Orientação Semântica *Versus* Orientação Sintática na Captura de Dados para Aplicações pelo Método dos Elementos Finitos

Isa Haro Martins
isa@inf.puc-rio.br

Clarisse Sieckenius de Souza
clarisse@inf.puc-rio.br

Marcelo Gattass
gattass@icad.puc-rio.br

PUC-Rio.Inf.MCC31/94

Resumo: Os Sistemas de Engenharia Baseados em Subdivisão Planar, mais precisamente os Pré-Processadores, constituem-se em poderosas ferramentas de captação de dados para análise e simulação pelo Método de Elementos Finitos. Parte da capacidade destes aplicativos está nas interfaces gráficas por manipulação direta, que permitem a entrada de dados visual, através da representação geométrica do objeto de engenharia. No entanto, as soluções atuais não se mostram adequadas no tocante ao modelo conceitual da interação usuário-sistema. Nestas interfaces identifica-se uma dicotomia entre os processos de manipulação da representação visual e captura de dados. Enquanto o primeiro pede flexibilidade para manipulação do traçado gráfico, o segundo pede rigidez na manipulação dos objetos semânticos no sentido de manter a consistência dos dados. A orientação sintática ou semantica do modelo conceitual de interação, permite um melhor atendimento a cada um destes aspectos, porém gerando conflitos com relação ao outro. Faz-se necessária a busca de um modelo de interação alternativo, capaz de minimizar os problemas encontrados nos modelos puros, permitindo uma linguagem de interface com alto poder de expressão. Este trabalho propõe um modelo alternativo, apresentando exemplos onde são contrastadas algumas soluções atuais com soluções propostas.

Palavras-chaves: interfaces gráficas, modelos conceituais em interfaces, entrada de dados visual, elementos finitos, pré-processadores.

Abstract: Engineering Systems Based on Planar Subdivision, more specifically Finite Element Pre-Processors, are powerful tools for capturing data for analysis and simulation systems. In this class of software applications direct manipulation interfaces play a very important role; they allow for visual data entry based on a geometric representation of the artifact. But conceptual models underlying current interface solutions are not adequate. We have identified some incongruities between the manipulation process of visual representations and the data capture process. The former looks for flexibility in the manipulation of the graphic drawings, whereas the latter looks for rigidity in the manipulation of semantic objects to maintain data consistency. Syntactic-driven and semantic-driven approaches favor one process at the expense of the other. Therefore, an alternative model is called for. This paper presents a proposal of one such alternative approach, including a number of contrastive examples between existing solutions and the proposed ones.

Keywords: graphical interfaces, conceptual models in interfaces, visual data entry, finite elements, pre-processors.

Sumário

1. Introdução 3

2. A Família de Sistemas Pré-Processadores 5

2.1. Funções de um Pré-Processador 6

2.2. A Geração da Representação Visual e a Captação de Dados 8

3. Os Modelos Sintático e Semântico nos Sistemas Pré-Processadores 10

3.1. Modelos Conceituais Básicos 11

3.2. A Funcionalidade dos Modelos Sintático e Semântico 12

3.3. Opção pela Representação ou pela Construção 13

4. Uma Abordagem Teórica à Entrada de Dados Visual 14

4.1. O Modelo Conceitual do Sistema e o Modelo Conceitual da Interação 15

4.2. Tradução do Modelo Conceitual do Sistema para o Usuário 16

5. Estudo de Caso 19

5.1. Um Estudo sobre as Operações de Definir e Desfazer Região 19

5.2. Um Estudo sobre o Problema do Furo 26

6. Proposta para um Modelo Sintático Estendido 27

6.1. Critérios de Extensão do Modelo Sintático 27

6.2. Especificação dos Modelos de Interação 28

6.3. Análise sobre a Gramática Especificada 31

6.4. Desenvolvimento das Heurísticas 33

7. Conclusões 41

Bibliografia 42

1. Introdução

As interfaces gráficas para Sistemas de Engenharia Baseados em Subdivisão Planar, mais especificamente os pré-processadores, têm sido tratadas sem uma metodologia particular de abordagem da questão. As pesquisas nesta classe de aplicativos têm privilegiado a ótica de desenvolvimento de programas, priorizando quase sempre aspectos ligados à estrutura de dados, ou algoritmos a ela associados. O resultado é o aumento da complexidade dos programas e da manipulação gráfica oferecida ao usuário, sem o correspondente acréscimo de qualidade na modelagem da interação usuário-sistema.

