, p.69)

específico, é perfeitamente condizente para mostrar como ele é gerado e para revelar o formato do desenho de sua trajetória.”

(KING, SCHATTSCHEIDER, 1997, p.xiii)

O instrumento da geometria dinâmica que possibilita a criação de lugares geométricos lembra o conceito de uma chave mestra. Se o compasso fosse uma chave comum, ele só abriria um cofre, em cujo interior estaria o conceito de círculos. Entretanto, a ferramenta *locus* pode abrir uma infinidade de cofres, que eventualmente guardam conceitos que sequer poderiam ser imaginados. Schumann e Green (1997) afirmam que a construção de lugares geométricos interativos deve ser efetivamente utilizada em algumas atividades do ensino da matemática, a saber:

- na fase heurística da tarefa de construção de problemas;
- na verificação experimental dos resultados de uma construção;
- em investigações sobre a posição e formato resultantes de figuras transformadas;
- na construção de seções cônicas e curvas algébricas de ordem > 2 ;
- na investigação de figuras geradas por *loci*, pelo movimento de determinados pontos especiais em um triângulo ou em uma configuração mais complexa.

A figura 18 mostra um exemplo singular do *locus* no *software* Cinderella, onde aparecem três barras interligadas, S0, S1 e S2. Os elementos que formam o modelo são os seguintes:

- os pontos A e C são fixos e correspondem respectivamente aos centros dos círculos C1 e C3 (de raios S0 e S1, invariantes);
- o ponto B pertence ao círculo C1 e corresponde ao centro do círculo C2 (de raio S2, invariante);
- o ponto D é resultante da intersecção do círculo C2 com o círculo C3;

- o ponto médio entre os pontos B e D (segmento S2) corresponde ao ponto E. O *locus* em forma de 8 é resultante de seu movimento.

Na figura 19 é apresentada outra aplicação, na qual o *locus* de um ponto sobre a roda – um ciclóide – adquire diferentes tamanhos de acordo com o raio escolhido. O *software* utilizado, neste caso, é o Cabri Géomètre II.

Figura 18: *Locus* na configuração de barras interligadas.

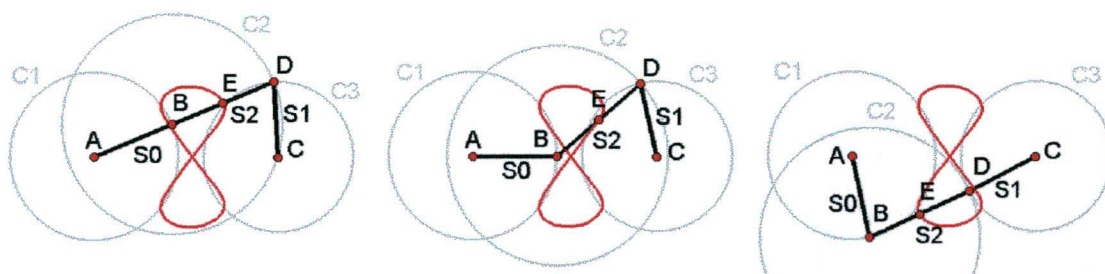
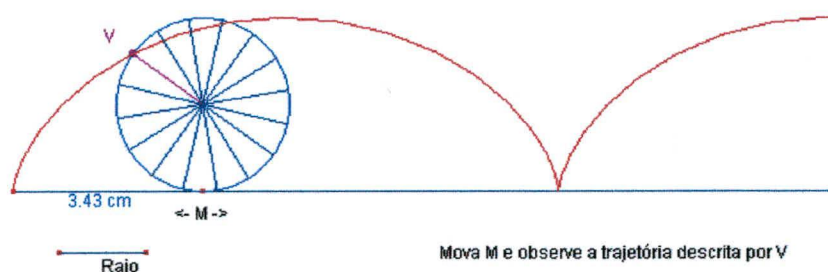


Figura 19: *Locus* representa uma ciclóide.



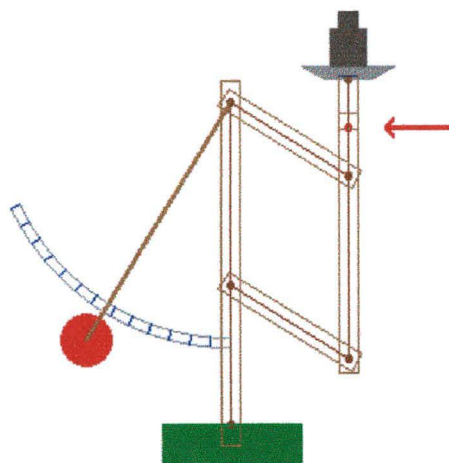
Fonte: Exemplo disponível junto ao software Cabri Géomètre II.

3.4.7 Simulação

As aplicações da geometria dinâmica não estão limitadas ao ensino e à aprendizagem da geometria plana. Com os recursos de animação, de traçado e de lugar geométrico, um grande número de modelos e de situações específicas da geometria espacial, da geometria descritiva, da ótica, da mecânica e de outras áreas que se utilizam da geometria plana podem ser construídos e

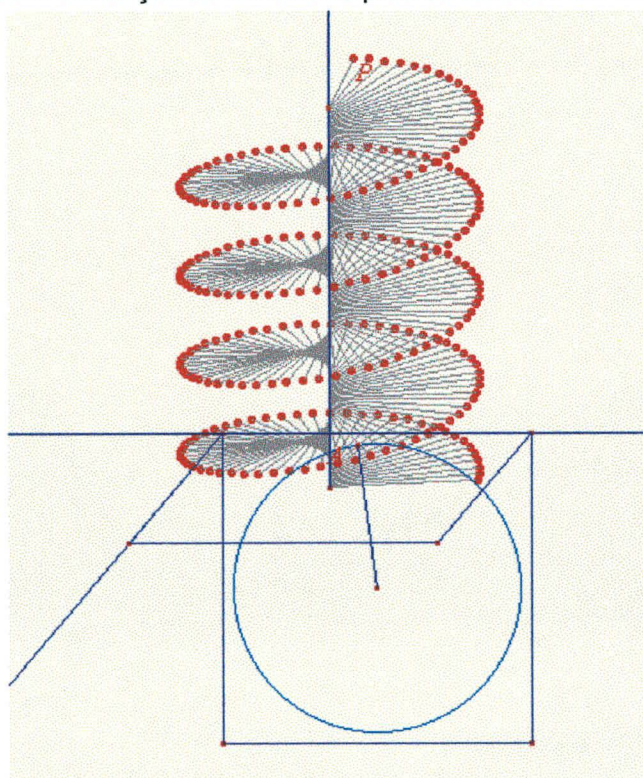
simulados. O site da aplicação *Cabri-Java*¹³ contém vários exemplos, dos quais apenas dois são aqui reproduzidos, o primeiro na figura 20 e o segundo na figura 21.

Figura 20: Exemplo de simulação de equilíbrio em mecânica.



Fonte: [on-line] Disponível na World Wide Web: <<http://www.cabri.net/cabrijava>>.

Figura 21: Simulação do movimento para se construir um helicóide.



Fonte: [on-line] Disponível na World Wide Web: <<http://www.cabri.net/cabrijava>>.

¹³ Trata-se de uma aplicação em *Java* que possibilita a visualização de arquivos do Cabri na internet. O endereço do site corresponde a <http://www.cabri.net/cabrijava>.

O exemplo da figura 20 simula o equilíbrio de um peso em uma balança. Conforme o peso “aumenta” (e cresce na figura), o prato desce e causa a subida do círculo (à esquerda) na escala. Já a figura 21 mostra que a criação de um helicóide é o resultado combinado de uma rotação (no plano x, y) e de uma translação (no eixo z) de um segmento de reta. Percebe-se ainda que, neste caso, a tridimensionalidade é simulada a partir dos recursos oferecidos pelo programa.

3.4.8 Micromundo

Aparentemente não existe uma definição universal para o conceito de micromundo. Harper (apud Maddux, Johnson, Willis, 1997) afirma que um sistema pode ser considerado um micromundo quando ele permite ao usuário o controle pessoal sobre o processo de criação e descoberta, enquanto Knupfer (apud Maddux, Johnson, Willis, 1993) o define como um ambiente de aprendizado que contém um conceito a ser explorado, um ambiente dinâmico para a exploração, e um produto estudantil.

Resumidamente, pode-se afirmar que um micromundo é um ambiente aberto de aprendizado, baseado em regras e que permite a manipulação e a exploração criativa pelos aprendizes. Como os programas de geometria dinâmica são baseados em restrições e possibilitam a exploração por manipulação direta dos elementos da geometria euclidiana, parece não restar dúvida que os mesmos constituem exemplos particulares de micromundos.

3.5 Discussões e problemas sobre interfaces conhecidas

Por melhores que sejam o processo de desenvolvimento e a tecnologia utilizados na elaboração de um produto, sua interface não será bem sucedida

se não corresponder às necessidades do usuário. Mas afinal, quais são os usuários e as tarefas previstas na geometria dinâmica? As facilidades previamente apresentadas dão uma idéia do vasto domínio de aplicações. Segundo Burgiel (1999), tanto o Cabri Géomètre II quanto o Geometer's Sketchpad 3 são melhor adaptados para o ensino médio e fundamental, sendo o Cinderella mais indicado para os cursos de graduação técnico-científicos. Como existe uma grande diversidade de usuários e de tarefas, a definição formal de uma "boa interface" torna-se utópica, sendo impossível determinar todas as combinações possíveis entre usuário, tarefa e tecnologia (Lewis, Rieman, 1994). Com relação à geometria dinâmica, três importantes aspectos de interface são discutidos a seguir:

- comandos em ordem pré e pós-fixada;
- *undo* ilimitado *versus* redefinição de lugares geométricos;
- problemas comuns de interação.

3.5.1 Comandos em ordem pré e pós-fixada

Existem quatro ordens de sintaxe para a execução de comandos (Foley et al., 1996):

- pré-fixada ("ferramenta então objeto", "verbo então nome");
- pós-fixada ("objeto então ferramenta", "nome então verbo");
- de formato livre;
- sensitiva ao modo.

Geralmente, um software de geometria dinâmica se utiliza de pelo menos uma das duas primeiras abordagens. Por exemplo, o Cabri Géomètre II se fundamenta na sintaxe pré-fixada, e o Geometer's Sketchpad 3 na sintaxe pós-fixada. O Cinderella constitui um caso à parte, pois algumas de suas ferramentas são acionadas tanto de uma forma quanto de outra (pela sintaxe

de formato livre). Cada solução possui vantagens e desvantagens, sendo a determinação da melhor alternativa um recorrente motivo para a pesquisa e o debate (Blanc-Brude, Bétrancourt, Laborde, 2001; Braviano, 2001).

Desde sua fundação, em outubro de 1994, o fórum sobre geometria dinâmica da faculdade Swarthmore¹⁴ tem sido palco de uma série de discussões sobre características das interfaces destes programas. Alguns diálogos apresentados nos próximos itens constituem trechos das mensagens cujos títulos são *“tool then object”* (maio de 1995) e *“geometer’s sketchpad or cabri geometry”* (julho de 1999). Nestes comentários, tanto os usuários quanto os criadores expõem suas razões e defendem seus pontos de vista.

3.5.1.1 Ferramenta então objeto

Na sintaxe pré-fixada, quando pretende-se realizar uma tarefa, primeiro é definida a ação e depois são escolhidos os objetos (tabela 5). Tal abordagem prende o usuário a um modo de operação, pois assim que o comando é determinado, somente seus parâmetros podem ser especificados, e em geral numa ordem previamente definida.

Tabela 5: Exemplos em sintaxe pré-fixada.

Verbo	Nome	Nome	Nome	
Comando	Objeto 1	Objeto 2	Objeto 3	Resultado: Novo Objeto
Refletir	Objeto A1	Segmento	-	Objeto B1
Transladar	Objeto A2	Vetor	-	Objeto B2
Rotacionar	Objeto A3	Ângulo	Ponto	Objeto B3
Ampliar	Objeto A4	Fator	Ponto	Objeto B4

Entretanto, se a tarefa envolve a criação de objetos, mas não a transformação de anteriores, a alternativa “ferramenta então objeto” representa uma solução adequada (Winroth, 1999). Ulrich Kortenkamp, um dos desenvolvedores do Cinderella, exemplifica as origens da afirmação anterior:

¹⁴ Acessado na internet pelo endereço <http://mathforum.org/epigone/geometry-software-dynamic/>

“Pense sobre isso: Como você trabalha com papel e lápis, régua e compasso? Se você quer desenhar uma linha através de dois pontos, você: 1) segura a régua e o lápis e 2) desenha a linha. Se você quer desenhar uma circunferência através de dois pontos, você: 1) segura o compasso e 2) desenha a circunferência.”

(KORTENKAMP, em “*geometer’s sketchpad or cabri geometry*”, 1999)

“Eu tenho usado o Sketchpad por 4 ou 5 anos e apenas ‘brincado’ com o Cabri por aproximadamente um ano. A abordagem ‘ferramenta então objeto’ (talvez ‘ação então objeto’ ou ‘operação então objeto’) parece mais intuitiva para mim. Afirmar este fato me faz lembrar de quantas vezes eu tenho visto estudantes irem ao menu e encontrarem a escolha prevista desativada, por não terem selecionado primeiro os objetos. Eu tenho curiosidade para descobrir qual a razão para a abordagem ‘objeto então ferramenta’ do Sketchpad.”

(STALLINGS, em “*tool then object*”, 1995)

Como os programas de geometria dinâmica utilizam a metáfora de *toolbox* (caixa de ferramentas), existem razões adicionais para tolerar o uso de modos de operação na sintaxe pré-fixada (Lewis, Rieman, 1994):

- o ícone muda para indicar em que modo o usuário se encontra, contribuindo para que ele não esqueça com que recurso está trabalhando (figura 22);
- só se ativa um modo quando o próprio usuário escolhe uma ferramenta, então ele sabe qual decisão tomou;
- é fácil sair de um modo pois basta selecionar outra ferramenta.

Figura 22: Ícone para criação de círculo ativado no Cabri Géomètre II.



3.5.1.2 Objeto então ferramenta

A abordagem pós-fixada traz vários outros benefícios. Diferentemente da anterior, um comando somente é disparado após a escolha dos objetos iniciais, evitando que o usuário fique preso a algum modo de operação (tabela 6).

Tabela 6: Exemplos em sintaxe pós-fixada.

Nome	Nome	Nome	Verbo	
Objeto 1	Objeto 2	Objeto 3	Comando	Resultado: Novo Objeto
Objeto A1	Segmento	-	Refletir	Objeto B1
Objeto A2	Vetor	-	Transladar	Objeto B2
Objeto A3	Ângulo	Ponto	Rotacionar	Objeto B3
Objeto A4	Fator	Ponto	Ampliar	Objeto B4

Segundo Raskin (2000), os principais motivos que levam os desenvolvedores de *software* a adotar esta alternativa são:

- redução de erros – na sintaxe pré-fixada, se houver uma distração entre selecionar um comando e realizar a seleção dos objetos, a ação seguinte poderá causar um efeito indesejado, pois cada modo de operação caracteriza uma resposta particular; na sintaxe pós-fixada, um comando é disparado no momento de sua requisição (quando ele constitui o foco de atenção), amenizando assim a ocorrência de erros dessa natureza;
- velocidade – se ao realizar uma operação, o usuário começa com o seu foco de atenção voltado para os objetos, faz mais sentido realizar primeiro a seleção dos mesmos para depois disparar a execução do comando; se este raciocínio estiver correto, supõe-se que na sintaxe pré-fixada o usuário perca tempo, pois começa desviando seu foco de atenção dos objetos para o comando, sendo o mesmo ativado no primeiro instante;
- simplicidade e reversibilidade – na sintaxe pré-fixada, quando um comando é ativado por engano, é necessária a existência de um

mecanismo para que ele seja cancelado, pois tudo que o sistema espera nesse momento é a seleção posterior de objetos; na sintaxe pós-fixada não há necessidade de um método para cancelar ou escapar de um modo operacional, pois ele não existe; uma vez que os objetos já estejam selecionados, pode-se escolher qualquer um dos comandos que a eles se aplicam.

Nicholas Jackiw, desenvolvedor e programador do Sketchpad, apresenta duas razões para adotar a abordagem “objeto então ferramenta”, questionada por Stallings na discussão intitulada “*tool then object*”. A primeira se refere ao padrão adotado pela indústria da informática, e a segunda trata da interferência positiva da interface deste programa no processo de aprendizado do usuário.

“É uma convenção de *software* Macintosh e Windows, independente do programa...Na minha opinião, a importância desta convenção não faz parte da minha justificativa, mas o fato de ela ter sido adotada como um padrão pela maioria dos programas de computador faz... Sob este aspecto, ‘o processo de aprendizado’ que o Sketchpad procura promover é muito prático e concreto. É uma forma de facilitar a integração da tecnologia na sala de aula, e de encorajar a exploração da geometria do programa sem requisitar muitas sessões de treinamento especial em como ‘dirigir’ sua interface.”

(JACKIW em “*tool then object*”, 1995)

“Se eu posso viajar em analogias absurdas por um momento, imagine ser servido por um menu ao jantar num restaurante. Existem várias escolhas (se a cozinha for ambiciosa, devem haver mais escolhas ainda). Algumas delas são claramente inapropriadas, em relação à fome que você estiver sentindo, suas restrições dietéticas, seu orçamento. A idéia por trás da interação não modal, por comandos no menu, é criar menus ideais para jantares por *software*. Você designa (pela seleção) os fatores de interesse – sua fome, sua dieta, seu orçamento – e o

estabelecimento o previne com um menu de escolhas customizado para suas necessidades. Você ainda tem escolha – mas apenas escolhas que combinam com seus interesses e necessidades. Compare isto com um *software* que seja inteiramente modal, no qual você possui dúzias de opções, sem consideração aos seus interesses e necessidades. Você não tem orientação alguma sobre o que você gostaria de comer (ou, em termos geométricos, o que é matematicamente apropriado para ser construído). Então você escolhe algo, esperando que ele possa caber na conta. Vamos esperar que ele caiba, porque senão, você será pego num ambiente dependente de um modo de operação, então você terá que engoli-lo de qualquer forma...”

(JACKIW em “*geometer’s sketchpad or cabri geometry*”, 1999)

3.5.2 *Undo* ilimitado versus redefinição de lugares geométricos

Em uma das mensagens sob o título “*geometer’s sketchpad or cabri geometry*”, de agosto de 1999, Walter Whiteley afirma:

“Há alguns meses, quando Ontário estava escolhendo o seu *software* de Geometria Dinâmica, os professores me disseram que o *undo* ilimitado era o que realmente importava! Isto aparenta ser um recurso que apenas o Cinderella e o Sketchpad compartilham, mas não as versões atuais do Cabri. Parece ser uma coisa pequena, mas tem um grande efeito no uso em sala de aula.”

(WHITELEY em “*geometer’s sketchpad or cabri geometry*”, 1999)

Undo é o termo da língua inglesa que significa *desfazer*. Com um *undo* ilimitado, garante-se ao usuário o retorno de várias etapas no traçado de um desenho, ficando ao seu critério a decisão de reconstruí-lo a partir de qualquer configuração anterior. Como Ulrich Kortenkamp e Nicholas Jackiw confirmam em mensagens posteriores, este recurso é extremamente desejável,

principalmente pelo fato de a geometria dinâmica ser um ambiente construtivo, no qual os “erros” fazem parte do processo de aprendizagem.

Não por acaso, Shneiderman (1998) considera a reversão por ações efetuadas como a sexta de suas oito regras de ouro para a avaliação heurística de IHC. Usuários podem diminuir a ansiedade, pois sabem que qualquer operação pode ser desfeita, sendo encorajados para a exploração de opções com as quais eles não estão familiarizados.

Em sua resposta à mensagem de Whiteley, Bernard Geneves, um dos pesquisadores do IMAG de Grenoble, comenta sobre outro recurso adequado, supostamente capaz de superar o anterior:

“O Cabri não possui o ‘undo ilimitado’ como o Word, o Excel e semelhantes. Ele possui uma ferramenta ‘undo/redo’. E esta talvez seja mais conveniente, pois o Cabri possui uma ferramenta geométrica que repõe o ‘undo’ na maioria dos casos, e é muito mais apropriada para o ensino e o aprendizado: a ferramenta ‘redefinir objeto’.”

(GENEVES em “*geometer’s sketchpad or cabri geometry*”, 1999)

A ferramenta de redefinição de objetos do Cabri Géomètre II possui um comportamento diferente do referente ao recurso anterior. Pelo seu uso, qualquer objeto geométrico pode mudar de configuração sob determinados aspectos. Por exemplo:

- um ponto sobre um segmento pode ser redefinido sobre uma circunferência;
- uma linha pode ser redefinida como paralela em relação à outra;
- uma linha pode ser redefinida como perpendicular em relação à outra;
- uma linha pode ser redefinida como a bissetriz de um ângulo;
- um triângulo pode ser identificado com qualquer outro triângulo.

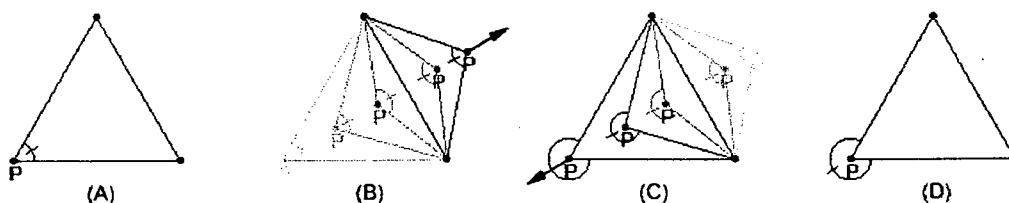
Embora a redefinição de objetos tenha o seu valor e constitua uma vantagem particular do Cabri em relação aos seus concorrentes, esta ferramenta não substitui o *undo* ilimitado, e também apresenta problemas. Existem várias redefinições impossíveis de serem realizadas, como no caso da criação de seqüências recursivas de construção, que embora sejam facilmente reconhecidas e canceladas pelo sistema, conduzem a mensagens de erro incompreensíveis para a maioria dos usuários.

Idealmente, tanto o *undo* infinito, quanto a redefinição de objetos deveriam ser implementados em um mesmo *software* de geometria dinâmica, pois ambos os recursos são importantes e insubstituíveis.

3.5.3 Problemas comuns de interação

Praticamente, todo *software* de geometria dinâmica apresenta algumas falhas de interação. Por exemplo, o Cabri Géomètre II nem sempre distingue um ângulo de seu replemento. Veja-se a figura 23: constrói-se um triângulo qualquer, no qual é marcado o ângulo relativo ao vértice *P*; quando *P* é 'clicado' e arrastado, de forma a deslocar-se para o lado oposto, este ângulo troca de sentido. Até então, nenhuma incoerência surge, pois o ângulo inicial continua interno ao polígono. Porém, no momento de *P* retornar ao seu local de origem, o ângulo "não se repete como antes", tornando-se o replemento do inicial.

Figura 23: Ângulo vira seu replemento.



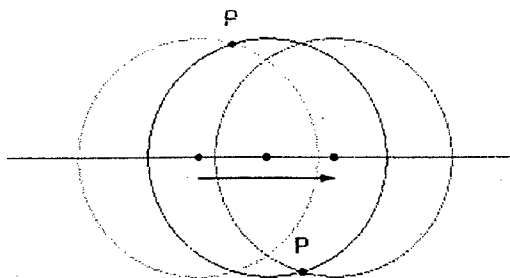
Um problema similar se refere aos objetos que saltam de um local para o outro através de variações sutis em manipulação direta. A figura 24 mostra um exemplo clássico, comum aos programas Cabri Géomètre II e The Geometer's Sketchpad 3. Em uma reta horizontal são marcados dois pontos distintos, correspondendo aos centros de dois círculos de mesmo raio. Como cada ponto central pode deslizar sobre a reta, move-se o da esquerda para que os dois círculos possam se "cortar", e marca-se outro ponto, rotulado como P , na interseção superior. Quando um círculo "escorrega" para cima do outro, acontece o inesperado. Logo após o encontro dos respectivos centros, P salta da interseção superior para a inferior, caracterizando uma violação do princípio de continuidade. (Richter-Gebert, Kortenkamp, 1999).

"Quando um objeto está sendo arrastado, a posição deste objeto e de todos os outros objetos dependentes deve ser uma função contínua da posição do cursor. Não se deve permitir que qualquer objeto realize saltos repentinos."

(WINROTH, 1999, p.103)

A correção de defeitos de continuidade não é uma tarefa trivial. O que à primeira vista pode parecer um pequeno defeito de interface, na realidade corresponde a questões de elevada complexidade, dependentes de diferentes teorias e métodos matemáticos para a resolução de problemas.

Figura 24: Ponto P salta de repente.