De modo geral, estas interfaces evoluíram de uma entrada de dados por *batch* para uma entrada de dados visual, por manipulação direta, alterando o paradigma de comunicação entre usuário e sistema. A natureza da manipulação direta traz problemas característicos, os quais não têm sido adequadamente tratados. A nível da interface soluções *ad hoc* têm sido adotadas para problemas comuns a toda esta classe de aplicativos, gerando frequentemente incoerências no comportamento dos sistemas, as quais impactam diretamente a interação com o usuário.

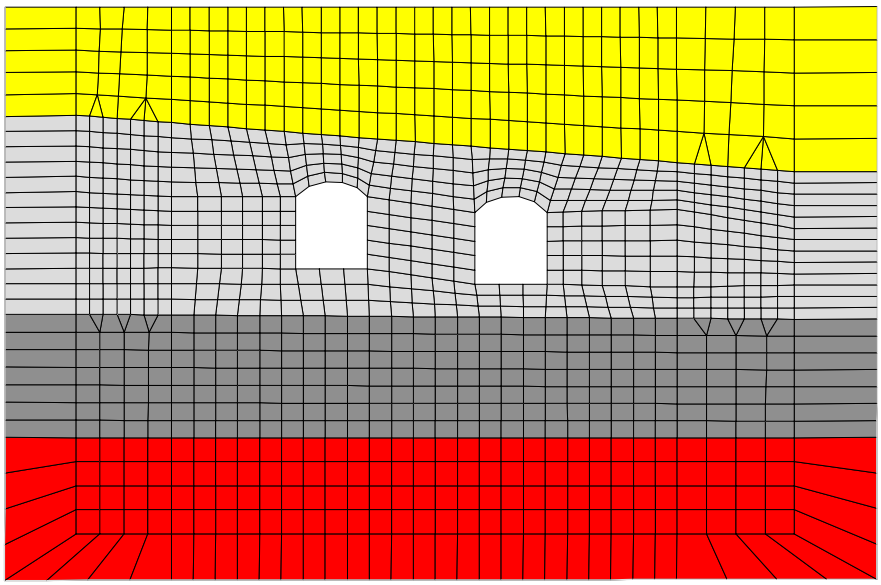


Figura 1. Exemplo de modelagem geométrica e traçado de malha

Os pré-processadores são sistemas captadores de dados para análise e simulação pelo método de elementos finitos. Enfocamos nossa pesquisa na classe de pré-processadores que faz a captação de dados através da modelagem geométrica do objeto de engenharia. Nestes sistemas é feita a

definição da geometria do objeto de engenharia, a definição de suas propriedades físicas e o traçado da malha de elementos finitos. A Figura 1 mostra um exemplo de modelagem geométrica e traçado da malha.

Um estudo sobre alguns modelos de interação com o sistema identificou uma dicotomia entre a representação visual do objeto de engenharia e sua interpretação. Sendo o pré-processador um captador de dados, a representação visual deve ser traduzível para o modelo de elementos finitos. No entanto, o processo de geração da representação visual está sujeito a regras de comportamento de cunho mais sintático, ou seja, que governam a geração de estruturas combinatórias de elementos visuais; já o processo de captação de dados está sujeito a regras de comportamento de cunho mais semântico, ou seja, que governam a sua interpretação segundo um modelo específico.