Quanto ao princípio da continuidade, o Cinderella constitui um considerável avanço em relação aos outros programas citados mas, mesmo assim, ele

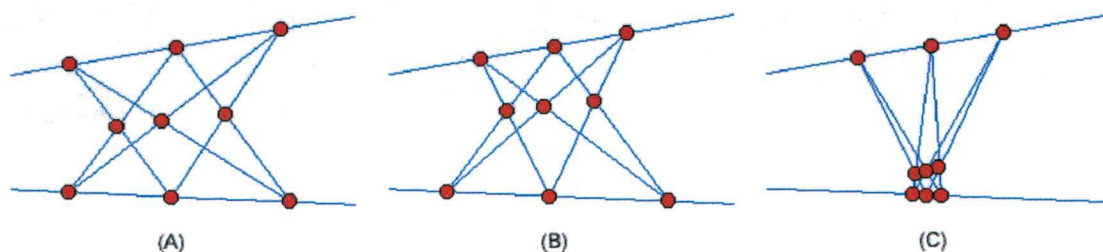
apresenta outra categoria de defeitos classificada por Winroth (1999) como o problema dos “objetos flutuantes”. Viola-se, neste caso, o princípio da repetição.

“Se um objeto é arrastado e devolvido à sua posição original, todos os objetos dependentes também devem retornar às suas posições originais. Não deve haver nenhum movimento ‘em rede’¹⁵ de qualquer objeto.”

(WINROTH, 1999, p.103)

No Cinderella, existem duas alternativas para criar uma reta qualquer. Uma delas exige a indicação de apenas um ponto de controle que, além de representar o seu centro de rotação, também determina a sua translação quando é ‘clicado’ e arrastado. Um ponto qualquer também pode ser ‘amarrado’ em uma reta, ficando restrito à mesma mas sendo capaz de deslizar com o auxílio da manipulação direta. O problema ocorre quando um ou mais pontos são criados sob uma reta e ela é rotacionada várias vezes. Considere-se, por exemplo, o esboço do teorema de Pappus na figura 25: o diagrama (A) apresenta três pontos bem espaçados na reta superior e mais três na inferior; quando a reta superior é rotacionada em 2π , os dois pontos laterais se aproximam do ponto central, como mostrado em (B). Este problema se agrava em (C), quando após três giros completos da reta inferior, os pontos que a ela pertencem se amontoam em torno de seu centro de rotação.

Figura 25: Pontos sob a mesma reta convergem para aquele que a define.



3.6 Conclusão do capítulo 3

A geometria dinâmica parece representar um grande avanço para o aprendizado de uma série de disciplinas gráficas e matemáticas. Através da construção de modelos geométricos precisos e interativos, os usuários podem usufruir de um amplo domínio de aplicações, dos quais destacam-se: a visualização de lugares geométricos, a descoberta de relações entre elementos e a obtenção de provas experimentais de teoremas para cada modelo criado.

Além das diferenças mostradas entre a geometria dinâmica e a geometria euclidiana (clássica), também se apresentou neste capítulo uma breve compilação sobre as características e problemas de IHC dos programas analisados. As informações levantadas até então formam a base de conhecimento utilizada para a realização desta pesquisa.

¹⁵ O original da língua inglesa indica "net movement". Trata-se provavelmente de uma figura de expressão, como se o objeto representasse uma aranha que se movimenta de um lado para o outro em sua rede (figura 4.15).

4 METODOLOGIA E DESENVOLVIMENTO DA PESQUISA

4.1 Introdução

A primeira parte do referencial teórico deste trabalho correspondeu ao estudo de tópicos selecionados da ergonomia e do *design* de interfaces, enquanto a segunda tratou dos recursos, aplicações e benefícios da geometria dinâmica. Diversas publicações, tanto dos projetistas dos programas avaliados quanto dos usuários que desenvolvem aplicações, foram observadas e analisadas. As informações obtidas através destes estudos foram de capital importância para se complementar os procedimentos descritos neste capítulo. Tornou-se imperativa não somente a coleta de informações sobre o tema, mas também uma estratégia de avaliação adequada e compatível com o tempo e com os recursos disponíveis.

Em linhas gerais, esta pesquisa parece estar mais próxima de ser classificada como descritiva, pois segundo Costa (2001, p.31), quando o pesquisador decide realizar uma pesquisa deste gênero, ele:

- já conhece algo sobre o assunto de interesse;
- quer divulgar o que já conhece e buscar, de certo modo, adesões de outros pesquisadores;
- busca também críticas construtivas que lhe possibilitem ampliar o conhecimento sobre o tema.

Compreende-se que os modelos formais para a avaliação de *software* tais como o uso apropriado de métricas e a testagem com usuários em laboratórios de usabilidade são de grande valia. Na prática, se tomadas as devidas precauções, estes métodos garantem uma maior precisão e conseqüente confiabilidade dos dados obtidos. Entretanto, tais alternativas, além de exigirem um alto custo financeiro, são geralmente aplicadas em várias etapas do

processo de desenvolvimento de um único *software* (Nielsen, 1993; Rubin, 1994; Shneiderman, 1998). Como a intenção inicial desta pesquisa era reconhecer e comparar os diferentes aspectos ergonômicos de IHC, encontrados em mais de um programa de geometria dinâmica, optou-se por uma estratégia alternativa de análise. Buscou-se, através de uma avaliação heurística e da aplicação de um questionário a usuários de programas de geometria dinâmica, o levantamento de informações que pudessem ser analisadas e comparadas entre si. Inicialmente foi determinado um grupo de 7 funcionalidades, ou características, comuns aos 3 programas estudados, Cabri Géomètre II, The Geometer's Sketchpad 3 e Cinderella. Os tópicos de interesse selecionados foram:

1. Ordem de comandos;
2. Mudança de atributos;
3. *Feedback*;
4. Manipulação Direta;
5. Animação;
6. Lugar Geométrico;
7. Macros e *Scripts*.

Prosseguiu-se então com a primeira etapa de avaliação (heurística) onde foram observados todos os programas, item por item, a partir dos conceitos e princípios apresentados no capítulo 2, ou mais especificamente, da síntese das recomendações de Bastien & Scapin (1993), Nielsen (1993) e Shneiderman (1998). Embora uma avaliação heurística deva ser realizada por um conjunto de avaliadores, onde cada um analisa a interface de um *software* isoladamente para depois serem compilados os resultados, neste caso contou-se com a presença de apenas um especialista (o próprio pesquisador). No entanto, justifica-se este procedimento pelo principal objetivo deste trabalho. Invés de se preocupar com o levantamento do maior número possível de falhas de usabilidade existentes em um software específico, procurou-se averiguar, discutir e comparar as diferentes soluções de interface existentes nos 3

programas selecionados. Além disso, sempre que possível, as hipóteses levantadas nesta primeira etapa foram relacionadas com as informações obtidas na segunda, sendo esta última caracterizada pela criação e pela aplicação de um questionário destinado a alunos e professores.

Respeitando-se as recomendações de Cervo & Bervian (1983), Gil (1994) e Lakatos & Marconi (1998), elaborou-se um questionário de 20 questões (apresentado em anexo) englobando os mesmos 7 tópicos de interesse previamente destacados. De forma abrangente, várias perguntas foram caracterizadas na forma de múltipla escolha, e praticamente todas foram acrescidas de um campo para a apresentação de comentários ou justificativas. Ao todo, 14 indivíduos colaboraram respondendo às perguntas do questionário durante o período do dia 26 de novembro até o dia 10 de dezembro de 2001. Divididos por categoria, os participantes foram:

- 4 alunos de graduação do curso de licenciatura em Educação Artística (habilitação em desenho) da UFRJ – EBA;
- 5 alunos de pós-graduação (lato-sensu) do curso de Especialização em Técnicas de Representação Gráfica da UFRJ – EBA;
- 5 professores, sendo 3 deles da UFRJ, um da Universidade Estadual de Londrina e outro do Colégio Pedro II, no Rio de Janeiro.

A tabela 7 mostra a familiaridade dos respondentes com os programas de geometria dinâmica avaliados e a(s) área(s) de conhecimento em que eles os aplicam. Os códigos de 1 a 9 (R1 a R9) correspondem aos 9 alunos e os códigos de 10 a 14 (R10 a R14) correspondem aos 5 professores participantes. Ao longo deste capítulo, estes mesmos códigos são utilizados para se referenciar os comentários de cada um destes indivíduos.

Tabela 7: Familiaridade dos respondentes com os programas avaliados.

Resp	Programa conhecido pelo respondente				Área de aplicação
	Cabri. Géomètre II	The Geo's Skecthpad 3	Cinderella	Outros	
R1	1 ano e meio	6 meses	só observação	-	Geom. Plana (M/Pós)
R2	1 ano	6 meses	só observação	-	Geom. Plana (Pós)
R3	6 meses	6 meses	só observação	-	Geom. Plana (M/Pós)
R4	1 ano e meio	6 meses	6 meses	-	Geom. Plana (Pós)
R5	6 meses	6 meses	só observação	-	Geom. Plana (Pós)
R6	9 meses	3 semanas	3 semanas	-	G. Plana, GD (U/Pós)
R7	5 meses	3 semanas	só observação	-	Geom. Plana (U)
R8	5 meses	3 semanas	só observação	-	Geom. Plana (U)
R9	1 ano	6 meses	só observação	-	Geom. Plana (U)
R10	3 anos e meio	2 anos e meio	1 ano	-	G. Plana, GD, Persp. e Sombras.; Axonometria.
R11	2 anos	3 anos	-	Cabri I – 2 anos Tabulae – 1 mês	G. Plana, GD (Ensino Fund. e Médio)
R12	4 anos	-	3 anos, porém sem usá-lo	-	Matem., G. Plana, GD (U e com professores)
R13	4 anos	4 anos	-	Tabulae – 1 ano	G. Plana e G.Esp. (U); GD (M)
R14	3 anos	6 meses	Alguns dias	-	G. Plana e Perspec. (U)

Os próximos itens deste capítulo apresentam, um a um, primeiro as explicações, hipóteses e exemplos resultantes da avaliação heurística e depois as informações filtradas a partir dos dados obtidos pelo questionário para cada tópico em discussão. Além das tabelas contendo as frequências e os percentuais associados às questões de múltipla escolha, no final também são mostrados os comentários e as justificativas mais significantes dos respondentes.

4.2 Ordem de comandos

Embora a ordem de comandos tenha sido amplamente discutida no capítulo 3 deste trabalho, este item trata de detalhes específicos das implementações observadas nos programas avaliados.

O programa Cabri Géomètre II utiliza sempre a sintaxe pré-fixada¹⁶, pois quando pretende-se realizar uma tarefa, primeiro é definida a ação e depois

¹⁶ Desconsiderando as operações de mudança de atributos, conforme será observado na seção 4.3.

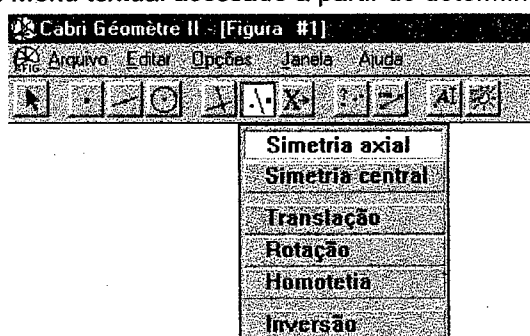
são escolhidos os objetos. Parte da solução de interface adotada pelo Cabri pode ser visualizada na figura 26.

Figura 26: Barra de ferramentas do Cabri Géomètre II.



Na parte superior da tela encontra-se uma barra de ferramentas com botões que mostram ícones, sendo que cada um deles é associado a uma operação particular. A princípio pode-se concluir, erroneamente, que a quantidade total de ferramentas corresponde ao número de botões presentes na barra, mas esta hipótese é falsa, pois cada botão esconde um determinado grupo de ferramentas. Por exemplo, o sexto botão da esquerda para a direita (figura 27) representa a operação de simetria axial mas faz parte de um conjunto específico de transformações. Ao repousar o cursor sobre uma opção, quando um usuário “clica” e mantém pressionado o botão esquerdo do *mouse* (durante aproximadamente 0.5s), surge um menu textual completo com várias opções. Se o usuário arrastar o cursor sobre as outras opções apresentadas, uma nova ferramenta será escolhida a partir do instante em que o botão do *mouse* for solto.

Figura 27: Menu textual acessado a partir de determinada opção.



Esta estratégia de pressionar, manter e soltar um dispositivo de controle é classificada por Raskin (2000) como sendo um semimodo¹⁷, pois invés de se ativar um modo de uma só vez, é necessária a manutenção do mesmo através de uma ação muscular contínua, tal como permanecer com uma alavanca

pressionada ou pisar num pedal. No caso do Cabri Géomètre II, os botões da barra de ferramentas não trazem nenhuma indicação de que os menus escondidos existem. Usualmente utiliza-se uma pequena seta no canto inferior direito dos ícones para informar que existem outras opções acessíveis a partir dali. Entretanto, podem-se compreender algumas das possíveis razões para que a solução adotada não obedeça este padrão.

Um programa de geometria dinâmica é primordialmente concebido para o ensino e aprendizagem em sala de aula, com alunos trabalhando em grupo e sob a orientação de um professor. Embora a própria interface não traga pistas sobre o acesso às outras ferramentas, basta que os alunos observem uma única vez como acessá-las, ainda que auxiliados pelo professor, para compreenderem o processo de funcionamento do sistema. Principalmente pelo fato de estarem sempre lidando com as mesmas operações, os alunos formam hábitos e passam as informações de uso da MCT para a MLT. Como todos os botões possuem menus associados, também não existe a necessidade de se distinguir as opções supostamente únicas das outras, evidenciando a provável redundância de informação.

O programa The Geometer's Sketchpad 3 utiliza a sintaxe pré-fixada apenas para a criação de primitivas geométricas básicas tais como ponto, reta e círculo. Para a execução destes comandos existe uma caixa de ferramentas sempre acessível à esquerda da tela e, assim como no Cabri Géomètre II, não existem pistas sobre a aparição de menus quando se mantém o botão esquerdo do mouse pressionado. O conjunto completo de ferramentas disponíveis na caixa é mostrado na figura 28. Nesta imagem, verifica-se que apenas a primeira e a quarta opções possuem menus associados. Diferentemente da situação encontrada no Cabri Géomètre II, nem todas as opções são extensíveis e a falta de uma diferenciação visual pode confundir o usuário iniciante. Por exemplo, levando-se em consideração que segmento, raio e reta estão agrupados, pode-se esperar que outros agrupamentos

¹⁷ Uma explicação detalhada sobre o conceito de modo de operação se encontra no capítulo 2.

também existam, um envolvendo ponto, ponto sobre objeto e pontos de interseção, e outro envolvendo círculo, arco e cônica. Na realidade estes agrupamentos não existem. Os ícones alterados e mostrados na figura 29 solucionariam este problema¹⁸.

Figura 28: Caixa de ferramentas com opções disponíveis.

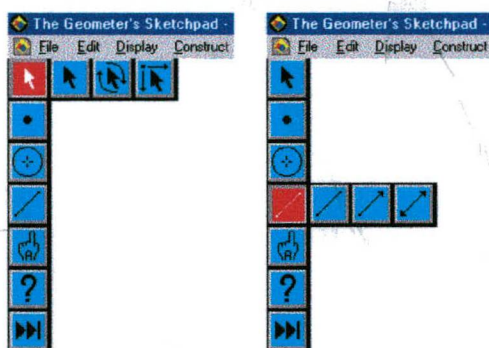
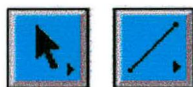


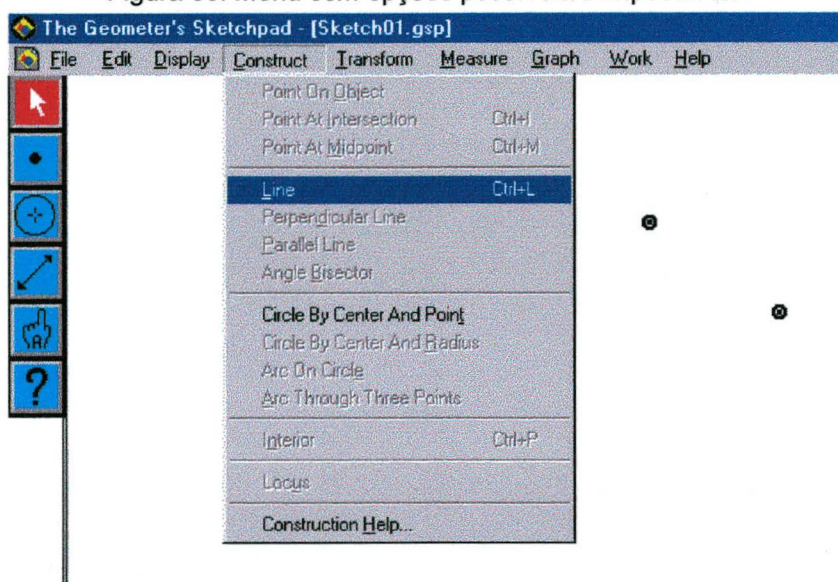
Figura 29: Ícones com pistas sobre os menus associados.



Neste mesmo programa, todos os objetos que dependem da definição de outros são construídos da maneira pós-fixada, havendo esta possibilidade até mesmo para as primitivas anteriormente citadas. Neste caso, o acesso às ferramentas ocorre após a escolha dos objetos iniciais. Na parte superior da tela existem palavras chave representando diferentes categorias de operações que, ao serem acionadas, mostram menus textuais com várias opções. Como os objetos são previamente escolhidos, o programa tem como averiguar que tipo de operação é aplicável, mantendo desabilitadas as alternativas que não se adequam à seleção efetuada. Por exemplo, após a escolha de dois pontos é possível optar pela criação de uma linha reta, entretanto não faria sentido acessar a ferramenta “arco passando por três pontos”, pois os parâmetros iniciais são incompatíveis com esta operação (figura 30).

¹⁸ Após este trabalho ser concluído, verificou-se que esta solução foi adotada na quarta versão deste programa.

Figura 30: Menu com opções possíveis e impossíveis.



Se a aplicação deste programa for destinada ao aprendizado dos conceitos básicos de geometria plana, esta característica de filtragem de opções deve ser considerada um benefício. Um aluno, ao selecionar objetos com o intuito de realizar uma operação específica, verificará que existem outras operações disponíveis. A sua curiosidade, associada a este recurso, pode levá-lo à descoberta de novos conceitos.

Outra vantagem da sintaxe pós-fixada presente no Sketchpad se refere à possibilidade de criação de mais de um objeto ao mesmo tempo. Por exemplo, considere-se a situação de quatro pontos e uma reta selecionados (figura 31). Quando o comando para a construção de uma reta perpendicular é escolhido, quatro novas retas surgem de forma simultânea (figura 32). Para se obter a mesma resposta no Cabri Géomètre II, ou no Cinderella, é necessário repetir quatro vezes a mesma operação.

Figura 31: Quatro pontos e uma reta selecionados.

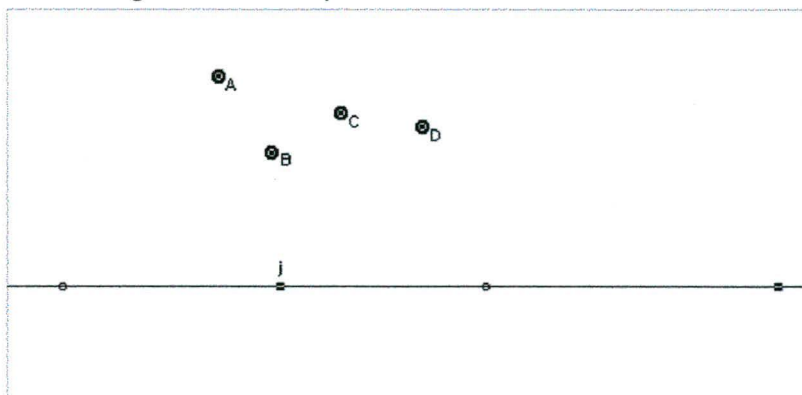
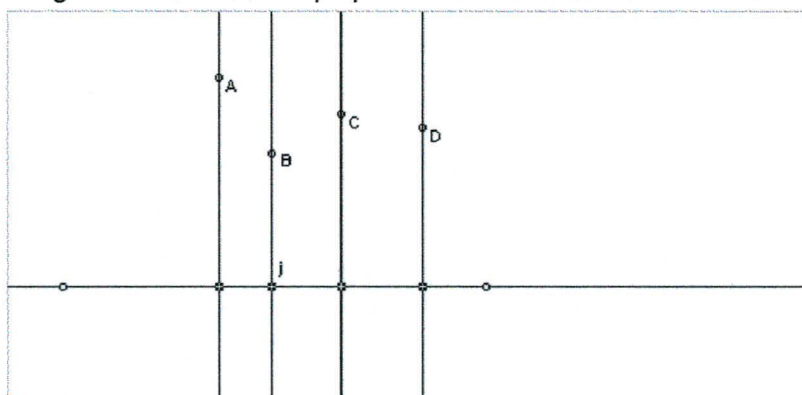
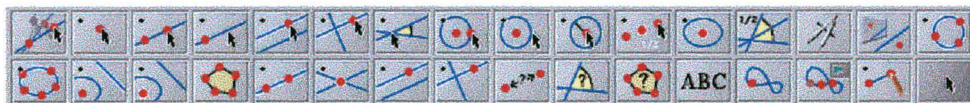


Figura 32: Quatro retas perpendiculares criadas de uma só vez.



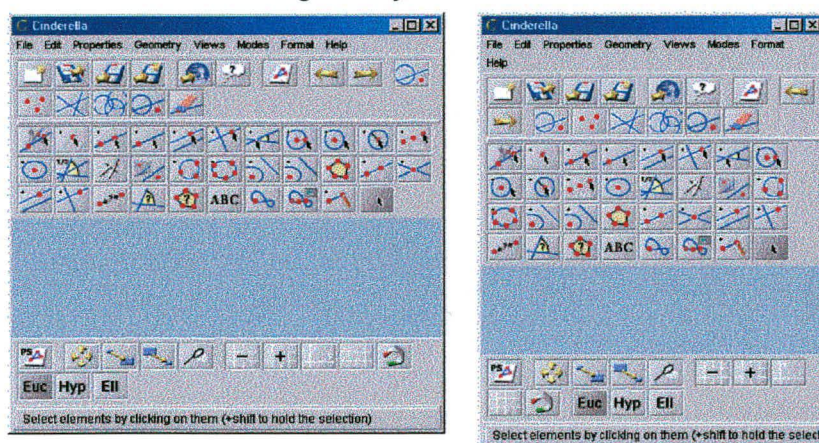
O programa Cinderella utiliza tanto a sintaxe pré-fixada quanto a pós-fixada. Algumas operações funcionam apenas pela primeira abordagem e outras somente pela segunda, porém, também existem opções que funcionam dos dois modos. A primeira diferença notável deste programa em relação aos outros dois observados diz respeito à quantidade de botões visíveis na barra de ferramentas (figura 33). Os onze primeiros ícones exibem a imagem da seta do cursor. Esta é uma solução inteligente para caracterizar as ferramentas que operam de maneira pré-fixada, pois são elas que criam os objetos através da manipulação direta. Outra solução merecedora de destaque é representada pelo minúsculo símbolo “+” no canto esquerdo superior de alguns botões. Somente as ferramentas que apresentam esta imagem criam novas instâncias de objetos.

Figura 33: Barra de ferramentas do Cinderella.



Todas as operações passíveis de execução no Cinderella estão disponíveis na barra apresentada, garantindo que qualquer ferramenta seja acessada sem a necessidade de procura em algum menu escondido. No entanto, além de ser ocupado um grande espaço na tela, torna-se difícil memorizar a posição dos itens quando a janela do programa é redimensionada. Na ocorrência deste evento, os ícones são reorganizados conforme o espaço disponível. Este comportamento é ilustrado na figura 34.

Figura 34: Últimos botões saltam para baixo quando a largura da janela diminui.



As respostas dadas à pergunta - COM QUAL DAS ALTERNATIVAS PARA SE IMPLEMENTAR A CONSTRUÇÃO DE ELEMENTOS GEOMÉTRICOS VOCÊ SE SENTE MAIS À VONTADE? - expressaram a opinião de cada respondente (num total de 14 indivíduos) sobre sua preferência quanto ao modo de acesso das ferramentas e decorrente execução de tarefas. Os dados obtidos podem ser visualizados na tabela 8.

Tabela 8: Resultados da pergunta 5 do questionário.

Categorias	nº	%
Ferramenta/objeto (sintaxe pré-fixada)	8	57,14
Opção condicionada à situação	2	14,29
Indiferença quanto à escolha	4	28,57
Total	14	100

Ao se observar a tabela 8, percebe-se a preferência dos usuários consultados pela sintaxe pré-fixada, em que primeiramente ativa-se a ferramenta para só então apontar-se o objeto com que se trabalhará (57,14% das respostas). A predominância por esta escolha torna-se mais significativa, ao saber-se que uma das alternativas oferecidas ao respondente neste item do questionário - optar isoladamente pelo modo pós-fixado - não foi apontada por qualquer dos participantes, embora admitida na segunda e terceira categorias (“depende da situação” e “tanto faz”) totalizando 42,86% das respostas.

As justificativas apresentadas pelos participantes para a escolha do “modo ferramenta/objeto” (primeira categoria da tabela) se originaram principalmente de sua familiarização com o programa Cabri, como mostram os comentários abaixo:

R6 – Tempo de uso maior do Cabri.

R11 – Acredito que seja uma questão de hábito; o programa que eu utilizo com mais frequência funciona assim.

R14 – Este é o modo mais utilizado pelo Cabri, com que estou mais acostumada.

Os que se manifestaram favoráveis tanto ao modo “ferramenta/objeto” quanto ao “objeto/ferramenta” apresentaram como motivo o fato de terem se adaptado facilmente aos dois, sendo-lhes portanto indiferente optar por qualquer deles.

Parte das hipóteses levantadas na avaliação heurística sobre a existência de dificuldades de aprendizado do acesso às ferramentas destes programas confirma-se em virtude dos seguintes comentários:

R1 – Sim, com a prática as dificuldades foram desaparecendo.

R3 – *Inicialmente sim. Mas diante da orientação recebida, a utilização dos programas que foi feita, uma associação entre eles funcionou de forma clara e dinâmica.*

R8 – *Não, porém foi necessário um tempo de adaptação ao uso do segundo programa¹⁹ utilizado (The Geometer's Sketchpad 3).*

R9 – *Minha primeira dificuldade foi descobrir os menus do Cabri. Depois, tive uma pequena dificuldade ao passar de um software para o outro.*

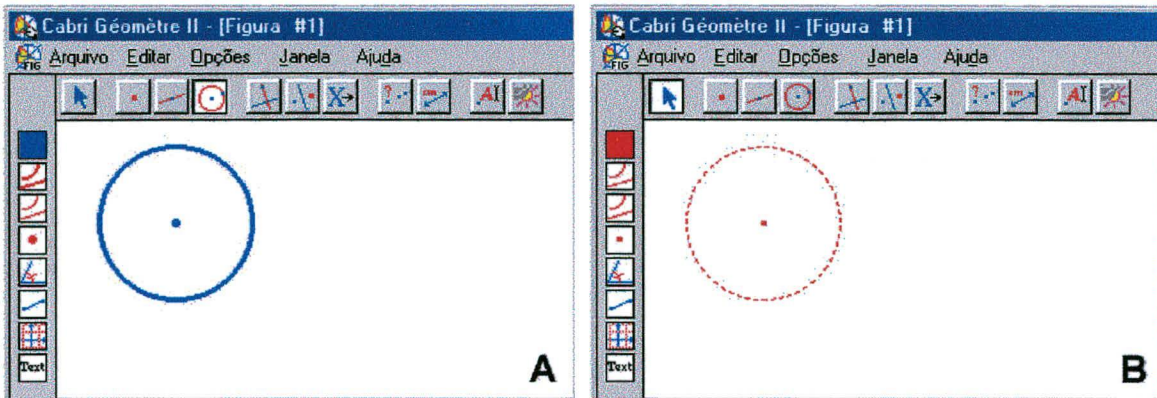
4.3 Mudança de atributos

Para a seleção e a modificação de atributos (ou estilos), o Cabri Géomètre II utiliza uma abordagem que Foley et al. (1996) classificam como sintaxe sensitiva ao modo. Cada ferramenta possui variáveis próprias, relacionadas com as características dos atributos, tais como cor, espessura da linha, tipo de tracejado e formato do ponto. Quando uma ferramenta é selecionada e os seus atributos são modificados, ela passa a armazenar estes novos valores como *default*. Neste caso, qualquer objeto criado por ela herdar os valores previamente estabelecidos. Porém, se primeiro for selecionado um objeto, a mudança de atributos refletirá somente sobre o mesmo, permanecendo inalterados os valores referentes à ferramenta que o originou.

A figura 35 exemplifica estas duas situações. No canto esquerdo das janelas aparece a barra com os atributos. Na primeira situação (A), os valores dos atributos correspondem à ferramenta selecionada. O círculo representado em azul é exemplo de um objeto criado por ela. Na segunda situação (B), o mesmo círculo criado anteriormente é selecionado (razão pela qual ele aparece tracejado), e somente os valores de seus atributos são modificados.

¹⁹ O primeiro programa utilizado por R8 foi o Cabri Géomètre II.

Figura 35: Atributos associados à ferramenta (A) e ao objeto (B).

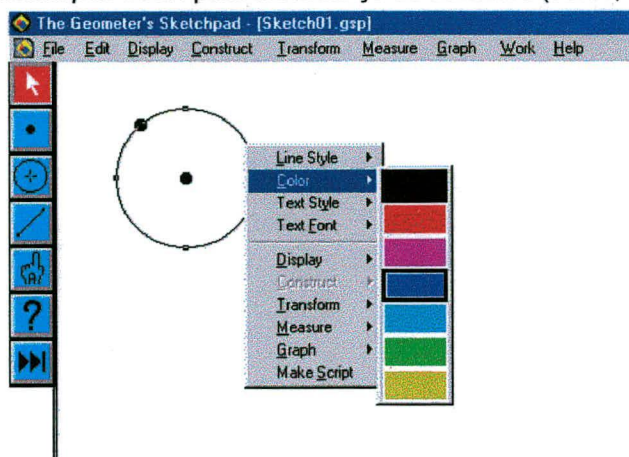


O programa The Geometer's Sketchpad 3 também trabalha com variáveis de atributos globais e locais. Entretanto, nele existe apenas um grupo de variáveis globais (invés de vários grupos como no Cabri Géomètre II, onde cada um pertence a uma ferramenta). Todo novo objeto criado, independente da ferramenta ou da operação executada, herda os estilos globais no momento de sua criação. Por exemplo, quando se estabelece uma nova cor *default*, ela é aplicada em toda e qualquer nova instância de um elemento geométrico, seja ele um ponto, uma reta ou um círculo. Assim como no Cabri, neste programa são necessários dois passos para se modificar os atributos de um determinado objeto:

1. selecionar o objeto a ser modificado;
2. modificar os estilos através do menu correspondente.

Quando o usuário sabe localizar o menu com a lista de atributos (acessível pela opção *display* ou pelo "clique" do botão direito do *mouse*), a realização desta tarefa torna-se trivial (figura 36). Porém, se a tecla *SHIFT* não estiver pressionada no momento da troca de estilos, a mudança não somente se refletirá nas variáveis locais do objeto, mas também nas variáveis globais do programa. Além de não ser intuitivo, o mecanismo da interface não traz nenhuma pista da necessidade de ativação desta tecla. Tal informação só pode ser obtida através da procura de ajuda ou da leitura do manual.

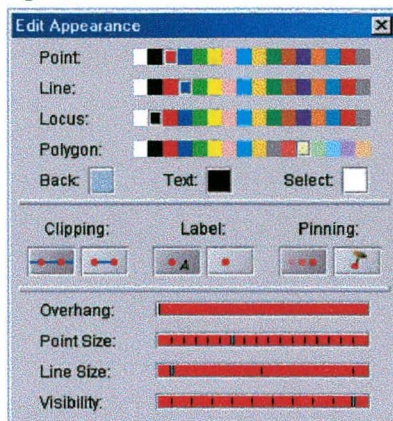
Figura 36: Menu *pull down* para a mudança de atributos (cores, neste caso).



O Cinderella apresenta um comportamento parecido com o do Sketchpad, pois qualquer mudança de atributos sobre um ou mais objetos determina, necessariamente, novos valores *default* para as variáveis globais correspondentes. Não existe sequer algum recurso, por menos evidente que seja, para se limitar o escopo da mudança em relação ao(s) objeto(s) selecionado(s); nem mesmo a tecla *SHIFT*.

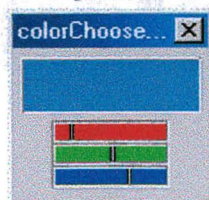
Conforme citado anteriormente, no Cabri Géomètre II os valores dos atributos (incluindo a cor) são supostamente associados às ferramentas, mas na realidade não são exatamente as ferramentas que guardam estes valores, mas sim os elementos geométricos criados por elas. Por exemplo, quando o usuário muda a cor associada à ferramenta específica para a criação de “reta paralela”, ele muda também, automaticamente, a cor relativa a todas as ferramentas que criam retas, tais como “reta perpendicular” e “reta” (solta). No caso do Cinderella, este tipo de “interferência” não ocorre pois a interface torna evidente uma correspondência direta entre cada elemento geométrico e a sua cor *default* (figura 37).

Figura 37: Atributos do Cinderella.



Ainda sobre o aspecto do uso de cores, é possível identificar duas falhas de usabilidade no programa Cinderella (apontadas por [1] e [2] ao longo do texto). A cor *default* de uma primitiva geométrica (linha por exemplo) pode ser escolhida entre 16 células (cada uma de uma cor) preestabelecidas em uma palheta (figura 37). Além disso, qualquer cor destas células pode ser modificada em função de suas componentes RGB (*red, green, blue*), totalizando 40^3 combinações possíveis para cada uma (a cor final é obtida através do posicionamento de três pequenas barras deslizantes – figura 38). Indubitavelmente, este amplo leque de possibilidades representa uma vantagem. [1] Entretanto, o estilo de interação utilizado para se modificar os valores RGB é consideravelmente inadequado. [2] Além da dificuldade de distinção entre uma cor e outra quando elas estão próximas, não se pode copiar células entre diferentes palhetas. Por exemplo, a cor *default* (modificada) associada a pontos não pode ser reproduzida como a cor *default* associada a linhas.

Figura 38: Interface para modificação das componentes RGB de uma cor.



As respostas dadas à pergunta – EM QUAL DOS PROGRAMAS UTILIZADOS VOCÊ SE SURPREENDEU PELO FATO DE NOVOS ATRIBUTOS TEREM SIDO ASSIMILADOS PELA FERRAMENTA QUE O

CRIOU? – expressaram a opinião de cada respondente (num total de 19 respostas assinaladas²⁰) sobre os programas em que se deparou com o fato de novos objetos criados herdarem as propriedades daqueles que foram construídos anteriormente. Os dados obtidos podem ser visualizados na tabela 9.

Tabela 9: Resultados das perguntas 6.1 e 6.2 do questionário.

Programas	n ^o	%
The Geometer's Sketchpad 3	11	57,89
Cinderella	7	36,84
Cabri Géomètre II	1	5,26
Total	19	100

Ao se observar a tabela 9, percebe-se que a maior ocorrência de surpresas quanto à mudança de atributos (57,89%) é referente ao programa The Geometer's Sketchpad 3. A avaliação heurística, por si só, mostrou a ausência de uma indicação sobre o uso correto do dispositivo para a mudança de variáveis locais de objetos. É provável que os usuários sequer tenham consciência da necessidade de se pressionar a tecla *SHIFT*, apesar da presença desta informação no *help* e no manual. Na prática, era de se esperar que o Cinderella apresentasse o maior percentual de surpresas, em virtude de não haver a menor possibilidade de uma mudança somente local dos valores de atributos. Contudo, uma possível razão para a sua segunda colocação (36,84 %) se deve ao fato de um pequeno número de usuários ter efetivamente trabalhado com o mesmo (conforme indicam os dados brutos, referentes à familiaridade dos respondentes com os programas, apresentado em anexo). É pertinente notar que 100% dos usuários que apontaram ter tido surpresas com o Cinderella, também indicaram tê-las tido com o Sketchpad.

As respostas dadas à questão – **INDIQUE A SUA PREFERÊNCIA QUANTO ÀS ABORDAGENS PARA A MUDANÇA DE PARÂMETROS APRESENTADAS PELOS PROGRAMAS RELACIONADOS** – expressaram os valores conferidos aos programas, numa “escala de graduação” (termo utilizado por Gil, 1999) por

²⁰ Deu-se a possibilidade de cada respondente assinalar mais de um programa.

parte de cada respondente, sobre a modificação de atributos. Os dados obtidos podem ser visualizados na tabela 10.

Tabela 10: Resultados da pergunta 6.3 do questionário.

Programas	Valores atribuídos	BOM	MÉDIO	FRACO
Cabri Géomètre II		13	1	-
The Geometer's Sketchpad 3		8	6	-
Cinderella		2	9	1
Tabulae		2	-	-

Na tabela 10, todos os 14 respondentes expressaram alguma opinião quanto ao Cabri (13+1) e ao Sketchpad (8+6). Porém, somente 12 indivíduos classificaram o Cinderella (2 deles declararam não conhecê-lo satisfatoriamente) e apenas 2 mencionaram o Tabulae (por serem os únicos a terem trabalhado com este). Pelo fato de a votação da tabela 10 não ser excludente, os percentuais não totalizam 100% mas podem ser comparados com os dados obtidos na tabela 9. Poderia se imaginar que quanto maior fosse o percentual de valores "BONS", menor seria o percentual de surpresas quanto ao fato de novos objetos criados herdarem os atributos daqueles que foram construídos anteriormente. No entanto, os dados abaixo não comprovam esta hipótese:

- Cabri: (1) 92,86% de "BONS" / (3) 5,26% de surpresas;
- Sketchpad: (2) 57,14% de "BONS" / (1) 57,89% de surpresas;
- Cinderella: (3) 16,67% de "BONS" / (2) 36,84% de surpresas.

Conforme comentado anteriormente, nesta avaliação observou-se que vários usuários apresentaram pouca prática com o Cinderella. Em caso contrário, é provável que o percentual de surpresas referentes à criação de novos objetos no Cinderella pudesse superar o valor encontrado para o Sketchpad.

As opiniões que melhor justificam estes votos são mostradas a seguir:

R9 – *Considero que a aparência dos objetos deva ser modificada apenas pelo usuário quando desejar modificar a imagem de objetos específicos.*

R10 – *Prefiro o Cabri por ter um acesso mais rápido à mudança de atributos, na coluna à esquerda.*

4.4 Feedback

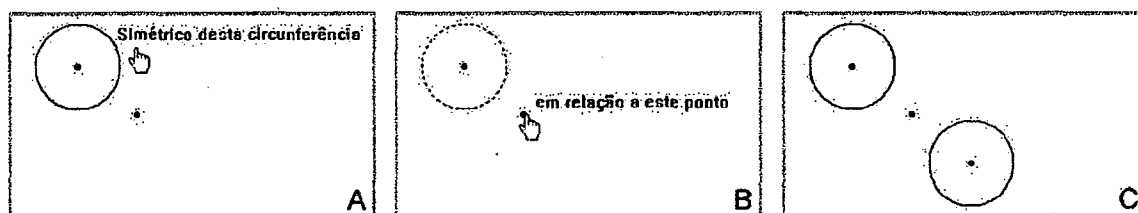
Feedback significa retorno. Sempre que o usuário realiza uma operação, ele deve receber um aviso, uma mensagem do sistema comunicando que houve uma mudança de contexto. Por exemplo, quando o usuário está com a ferramenta para criar pontos habilitada e ele passa o cursor sobre uma reta, o programa deve identificar este evento como sendo uma possibilidade real para a criação de um ponto sobre a reta. Nesta situação, é desejável o surgimento de alguma mensagem, ou sinal de retorno significativo para auxiliar o usuário neste processo.

Aparentemente, o Cabri Géomètre II trabalha esta questão de uma forma bastante satisfatória. Neste programa, a ocorrência de eventos (provenientes das ações do usuário através da manipulação direta) dispara mensagens de retorno próximas ao cursor. Frequentemente, o usuário está com o seu foco de atenção voltado para os objetos com os quais ele está trabalhando. Portanto, a proximidade das mensagens de retorno em relação ao cursor (e conseqüentemente em relação ao objetos observados e manipulados) deve ser um fator importante para que as informações sejam rapidamente percebidas e processadas pelo cérebro.

No exemplo da figura 39, pode-se verificar que as mensagens apresentadas não somente auxiliam o usuário na conclusão da tarefa, mas também servem como apoio ao aprendizado do conceito de simetria. Independente da precisão

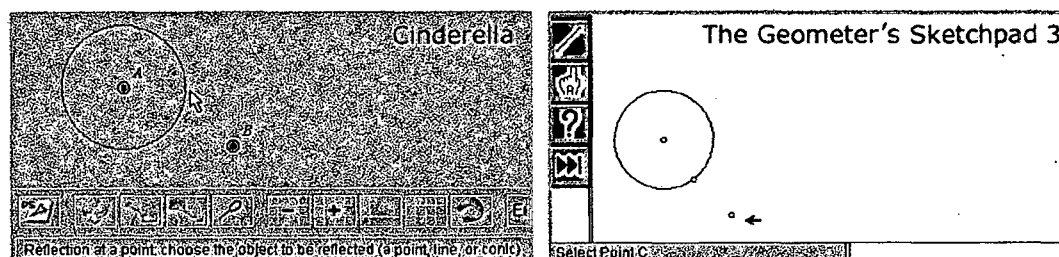
da figura, basta que as mensagens sejam lidas para se afirmar: “o novo círculo é o simétrico do primeiro em relação ao ponto indicado”.

Figura 39: Etapas para a reflexão de um círculo em relação a um ponto.



Os programas The Geometer's Sketchpad 3 e Cinderella também apresentam mensagens de retorno para o usuário. Porém, no que se refere à localização das mesmas, ambos se diferenciam do Cabri. A figura 40 mostra as soluções adotadas por eles, onde uma barra para a divulgação de mensagens se encontra no canto inferior esquerdo da tela. Neste caso, a mudança do foco de atenção dos objetos para a barra pode interromper o fluxo natural de realização da tarefa, causando distração ou atraso para se completar o objetivo proposto.

Figura 40: Barra de mensagens no canto inferior esquerdo da tela.



As respostas dadas à pergunta – COM QUAL DAS ALTERNATIVAS PARA SE IMPLEMENTAR AS MENSAGENS DE RETORNO PARA O USUÁRIO VOCÊ SE SENTE MAIS À VONTADE? – indicaram a opinião de cada respondente (num total de 14 indivíduos) sobre a forma utilizada em cada um dos três programas de geometria dinâmica para dialogar com o usuário. Os dados obtidos podem ser visualizados na tabela 11.

Tabela 11: Resultados da pergunta 7 do questionário.

Categories	n ^o	%
Transmissão de mensagens ao lado do cursor	12	85,71
Transmissão por barra na parte inferior da tela	1	07,14
Transmissão por ambas as maneiras	1	07,14
Total	14	100

Conforme a tabela 11 indica, parece não restar dúvida quanto ao benefício de se implementar o *feedback* em um local que corresponda ao foco de atenção visual do usuário. Enquanto 12 (85,71%) dos indivíduos pesquisados são a favor da transmissão de mensagens ao lado do cursor, apenas 1 usuário (7,14%) julga mais adequada a transmissão pela barra na parte inferior da tela, e outro (7,14%) considera esta uma mera questão de costume, sendo indiferente quanto à escolha de uma entre as duas alternativas. Praticamente todas as justificativas oferecidas representam com fidelidade os resultados da avaliação heurística:

R2 – *Sempre esqueço de olhar para o canto da tela.*

R3 – *No momento em que se clica a ferramenta desejada, automaticamente você estará visualizando a informação necessária. A visualização em resposta é imediata.*

R4 – *Visualização imediata do que se faz.*

R6 – *Mais visível.*

R7 – *As mensagens ao lado do próprio cursor facilitam a visualização do usuário.*

R8 – *A mensagem expressa ao lado do cursor é captada automaticamente conforme a manipulação, o que facilita a visualização do usuário.*

R10 – *Esse modo de dialogar com o usuário me parece ser bem mais eficiente; não há como deixar de ver.*

R11 – *A mensagem ao lado do cursor permite a leitura imediata, sem que seja necessário desviar a atenção do desenho.*

R12 – *Às vezes a gente se esquece de olhar a barra. Quando aparece ao lado do cursor, é mais imediato.*

O único respondente (R9) que considera mais adequada a transmissão de mensagens pela barra justifica que este método é *similar ao utilizado no software AUTOCAD*. Conforme explicado na seção 4.2, a formação de hábitos é um fator que pode interferir negativamente na quebra de paradigmas de interface. Eventualmente, por mais adequada que seja uma nova solução, o usuário, estando acostumado com a anterior, pode encontrar dificuldades para se adaptar à nova.

4.5 Manipulação direta

Sem dúvida, a manipulação direta é o principal estilo de interação utilizado pelos programas de geometria dinâmica. Poderoso e fácil de aprender, este estilo apresenta relações, objetos e atributos que podem ser selecionados e operados com o auxílio de um *mouse* ou de qualquer outro tipo de dispositivo de entrada gráfico (*GID*). Nos programas analisados, apresentam-se duas alternativas para se implementar a seleção e a manipulação de objetos:

- A. existe uma única ferramenta que serve tanto para selecionar os objetos quanto para manipulá-los (o Cabri Géomètre II e o The Geometer's Sketchpad 3 se utilizam desta alternativa);
- B. existem duas ferramentas, uma específica para selecionar os objetos e outra específica para manipulá-los (o Cinderella se enquadra nesta alternativa).

As ações de selecionar e manipular objetos geralmente aparecem em conjunto, ou seja, para atingir seus objetivos, os usuários estão acostumados a executar tarefas que necessitam primeiro da seleção e, logo em seguida da manipulação de objetos. Este é o motivo pelo qual, não somente os programas voltados para a geometria dinâmica, mas também aqueles destinados à edição

de imagens, editoração eletrônica e CAD costumam adotar a primeira alternativa como um padrão.

Em certos casos é válida a mudança de paradigma em relação a algum recurso tradicional de interface pois, às vezes, à medida que o usuário vai se tornando especialista, um novo método pode se apresentar mais adequado e eficiente em relação ao tradicional (Raskin, 2000). Porém, este não parece ser o caso da alternativa adotada pelo Cinderella, porque não se trata apenas da impossibilidade de se desfazer um hábito anterior mas da real inadequação do acesso à funcionalidade oferecida, evidenciando, portanto, uma falha de usabilidade.

As respostas dadas à pergunta – COM QUAL DAS ALTERNATIVAS PARA SELECIONAR E MANIPULAR OBJETOS VOCÊ SE SENTE MAIS À VONTADE? – registraram a opinião de cada participante (num total de 14 indivíduos) sobre o ferramental utilizado em cada um dos programas de geometria dinâmica para a seleção e/ou a manipulação de um objeto. Os dados obtidos podem ser visualizados na tabela 12.

Tabela 12: Resultados da pergunta 8 do questionário.

Categorias	n ^o	%
Uma única ferramenta para ambas as funções	12	85,71
Ferramentas diferentes para cada função	2	14,29
Total	14	100

A tabela 12 mostra que os resultados obtidos estão plenamente de acordo com a hipótese levantada pela avaliação heurística, pois a maioria dos respondentes (85,71%) afirmou preferir o uso de uma única ferramenta tanto para a seleção quanto para a manipulação de objetos. Os comentários abaixo expressam as razões para se adotar este método:

R4 – *É mais simples.*

R5 – *Existe uma simplificação do trabalho a ser realizado.*

R6 – *Maior velocidade na construção.*

R7 – *Uma única ferramenta facilita ao usuário.*

R8 – *Fica mais prático para o usuário.*

R9 – *Mesmo ainda não usando o Cinderella, acho mais fácil o uso de apenas uma ferramenta.*

R10 – *É mais comum em programas gráficos e a gente se acostuma.*

R12 – *Não sei se por ter aprendido primeiro assim ou se é pela simplicidade realmente. Quanto menos ‘operações’ a memorizar, tanto melhor.*

Os 2 respondentes (14,29%) que afirmaram preferir o uso de ferramentas diferentes para cada função não apresentaram justificativas. Eles correspondem aos indivíduos R1 e R3, e pode-se verificar (na tabela de dados brutos apresentada em anexo) que ambos nunca utilizaram o Cinderella, sendo apenas espectadores de apresentações realizadas pelo professor.

Muitas vezes, na construção de determinados desenhos, é comum a necessidade de selecionar mais de um objeto ao mesmo tempo. Este pode ser o caso de uma mudança única de atributos para um grupo de elementos, ou da indicação dos objetos necessários para a construção de um terceiro. Para se concluir a seleção de um conjunto de objetos, existem 4 alternativas possíveis:

- A. é preciso “clicar, arrastar e soltar” o botão esquerdo do *mouse* para envolver os objetos com um retângulo imaginário;
- B. é necessário “clicar no primeiro objeto”, manter o botão SHIFT pressionado, e ir “indicando sucessivamente os próximos” até que todos sejam selecionados;
- C. em outra janela existe uma lista com o nome dos objetos desenhados na tela; é possível usar a mesma abordagem da alternativa B, porém invés de se “clicar diretamente em cima dos elementos gráficos”, são escolhidos os nomes dos objetos pela lista textual;

D. botões com ícones, representando categorias específicas de objetos, podem ser “clicados”; cada botão seleciona somente os objetos pertencentes à categoria escolhida, por exemplo: “selecionar todos os pontos”, “selecionar todas as retas”, “selecionar todas as cônicas”.