Cada um desses processos tem exigências próprias. Supondo-se situações extremas, a geração da representação visual e a captação de dados podem ser feitas em paralelo ou em momentos distintos. Executá-las em momentos distintos significa gerar a representação visual com uma orientação exclusivamente sintática, e em uma etapa posterior validar essa representação para a captação de dados. Neste caso o usuário ganha em flexibilidade na manipulação da representação visual, mas pode perder em consistência, se algumas estruturas visuais geram interpretações ambíguas ou errôneas. Executá-las em paralelo significa gerar uma representação visual válida para o modelo de elementos finitos, podendo o usuário perder em flexibilidade, já que toda manipulação visual é avaliada concomitantemente, a cada passo, face a um modelo de construção e modificação de um objeto de engenharia fisicamente determinado. Isto é, todo traçado equivale a um estado físico válido do objeto.

Na maioria dos casos o sistema terá dificuldades em recuperar a consistência de uma representação visual gerada em um modelo exclusivamente sintático. De forma semelhante, o usuário terá dificuldades em manipular a representação visual através de um modelo exclusivamente semântico. Neste caso, aquilo que o usuário poderia lucrar tendo uma entrada de dados visual estaria potencialmente perdido pela rigidez dos princípios de manipulação. O equilíbrio entre as duas orientações no modelo de interação usuário-sistema é fundamental para uma linguagem de comunicação com alto poder de expressão.

Este trabalho discute os problemas inerentes a cada uma dessas abordagens na interação usuário-sistema, e propõe uma abordagem alternativa baseada em heurísticas geradas a partir da modelagem de tarefas do usuário, capaz de contornar situações insolúveis em cada um dos modelos puros, evitando soluções *ad hoc.* Com isso pode-se oferecer ao usuário uma coerência conceitual no comportamento da interface.

Nos próximos itens será detalhada uma descrição de aspectos dos pré-processadores, com influência no modelo de interação usuário-sistema. Serão discutidas também a dicotomia entre modelo visual e modelo de interpretação nesta família de sistemas, e os problemas dela decorrentes. Será então apresentado um estudo de caso exemplificando as questões levantadas. Por último, será oferecida a proposta de um modelo conceitual de interação usuário-sistema alternativo.

Ao longo do trabalho serão utilizadas as seguintes siglas referentes a modelos: modelo de elementos finitos (MEF), modelo de engenharia (ME), modelo visual (MV), modelo de interpretação (MI), modelo sintaticamente orientado (MSiO), modelo semanticamente orientado (MSeO).

2. A Família de Sistemas Pré-Processadores

Esta pesquisa dirigiu-se à classe de Sistemas de Engenharia Baseados em Subdivisão Planar (SBSP), mais especificamente sistemas pré-processadores, com entrada de dados visual, através da representação gráfica da geometria do objeto e de sua manipulação direta.

A captura de dados num sistema pré-processador pode se dar através de uma linguagem de entrada de dados ou através de uma representação visual do objeto em análise. Pela natureza numérica dos dados, pelo volume e pela precisão necessária, o processo de captação através de uma linguagem de entrada de dados é oneroso e sujeito a erros. Isto justificou a evolução dos sistemas no sentido de uma entrada de dados visual com manipulação direta, que oferece ao usuário um acréscimo de qualidade com relação à entrada de dados textual. No entanto, se as peculiaridades da entrada de dados visual não forem adequadamente tratadas, as vantagens que ela poderia proporcionar ficam inertes, no sentido de que o usuário não encontra recursos para expressar suas intenções. O enfoque da pesquisa nesta classe de sistemas teve como objetivo a avaliação da capacidade de expressão da linguagem de interface, no tocante à manipulação direta sobre a representação visual. Com poucas adaptações, o resultado desta pesquisa poderá ser estendido a outras classes de sistemas que atendam às mesmas premissas.

A seguir serão detalhadas as características desta família de sistemas, com o intuito de caracterizar claramente o problema em foco.

2.1. Funções de um Pré-Processador

Os pré-processadores são captadores de dados para análise e simulação sobre objetos de engenharia (OE). A análise e simulação são executadas com base no MEF.