A tabela 13 mostra quais são as abordagens utilizadas nos programas em análise. Do ponto de vista da necessidade de aplicação pelos usuários, as alternativas A e B parecem representar as mais significativas, pois assim como no caso anterior, outros programas destinados à edição de imagens, editoração eletrônica e CAD, costumam adotá-las como padrão. Não por acaso, a alternativa A está presente em 2 dos 3 programas analisados e a B está em todos.

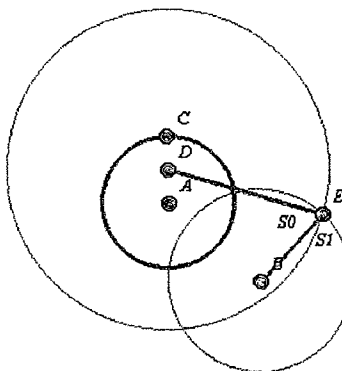
Tabela 13: Programas e abordagens para a seleção de um grupo de objetos.

	A	B	C	D
Cabri Géomètre II	x	x		
The Geometer's Sketchpad 3	x	x		
Cinderella		x	x	x

Quando se decide utilizar a alternativa B, e existem vários elementos geométricos próximos uns dos outros, é comum que algum objeto acabe sendo selecionado por engano. Nesta situação específica, a alternativa C se mostra mais prática e eficiente (figura 41). Quanto à alternativa D, acredita-se que ela seja a menos significativa, pois raras são as situações em que se necessita da seleção de todos os elementos de uma mesma categoria.

Figura 41: Lista textual dos objetos na tela.

	Who?	What?	Where?
●	A	Point(4,68 -2)	(4,68 -2)
○	C0	Circle(A;0)	$(x - 4,68)^2 + (y + 2)^2 = 1,88^2$
●	B	Point(7,28 -4,24)	(7,28 -4,24)
○	C	PointOn(C0;91,97°)	(4,62 -0,12)
●	D	Mid(C;A)	(4,65 -1,06)
○	C1	Circle(D;0)	$(x - 4,65)^2 + (y + 1,06)^2 = 4,59^2$
○	C2	Circle(B;0)	$(x - 7,28)^2 + (y + 4,24)^2 = 2,62^2$
●	E	Intersection(C1,C2)	(9,06 -2,32)
/	S0	Segment(D;E)	-
/	S1	Segment(E;B)	-



As respostas dadas à questão – **INDIQUE A SUA OPINIÃO QUANTO À RELEVÂNCIA DAS ALTERNATIVAS PARA SE IMPLEMENTAR A SELEÇÃO DE UM GRUPO DE OBJETOS** – expressaram os valores atribuídos, numa “escala de graduação” (termo utilizado por Gil, 1999) por parte de cada respondente, às opções oferecidas pelos programas de geometria dinâmica para a implementação da seleção de um determinado grupo de objetos. Os dados obtidos podem ser visualizados na tabela 14.

Tabela 14: Resultados da pergunta 9 do questionário.

Valores atribuídos Alternativas	PRIMORDIAL	SECUNDÁRIA	DESNECESSÁRIA
A	14	-	-
B	10	4	-
C	4	9	1
D	5	5	4

A tabela 14 parece estar de acordo com todas as hipóteses levantadas na avaliação heurística. Claramente, as alternativas A e B constituem-se as mais significativas segundo a opinião dos participantes. A criação de um retângulo imaginário (alternativa A) foi 100% considerada como primordial, somando 14 indicações ao todo. Quanto ao uso da múltipla seleção com o auxílio da tecla *SHIFT* (alternativa B), houve 10 indicações como primordial (71,43%) e mais 4 como secundária (28,57%). A alternativa C, provavelmente por ser a mais útil quando os elementos geométricos estão muito próximos uns dos outros, obteve somente 4 indicações como sendo primordial (28,57%), porém 9 como sendo secundária (64,3%) e uma (7,14%) como desnecessária. A alternativa D, conforme o previsto, obteve o maior número de indicações, 4 (28,57%), como sendo desnecessária.

Para se criar e manipular retas e círculos também existem variações significativas com relação à manipulação direta. As tabelas 15 e 16 mostram duas abordagens diferentes a respeito das ações físicas necessárias para se traçar uma reta. A tabela 15 mostra qual das abordagens é adotada em cada programa analisado.

Tabela 15: Alternativa A para criação de reta.

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar reta.	
Pressionar e soltar o botão esquerdo do mouse.	Ponto é criado.
Arrastar o mouse.	Reta é puxada.
Pressionar e soltar o botão esquerdo do mouse.	Reta é definida.

Tabela 16: Alternativa B para criação de reta.

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar reta.	
Pressionar o botão esquerdo do mouse.	Ponto é criado.
Arrastar o mouse.	Reta é puxada.
Soltar o botão esquerdo do mouse.	Reta é definida.

Tabela 17: Programas e respectivas abordagens para a criação de retas.

	A	B
Cabri Géomètre II	x	
The Geometer's Sketchpad 3		x
Cinderella		x

A alternativa B adota o padrão semimodo, obrigando o usuário a manter, freqüentemente, seu dedo pressionado durante uma única sessão de uso. Por uma questão de coerência, justifica-se esta alternativa pelo fato de a ferramenta para a manipulação de objetos adotar o mesmo padrão de funcionamento: selecionar, pressionar, arrastar e soltar. Entretanto, faltam argumentos que justifiquem a melhor estratégia a ser aplicada. Uma avaliação heurística, por si só, não garante a validade de qualquer hipótese levantada nestas circunstâncias.

As respostas dadas à pergunta – COM QUAL DAS ALTERNATIVAS PARA CONSTRUIR UMA RETA VOCÊ SE SENTE MAIS À VONTADE ATUALMENTE? – registraram a opinião de cada respondente (num total de 14 indivíduos) sobre a opção de sua preferência para a construção de uma reta em programas de geometria dinâmica. Os dados obtidos podem ser visualizados na tabela 18.

Tabela 18: Resultados da pergunta 10.1 do questionário.

Categorias	n ^o	%
Indiferença quanto à escolha.	5	35,71
Opção condicionada à situação.	4	28,57
Alternativa A para criação de reta.	3	21,43
Alternativa B para criação de reta.	2	14,28
Total	14	100

Ao se observar a tabela 18 pode-se perceber que as opiniões foram bem distribuídas entre os respondentes. A maioria considerou a escolha como sendo uma questão irrelevante; afinal 5 indivíduos (35,71%) selecionaram a primeira categoria. Algumas justificativas recolhidas foram:

R8 - *As duas são muito parecidas, que não faz diferença usar uma ou outra.*

R12 - *Criar retas é uma coisa tão fácil que a gente praticamente não precisa pensar sobre... faz-se automaticamente.*

Embora se possa pensar numa aparente concordância entre as opiniões apresentadas acima, as duas mostram-se controversas. Pelo fato de a criação de retas ser uma operação executada várias vezes em qualquer sessão de uso, acaba formando-se um hábito que transforma a seqüência de ações necessárias em um único gesto aplicado de forma inconsciente. Por ser assim, concorda-se com a afirmação de “ser tão fácil, não sendo preciso pensar sobre... faz-se automaticamente”. Justamente por tornar-se inconsciente, a execução deste gesto pode levar a erros quando se troca de contexto, o que na prática é verificado quando se muda de programa (se o segundo adotar um método oposto ao existente no primeiro). Por mais parecidos que possam ser os dois métodos propostos, há diferença entre um e outro quando se acostuma ao uso do primeiro e troca-se o contexto para o segundo, e vice-versa.

A segunda categoria foi escolhida por 4 indivíduos (28,57%), que julgaram a forma de se criar retas dependente do contexto de uso. Porém, nenhum dos respondentes informou qual o tipo de situação adequada para cada método proposto. Curiosamente, não parece fazer sentido que o tipo de elemento a ser

criado seja condicionado ao método físico de construção do mesmo. Neste caso, nenhuma das justificativas apresentadas inspira precisão ou confiança:

R1 - *Depende da metodologia que vamos empregar (!).*

R2 - *Depende do objeto que se vá construir (!).*

R3 - *Tudo dependerá da metodologia que se pretende usar no trabalho (!).*

Os respondentes que efetivamente escolheram um método se dividiram em 3 (21,43%) a favor da alternativa A (pressionar e soltar, arrastar, pressionar e soltar) - característica do programa Cabri Géomètre II, e apenas 2 (14,29%) a favor da alternativa B (pressionar, arrastar, soltar) – característica do programa The Geometer's Sketchpad 3. Excetuando-se a última, as seguintes justificativas mostram que o hábito de uso do Cabri parece ter sido um fator significativo para esta escolha.

R6 - *Talvez, a maior intimidade com o Cabri.*

R7 - *Trabalhamos mais com o Cabri.*

R10 - *É mais rápida e intuitiva (alternativa A).*

R11 - *É mais confortável decidir a direção da reta sem estar pressionando o botão.*

Além das abordagens anteriores, existem outras duas alternativas lógicas (podendo ser combinadas com as primeiras) para se definir uma reta (a tabela 19 mostra quais delas são adotadas em cada programa analisado):

A. a reta é definida por um único ponto, uma direção e sentido dados (figura 42);

B. a reta é definida pelo primeiro e por um segundo ponto, criado arbitrariamente no instante de conclusão da tarefa (figura 43).

Tabela 19: Programas e respectivas abordagens para a definição de retas.

	A	B
Cabri Géomètre II	x	
The Geometer's Sketchpad 3		x
Cinderella	x	x

Figura 42: Reta definida pelo método A.

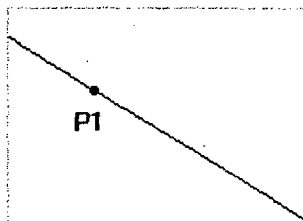
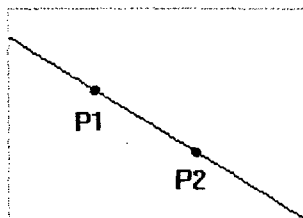


Figura 43: Reta definida pelo método B.



As duas estratégias utilizadas criam retas cujos parâmetros de definição e de controle não são equivalentes entre si. Isto significa que uma ação efetuada sobre uma reta poderá ter uma reação diferente em virtude do método de traçado inicial (A ou B). A tabela 20 mostra quais são as reações esperadas para cada ação e método proposto.

Tabela 20: Ações e reações em relação à manipulação direta de retas.

	A	B
Usuário (ação)	Programa (reação)	Programa (reação)
Selecionar um ponto que define a reta e transladá-lo para um local diferente.	A reta é transladada de acordo com a movimentação do ponto que a define.	A reta muda de direção e sentido, sendo rotacionada de acordo com a movimentação do ponto móvel, e com o centro de rotação no ponto fixo.
Selecionar diretamente a reta e transladar o cursor para um local diferente.	A reta muda de direção e sentido, sendo rotacionada de acordo com a movimentação do cursor, e com o centro de rotação no ponto que a define (no Cabri II).	A reta é transladada de acordo com a movimentação do cursor (no Sketchpad 3).

As respostas dadas à pergunta – VOCÊ CONSIDERA DESEJÁVEL QUE A FERRAMENTA PARA CONSTRUIR UMA RETA CRIE UM SEGUNDO PONTO

SOBRE A MESMA? – indicaram a opinião de cada respondente (num total de 14 indivíduos) sobre a vantagem ou não de um segundo ponto ser alocado numa reta no momento de sua construção, em programas de geometria dinâmica. Os dados obtidos podem ser visualizados na tabela 21.

Tabela 21: Resultados da pergunta 10.2 do questionário.

Categorias	n ^o	%
Criação desnecessária	9	64,29
Condicionamento da criação à situação	4	28,57
Indiferença quanto à criação (ou não)	1	7,14
Total	14	100

Na tabela 21, pode-se verificar que a maioria dos respondentes, 9 em 14 (64,29%), optou pela definição da reta por um único ponto, com uma direção e sentido dados. Conforme as opiniões apresentadas, torna-se evidente a insatisfação destes usuários em se amarrar a direção da reta pelo segundo ponto (embora houvesse o item criação desejável no questionário, este não foi assinalado por ninguém isoladamente):

R1 - *O segundo ponto amarra a reta.*

R9 - *Porque se marcamos um ponto na reta, para não nos confundirmos, temos que mudá-lo de cor.*

R10 - *Me obrigaria a escondê-lo depois.*

R11 - *A criação de um segundo ponto prende a mudança de direção da reta; acho interessante a opção de criar uma reta por um ponto e uma direção, ou por 2 pontos.*

R12 - *Não há necessidade do segundo ponto.*

As tabelas 22 e 23 mostram duas abordagens diferentes a respeito das ações físicas necessárias para se traçar um círculo. A tabela 24 mostra qual das abordagens é adotada em cada programa analisado.

Tabela 22: Alternativa A para criação de círculo
(pressionar e soltar, arrastar, pressionar e soltar).

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar círculo.	
Pressionar e soltar o botão esquerdo do mouse.	Ponto é criado.
Arrastar o mouse.	Círculo é puxado.
Pressionar e soltar o botão esquerdo do mouse.	Círculo é definido.

Tabela 23: Alternativa B para criação de círculo
(pressionar, arrastar e soltar).

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar círculo.	
Pressionar o botão esquerdo do mouse.	Ponto é criado.
Arrastar o mouse.	Círculo é puxado.
Soltar o botão esquerdo do mouse.	Círculo é definido.

Tabela 24: Programas e abordagens para a criação de círculos.

	A	B
Cabri Géomètre II	x	
The Geometer's Sketchpad 3		x
Cinderella		x

De forma equivalente ao caso da criação de retas, a alternativa B adota o padrão semimodo. Portanto, as mesmas considerações anteriores também se aplicam a esta situação específica.

As respostas dadas à pergunta – COM QUAL DAS ALTERNATIVAS PARA CONSTRUIR UM CÍRCULO VOCÊ SE SENTE MAIS À VONTADE ATUALMENTE? – sinalizaram a opinião de cada respondente (num total de 14 indivíduos) sobre a opção de sua preferência para a construção de um círculo em programas de geometria dinâmica. Os dados obtidos podem ser visualizados na tabela 25.

Tabela 25: Resultados da pergunta 11.1 do questionário.

Categorias	n ^o	%
Indiferença quanto à escolha.	5	35,71
Alternativa A para criação de círculo.	4	28,57
Opção condicionada à situação.	3	21,43
Alternativa B para criação de círculo.	2	14,29
Total	14	100

Ao se observar a tabela 25 pode-se perceber que, assim como na tabela 18, as opiniões foram igualmente bem distribuídas entre os respondentes.

Novamente a maioria considerou a escolha sendo uma questão irrelevante, pois exatamente 5 indivíduos (35,71%) selecionaram a primeira categoria. Como no caso da reta, a preferência pela alternativa A (pressionar e soltar, arrastar, pressionar e soltar) prevaleceu sobre a alternativa B (pressionar, arrastar e soltar). Algumas das justificativas apresentadas evidenciam não somente a coerência de opinião dos usuários quanto à criação de retas e círculos, mas também a provável influência do hábito de uso do programa Cabri Géomètre II:

R10 - *É mais rápida e intuitiva (alternativa A).*

R11 - *É mais confortável decidir o raio do círculo sem estar pressionando o botão.*

Além destas abordagens, existem outras duas alternativas lógicas (podendo ser combinadas com as primeiras) para se definir um círculo (a tabela 26 mostra quais delas são adotadas pelos respectivos programas):

- A. o círculo é definido por um ponto central e pela determinação do raio através da manipulação direta, causando o traçado da circunferência mas sem a existência de um segundo ponto sobre a mesma (figura 44);
- B. o círculo é definido por um ponto central e por um segundo ponto que delimita o traçado da circunferência (criado arbitrariamente no instante de conclusão da tarefa, conforme mostra a figura 45).

Tabela 26: Programas e abordagens para a definição de círculos.

	A	B
Cabri Géomètre II	x	
The Geometer's Sketchpad 3		x
Cinderella	x	x

Figura 44: Círculo definido pelo método A.

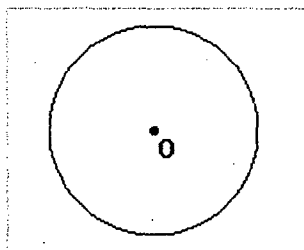
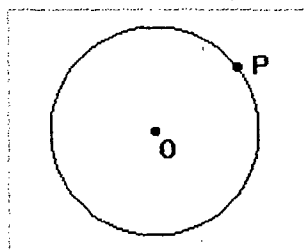


Figura 45: Círculo definido pelo método B.



Assim como no caso das retas, as duas estratégias utilizadas criam círculos cujos parâmetros de definição e de controle não são equivalentes entre si. Isto significa que uma ação efetuada sobre um círculo poderá ter uma reação diferente em função do método de traçado inicial (A ou B). A tabela 27 mostra quais são as reações esperadas para cada ação e método proposto.

Tabela 27: Ações e reações em relação à manipulação direta de círculos.

	A	B
Usuário (ação)	Programa (reação)	Programa (reação)
Selecionar o ponto que define o centro do círculo e transladá-lo para um local diferente.	O círculo é transladado de acordo com a movimentação do seu ponto central, e o valor de seu raio é mantido constante.	O centro do círculo é transladado de acordo com a movimentação do cursor, mas o valor do raio também muda, pois o segundo ponto que delimita o traçado da circunferência é mantido fixo.
Selecionar diretamente o círculo e transladar o cursor para um local diferente.	Muda-se o valor do raio conforme a movimentação do cursor, enquanto o ponto que define o centro do círculo é mantido fixo.	O círculo é transladado de acordo com a movimentação do cursor, e o valor de seu raio é mantido constante. (no Sketchpad 3).
Selecionar o segundo ponto, aquele que delimita o traçado da circunferência, e transladá-lo para um local diferente.	Esta ação não se aplica em um círculo criado pelo método A, pois não existe um Segundo ponto neste caso.	Muda-se o valor do raio conforme a movimentação deste segundo ponto, enquanto o primeiro (ponto que define o centro do círculo) é mantido fixo.

As respostas dadas à pergunta – VOCÊ CONSIDERA DESEJÁVEL QUE A FERRAMENTA PARA CONSTRUIR UM CÍRCULO CRIE UM SEGUNDO PONTO SOBRE O MESMO? – indicaram a opinião de cada respondente (num total de 14 indivíduos) sobre a vantagem ou não de um segundo ponto ser alocado num círculo no momento de sua construção, em programas de geometria dinâmica. Os dados obtidos podem ser visualizados na tabela 28.

Tabela 28: Resultados da pergunta 11.2 do questionário.

Categorias	n ^o	%
Criação desnecessária	7	50,00
Condicionamento da criação à situação	6	42,86
Indiferença quanto à criação (ou não)	1	7,14
Total	14	100

Na tabela 28, verifica-se grande semelhança com a tabela 21, na qual maioria dos respondentes optou pela definição da reta por um único ponto, com uma direção e sentido dados. Neste caso, a metade - 7 entre 14 indivíduos - optou pela definição de um círculo sem a existência de um segundo ponto sobre o mesmo. Embora houvesse o item “criação desejável” no questionário, este não foi assinalado por ninguém isoladamente. Algumas das opiniões oferecidas por estes usuários são apresentadas a seguir:

R9 - *Não, porque se marcamos um ponto no círculo, para não nos confundirmos, temos que mudá-lo de cor.*

R10 - *Me obrigaria a escondê-lo depois.*

R11 - *Da mesma forma que acontece com a reta, é interessante a liberdade de modificar o raio por qualquer ponto da circunferência.*

R12 - *Quando a gente quer que o círculo fique preso em um ponto, é interessante esse segundo ponto. Se não houver essa necessidade, não precisa ter o ponto.*

4.6 Animação

Existem diversas maneiras e recursos associados para se implementar a animação de objetos em programas de geometria dinâmica (tabela 29):

- A. múltipla animação (animação de mais de um objeto ao mesmo tempo);
- B. associação de um botão específico para cada animação preestabelecida;
- C. controle da velocidade de animação;
- D. escolha do sentido sobre o trajeto (no caso de ser selecionado um ponto e um caminho).

Tabela 29: Programas e respectivos recursos para a animação de objetos.

	A	B	C	D
Cabri Géomètre II	x		x	X
The Geometer's Sketchpad 3	x	x	x	
Cinderella			x	

As respostas dadas à questão – **INDIQUE A SUA OPINIÃO QUANTO À RELEVÂNCIA DOS DIFERENTES RECURSOS PARA SE IMPLEMENTAR A ANIMAÇÃO** – sinalizaram os valores atribuídos, numa “escala de graduação” (Gil, 1999) por parte de cada respondente, às alternativas oferecidas pelos programas de geometria dinâmica para a animação de objetos. Os dados obtidos podem ser visualizados na tabela 30.

Tabela 30: Resultados da pergunta 12 do questionário.

Recursos	Valores atribuídos	ESSENCIAL	SECUNDÁRIO	DESNECESSÁRIO
A		12	2	-
B		3	8	3
C		11	3	-
D		12	2	-

Pelos resultados de frequência obtidos na tabela 30, pode-se verificar que os recursos A, C e D parecem ser os mais importantes, pois todos receberam um grande número de indicações favoráveis (de 11 a 12, ou seja, de 78,57% a 85,71%). No entanto, o recurso B (associação de um botão específico para

cada animação preestabelecida) recebeu 8 indicações como sendo secundário, 3 como sendo desnecessário e também apenas 3 como sendo essencial. Das opiniões recolhidas, as mais relevantes são apresentadas a seguir:

R6 - O recurso B é bastante útil desde que adequado às demais opções (sentido etc).

R8 - Todos os recursos são necessários conforme a utilização.

R10 - Seria ideal que todos os programas oferecessem as várias possibilidades.

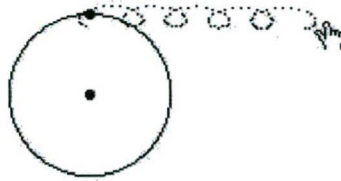
R11 - O recurso B é prático pois permite que se possa salvar uma animação (às vezes temos vários elementos para animar) mas não chega a ser essencial.

R13 - Não considero a animação como uma das características importantes do software de geometria dinâmica. Acho que ela induz o professor a realizar o trabalho do aluno que pode ficar reduzido ao papel de espectador de uma animação, o que não é um uso relevante dos software de geometria dinâmica.

Mesmo quando dois ou mais programas contêm um recurso em comum, a qualidade de acesso e de uso do mesmo costuma diferir significativamente entre eles. O controle da velocidade de animação é um exemplo deste fato.

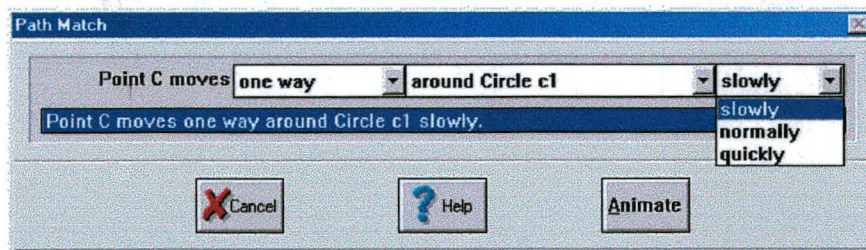
No Cabri Géomètre II, ao utilizar a técnica de semimodo (pressionar, arrastar e soltar), o usuário cria um dispositivo virtual que lembra uma mola sendo esticada. Este dispositivo, além de representar um mapeamento natural que facilita o reconhecimento por analogia, agrega dois benefícios consideráveis à animação (figura 46): é possível tanto a escolha do sentido sobre o trajeto quanto o controle da sua velocidade (quanto mais esticada estiver a mola, mais veloz será o movimento; quanto menos esticada estiver a mola, mais lento será o movimento).

Figura 46: Controle de velocidade da animação no Cabri Géomètre II.



No The Geometer's Sketchpad 3, o método de controle da velocidade varia significativamente do correspondente anterior. Conforme mostra a figura 47, é oferecido um menu com três opções de velocidade.

Figura 47: Controle de velocidade da animação no Sketchpad.



Ao utilizar o Cabri, o usuário detém um maior controle sobre a velocidade que o objeto pode alcançar. O dispositivo que imita a mola, além de permitir uma estimativa inicial do percentual de velocidade da animação, também garante a escolha de valores intermediários entre os limites inferior e superior. No caso do Sketchpad só existem três valores disponíveis. A velocidade pode ser: (1) lenta, (2) normal ou (3) rápida. O problema se agrava quando percebe-se que estes valores são relativos, e dependem da maneira pela qual a figura foi construída. Figuras simples determinam grandes velocidades²¹, mesmo quando se escolhe a primeira opção (lenta).

Ao ser comparado com o Cabri e com o Sketchpad, verifica-se que o Cinderella possui um controle da velocidade mais simples, prático e adequado aos propósitos do usuário. A figura 48 mostra o diálogo de controle da animação, no qual utiliza-se uma barra horizontal deslizante. Quando o usuário "clica" e arrasta o pequeno traço em branco sobre esta barra, ele pode escolher

²¹ Dependendo do tamanho das figuras e até mesmo da capacidade de processamento do computador.

qualquer velocidade desejada, indo de zero até seu valor máximo. Como o controle é executado em paralelo com a animação, pode-se começar com uma velocidade e aumentá-la ou diminuí-la conforme seus propósitos. Tem-se também a alternativa de parar o movimento e retomá-lo com a mesma intensidade a partir de qualquer posição pré-determinada.

Figura 48: Controle de velocidade da animação no Cinderella.



Quanto à ordem de comandos para se animar um elemento no Cabri, primeiro deve-se ativar a ferramenta de animação para depois se escolher o objeto desejado. Já no Sketchpad, primeiro são indicados o objeto e o caminho, para depois ser acessado o comando de animação. No Cabri não há necessidade de se indicar um caminho, pois o programa deduz qual o trajeto a ser percorrido pela própria geometria da figura. Por exemplo, um ponto criado sobre um círculo só pode ser animado sobre este próprio círculo. Neste caso específico, o Cinderella segue a mesma abordagem adotada pelo Cabri: se a dedução do trajeto for possível, o programa se satisfaz com a indicação de apenas um parâmetro (o ponto). Os resultados das animações destes programas variam significativamente de acordo com as situações apresentadas. Observe-se a tabela 31.

Tabela 31: Possibilidades de animação entre os programas analisados.

Parâmetros oferecidos	Cabri Géomètre II	Sketchpad 3	Cinderella
[1] Somente o ponto (completamente solto).	Anima sobre uma reta, de acordo com a direção e sentido indicados.	Não anima, pois é necessária a indicação de um caminho.	Não anima, pois para o Cinderella o ponto precisa ter ao menos um caminho.
[1] Somente o ponto (pertencente a um único caminho).	Anima sobre este caminho.	Não anima, pois é necessária a indicação de um caminho.	Anima sobre este caminho.
[1] Ponto e [2] caminho único ao qual ele pertence.	Não anima, pois não faz sentido a seleção de mais de um parâmetro. O item anterior é uma alternativa possível.	Anima sobre este caminho.	Não anima, pois não faz sentido a seleção de mais de um parâmetro se o ponto só pertence a um caminho. O item anterior é uma alternativa possível.
[1] Somente o ponto (solto mas com um caminho cuja definição depende dele).	Anima sobre uma reta, de acordo com a direção e sentido indicados.	Não anima, pois é necessária a indicação de um caminho.	Anima sobre este caminho.
[1] Ponto (solto) e [2] caminho cuja definição depende deste ponto (referente ao 1º exemplo mostrado adiante).	Não anima, pois não faz sentido a seleção de mais de um parâmetro. O item anterior é uma alternativa possível.	Não anima, justamente pelo fato de o caminho depender do ponto.	Anima sobre este caminho, desde que exista ao menos um outro caminho que também depende deste ponto.
[1] Ponto (completamente solto) e [2] caminho ao qual ele não pertence (referente ao 2º exemplo mostrado adiante).	Não anima, pois faz sentido a seleção de mais de um parâmetro.	Tira o ponto de onde ele está e anima sobre este caminho.	Não anima. Não se permite a seleção de um caminho ao qual o ponto não pertence, ou de um que ele não define.
[1] Ponto (pertencente a outro caminho) e [2] caminho ao qual ele não pertence (referente ao 3º exemplo mostrado adiante).	Não anima, pois não faz sentido a seleção de mais de um parâmetro.	Anima (mas de forma peculiar sobre o caminho ao qual ele pertence).	Não anima. Não se permite a seleção de um caminho ao qual ele não pertence, ou de um que ele não define.

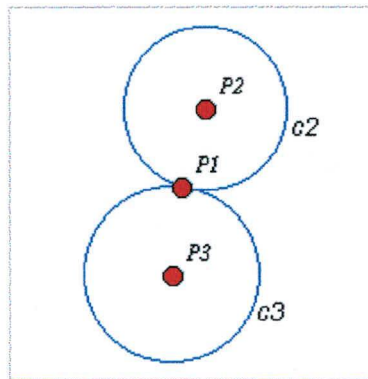
As três últimas situações mostradas na tabela 31 evidenciam claramente as diferentes concepções dos projetistas quanto a um modelo ideal para se representar a geometria dinâmica. Para se investigar as reações dos usuários a respeito destas concepções, foram elaborados três exemplos correspondentes, sendo um para cada situação.

4.6.1 Exemplo 1

Observe-se a seguinte construção (figura 49):

1. criação de um ponto P1 qualquer;
2. criação de um círculo c2, passando por P1, de centro P2 qualquer (diferente de P1);
3. criação de um círculo c3, passando por P1, de centro P3 qualquer (diferente de P1 e P2).

Figura 49: Primeiro exemplo de animação.



Na situação apresentada, levando-se em conta as características dos programas analisados, existem três possibilidades para se trabalhar com o recurso de animação:

- A. não ser possível a animação de P1;
- B. ser possível a animação de P1 mas sem associar seu trajeto ao círculo c2 ou c3;
- C. ser possível a animação de P1 ao longo do círculo c2 e também no círculo c3, sendo apenas estes trajetos possíveis.

As respostas dadas à pergunta – QUE COMPORTAMENTO VOCÊ CONSIDERA COERENTE NO QUE RESPEITA A ANIMAÇÃO DO PONTO DE INTERSEÇÃO DE 2 CÍRCULOS? – expressaram a opinião de cada respondente (num total de 14 indivíduos) quanto à coerência das opções de

animação para um ponto, construído inicialmente, pertencente a dois círculos traçados a posteriori. Os dados obtidos podem ser visualizados na tabela 32.

Tabela 32: Resultados da pergunta 13 do questionário.

Categorias	n ^o	%
Alternativa C	11	78,57
Alternativa B	2	14,29
Alternativa A	1	07,14
Total	14	100

Como se pode observar na tabela 32, a maioria dos respondentes - 11 em 14 (78,57%) - optou pela alternativa C, indicando que a animação do ponto sobre os dois círculos deve ser possível. No entanto esta é uma questão polêmica, pois deve ser encarada de duas formas distintas. Se por um lado é dada uma maior liberdade para o usuário através da possibilidade de escolha entre os dois trajetos possíveis, por outro existe algo de incompatível com o modelo conceitual imposto pela geometria dinâmica. Os comentários apresentados abaixo parecem demonstrar este último ponto de vista:

R11 - O ponto é livre e deve se movimentar livremente; as circunferências é que devem acompanhar o movimento do ponto. Penso que, se o programa tem por objetivo ensinar a geometria, é importante manter as propriedades geométricas dos elementos; entretanto, se usado para outras finalidades, esses recursos podem facilitar a execução de um desenho. Ainda assim, dou preferência ao recurso de "redefinição do objeto", disponível no Cabri.

R12 - O primeiro item é possível, pois se P1 é um ponto qualquer, pode ser movimentado para onde se queira, como no Cabri (se bem que aí os 2 círculos variam). Não sei animar no Sketchpad mas pude mover o ponto arbitrariamente. Idem no Cinderella. O terceiro item é possível no Cinderella, apesar de eu achar meio estranho. No Cabri isso não é possível pois é o círculo que está amarrado no ponto e não o inverso.

R13 - P1 é um ponto livre.

Os comentários acima foram feitos pelos respondentes que não optaram pela alternativa A, sendo os 2 primeiros (14,29%) indicadores da alternativa B e

o terceiro (7,14%) da alternativa C. É pertinente assinalar que todas as 3 opiniões foram expressas por professores que conhecem e utilizam a geometria dinâmica há vários anos (como pode-se observar na tabela de dados brutos apresentada em anexo), portanto, os usuários mais experientes que contribuíram para a realização desta pesquisa.

A maioria dos comentários apresentados pelos outros respondentes - aqueles que optaram pela alternativa C - parece demonstrar uma valorização deste comportamento pelas possibilidades de aplicação do mesmo, deixando em segundo plano a visão da geometria dinâmica como um sistema baseado em restrições:

R3 - *O terceiro item lhe dá possibilidade de animação sem nenhuma dificuldade.*

R6 - *O fato de termos mais uma possibilidade de animação.*

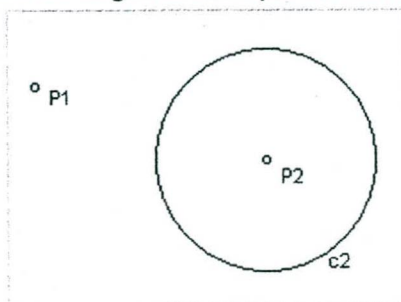
R10 - *Parece-me ser mais útil na resolução de problemas.*

4.6.2 Exemplo 2

Observe-se a seguinte construção (figura 50):

1. criação de um ponto P1 qualquer;
2. criação de um círculo c2, não contendo P1, de centro P2 qualquer (diferente de P1).

Figura 50: Segundo exemplo de animação.



Na situação apresentada, levando-se em conta as características dos programas analisados, existem duas possibilidades para se trabalhar com o recurso de animação:

- A. ser possível a animação de P1 ao longo do círculo c2;
- B. não ser possível a animação de P1 ao longo do círculo c2.

As respostas dadas à pergunta – QUE COMPORTAMENTO VOCÊ CONSIDERA COERENTE PARA SER IMPLEMENTADO NUM PROGRAMA DE GEOMETRIA DINÂMICA IDEAL NO QUE RESPEITA A ANIMAÇÃO DE UM PONTO EXTERIOR A UM CÍRCULO? – registraram a opinião de cada respondente (num total de 14 indivíduos) quanto à possibilidade de animar-se um ponto, construído inicialmente como exterior ao círculo e, automaticamente redefinido, usar a circunferência de círculo como trajetória. Os dados obtidos podem ser visualizados na tabela 33.

Tabela 33: Resultados da pergunta 14 do questionário.

Categorias	n ^o	%
Alternativa A	7	50
Alternativa B	7	50
Total	14	100

A tabela 33 mostra que 7 usuários escolheram a alternativa A (50%) e 7 também escolheram a alternativa B (50%), evidenciando um resultado totalmente equilibrado. Ao se afirmar que é possível a animação de P1 ao longo do círculo c2, deturpa-se o conceito da geometria dinâmica como sendo um sistema baseado em restrições (assim como no exemplo 2), pois o ponto P1 está livre e conseqüentemente não pertence ao círculo c2. Este deve ser o motivo pelo qual todos os 5 professores (além de 2 alunos), que conhecem e utilizam a geometria dinâmica há vários anos, optaram pela alternativa B. Os comentários apresentados abaixo aparentam estar de acordo com este último ponto de vista:

R9 - *A não ser que o comportamento do ponto P1 ao longo do círculo c2 seja justificado graficamente / matematicamente. Não sei nada sobre isso.*

R10 - *Seria um recurso a mais que possibilitaria redefinir o objeto, porém acho-o perfeitamente dispensável e pode confundir o usuário.*

R11 - *O ponto P1 não tem nenhuma relação geométrica com o círculo; sua movimentação deve ser independente.*

R12 - *Não tem por que isso acontecer. Se algum dos programas fizer isso, gostaria de saber SE e POR QUE faz isso.*

R13 - *P1 é um ponto livre, não há caminho para ser percorrido, logo não faz sentido falar em sua animação.*

No entanto, os respondentes que se mostraram a favor da alternativa A apresentaram uma outra visão do problema, conforme mostram as seguintes opiniões:

R2 - *Mais coerente à dinâmica do exercício.*

R5 - *É importante que ele possa ser animado para que existam possibilidades de diferentes estudos na figura.*

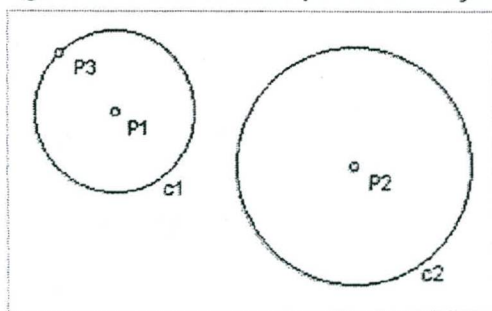
R8 - *Seria melhor para o usuário que o ponto fosse animado no círculo, mesmo não estando contido neste, porém mostrando o caminho percorrido pelo ponto.*

4.6.3 Exemplo 3

Observe-se a seguinte construção (figura 51):

1. criação de um círculo c1 qualquer, de centro P1;
2. criação de um círculo c2 qualquer, diferente de c1, de centro P2;
3. criação de um ponto P3 qualquer, pertencente ao círculo c1.

Figura 51: Terceiro exemplo de animação.



Na situação apresentada, levando-se em conta as características dos programas analisados, existem duas possibilidades para se trabalhar com o recurso de animação:

- A. ser possível a animação de P3 somente ao longo do círculo c1;
- B. ser possível a animação de P3 em torno do círculo c1 e também quando se aponta o círculo c2 como caminho.

As respostas dadas à pergunta – QUE COMPORTAMENTO VOCÊ CONSIDERA COERENTE PARA SER IMPLEMENTADO NUM PROGRAMA DE GEOMETRIA DINÂMICA IDEAL NO QUE RESPEITA A ANIMAÇÃO DE UM PONTO PERTENCENTE A UM CÍRCULO USANDO-SE OUTRO CÍRCULO COMO REFERENCIAL DE TRAJETÓRIA? – expressaram a opinião de cada respondente (num total de 14 indivíduos) quanto à possibilidade de animar-se um ponto pertencente a um círculo quando se indica um segundo círculo como caminho. Os dados obtidos podem ser visualizados na tabela 34.

Tabela 34: Resultados da pergunta 15 do questionário.

Categorias	n ^o	%
Alternativa A	8	57,14
Alternativa B	6	42,86
Total	14	100

A tabela 34 mostra que 8 usuários escolheram a alternativa A (57,14%) e 6 escolheram a alternativa B (42,86%), evidenciando um resultado muito próximo ao encontrado no exemplo anterior. A segunda alternativa aparenta incoerência, dando margem a um problema de usabilidade, pois o ponto P3

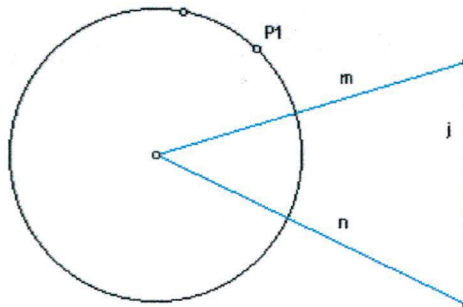
pertence ao círculo c_1 , o que deveria impossibilitar a sua animação em torno de c_2 . Provavelmente, este constitui o motivo pelo qual todos os 5 professores, que conhecem e utilizam a geometria dinâmica há vários anos, optaram pela alternativa A. Os comentários apresentados abaixo parecem demonstrar este ponto de vista:

R10 – *Apesar de provavelmente ter alguma utilidade, acho que confunde o usuário, pois o ponto não fica pertencendo ao segundo círculo.*

R12 – *Não consigo ver uma lógica para a segunda opção, nos moldes em que foi posta a questão. Talvez até seja interessante que aconteça essa dupla rotação, como por exemplo para simular a rotação da lua em torno da Terra e esta em torno do Sol, mas da maneira como foi colocada a questão, acho incoerente.*

No entanto, um dos alunos respondentes, ao explorar uma variação deste exemplo no programa The Geometer's Sketchpad 3, descobriu uma possível aplicação quando se anima um ponto solto “sobre um trajeto” ao qual ele não pertence. A figura 52 ilustra esta situação. De acordo com uma das observações feitas pelo aplicador do questionário, o estudante experimentou animar um ponto pertencente a um círculo (P_1), dando como caminho um segmento de reta (j). Daí, descobriu que a trajetória do ponto ficava limitada ao arco subtendido entre os segmentos que iam do centro do círculo aos extremos do segmento traçado (m e n).

Figura 52: Possível aplicação apontada por um estudante.



Ao averiguarem esta possibilidade, segundo um comentário anotado pelo aplicador, vários outros alunos parecem ter sido influenciados pelo colega, julgando (erroneamente) haver alguma aplicação também para o exemplo 3, ao optarem pela alternativa B. As justificativas apresentadas foram:

R2 – *Interesse didático.*

R3 – *Sim, levando para física. Foco de luz.*

R4 – *Sim, em desenho – projeção cônica.*

R5 – *Há interesse nesta fase pois os recursos têm forte apelo didático.*

R7 – *Desde que se mostre o caminho a ser percorrido pelo ponto.*

Nada parece ilustrar melhor esta situação do que a irônica analogia que Cooper (1995) oferece. O autor comenta que o projeto de *software* centrado em usuários deve ser encarado como um projeto de aquário centrado em peixes. Não se costuma perguntar aos peixes como eles gostariam que fosse construído o seu aquário, significando que nem sempre as opiniões dos usuários a respeito do *software* a ser utilizado devem ser levadas ao pé da letra. Cabe aos especialistas em IHC, observar e compreender as suas necessidades e intenções.

4.7 Lugar Geométrico

No capítulo 3 foi mostrado que um dos benefícios da geometria dinâmica corresponde à possibilidade de construção de lugares geométricos (*loci*, plural de *locus*), mas até então nenhuma diferença específica entre as interfaces dos programas avaliados foi motivo de discussão. Retoma-se aqui este assunto com base nas reações daqueles que responderam ao instrumento de pesquisa utilizado.

Os programas Cabri Géomètre II e The Geometer's Sketchpad 3, ao contrário do Cinderella, possibilitam a colocação de pontos sobre um *locus*. Assim, se em determinada situação um *locus* parece resultar em um elemento geométrico conhecido, tal como uma reta, uma cônica, ou um círculo, pode-se comprovar esta informação quando se constrói o elemento correspondente a partir de pontos amarrados sobre o mesmo. Para ilustrar esta situação, mostra-se o seguinte diálogo, ocorrido através de correio eletrônico (em 18 de agosto de 2001), entre o pesquisador, autor deste trabalho, e um dos criadores do Cinderella, Ulrich Kortenkamp:

Pesquisador - A criação de pontos amarrados a qualquer locus é permitida no Cabri e no Sketchpad, mas não é possível no Cinderella (estou errado?). Eu tento imaginar o porquê disso. Este recurso é contra a fundamentação matemática do programa?

Ulrich Kortenkamp - Por um lado sim, por outro não. A razão de este recurso não ser possível é parcialmente causada por nossa atual implementação e pela nossa crença de que não é bom introduzir características que não sejam completamente compreendidas por nós.

Pesquisador - Por exemplo: Imagine que seja o caso de você ter descoberto um locus "que aparentemente representa uma curva cônica", e você quisesse achar a interseção entre este locus e uma reta isolada. Como você faria isso? Usando o Cabri, eu posso criar 5 pontos sobre o locus, então definir a cônica (escondendo o locus), e finalmente descobrir os pontos de interseção. Infelizmente o Sketchpad não implementa o recurso de se criar cônicas, mas ao menos é possível que sejam definidos pontos sobre o locus.

Ulrich Kortenkamp - Felizmente para nós, seu exemplo mostra algo particularmente interessante a respeito do Cinderella. Se você cria uma curva cônica como um locus, o Cinderella mostra a equação que define a

curva no texto de construção. Então você pode estar bem certo de que este locus representa uma cônica. Também, se criar cinco pontos no locus (por algum outro método que não signifique simplesmente amarrá-los nele), o verificador de integridade de teoremas irá mostrar que a cônica através destes cinco pontos já terá sido determinada.

As respostas dadas à pergunta – UM PROGRAMA DE GEOMETRIA DINÂMICA DEVE PERMITIR A COLOCAÇÃO DE PONTOS SOBRE UM LOCUS (LUGAR GEOMÉTRICO)? – indicaram a opinião de cada respondente (num total de 14 indivíduos) sobre a vantagem ou não de ser possível alocar pontos sobre um lugar geométrico, em programas de Geometria dinâmica. Os dados obtidos podem ser visualizados na tabela 35.

Tabela 35: Resultados da pergunta 16 do questionário.

Comportamentos	n ^o	%
Deve permitir pontos sobre o <i>locus</i> .	11	78,57
Não deve permitir pontos sobre o <i>locus</i> .	3	21,43
Total	14	100

A tabela 35 indica que a maioria dos respondentes, 11 em 14 (78,57%), julga conveniente a possibilidade de se amarrar pontos sobre lugares geométricos, enquanto 3 indivíduos (21,43%) acreditam que não há necessidade de se implementar este recurso. Não por acaso, 2 destes últimos afirmam nunca tê-lo utilizado em qualquer uma de suas construções.

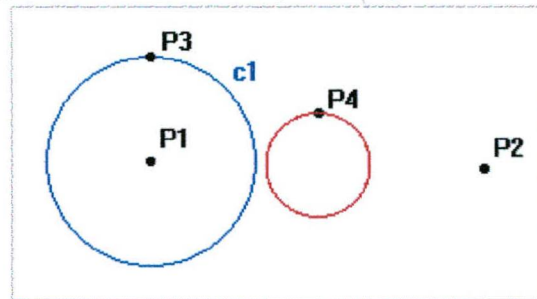
Em geral, um *locus* é determinado através da indicação de dois pontos: (1) um ponto móvel e (2) outro dependente deste primeiro, sendo o último responsável pela descrição de seu traçado. O exemplo a seguir mostra que existem 3 abordagens possíveis para se identificar estes pontos.

4.7.1 Exemplo 1

Observe-se a seguinte construção (figura 53):

1. criação de um círculo c_1 qualquer, de centro P_1 ;
2. criação de um ponto P_2 qualquer, diferente de P_1 e não pertencente ao círculo c_1 ;
3. criação de um ponto P_3 qualquer, pertencente ao círculo c_1 ;
4. criação do ponto P_4 , ponto médio entre P_2 e P_3 ;
5. criação do *locus* de P_4 em relação a P_3 .

Figura 53: Primeiro exemplo de Lugar Geométrico.



Na situação apresentada, levando-se em conta as características dos programas analisados, existem 3 possibilidades para se criar um *locus*:

- A. primeiro deve haver a indicação de P_4 (ponto dependente, descreve o traçado) e depois de P_3 (ponto móvel);
- B. primeiro deve haver a indicação de P_3 (ponto móvel) e depois de P_4 (ponto dependente, caracteriza o traçado);
- C. tanto faz a ordem de indicação (o programa deve descobrir que pontos são esses); o importante é que os dois sejam selecionados.

As respostas dadas à pergunta – QUE ORDEM DE PARÂMETROS VOCÊ CONSIDERARIA COERENTE EM UM PROGRAMA DE GEOMETRIA DINÂMICA IDEAL PARA A CONSTRUÇÃO DO *LOCUS*? – sinalizaram a opinião de cada respondente (num total de 14 indivíduos) quanto à coerência das opções de criação de *locus* no que respeita a ordem de seleção dos pontos móvel e dependente. Os dados obtidos podem ser visualizados na tabela 36.

Tabela 36: Resultados da pergunta 17 do questionário.

Comportamentos	n ^o	%
Alternativa C	9	64,29
Alternativa A	3	21,43
Alternativa B	2	14,29
Total	14	100

A tabela 36 mostra que a maioria dos respondentes, 9 em 14 (64,29%), optou pela alternativa C, sendo que as alternativas A e B receberam consecutivamente 3 (21,43%) e 2 (14,29%) indicações ao todo. O Cabri Géomètre II se utiliza da alternativa A, o Cinderella da alternativa B e o Sketchpad da alternativa C.

A indiferença quanto à ordem de indicação dos pontos é uma solução que traz como principal benefício a proteção contra erros acidentais, pois mesmo que o usuário acredite existir uma ordem lógica específica, na ocorrência da troca de parâmetros ele verificará que mesmo assim o programa é capaz de criar o *locus*. No entanto, neste caso é questionável se o usuário realmente compreende o conceito e como estão relacionados estes elementos na criação de um *locus*. As duas outras alternativas, ao forçarem a utilização de uma lógica de uso, exigem que o usuário saiba exatamente qual é o ponto móvel e qual é o ponto dependente. As justificativas abaixo mostram como estes dois pontos de vista foram manifestados pelos usuários:

R3 - *É necessário seguir um raciocínio diante da construção.*

R4 - *Prefiro seguir uma lógica de raciocínio.*

R2 - *O fato importante é que o locus é dado, não importa o processo.*

R11 - *Se tiver que ter uma ordem, que seja a primeira opção; no entanto, a melhor opção é a de selecionar "sem ordem"; o lugar geométrico é uma consequência única da relação entre esses elementos.*

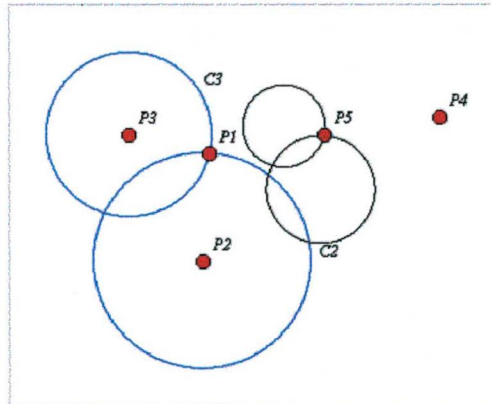
O próximo exemplo mostrado é uma variação do primeiro exemplo pertencente à seção 4.6, sobre animação. O contexto é semelhante pois lida basicamente com o mesmo problema. A apresentação desta questão aos respondentes foi proposital para se investigar a coerência de suas opiniões.