A tarefa de análise como um todo divide-se em três etapas, apresentadas na Figura 2. A etapa de pré-processamento faz a captação de dados para a etapa de simulação pelo MEF. Nela é feita a especificação do objeto analisado e da malha de elementos finitos. A etapa de análise/simulação utiliza-se dos dados especificados. A etapa de pós-processamento lança mão de recursos gráficos para visualização dos resultados. Os nomes, pré e pós-processador, devem-se ao fato da cultura na área estar centrada no processo de análise/simulação.

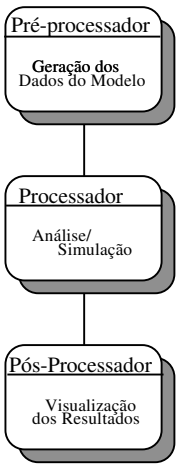


Figura 2. Arquitetura de um sistema de análise de elementos finitos

A tarefa do pré-processador divide-se em duas grandes etapas: a definição do OE sobre o qual será traçada a malha de elementos finitos e a definição da malha de elementos finitos. A definição do OE cabe ao usuário, o traçado da malha é gerado automaticamente pelo sistema. Em geral, é permitida uma interferência do usuário sobre a malha gerada pelo sistema. Essa interferência deve-se a que o estado da arte, em termos de algoritmos de traçado de malha, ainda não permite a automação completa do processo. Devido a isto, a definição da malha pode se dividir também em duas etapas, o que permite que após a interferência do usuário o sistema faça uma verificação da malha que será passada ao analisador/simulador. Na primeira etapa de traçado da malha são então executadas a geração automática da malha e as alterações do usuário sobre ela, na segunda etapa é feita a verificação da interpretabilidade da malha, pelo sistema.

De acordo com a especificação acima, consideramos que a tarefa do pré-processador como um todo divide-se em três etapas, ilustradas na Figura 3: a definição do OE, a geração automática da malha e a definição da malha propriamente dita. A sequência de etapas não significa que o usuário não possa voltar atrás. Ele pode retornar às primeira e segunda etapas quantas vezes for necessário para obter o produto desejado, evidentemente refazendo as etapas posteriores. Ou seja, a cada vez que o usuário alterar a definição do OE a malha deverá ser retraçada.

Existem diferentes alternativas para a representação visual do OE, por exemplo um modelo de reticulados ou um modelo geométrico. Trabalhamos com a classe de pré-processadores que fazem a captação de dados através de uma representação geométrica do OE, com base em subdivisão planar [Carvalho-Gattass-Martha (1990), Campos (1991)]. Em outras palavras, trabalhamos com sistemas de geração de modelos de elementos finitos através do traçado geométrico.

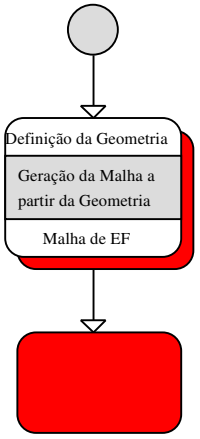


Figura 3. Etapas de um Pré-Processador

A etapa de definição do OE compreende a definição da geometria e topologia, e de suas propriedades físicas. As propriedades físicas do objeto são definidas através de seus atributos. Por atributos entende-se: materiais, excitações, restrições. Ainda nesta etapa, para efeito de análise, a geometria do OE pode ser dividida em regiões, às quais são associados os atributos. A nível da interface é uma etapa de desenho, ela produz a representação visual do objeto. A nível da estrutura de dados é uma etapa de captação do domínio da equação diferencial do problema, que tem solução pelo método de elementos finitos.

Na segunda etapa do pré-processador é gerado o traçado automático da malha de elementos finitos, sobre a geometria especificada. As malhas são traçadas sobre regiões. Entre o modelo

geométrico e a malha traçada existe uma relação hierárquica, onde os elementos da malha herdam as características da região sobre a qual foram traçados. Isto é, atributos como materiais, excitações e restrições que foram associados às regiões definidas no modelo geométrico, são herdados pelos elementos da malha sobre elas traçados.