4.7.2 Exemplo 2

Observe-se a seguinte construção (figura 54):

1. criação de um ponto P1 qualquer;
2. criação de um círculo c2, passando por P1, de centro P2 qualquer (diferente de P1);
3. criação de um círculo c3, passando por P1, de centro P3 qualquer (diferente de P1 e P2);
4. criação de um ponto P4 qualquer, diferente de P1, P2 e P3;
5. criação do ponto P5, ponto médio entre P1 e P4.

Figura 54: Segundo exemplo de Lugar Geométrico.



Na situação apresentada, levando-se em conta as características dos programas analisados, se for desejado o *locus* de P5 em relação a P1 pode-se pensar em duas possibilidades diferentes:

- A. não pode ser traçado o *locus* pedido;
- B. além do ponto móvel (P1) e do ponto dependente (P5), deve ser necessário informar o caminho percorrido pelo ponto móvel, que neste exemplo pode ser c2 ou c3; cada escolha determinará um *locus* diferente.

As respostas dadas à pergunta – QUE COMPORTAMENTO VOCÊ CONSIDERARIA COERENTE PARA SER IMPLEMENTADO NUM PROGRAMA DE GEOMETRIA DINÂMICA IDEAL NO CASO DA CONSTRUÇÃO DE UM *LOCUS* QUE ADMITE DOIS PONTOS MÓVEIS DISTINTOS? – indicaram a opinião de cada respondente (num total de 14 indivíduos) quanto à coerência das opções de criação do *locus* de um elemento geométrico, quando este pode ser considerado dependente de dois pontos distintos. Os dados obtidos podem ser visualizados na tabela 37.

Tabela 37: Resultados da pergunta 18 do questionário.

Comportamentos	nº	%
Alternativa B	9	64,29
Alternativa A	5	35,71
Total	14	100

Ao se comparar os resultados da tabela 37 com os da tabela 32 pode-se perceber que entre os 14 participantes, 4 parecem ter mudado de opinião em relação à possibilidade de se movimentar o ponto P1, pertencente aos círculos c1 e c2. Afinal, os 2 lugares geométricos indicados na figura 55 são resultados indiretos de supostas animações de P1 sobre os círculos c1 e c2. Mesmo assim, acredita-se que o resultado esteja de acordo com o obtido anteriormente pois a maioria dos respondentes, 9 em 14 (64,29%), julgou ser possível a criação dos 2 *locus* (alternativa B). Algumas das justificativas apresentadas foram:

R10 - *Proporciona uma visão mais ampla das relações entre o ponto móvel e o ponto dependente.*

R11 - *Mantenho a opinião de que, se desejo que o ponto P1 esteja sobre c2, devo construí-lo sobre ele mesmo. Lembro, porém, que depende do objeto para o qual o programa esteja sendo utilizado.*

R13 - *P1 e P5 são pontos livres.*

4.8 Macros e Scripts

Entre os programas de geometria dinâmica analisados, apenas o Cabri Géomètre II e o The Geometer's Sketchpad 3 possuem recursos para a construção de ferramentas personalizadas, denominados *macros* no Cabri e *scripts* no Sketchpad. Em termos conceituais, ambos têm um significado em comum, e correspondem à gravação da seqüência de construções geométricas de objetos e das relações entre os mesmos. Quando lida-se com a criação e o uso de ferramentas (que disparam a execução de algoritmos estruturados), existem duas etapas com propósitos diferentes:

1. Criação da ferramenta - o usuário gasta o tempo necessário para raciocinar em cima de um problema que deseja resolver. Ele interage com o sistema, reflete, faz questionamentos e descobre uma solução com a ajuda dos recursos que lhe são oferecidos. Uma vez que ele tenha descoberto e compreendido todo o processo de construção, é possível então criar uma ferramenta que esconda e englobe a seqüência completa dos passos para a resolução do modelo.
2. Uso da ferramenta – o seu processo de criação foi elaborado na etapa anterior, e o usuário não está mais envolvido com o raciocínio que o levou à construção do algoritmo inicial. A sua intenção, ao selecionar a nova ferramenta, pode ser a obtenção do traçado automaticamente pela indicação de poucos parâmetros iniciais, não deixando vestígios da maneira pela qual foi elaborado. Ele pode até mesmo construir novos elementos a partir do modelo obtido. Entretanto, se o algoritmo inicial for ineficaz, à medida que novos elementos forem surgindo poderá haver uma queda na qualidade de serviço da aplicação. O próprio usuário deverá pensar em uma solução mais adequada.

Existem basicamente três alternativas para se implementar a criação de *macros* (a tabela 38 mostra quais delas são adotadas pelos respectivos programas):

- A. usuário constrói o modelo, seleciona todos os seus elementos, e depois identifica a nova *macro*;
- B. usuário começa a gravação de seus comandos, constrói o modelo e depois pára de gravar (englobando todos os passos), identificando a nova *macro*;
- C. usuário deve escolher os objetos iniciais, depois os objetos finais e então identificar a nova *macro*.

Tabela 38: Programas e respectivas alternativas para a criação de macros.

	A	B	C
Cabri Géomètre II			x
The Geometer's Sketchpad 3	x	x	

As respostas dadas à pergunta – COM QUAL DAS ALTERNATIVAS PARA SE IMPLEMENTAR A CRIAÇÃO DE MACROS VOCÊ SE SENTE MAIS À VONTADE ATUALMENTE? – registraram a opinião de cada respondente (num total de 14 indivíduos) quanto ao procedimento de preferência para a criação de macro-construções nos programas de geometria dinâmica. Os dados obtidos podem ser visualizados na tabela 39.

Tabela 39: Resultados da pergunta 19 do questionário.

Categorias	nº	%
Alternativa C	11	78,57
Opção do procedimento de acordo com a situação.	3	21,43
Total	14	100

A tabela 39 indica que a maioria dos respondentes, 11 em 14 (78,57%), optou pela alternativa C, onde o usuário deve escolher os objetos iniciais, depois os objetos finais e somente então identificar a nova *macro*. Embora também tenham sido oferecidas mais opções, tais como as alternativas A e B,

os 3 respondentes restantes (21,43%) optaram pela categoria de dependência de acordo com a situação. Como pode-se verificar na tabela 38, o Cabri é o único programa que possui a alternativa C implementada. Algumas das justificativas apresentadas foram:

R2 - *Mais rápida* (a alternativa C, característica do Cabri).

R8 - *No caso C é mais prático. Não são precisos tantos passos para construir a macro. É mais direta a construção.*

R9 - *No Cabri, mesmo a construção de figuras é feita de forma mais direta, de forma que todo o processo é mais simples e menos trabalhoso.*

Quanto à forma de execução de *macros*, existem 2 alternativas distintas (a tabela 40 mostra quais delas são adotadas pelos respectivos programas):

- A. A *macro* não mostra os passos de execução quando é utilizada e o programa desenha automaticamente o modelo na tela.
- B. programa executa a *macro* lentamente, mostrando passo a passo a execução do algoritmo.

Tabela 40: Programas e alternativas para a execução de macros.

	A	B
Cabri Géomètre II	x	
The Geometer's Sketchpad 3	x	x

As respostas dadas à pergunta – COM QUAL DAS ALTERNATIVAS PARA SE IMPLEMENTAR A EXECUÇÃO DE MACROS VOCÊ SE SENTE MAIS À VONTADE ATUALMENTE? – registraram a opinião de cada respondente (num total de 14 indivíduos) quanto ao procedimento de preferência para a ativação de macro-construções nos programas de geometria dinâmica. Os dados obtidos podem ser visualizados na tabela 41.

Tabela 41: Resultados da pergunta 20 do questionário.

Categorias	n ^o	%
Opção do procedimento de acordo com a situação.	7	50
Alternativa B	4	28,57
Alternativa A	3	21,43
Total	14	100

Ao que parece, a tabela 41 indica que a metade dos respondentes -7 em 14 - considerou prudente escolher a categoria do procedimento em função de um determinado contexto, ou seja, achou melhor ter ambas as possibilidades para poder escolher uma delas quando fosse necessário. Conforme mostra a tabela 40, o The Geometer's Sketchpad 3 possui estas duas alternativas para execução de macros. No entanto, de acordo com os resultados da questão anterior sobre a criação de macros, ele não possui a alternativa escolhida pela maioria dos respondentes. Os outros percentuais obtidos, 28,57% para a alternativa B e 21,43% para a alternativa A, colaboram para se pensar na validade de implementação dos dois métodos em um mesmo programa. Algumas das justificativas apresentadas foram:

R8 - *O ideal seria o modo de construção prático, como na alternativa C do item anterior, sendo mostrados os passos se o usuário pedir ou não, conforme sua necessidade.*

R10 - *Não vejo necessidade de mostrar passos. O importante é ter uma nova "ferramenta" à disposição, construída pelo próprio usuário. As outras já existentes não mostram passos, afinal, e nem sentimos falta deles.*

R11 - *Para utilizar a macro, é interessante que esta surja automaticamente na tela; no entanto, é importante poder rever a execução, passo a passo.*

R14 - *Depende da situação, pois em alguns casos a descrição dos passos a serem seguidos nos faz lembrar a maneira de como construir determinada figura. O objetivo principal da utilização desse recurso não deve ser esse, mas poderia ser usado pelo aluno para consulta e verificação do seu trabalho.*

4.9 Conclusão do capítulo 4

A escolha e a aplicação de métodos para a avaliação comparativa de IHC em programas de geometria dinâmica não são tarefas triviais. Os 3 programas analisados têm várias aplicações, em diversas áreas do conhecimento e para diversos tipos de usuário. Por isso, procurou-se determinar um conjunto de recursos fundamentais, comuns a todos eles e utilizados por praticamente qualquer indivíduo (professor ou aprendiz), como um mínimo denominador aproximado para o estudo de questões de usabilidade. Foram escolhidos sete itens: ordem de comandos, mudança de atributos, *feedback*, manipulação direta, animação, lugar geométrico, *macros* e *scripts*.

A primeira das avaliações foi realizada de forma heurística, sem a participação de usuários e baseada nos princípios e conceitos da ergonomia e do *design* de IHC, enquanto a segunda foi resultante de um questionário, distribuído para 5 professores e 9 alunos. A referência cruzada das informações obtidas nestas duas etapas parece ter demonstrado que ambos os métodos se complementaram entre si, pois várias vezes descobriu-se uma clara correspondência entre as opiniões dos usuários e as hipóteses levantadas pelo pesquisador. No próximo capítulo é apresentada uma síntese destes resultados e das conclusões desta pesquisa.

5 CONCLUSÕES

5.1 Conclusões finais

Para se apresentar uma síntese dos resultados obtidos, e as respectivas conclusões deste trabalho, julga-se conveniente realizar uma breve recapitulação sobre as origens dos programas escolhidos para serem analisados. Em meados da década de 80, as primeiras versões do Cabri e do Sketchpad parecem ter surgido antes de seus criadores compreenderem como seriam as reações de seus futuros usuários e o que poderia significar, de fato, a geometria dinâmica.

Os planos iniciais de desenvolvimento do Sketchpad sequer previam o uso da manipulação direta, pois a idéia básica era a de se criar um programa que pudesse servir apenas para a construção de figuras estáticas e precisas da geometria euclidiana. Com o tempo, Nicholas Jackiw, programador do Sketchpad, acabou mudando os rumos do projeto pela fascinação que os mecanismos de interação dos jogos de computador exerciam sobre ele, optando portanto pela adoção do *mouse* como dispositivo de manipulação direta.

Jean-Marie Laborde, idealizador do Cabri Géomètre, desde o princípio havia pensado como seria interessante a característica de se puxar e arrastar objetos para se obter um número infinito de situações para cada modelo proposto. Ao contrário do que se possa imaginar, suas motivações iniciais para o uso da manipulação direta não tinham relação alguma com os objetivos educacionais ou pedagógicos que poderiam ser explorados a partir de então. Nesta ocasião, Laborde pensou em seu próprio entretenimento, entusiasmando-se com a possibilidade de se "divertir" com esta nova forma de trabalho com a geometria.

O Cinderella, o mais recente dos 3 programas estudados, resultado de uma seqüência de projetos realizados entre 1993 e 1998, surgiu a partir de uma idéia ambiciosa compartilhada por 2 pesquisadores. Henry Crapo e Jürgen Richter-Gebert sonharam com a criação de um sistema no qual um indivíduo pudesse entrar com algumas configurações geométricas, através de poucos “cliques” do *mouse*, e depois obtivesse respostas do computador sobre as propriedades embutidas nestas construções.

Independente das diferenças encontradas no Cabri, Skecthpad e Cinderella, estes programas surgiram de novas idéias que foram sendo reavaliadas e refinadas no decorrer de cada processo de desenvolvimento, no meio acadêmico e de forma experimental. Pelo que se pode verificar, nenhum deles parece ter sido elaborado através de um processo de *design* efetivamente centrado no usuário e suas implementações, mesmo nas versões mais recentes, trazem uma série de características que dificilmente podem ser julgadas e determinadas sob um mesmo ponto de vista. Como existe uma grande diversidade de usuários e de tarefas, a definição formal da “melhor interface” torna-se utópica. Por isso, no decorrer deste trabalho, os usuários foram considerados como seres humanos, possuidores de capacidades e de limitações quanto às formas de processamento e de armazenamento de informações.

No que se refere aos quesitos fundamentais da ergonomia e do *design* de interfaces, cada programa possui seus aspectos positivos e negativos. Algumas falhas de usabilidade são meramente conceituais, justamente por serem abrangentes e comuns a outros programas que não tratam especificamente da geometria dinâmica. Outras são consideravelmente mais complexas, pois se misturam com as funcionalidades características deste tipo de aplicação.

No decorrer do processo de pesquisa, para cada item escolhido, procurou-se observar as relações com as bases provenientes da ergonomia cognitiva,

tais como o uso de gestos e de modos de operação, o princípio da formação de hábitos e a conseqüente automatização de tarefas.

Os programas avaliados são voltados para o ensino e a aprendizagem, e a ocorrência de modos de operação nem sempre deve ser considerada prejudicial. Por exemplo, no capítulo anterior foram averiguadas duas formas lógicas para se construir uma reta, e ambas estão disponíveis no Cinderella. A primeira forma é definida por um único ponto e uma direção dada, e a segunda é definida por dois pontos, sendo o último criado arbitrariamente no instante de conclusão da tarefa. Se uma reta for criada pelo primeiro método e depois alocar-se um ponto sobre a mesma, numa ocasião posterior o usuário poderá se confundir, ao imaginar que ela tenha sido criada pelo segundo método. Pela falta da existência de uma pista de qual tenha sido o método de construção utilizado, o usuário é obrigado a manipular a própria reta (ou os pontos) para compreender as propriedades que estão embutidas na construção. A exploração do contexto, neste caso, parece ser um benefício para o próprio aprendizado do aluno.

A formação de hábitos facilita a execução automática e inconsciente de tarefas. Ganha-se a prática no uso da interface e acostuma-se com as formas de utilização dos recursos oferecidos. Grande parte dos usuários respondentes ao questionário pareceu demonstrar esta hipótese ao escolher as soluções de usabilidade utilizadas pelo Cabri Géomètre II como sendo as mais satisfatórias. Este é um dos vários motivos que colaboraram para se agir com cautela quando as opiniões dos usuários foram interpretadas.

Muitos dos possíveis problemas de interface são reflexos superficiais de questões mais complexas. O principal estilo de interação utilizado é a manipulação direta e os requisitos de implementação são fundamentais, pois prima-se pelo equilíbrio entre a precisão dos modelos representados e a rapidez das respostas do sistema. Uma equipe de criadores de um *software* de geometria dinâmica, quando decide a linguagem de programação e a

plataforma de desenvolvimento a serem utilizadas, abre caminhos para algumas soluções mas também é obrigada a abandonar outras alternativas, em função das dificuldades de implementação impostas pelas ferramentas de trabalho escolhidas.

Em função da revisão de literatura e do contato do pesquisador com alguns dos projetistas dos 3 programas analisados, percebeu-se que existe uma grande preocupação em se implementar sistemas de representação coerentes com os modelos matemáticos propostos. Para um certo número de situações específicas podem surgir aparentes “inconsistências matemáticas”. Seja qual for o modelo matemático escolhido, nenhum programa de geometria dinâmica estará isento de alguns destes problemas: marcações erradas de ângulos, degeneração de cônicas, interseções de retas com círculos, falhas em lugares geométricos, erros de aproximação numérica e outros mais.

Dentre os 14 usuários colaboradores da avaliação por questionário, apenas 3 indivíduos (21,43%, todos professores) afirmaram ter se deparado com algum tipo de falha desta categoria. Inclusive, 2 deles apresentaram opiniões positivas a respeito destas ocorrências. O primeiro comentou que não crê na “perfeição” de qualquer programa de geometria dinâmica, mantendo-se atento para comparar os resultados obtidos com suas expectativas; o segundo chegou a afirmar ser válido o fato de estes programas apresentarem tais falhas, pois a decisão sobre o que é correto passa a ser dos usuários, ajudando na construção dos objetos matemáticos, de forma independente do modelo utilizado.

Supõe-se que um programa de geometria dinâmica deva ser aplicado dentro de um laboratório para pequenos grupos de alunos e sob a orientação de um professor preparado e consciente, que adapte as tarefas ao conteúdo proposto. Neste caso, as descobertas de incompatibilidades dos sistemas de representação com os modelos matemáticos escolhidos não necessariamente devem ser encaradas como falhas. Se os professores souberem explorar estas

ocorrências, debatendo e discutindo com seus alunos sobre as mesmas, os supostos defeitos se transformarão em benefícios.

No escopo deste estudo, embora tenham sido pesquisados apenas 7 itens dos 3 programas avaliados, espera-se que estes resultados e conclusões sirvam como um ponto de referência para o discernimento dos usuários entre os possíveis aspectos positivos e negativos das soluções de usabilidade observadas. Quanto aos desenvolvedores, acredita-se que o relato possa servir de base para a correção e o aprimoramento de versões anteriores dos mesmos programas, assim como para a criação de novos produtos que melhor se adequem às expectativas e às tarefas dos usuários.

5.2 Sugestões para trabalhos futuros

- Realização de um levantamento sobre o uso da geometria dinâmica no Brasil e no mundo, procurando-se descobrir quais são os percentuais de utilização dos respectivos programas para cada região específica e em quais segmentos ou áreas do conhecimento eles estão sendo aplicados;
- realização de pesquisas sobre as IHC dos programas de geometria dinâmica utilizando grupos homogêneos, como por exemplo: somente estudantes do ensino médio, estudantes de matemática (ou de artes), adultos e crianças;
- estudo mais aprofundado sobre a criação de interfaces para programas de geometria dinâmica, traçando paralelos entre os fatores de usabilidade, matemática, computação e pedagogia.
- criação de programas voltados especificamente para a arte eletrônica e para o *design* gráfico baseados em um subconjunto de características específicas dos programas de geometria dinâmica.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Adler P., Winograd T. **Turning Technologies into Tools**. NY: Oxford University Press, 1992.

Bastien C., Scapin D. **Validation des critères ergonomiques pour l'évaluation d'interfaces utilisateurs**. (Rapport de Recherche No. 1427). INRIA, 1993.

Bellemain F. **Geometria Dinâmica: diferentes implementações, papel da manipulação direta e usos na aprendizagem**. In: Anais do GRAPHICA 2001, 15º Simpósio Nacional de Geometria Descritiva e Desenho Técnico – IV International Conference on Graphics Engineering for Arts and Design, São Paulo, p.1314-1329, 2001.

Blanc-Brude T., Bétrancourt M., Laborde C. **Action d'abord vs selection d'abord: Analyse comparative de deux methods syntaxiques dans deux logiciels de géométrie dynamique (Geometer's Sketchpad et Cabri Géomètre)**. In: Actes du 5eme Colloque Hypermédias et Apprentissages, Grenoble, p.59-61, 2001.

Braviano G. **Ação+Seleção ou Seleção+Ação: Qual estratégia é a mais intuitiva?** In: Anais do GRAPHICA 2001, 15º Simpósio Nacional de Geometria Descritiva e Desenho Técnico – IV International Conference on Graphics Engineering for Arts and Design – Workshop "Uso de software de Geometria Dinâmica na Área Gráfica", São Paulo, p.1253, 2001.

Burgiel H. **Review of the interactive software Cinderella**, Mathematical Association of America, 1999. [online] Disponível na Internet via WWW. URL: <http://www.math.uic.edu/~burgiel/Cinderella/review.pdf>.

de Souza, C.S. **The Semiotic Engineering of User Interfaces Languages.** *International Journal of Man-Machine Studies*, 39, p.753-773, 1993.

de Souza, C.S., Leite, J.C., Prates, R.O., Barbosa, S.D.J. **Projeto de Interfaces de Usuário: Perspectivas Cognitiva e Semiótica.** Anais da Jornada de Atualização em Informática, XIX Congresso da Sociedade Brasileira de Computação, Rio de Janeiro, julho de 1999. [online] Disponível na Internet via WWW. URL: <http://www.dimap.ufrn.br/~jair/piu/index.html>.

de Villiers M. **The Role of Proof in Investigative, Computer-based Geometry: Some Personal Reflections.** In: King, J.; Schattschneider, D. *Geometry Turned On! Dynamic Software In Learning, Teaching and Research.* Washington: Mathematical Association of America, p.15-24, 1997.

Castilho J.C., Hartson H.R, Hix D. **The User-Reported Critical Incident Method at a Glance,** 1998. [on-line] Disponível na Internet via WWW. URL: http://miso.cs.vt.edu/~usab/remote/docs/TR_user_reported_CI_method.pdf.

Cervo, A. L., Bervian, P. A. **Metodologia científica para uso dos estudantes universitários.** São Paulo: MAKRON Books do Brasil, McGraw-Hill Ltda, 1983.

Cooper, A. **About Face – The Essentials of User Interface Design.** Foster City, CA: IDG Books WorldWide, 1995.

Costa, S. F. **Método Científico - Os Caminhos da Investigação.** São Paulo: Editora HARBRA, 2001.

Foley J.D. et al. **Computer Graphics: Principles and Practice (Second Edition in C).** Reading, MA: Addison-Wesley, 1996.

Garry T. **Geometry Sketchpad in Classroom**. In: King, J.; Schattschneider, D. **Geometry Turned On! Dynamic Software In Learning, Teaching and Research**. Washington: Mathematical Association of America, p.55-62, 1997.

Gil, A. C. **Métodos e Técnicas de Pesquisa Social**. São Paulo: Atlas, 1994.

Hansen W. **User engineering principles for interactive systems**. AFIPS Conference Proceedings 39, AFIPS Press, p.523-532, 1971.

Jackiw N.R. **The Geometer's Sketchpad**. California: Key Curriculum Press, 1995.

King J., Schattschneider D. **Geometry Turned On! Dynamic Software In Learning, Teaching and Research**. Washington: Mathematical Association of America, 1997.

Kortenkamp U. **Foundations Of Dynamic Geometry**. Dissertação de Doutorado, Swiss Federal Institute of Technology Zurich, 1999. [online] Disponível na Internet via WWW:
<http://geometry.inf.ethz.ch/~kortenka/Papers/diss.pdf>.

Laborde J.M., Bellemain F. **Cabri-Geometry II**. Dallas: Texas Instruments, 1994.

Lakatos, E. M., Marconi, M. de A. **Metodologia do Trabalho Científico**. São Paulo: Atlas, 1998.

Lewis C., Rieman J. **Task-Centered User Interface Design – A Pratical Introduction**. 1994. [online] Disponível na Internet via WWW:
<http://www.acm.org/~perlman/uidesign.html>.

Pinheiro V.A. **Geometrografia 1**. Gráfica Editora Bahiense, 1974.

Maddux C.L., Johnson D.L., Willis. J. **Educational Computing – Learning with Tomorrow’s Technologies**. Second Edition. Needham Heights, MA: Allyn & Bacon, 1997.

Mandel T. **The elements of user interface design**. NY: John Wiley & Sons, Inc, 1997.

Mayhew, D. **Principles and Guidelines in Software User Interface Design**. Englewood Cliffs, NJ: Prentice-Hall, 1992.

Mullet, K., Sano, D. **Designing Visual Interfaces – Communication Oriented Techniques**. California: Sunsoft Press, 1995.

Munzner T. **Mathematical visualization: standing at the crossroads (panel)**. In: *Proceedings of the conference on Visualization '96*, p.451-453, 1996.

Nielsen, J. **Usability Engineering**. California: Academic Press, 1993.

Norman, D.A. **The Design of Everyday Things**. New York: Basic Books, 1990.

Raskin J. **The Humane Interface**. Reading, MA: Addison-Wesley / ACM Press, 2000.

Ritcher-Gebert J., Kortenkamp U.H. **The Interactive Geometry Software Cinderella**. Berlin: Springer-Verlag, 1999.

Rodrigues M.H.L., Rodrigues D.W.L. **“Transpontuais”: Uma Alternativa Dinâmica para o Estudo Interdisciplinar de Conceitos Geométricos**. In: *Revista Educação Gráfica, Bauru*, n.4, p.141-142, novembro de 2000.

Rubenstein R., Hersch H. **The Human Factor: Designing Computer Systems for People**. Newton, MA: Digital Press, 1984.

Rubin J. **Handbook of usability testing: how to plan, design and conduct effective tests**. NY: John Wiley & Sons, Inc., 1994.

Scher D. **Lifting the Curtain: Evolution of The Geometer's Sketchpad**. In: The Mathematics Educator, volume 10, Number 2, p.42-48, 2000. [online] Disponível na Internet via WWW: <http://www15.addr.com/~dscher/history.pdf>.

Schumann H., Green D. **Producing and using Loci with Dynamic Geometry Software**. In: King, J.; Schattschneider, D. Geometry Turned On! Dynamic Software In Learning, Teaching and Research. Washington: Mathematical Association of America, p.79-88, 1997.

Shneiderman B. **Designing the User Interface**. Reading, MA: Addison-Wesley, 1998.

Soloway E., Guzdial M., Hay K.E. **Learner-centered design: the challenge for HCI in the 21st century**. In: *Interactions*, p.36-48, Abril de 1994.

Winroth H. **Dynamic Projective Geometry**. Dissertação de Doutorado, Computacional Vision and Active Perception Laboratory. StockHolms Universitet: NADA, KTH, 1999. [online] Disponível na Internet via WWW: <http://www.lib.kth.se/fulltext/winroth990324.pdf>.

7 ANEXOS

7.1 Questionário p/ avaliação das IHC dos programas de G.din.

Este questionário contribui para a avaliação de três programas de geometria dinâmica. Suas informações são vitais para que se possa investigar a usabilidade dos recursos oferecidos por eles. Por favor, leia com atenção as questões abaixo antes de respondê-las. Procure ser espontâneo, representando suas expectativas e reações como usuário. A identificação do respondente neste instrumento é dispensável, mas pode ser efetuada se assim o desejar.

Este trabalho faz parte de uma dissertação de mestrado e recebe o financiamento da CAPES. Desde já, agradecemos pela participação.

Nome:

Instituição:

Departamento:

Aluno(a) Professor(a)

1. Quais são (ou foram) os programas de geometria dinâmica utilizados por você?

- Cabri Géomètre II
 The Geometer's Sketchpad 3
 Cinderella
 Outro (especificar):

2. Há quanto tempo você utiliza (ou utilizou) estes programas?

Cabri Géomètre II
The Geometer's Sketchpad 3
Cinderella
Outro (especificar)

3. Em qual área do conhecimento você tem aplicado a geometria dinâmica (por exemplo: geometria plana, geometria descritiva, cálculo...)? Especifique o nível de ensino (fundamental, médio ou universitário).

4. No uso da geometria dinâmica, você já se deparou com alguma falha ou aparente situação de inconsistência matemática em algum programa utilizado? Em caso positivo:

1. especifique o programa;
2. descreva o(s) acidente(s) ocorrido(s);
3. cite a sua reação ao se deparar com o(s) problema(s).

5. Existem duas alternativas para se implementar a construção de elementos geométricos:

A: é selecionada a ferramenta primeiro e os objetos depois.

B: são selecionados os objetos primeiro e a ferramenta depois.

Você teve alguma(s) dificuldade(s) com relação ao 'modo' utilizado, quando começou a se familiarizar com o(s) programa(s) de G. din.? Em caso positivo, descreva-a(s):

Com qual das duas alternativas você "se sente mais à vontade", atualmente?

- Ferramenta primeiro e objetos depois (exemplo: perpendicular → uma reta e um ponto).
 Objetos primeiro e ferramenta depois (exemplo: uma reta e um ponto → perpendicular).
 Depende da situação.
 Tanto faz.

Justifique se desejar:

6. Eventualmente são realizadas mudanças de atributos em objetos, tais como a cor, o tamanho de pontos, a grossura e o tracejado de linhas. Cada programa de geometria dinâmica utiliza uma abordagem e um método de acesso diferente para realizar estas operações.

Você teve alguma(s) dificuldade(s) para comandar estes tipos de modificação quando começou a se familiarizar com o(s) programa(s) de G. din.? Em caso positivo, descreva-a(s):

Ao modificar a aparência de apenas um objeto específico, você já se surpreendeu pelo fato de os novos atributos terem sido assimilados pela ferramenta que o criou (quando todos os novos objetos criados herdaram as novas propriedades)?

- Sim
 Não

Em caso afirmativo, em qual (ou quais) programa(s) utilizados já ocorreu este tipo de problema?

- Cabri Géomètre II
 The Geometer's Sketchpad 3
 Cinderella
 Outro (especificar)

Complete a tabela abaixo, marcando X nos campos. Eles devem corresponder à sua preferência quanto às abordagens para a mudança de parâmetros apresentadas por estes programas.

	Boa	Média	Fraca
Cabri	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sketchpad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cinderella	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(outro)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Justifique se desejar:

7. Nos programas de geometria dinâmica observados, existem basicamente duas alternativas para se implementar as mensagens de retorno para o usuário:

A: Existe uma barra para a transmissão de mensagens no canto inferior esquerdo da tela.

B: Ao invés da barra, as mensagens são expressas ao lado do próprio cursor.

Com qual das duas alternativas você "se sente mais à vontade"?

Alternativa A.

Alternativa B.

Justifique se desejar:

8. Há duas alternativas para se implementar a seleção e a manipulação de objetos:

A: existe uma única ferramenta que serve tanto para selecionar os objetos quanto para manipulá-los.

B: existem duas ferramentas: uma específica para selecionar os objetos e outra específica para manipulá-los.

Você teve alguma(s) dificuldade(s) para selecionar e manipular os elementos geométricos, quando começou a se familiarizar com o(s) programa(s) de G. din.? Em caso positivo, descreva-a(s):

Com qual das duas alternativas você "se sente mais à vontade", atualmente?

Alternativa A.

Alternativa B.

Justifique se desejar:

9. Existem quatro alternativas para se implementar a seleção de um determinado grupo de objetos:

A: é preciso "clicar, arrastar e soltar" o botão esquerdo do mouse para envolver os objetos com um retângulo imaginário.

B: é necessário "clicar no primeiro objeto", manter o botão SHIFT pressionado, e ir "indicando sucessivamente os próximos" até que todos sejam selecionados.

C: em outra janela existe uma lista, com o nome dos objetos desenhados na tela. É possível usar a mesma abordagem da alternativa B, porém ao invés de se "clicar diretamente em cima dos elementos gráficos", são escolhidos os nomes dos objetos pela lista textual.

D: botões com ícones, representando categorias específicas de objetos, podem ser clicados; cada botão seleciona somente os objetos pertencentes à categoria escolhida, por exemplo: "selecionar todos os pontos", "selecionar todas as retas", "selecionar todas as cônicas".

Você teve alguma(s) dificuldade(s) para selecionar grupos de elementos geométricos, quando começou a se familiarizar com o(s) programa(s) de G.din.? Em caso positivo, descreva-a(s):

Complete a tabela abaixo, marcando X nos campos que correspondem à relevância destas alternativas em um programa de geometria dinâmica que você consideraria ideal.

	Primordial	Secundária	Desnecessária
Alternativa A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alternativa B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alternativa C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alternativa D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Justifique se desejar:

10. Observe as seguintes alternativas para se construir uma reta:

A

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar reta	
Pressionar e soltar o botão esquerdo do mouse	Ponto é criado
Arrastar o mouse	Reta é puxada
Pressionar e soltar o botão esquerdo do mouse	Reta é definida

B

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar reta	
Pressionar o botão esquerdo do mouse	Ponto é criado
Arrastar o mouse	Reta é puxada
Soltar o botão esquerdo do mouse	Reta é definida

Você teve alguma(s) dificuldade(s) para construir retas, quando começou a se familiarizar com o(s) programa(s) de G.din.? Em caso positivo, descreva-a(s):

Com qual das duas alternativas você "se sente mais à vontade", atualmente?

- Alternativa A.
 Alternativa B.
 Depende da situação.
 Tanto faz.

Justifique se desejar:

Você considera desejável que a ferramenta para construir uma reta crie um segundo ponto sobre a mesma?

- Sim (figura 1).
 Não (figura 2).
 Depende da situação.
 Tanto faz.

Justifique se desejar:

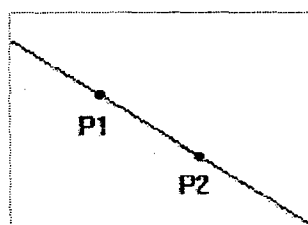


figura 1

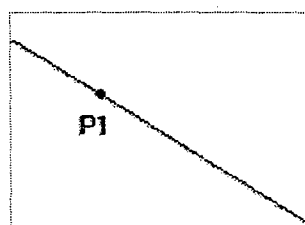


figura 2

11. Observe as seguintes alternativas para se construir um círculo:

A

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar círculo	
Pressionar e soltar o botão esquerdo do mouse	ponto é criado
Arrastar o mouse	círculo é puxado
Pressionar e soltar o botão esquerdo do mouse	círculo é definido

B

Usuário (ação)	Programa (reação)
Selecionar a ferramenta para criar círculo	
Pressionar o botão esquerdo do mouse	ponto é criado
Arrastar o mouse	círculo é puxado
soltar o botão esquerdo do mouse	círculo é definido

Você teve alguma(s) dificuldade(s) para construir círculos, quando começou a se familiarizar com o(s) programa(s) de G. din.? Em caso positivo, descreva-a(s):

Com qual das duas alternativas você "se sente mais à vontade", atualmente?

- Alternativa A.
 Alternativa B.
 Depende da situação.
 Tanto faz.

Justifique se desejar:

Você considera desejável que a ferramenta para construir um círculo crie um ponto sobre o mesmo (além do ponto central)?

- Sim (figura 3).
 Não (figura 4).
 Depende da situação.
 Tanto faz.

Justifique se desejar:

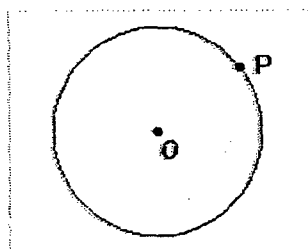


figura 3

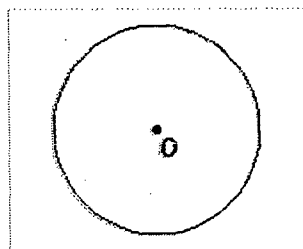


figura 4

ANIMAÇÃO

12. Considere os seguintes recursos referentes à animação:

- A:** múltipla animação (animação de mais de um objeto ao mesmo tempo).
B: associação de um botão específico para cada animação preestabelecida.
C: controle de velocidade da animação.
D: escolha do sentido sobre o trajeto (no caso de ser selecionado um ponto e um caminho).

Você teve alguma(s) dificuldade(s) para animar elementos geométricos, quando começou a se familiarizar com o(s) programa(s) de G.din.? Em caso positivo, descreva-a(s):

Complete a tabela abaixo, marcando X nos campos que correspondem à relevância destas alternativas em um programa de geometria dinâmica que você, atualmente, consideraria ideal.

	Essencial	Secundário	Desnecessário
Recurso A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Recurso B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Recurso C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Recurso D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Justifique se desejar:

13. Acompanhe a seguinte construção (figura 5):

1. Criação de um ponto P1 qualquer.
2. Criação de um círculo c2, passando por P1, de centro P2 qualquer (diferente de P1).
3. Criação de um círculo c3, passando por P1, de centro P3 qualquer (dif. de P1 e P2).

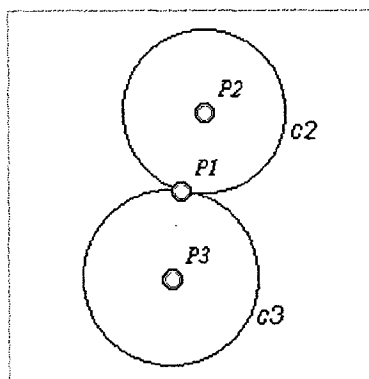


figura 5.

Neste caso, qual comportamento você consideraria coerente para ser implementado em um programa de geometria dinâmica ideal?

- Não ser possível a animação de P1.
 Ser possível a animação de P1 mas sem associar seu trajeto ao círculo c2 ou c3.
 Ser possível a animação de P1 ao longo do círculo c2 e também no círculo c3, sendo apenas estes trajetos possíveis.

Justifique se desejar:

14. Acompanhe a seguinte construção (figura 6):

1. Criação de um ponto P1 qualquer.
2. Criação de um círculo c2, não contendo P1, de centro P2 qualquer (diferente de P1).

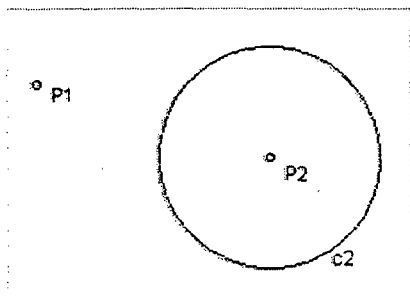


figura 6.

Neste caso, qual comportamento você consideraria coerente para ser implementado em um programa de geometria dinâmica ideal?

- Ser possível a animação de P1 ao longo do círculo c2.
- Não ser possível a animação de P1 ao longo do círculo c2.

Justifique se desejar:

15. Acompanhe a seguinte construção (figura 7):

1. Criação de um círculo c1 qualquer, de centro P1.
2. Criação de um círculo c2 qualquer, diferente de c1, de centro P2.
3. Criação de um ponto P3 qualquer, pertencente ao círculo c1.

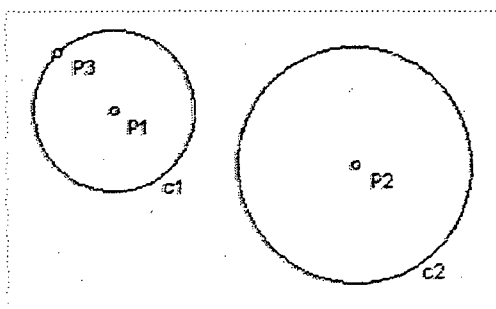


figura 7.

Neste caso, qual comportamento você consideraria coerente para ser implementado em um programa de geometria dinâmica ideal?

- Ser possível a animação de P3 somente ao longo do círculo c1.
- Ser possível a animação de P3 em torno do círculo c1 e também quando se aponta o círculo c2 como caminho.

Justifique se desejar:

LUGAR GEOMÉTRICO

16. Na sua opinião, um programa de geometria dinâmica deve permitir a colocação de pontos sobre um locus (lugar geométrico)?

- Sim.
 Não.

Justifique se desejar:

Este recurso (colocar pontos sobre um locus) foi utilizado por você em alguma construção?

- Sim.
 Não.

Você teve alguma(s) dificuldade(s) para utilizar o 'locus', quando começou a se familiarizar com o(s) programa(s) de G. din.? Em caso positivo, descreva-a(s):

17. Acompanhe a seguinte construção (figura 8):

1. Criação de um círculo c_1 qualquer, de centro P_1 .
2. Criação de um ponto P_2 qualquer, diferente de P_1 e não pertencente ao círculo c_1 .
3. Criação de um ponto P_3 qualquer, pertencente ao círculo c_1 .
4. Criação do ponto P_4 , ponto médio entre P_2 e P_3 .
5. Criação do locus de P_4 em relação a P_3 .

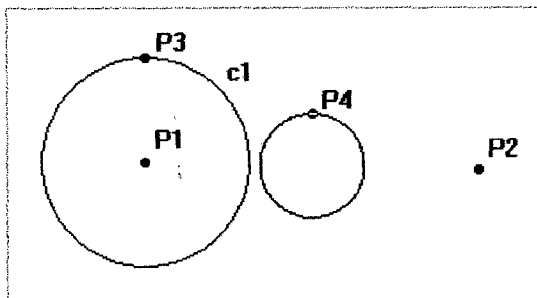


figura 8.

Neste caso, qual a ordem de parâmetros que você consideraria coerente para a criação do locus em um programa de geometria dinâmica ideal?

- Primeiro deve haver a indicação de P_4 (ponto dependente, descreve o traçado) e depois de P_3 (ponto móvel).
 Primeiro deve haver a indicação de P_3 (ponto móvel) e depois de P_4 (ponto dependente, caracteriza o traçado).
 Tanto faz a ordem de indicação (o programa deve descobrir que pontos são esses). O importante é que os dois sejam selecionados.

Justifique se desejar:

18. Acompanhe a seguinte construção (figura 9):

1. Criação de um ponto $P1$ qualquer.
2. Criação de um círculo $c2$, passando por $P1$, de centro $P2$ qualquer (diferente de $P1$).
3. Criação de um círculo $c3$, passando por $P1$, de centro $P3$ qualquer (diferente de $P1$ e $P2$).
4. Criação de um ponto $P4$ qualquer, diferente de $P1$, $P2$ e $P3$.
5. Criação do ponto $P5$, ponto médio entre $P1$ e $P4$.

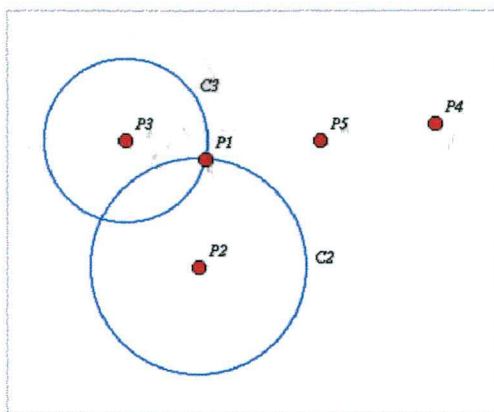


figura 9.

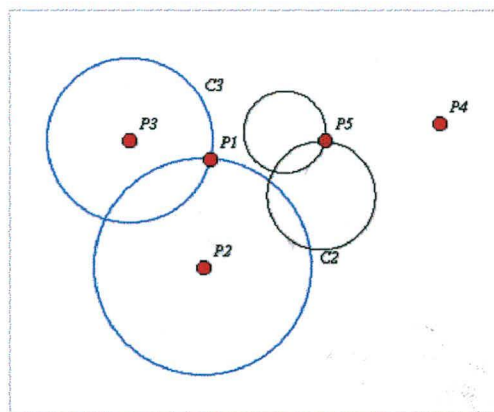


figura 10.

Neste caso, se for desejado o locus de $P5$ em relação a $P1$, qual comportamento você consideraria coerente para ser implementado em um programa de geometria dinâmica ideal?

- Não pode ser traçado o locus pedido.
- Além do ponto móvel ($P1$) e do ponto dependente ($P5$), deve ser necessário informar o caminho percorrido pelo ponto móvel, que neste exemplo pode ser $c2$ ou $c3$. Cada escolha determinará um locus diferente (figura 10).

Justifique se quiser:

19. Nos programas de geometria dinâmica observados, existem basicamente três alternativas para se implementar a criação de macros:

A: O usuário constrói o modelo, seleciona todos os seus elementos, e depois identifica a nova *macro*.

B: O usuário começa a gravação de seus comandos, constrói o modelo e depois pára de gravar (englobando todos os passos), identificando a nova *macro*.

C: O usuário deve escolher os objetos iniciais, depois os objetos finais e então identificar a nova *macro*.

Você teve alguma(s) dificuldade(s) para construir 'macros', quando começou a se familiarizar com o(s) programa(s) de G. din.? Em caso positivo, descreva-a(s):

Com qual das duas alternativas você "se sente mais à vontade", atualmente?

- Alternativa A.
- Alternativa B.
- Alternativa C.
- Depende da situação.
- Tanto faz.

Justifique se desejar:

20. Nos programas de geometria dinâmica observados, existem basicamente duas alternativas para se implementar a execução de macros:

A: A *macro* não mostra os passos de execução quando é utilizada e o programa desenha automaticamente o modelo na tela.

B: O programa executa a *macro* lentamente, mostrando passo a passo a execução do algoritmo.

Com qual das duas alternativas você "se sente mais à vontade"?

- Alternativa A.
- Alternativa B.
- Depende da situação.
- Tanto faz.

Justifique se desejar:

7.2 Dados brutos obtidos pelo instrumento de pesquisa

Resp	Programa conhecido pelo respondente				Área do conhecimento em que o aplica	Identificação de inconsistência matemática	Progr.
	CABRI	SKETCHPAD	CINDERELLA	OUTRO			
R1	1 ano e meio	6 meses	só observação	-	Geom. Plana (M/Pós)	-----	-
R2	1 ano	6 meses	só observação	-	Geom. Plana (Pós)	-----	-
R3	6 meses	6 meses	só observação	-	Geom. Plana (M/Pós)	-----	-
R4	1 ano e meio	6 meses	6 meses	-	Geom. Plana (Pós)	-----	-
R5	6 meses	6 meses	só observação	-	Geom. Plana (Pós)	-----	-
R6	9 meses	3 semanas	3 semanas	-	G. Plana, GD (U/Pós)	-----	-
R7	5 meses	3 semanas	só observação	-	Geom. Plana (U)	-----	-
R8	5 meses	3 semanas	só observação	-	Geom. Plana (U)	-----	-
R9	1 ano	6 meses	só observação	-	Geom. Plana (U)	-----	-
R10	3 anos e meio	2 anos e meio	1 ano	-	G. Plana, GD, Persp. e Sombras.; Axonometria. (U, Pós)	Marcação do ângulo. Locus e trace incompleto (hipérbole)	CABRI
R11	2 anos	3 anos	-----	Cabri I – 2 anos Tabulae – 1 mês	G. Plana, GD (Ensino Fund. e Médio)	A soma dos ângulos internos de um triâng. As falhas gráficas ficam por conta do não reconh. do infinito ²² .	GSP
R12	4 anos	-----	3 anos, porém sem usá-lo.	-	Matem., G. Plana, GD (U e com professores)	Não que eu me lembre.	-
R13	4anos	4anos	-----	Tabulae – 1 ano	G. Plana e G.Esp. (U); GD (M)	Sim, em todos eles ²³ .	Cabri Sketch Tabulae
R14	2 anos	6 meses	alguns dias	-	G. Plana e Perspec. (U)	-----	-

²² Comentário completo: P2- Com certeza isso já ocorreu, no entanto, não me recordo exatamente quais foram as falhas. Em geral, por ser da disciplina de Desenho, eu não costumo trabalhar com medidas, porém, me lembro de já ter tido problemas em relação à soma dos ângulos internos de um triângulo, por exemplo. As falhas gráficas ficam por conta do não reconhecimento do ponto no infinito (Sketchpad) e da falta de precisão do lugar geométrico. Quando me deparo com um problema deste tipo, procuro resolvê-lo por outro caminho; e devo confessar que, às vezes, até gosto. Eu não creio na "perfeição" do programa e me mantenho atenta em comparar os resultados com as minhas expectativas.

²³ São diversos: degeneração de cônicas; interseções de retas com círculos, degenerações de lugares geométricos, erros de aproximação numérica... Reação: pensei como poderia utilizá-los como recursos didáticos. Acho que é muito bom que os programas falhem e que a decisão do que é correto passe a ser do usuário, pois isso ajuda na construção dos objetos matemáticos, independentemente do modelo utilizado.

ORDEM DE PREFERÊNCIA DOS PROGRAMAS

Resp	Dificuldades iniciais?	Alternativa preferida				Justificativa
		F-O	O-F	Depende	Indif.	
R1	Sim, com a prática foram desaparecendo.	X				-----
R2	Sim, mas no Cabri o acesso é mais fácil.			X		-----
R3	Inicialmente sim, mas diante da orientação e a associação entre os programas, funcionou de forma clara e dinâmica.				X	Depois de se familiarizar com os programas, fica muito fácil de ser utilizado.
R4	Inicialmente tive pequenas dificuldades.				X	Após a familiarização torna-se mais fácil de ser utilizado.
R5	-----			X		-----
R6	Não.	X				Tenho tempo de uso maior do Cabri.
R7	Não.	X				-----
R8	Não, porém foi necessário um tempo para adaptação ao uso do segundo programa utilizado (Sketchpad)	X				-----
R9	Como utilizei inicialmente o Cabri, senti alguma dificuldade para selecionar primeiro os objetos de depois a ferramenta.				X	Essa foi uma dificuldade inicial. Com o uso do Sketchpad o outro modo ficou familiar.
R10	A primeira foi descobrir os menus do Cabri. Pequena dificuldade ao passar de um software para outro.	X				Sempre sei que ferramenta usar, por isso posso tê-las todas disponíveis sem precisar selecionar os objetos.
R11	-----	X				Acredito que seja simplesmente uma questão de hábito; o programa que eu utilizo com mais frequência funciona assim.
R12	Não.	X				Quando aprendi a usar o Cabri, achei 'lógico' selecionar primeiro a ferramenta e depois o que quero fazer com ela. Como me habituei com esse modo, estranho um pouco ter que selecionar os objetos primeiro.
R13	Não; acho as duas opções interessantes.				X	-----
R14	Não.	X				Este é o modo utilizado pelo Cabri, com que estou mais acostumada.

MUDANÇA DE ATRIBUTOS INICIAIS

Resp	Dificuldades iniciais?	Ao modificar				Pref. e classificação				Justificativa
		CABRI	GSP	CIND	Outro	CABRI	GSP	CIND	TAB	
R1	-----		X	X		B	M	M		-----
R2	Não.					B	M	M		-----
R3	Não.		X	X		B	B	M		-----
R4	Não, porque alguns software são semelhantes.		X	X		B	M	M		Ao iniciar o curso de geometria dinâmica tive mais oportunidade de trabalhar com o Cabri e acabei me adaptando mais a ele.
R5	Não.		-	-		B	B	B		-----
R6	Não.		X			B	M	F		-----
R7	Sim, apenas no período de adaptação.		X			B	B	M		-----
R8	Não.		X	X		B	B	M		-----
R9	Sim, a visualização dos pontos é prejudicada por não podermos modificar o tamanho dos pontos (GSP)		X	X		B	M	M		Considero que a aparência dos objetos deva ser modificada apenas pelo usuário quando desejar modificar a imagem de objetos específicos.
R10	Não, apesar de estranhar a mudança de um para outro programa.		X	X		B	M	M		Prefiro o Cabri por ter um acesso mais rápido à mudança de atributos, na coluna à esquerda.
R11	-----		X			M	B	-	B	O Cabri não permite a seleção de mais de um objeto.
R12	-----	x24				B	B	B		-----
R13	Não, é apenas uma questão de se acostumar ao procedimento.		X			B	B	-	B	-----
R14	Não.		X	X		B	B	M		-----

24 Quando quis dar o nome de ângulo " α " e todo o texto ficou escrito com letra Symbol.

QUESTIONÁRIO

Resp	Preferência		Justificativa
	Barra	Cursor	
R1		X	-----
R2		X	Sempre esqueço de olhar para o canto da tela.
R3		X	No momento em que se clica a ferramenta desejada, automaticamente vocês estará visualizando a informação necessária. A visualização em resposta é imediata.
R4		X	Visualização imediata do que se faz.
R5	X		Familiar ao autocad.
R6		X	Mais visível.
R7		X	As mensagens ao lado do próprio cursor facilitam a visualização do usuário.
R8		X	A mensagem expressa ao lado do cursor é captada automaticamente conforme a manipulação, o que facilita na visualização do usuário.
R9		X	-----
R10		X	Esse modo de dialogar com o usuário me parece ser bem mais eficiente; não há como deixar de ver.
R11		X	A mensagem ao lado do cursor permite a leitura imediata, sem que seja necessário desviar a atenção do desenho.
R12		X	Às vezes a gente se esquece de olhar a barra. Quando aparece ao lado do cursor, é mais imediato.
R13	X	X	Como na pergunta anterior, é apenas uma questão de se acostumar ao processo. Uma vez aprendido me é indiferente.
R14		X	-----

MANIPULAÇÃO DIFERENÇA DE TAREFAS: Questão 9 de um grupo de 20 questões

Resp	Seleção e manipulação de um objeto			Seleção e manipulação de um grupo de objetos					
	Dificuldade?	Pref.	Justificativa	Dificuldade?	A	B	C	D	Justificativa
R1	Não.	B	-----	Não	P	P	S	D	-----
R2	Sim, o contato com o primeiro progr. vicia e por ser + fácil você esquece para o segundo.	A	-----	Sim, com a janela você pode envolver algo não necessário...	P	S	S	D	-----
R3	Não.	B	-----	Não.	P	P	P	P	-----
R4	Não; acredito haver uma grande similaridade entre os software.	A	É mais simples.	Não.	P	S	P	P	-----
R5	Não.	A	Existe uma simplificação do trabalho a ser realizado.	Não.	P	S	S	S	-----
R6	Não.	A	Maior velocidade na constr.	Não.	P	P	D	S	-----
R7	Não.	A	Uma única ferramenta facilita ao usuário.	Não.	P	P	P	P	Todas são necessárias. Dependendo do tipo de tarefa, uma é mais do que a outra.
R8	-----	A	Fica mais prático para o usuário.	Sim, no Cinderella o item C é direcionado à matem. difícil ao leigo.	P	P	S	P	-----
R9	Apenas observei a alternativa B no Cinderella.	A	Mesmo ainda não usando o Cinderella, acho mais fácil o uso de apenas uma ferramenta.	----	P	P	S	S	Acho mais prático resolver essas seleções diretamente no desenho.
R10	Sim, até hoje tenho difícil com o Cinderella; me esqueço.	A	É mais comum em programas gráficos e a gente se acostuma.	Não.	P	P	S	D	-----
R11	-----	A	-----	-----	P	P	P	P	Considero todas primordiais, pois as necessid. São diversas em função do tipo de desenho que estiver sendo executado.
R12	No Cabri, a aprend. foi natural, mas tive um pouco de dificuldade para sel. e mov. principal/ no Cind. No GSP eu demorei p/ descobrir como girar uma reta.	A	Não sei se por ter aprendido primeiro assim ou se é pela simplicidade realmente. Quanto menos 'operações' a memorizar, tanto melhor.	Não, não tive.	P	P	S	S	-----
R13	Apenas a usual, de man. c/ o mouse em novas sit. Em alguns mom., acho q. os pr. de G.din.são de fácil utiliz.	A	-----	Não, foi apenas uma questão de aprender a regra do programa.	P	P	S	S	-----
R14	-----	A	-----	-----	P	S	S	D	-----

MANEJO DO PUNTO (CONSTRUÇÃO DE UM PONTO)

Resp	Alternativas para construção de uma reta					Criação de um segundo ponto?					
	Dificuldades?	A	B	Dep	Indit	Justificativa	Sim	Não	Dep	Indit	Justificativa
R1	Não.			X		Depende da metodologia que vamos empregar.			X		O segundo ponto amarra a reta.
R2	Sim, porém acho que a intimidade com o software.				X	É incômodo lembrar de segurar o botão e saber qual o ponto que foi feito.		X			Inconveniente de se necessitar um ponto na reta, ter que identificá-lo.
R3	Sim. Inicialmente você irá fazer comp. Entre progr. De acordo com o que se deseja trabalhar a constr. do obj.			X		Depende do objeto que se vá construir.			X		Se este ponto for necessário, sim.
R4	Não.			X		Será de acordo com a necessidade da tarefa. Se for necessário ou não passar por algum ponto específico.			X		Depende do que se pretende fazer. Que tipo de construção se queira.
R5	Não, as oper. são simples basta atenção nas barras de fer. e ainda dep. do partido q. se deseja tomar.			X		Tudo dependerá da metodologia que se pretende usar no trabalho.			X		Depende do objetivo que se pretende alcançar.
R6	Não.	X				Talvez, a maior intimidade com o Cabri.		X			-----
R7	Não.	X				Trabalhamos mais com o Cabri.		X			Existindo a necessidade de 2 pontos, podemos construir um segmento ao invés de uma reta.
R8	Não.				X	As duas são muito parecidas. Não faz diferença usar uma ou outra.		X			Acho desnecessário.
R9	Não.				X	-----		X			Porque se marcarmos um ponto na reta, para não nos confundirmos, temos que mudá-lo de cor.
R10	Não.		X			É mais rápida e intuitiva.		X			Me obrigaria a escondê-lo depois.
R11	-----	X				É mais confortável decidir a direção da reta sem estar pressionando o botão.		X			A criação de um 2º ponto prende a mudança de direção da reta; acho interessante a opção de criar uma reta por um ponto e uma direção, ou por 2 pontos.
R12	Não.				X	Criar retas é uma coisa tão fácil que a gente pratica/ não precisa pensar sobre... faz-se automaticamente.		X			Não há necessidade do segundo ponto.
R13	Não.				X	-----				X	-----
R14	Não.		X			-----		X			-----

MANIPULAÇÃO DIRETA GEOMETRIA DO CÍRCULO (PUNTO)

Resp	Alternativas para construção de uma reta					Criação de um segundo ponto?					
	Dificuldades?	A	B	Dep	Indif	Justificativa	Sim	Não	Dep	Indif	Justificativa
R1	Não.			X		-----			X		Considero que o segundo ponto é necessário quando desejamos definir o raio do círculo.
R2	Sim, porém acho que a intimidade com o software.				X	-----		X			Inconveniente de se necessitar um ponto no círculo, ter que identificá-lo
R3	Não.			X		-----			X		Depende do que se vai construir.
R4	Não.				X	-----			X		Depende do que queira-se construir.
R5	Não, esta questão recai na anterior; tudo depende do que se deseja construir.			X		-----			X		Depende do objetivo que se pretende alcançar.
R6	Não.	X				-----		X			-----
R7	Não.	X				-----		X			-----
R8	Não.				X	As duas formas são muito parecidas e simples de se usar. Tanto faz.			X		-----
R9	Não.				X	-----		X			Não, porque se marcarmos um ponto no círculo, para não nos confundirmos, temos que mudá-lo de cor.
R10	Não.		X			É mais rápida e intuitiva.		X			Me obrigaria a escondê-lo depois.
R11	-----	X				É mais confortável decidir o raio do círculo sem estar pressionando o botão.		X			Da mesma forma que acontece c/ a reta, é interessante a liberdade de modificar o raio por qualquer ponto da circunferência.
R12	Um pouco quando comecei a usar o GSP e o Cinderella pois precisam de mais opções que pratica/ são feitas com um só comando no Cabri.	X				Talvez pelo fato de ter aprendido primeiro, mas as outras não são complicadas.			X		Quando a gente quer que o círculo fique preso em um ponto, esse segundo ponto é interessante Se não houver essa necessidade, não precisa ter o ponto.
R13	-----				X	-----				X	-----
R14	-----		X			-----		X			-----

ANÁLISE DE DIFICULDADES INICIAIS DE ANIMAÇÃO

Resp	Dificuldades iniciais?	Relevância das alternativas				Justificativa
		A	B	C	D	
R1	Sim, a falta de uma orientação inicial criou dificuldades.	E	S	E	E	-----
R2	Não, fácil assimilação após a orientação.	E	D	E	E	Visualizações diversas.
R3	Adaptação inicial sim. Logo se tem uma dinâmica maior no uso de ferramenta devido a uma boa orientação recebida.	E	S	E	E	-----
R4	Não, porque tive orientação de como fazê-las.	E	S	E	E	-----
R5	Não. Por já haver um conhecimento de outros programas; vão existir correlações que ajudam no entendimento do processo de animação.	S	E	E	E	-----
R6	Sim, as seqüências de ações são bem distintas.	E	S	E	E	O recurso B é bastante útil desde que adequado às demais opções (sentido etc.)
R7	Não.	E	E	E	E	O domínio do Cabri, do GSP e do Cinderella. É necessário pois, de acordo com cada tipo de problema, um se torna mais interessante em seu uso do que o outro.
R8	Não.	E	E	E	E	Todos os recursos são necessários conforme a utilização.
R9	Não.	E	S	E	E	-----
R10	Sim, até descobrir os vários recursos.	E	S	E	E	Seria ideal que todos os programas oferecessem as várias possibilidades.
R11	-----	E	S	E	E	O recurso B é prático pois permite que se possa salvar uma animação (às vezes temos vários elementos p/ animar) mas não chega a ser essencial.
R12	Sim, muita dificuldade. Eu não sei fazer animação no Cinderella e no Sket. Não sei se a versão demo do Sket não permite fazer animação ou se eu é que não consegui descobrir. Estou descobrindo tudo por analogia ao que sei do Cabri.	E	D	S	S	-----
R13	-----	S	S	S	S	Não considero a animação como uma das características importantes do software de geometria dinâmica. Acho que ela induz o professor a realizar o trabalho do aluno que pode ficar reduzido ao papel de espectador de uma animação, o que não é um uso relevante dos software de geometria dinâmica.
R14	-----	E	D	S	E	-----

ANÁLISE DE CONTRIBUIÇÃO DE DIFERENÇAS ENTRE OS SIMULADORES CINDERELLA E CABRI

Resp	Animação de P1			Justificativa
	Não	Sim/c1,c2	Sim/Não.c1,c2	
R1			X	-----
R2			X	Mostra um delineamento espacial de fácil compreensão e lógica.
R3			X	O terceiro item lhe dá possibilidade de animação sem nenhuma dificuldade.
R4			X	Foi uma forma bem elaborada do software.
R5			X	Fica mais clara a forma como ocorre a animação. Outro ponto importante: são informações que o programa Cinderella traz na barra de ajuda.
R6			X	O fato de termos mais uma possibilidade de animação.
R7			X	-----
R8			X	-----
R9			X	Observando também a animação de um ponto comum a duas retas, podemos visualizar a sua trajetória na esfera escolhendo um ou outro caminho, percebendo o que acontece no espaço.
R10			X	Parece-me ser mais útil na resolução de problemas.
R11		X		O ponto é livre e deve se movimentar livremente; as circunferências é que devem acompanhar o movimento do ponto. Penso que, se o programa tem por objetivo ensinar a geometria é importante manter as propriedades geométricas dos elementos; entretanto, se usado para outras finalidades, esses recursos podem facilitar a execução de um desenho. Ainda assim, dou preferência ao recurso de "redefinição do objeto", disponível no Cabri.
R12		X		O prim. Item é possível, pois se P1 é um ponto qualquer, pode ser movimentado para onde se queira, como no Cabri (se bem que aí os 2 círculos variam). Não sei animar no GSP mas pude mover o ponto arbitrariamente. Idem no Cinderella. O terceiro item é possível no Cinderella, apesar de eu achar meio estranho. No Cabri isso não é possível pois é o círculo que está amarrado no ponto e não o inverso.
R13	X			P1 é um ponto livre.
R14			X	-----

ANIMAÇÃO DE P1 AO LONGO DO CÍRCULO E P3 AO LONGO DOS CÍRCULOS

Resp	Animação de P1 ao longo do círculo			Animação de P3 ao longo dos círculos		
	Sim	Não	Justificativa	Sim/c1	Sim/c1 c2	Justificativa
R1	x		-----		x	-----
R2	x		Mais coerente à dinâmica do exercício.		x	Interesse didático (a aluna viu a possibilidade de trabalhar com o centro do primeiro círculo como se ele fosse um foco de luz)
R3	x		Desejar que o ponto seja animado para que tenha uma função necessária.		x	Sim, levando para física. Foco de luz.
R4	x		Não tem função nenhuma um ponto percorrer por um espaço qualquer (CABRI)		x	Sim, em desenho – projeção cônica.
R5	x		É importante que ele possa ser animado para que existam possibilidades de diferentes estudos na figura.		x	Há interesse nesta fase pois os recursos têm forte apelo didático.
R6	x		-----		x	Obs. do aplicador: o participante explorou o caso em diferentes situações, fazendo descobertas. *
R7		x	-----		x	-----
R8	x		Seria melhor p/ o usuário que o ponto fosse animado no círculo, mesmo não estando contido neste, porém mostrando o caminho percorrido pelo ponto.		x	Desde que se mostre o caminho a ser percorrido pelo ponto.
R9		x	A não ser que o comportamento do ponto P1 ao longo do círculo c2 seja justificado graficamente/matematicamente. N1 ao sei nada sobre isso.	x		-----
R10		x	Seria um recurso a mais que possibilitaria redefinir o objeto, porém acho-o perfeitamente dispensável e pode confundir o usuário.	x		Apesar de provavelmente ter alguma utilidade, acho que confunde o usuário, pois o ponto não fica pertencendo ao segundo círculo.
R11		x	O ponto P1 não tem nenhuma relação geométrica com o círculo; sua movimentação deve ser independente.	x		Está justificado nas questões anteriores.
R12		x	Não tem por que isso acontecer. Se algum dos programas fizer isso, gostaria de saber SE e POR QUE faz isso.	x		Não consigo ver uma lógica para a segunda opção, nos moldes em que foi posta a questão. Talvez até seja interessante que aconteça essa dupla rotação, como por exemplo para simular a rotação da lua em torno da Terra e esta em torno do Sol, mas da maneira como foi colocada a questão, acho incoerente.
R13		x	P1 é um ponto livre, não há caminho para ser percorrido, logo não faz sentido falar em sua animação.	x		-----
R14		x	-----	x		-----

* Comentário do aplicador: Um aluno de especialização (R6) descobriu uma utilidade para o caso de animação descrito, o que acabou influenciando os demais na questão 15.

NUCLEO DE GESTÃO DE RECURSOS EDUCACIONAIS

Resp	Dificuldades iniciais?	Ponto?		Justificativa	Usou ponto?	
		Sim	Não		Sim	Não
R1	Não.	X		-----	X	
R2	Não.		X	-----	X	
R3	Não.	X		-----	X	
R4	-----	X		-----	X	
R5	-----		X	-----	X	
R6	-----	X		-----		X
R7	As dificuldades foram mínimas, a orientação do professor foi muito boa, eliminando as dificuldades.	X			X	
R8	Não.	X		-----	X	
R9	Não, porque fui orientada.	X		-----	X	
R10	Custei um pouco a perceber a relação entre o ponto que se movia e o dependente.	X		O ponto-chave de um problema pode estar ali. Além disso, neste caso, todo recurso a mais é desejável.	X	
R11	-----	X		-----	X	
R12	Sim, tive muita dificuldade pois ela não era tão evidente para mim no início. Tive que ter umas boas aulas para entender o espírito da coisa. Agora sou capaz de fazer no Cabri, mas ainda não descobri no Cinderella e no Sketchpad (este por ser demo e não permitir um monte de opções). Aceito ajuda.	X		Essa ferramenta é muito útil e muito interessante. Já li algo que diz que o Cabri se justifica só pela possibilidade de traçar os lugares geométricos. É uma ferramenta muito rica.	X	
R13	Não.	X		É uma ferramenta útil.	X	
R14	-----		X	-----		X

Figura 10 - Resultados da seleção antes da realização da construção do locus

Resp	Ordem na seleção dos pontos para o traçado do locus				Possibilidade de traçado do locus na situação dada		
	D/M	M/D	Indif.	Justificativa	Não	Sim	Justificativa
R1			X	Raciocinar logicamente é o melhor.*		X	Possível somente no Cinderella.
R2			X	O fato importante é que o locus é dado, não importa o processo.	X		O ponto depende dos dois círculos e sendo assim o caminho relativo a apenas um sai do pedido.
R3		X		É necessário seguir um raciocínio diante da construção.		X	No Cabri e no Sketchpad não temos a possibilidade de construção do locus pedido. Já o Cinderella nos dá a possibilidade independente do ponto estar no círculo c3 ou no c2.
R4		X		Prefiro seguir uma lógica de raciocínio.		X	-----
R5	X			Temos familiaridade com o programa, não há problemas com a marcação dos pontos, mas por uma questão metodológica, acredito que a definição dos parâmetros construtivos são fundamentais.	X		-----
R6			X	-----		X	-----
R7			X	-----		X	Neste caso, o Cinderella tem mais recurso.
R8			X	Por ser mais prático.		X	Porque é mais completo, nos dá mais recurso.
R9			X	-----		X	Porque em função disso, terei locus diferentes.
R10	X			Acho mais coerente, já que se pede o locus do ponto dependente.		X	Proporciona uma visão mais ampla das relações entre o ponto móvel (pertencente aos dois LLGG) e o ponto dependente.
R11			X	Se tiver que ter uma ordem, que seja a primeira opção; no entanto, a melhor opção é a de selecionar "sem ordem"; o locus é uma consequência única da relação entre esses elementos.	X		Mantenho a opinião de que, se desejo que o pt P1 esteja sobre c2, devo construí-lo sobre ele mesmo. Lembro, porém, que depende do objeto para o qual o programa esteja sendo utilizado.
R12	X			O próprio enunciado já diz: o P4 depende do P3, então é o que vai acontecer com ele quando P3 se desloca sobre o objeto sobre o qual foi criado.	X	X	Tentei no Cabri e não consegui. No Cinderella consegui. Esses 2 processos partem de premissas bem diferentes. Acho interessante que essas possibilidades diferentes existam.
R13			X	-----	X		P1 e P5 são pontos livres.
R14			X	-----	X		-----

* Comentário do aplicador: é bem provável que o aluno R1 tenha copiado (embora disfarçadamente) a justificativa dada pelo aluno R4.

MACROS EM GEOMETRIA (cabri e sketchpad)

Resp	Dificuldades iniciais?	Preferência de alternativa					Justificativa
		A	B	C	Dep.	Indif	
R1	Não.			X			-----
R2	Não.			X			Mais rápido.
R3	Somente a princípio. Diante da orientação contínua você estará familiarizado mediante os comandos estabelecidos e exercitados.				X		Depende do grau de conhecimento do programa.
R4	Um pouco de dificuldade no Sketchpad, pois já tinha me familiarizado com o Cabri...*			X			... trabalhar com o Cabri é mais simples.*
R5	Não.				X		Todo o trabalho dependerá do grau de conhecimento e familiaridade dos recursos que o programa possui.
R6	-----			X			-----
R7	Sim, as dificuldades foram solucionadas depois de uma boa orientação.			X			-----
R8	Sim. Foi preciso orientação do professor.			X			No caso C é mais prático. Não são precisos tantos passos para construir a macro. É mais direta a construção.
R9	Sim, no Sketchpad, foi necessária uma orientação pois as ferramentas / comandos são mais complexos e sem uma ajuda dificilmente chegaria a construir e definir uma macro.			X			No Cabri, mesmo a construção de figuras é feita de forma mais direta, de forma que todo o processo é mais simples e menos trabalhoso.
R10	Precisei que alguém me informasse como fazer um script no Geometer's Sketchpad.			X			É mais funcional pois opera rápido na figura desejada, além de me deixar escrever um help e criar um ícone próprio.
R11	-----			X			Exclusivamente por questão de hábito.
R12	Sim, tive, mas depois tomei jeito, ou seja, aprendi. Só sei fazer no Cabri. Gostaria de aprender nos outros. Você poderia me dar as dicas?			X			É a única que conheço.
R13	Não.				X		-----
R14	-----			X			-----

* Comentário do aplicador: O respondente AEP4, escreveu num só período as respostas que daria a dois itens da pergunta: se teve dificuldades iniciais e qual a justificativa para sua escolha de alternativa para MACROS.

QUESTIONÁRIO DE AVALIAÇÃO DE RECURSOS

Resp	Preferência de alternativa				Justificativa
	A	B	Dep.	Indif.	
R1			X		-----
R2		X			Visualização do caminho percorrido permitindo melhor compreensão.
R3		X			Proporciona melhor informação para o caminho desejado na construção do objeto.
R4		X			Apesar de ser inicialmente mais trabalhoso, pode-se verificar passo a passo.
R5			X		Idem da questão 19. Todo o trabalho dependerá do grau de conhecimento e familiaridade dos recursos que o programa possui.
R6		X			Seria interessante a interrupção da execução em qualquer "passo".
R7	X				-----
R8			X		O ideal seria o modo de construção prático, como na alternativa C do item anterior, sendo mostrados os passos se o usuário pedir ou não, conforme sua necessidade.
R9			X		
R10	X				Não vejo necessidade de mostrar passos. O importante é ter uma nova "ferramenta" à disposição, construída pelo próprio usuário. As outras já existentes não mostram passos, afinal, e nem sentimos falta deles.
R11			X		Para utilizar a macro, é interessante que esta surja automaticamente na tela; no entanto, é importante poder rever a execução, passo a passo.
R12	X				Por ser a única que conheço. Acho a outra muito interessante, pois já vi alguém demonstrando. Gostaria de conhecer melhor ou saber fazer.
R13			X		-----
R14			X		Depende da situação, pois em alguns casos a descrição dos passos a serem seguidos nos faz lembrar a maneira de como construir determinada figura. O objetivo principal da utilização desse recurso não deve ser esse, mas poderia ser usado pelo aluno para consulta e verificação do seu trabalho.