Nesta etapa o usuário pode interferir sobre a malha, tanto no traçado gerado pelo sistema, quanto nas propriedades herdadas da geometria, associando novos atributos aos elementos da malha. Nesta etapa também ocorre manipulação da representação visual, agora não mais sobre o traçado da geometria mas sobre o traçado da malha. O tipo de interferência permitido ao usuário também pode variar de um sistema para outro, de acordo com a capacidade de geração de malha do pré-processador. Portanto, na primeira e segunda etapas ocorre manipulação direta sobre a representação visual, da geometria e da malha, respectivamente.

Na última etapa, executada pelo sistema, é feita a verificação da interpretabilidade da malha. Após sua execução o usuário não tem mais acesso de edição ao traçado da malha.

2.2. A Geração da Representação Visual e a Captação de Dados

Nas etapas apresentadas para o pré-processador ocorrem dois processos: um processo de representação visual e um processo de captação de dados. A representação visual corresponde ao traçado, tanto da geometria como da malha. A captação de dados é feita através da representação visual.

Na etapa de definição do OE ocorre um *pre-editing,* pois através da manipulação da representação visual o usuário gera a entrada de dados para o sistema. Na etapa de definição da malha ocorre um *post-editing*, porque o usuário atua sobre um traçado gerado pelo sistema, produzindo uma alteração sobre os dados da malha já existentes.

Os processos de geração da representação visual e captação de dados podem ocorrer em paralelo ou em sequência. Se ocorrem em sequência dizemos que o OE é **representado**, porque o traçado é definido e posteriormente submetido a validação. Se os processos ocorrem em paralelo dizemos que o OE é **construído**, porque ele é representado e validado simultaneamente. Na representação a captura de dados é posterior à validação, na construção ela é simultânea.

O processo de representação é executado com base em um modelo de subdivisão planar. Nele o traçado é tratado como um conjunto de faces, arestas e vértices. O significado do traçado é

meramente visual. No processo de construção da geometria o traçado é interpretado como um conjunto de regiões, fronteiras e nós, onde uma fronteira delimita uma região, e um nó delimita uma fronteira. Neste processo é atribuído à representação visual um significado referente a um ME, através de um mapeamento entre as primitivas do ME e do MV: região-face, fronteira-aresta, nó-vértice.

Da mesma forma, no processo de construção da malha, o traçado é interpretado como um conjunto de elementos e nós. Neste processo é atribuído à representação visual um significado referente a um MEF, através de um mapeamento entre as primitivas do MEF e do MV: elemento-face e nó-vertice. A aresta, que corresponderia ao lado do elemento finito, não é tratada no MEF. Na manipulação da malha, a geometria do OE que serviu de base ao traçado da malha deixa de existir do ponto de vista conceitual. Embora a representação visual seja a mesma, o usuário não tem acesso ao traçado geométrico, mas apenas ao traçado da malha.

Existe uma coerência entre o traçado e suas diferentes interpretações. O elo de ligação entre eles é definido pelas primitivas. A Tabela 1 ilustra a relação entre as primitivas, no processo de representação visual e no processo de construção nos diferentes modelos.

Primitivas do Modelo de Subdivisão Planar	Primitivas do Modelo de Engenharia	Primitivas do Modelo Elementos Finitos
Face	Região	Elemento
Aresta	Fronteira	
Vértice	Nó	Nó

Tabela 1. Relação entre as primitivas dos diferentes modelos

A representação visual do objeto deve ser coerente com o ME, para que possa ser interpretada de acordo com ele. Mas também deve ser coerente com o MEF, porque os dados geométricos e topológicos recolhidos através dela são a base para o traçado da malha de elementos finitos, e a consistência da malha exige geometria e topologia adequadas. A Figura 4 esclarece a relação entre a representação visual do OE e os MI.

O ponto que nos interessa nos diferentes processos de manipulação apresentados é a dicotomia entre MV e MI. A presença de dois modelos de interpretação apenas facilita a compreensão do problema. Embora todos os processos utilizem as mesmas primitivas visuais, existe uma distinção básica entre a manipulação dessas primitivas na representação e na construção. Ou seja, a representação visual é manipulada de forma distinta, de acordo com o processo ativo.

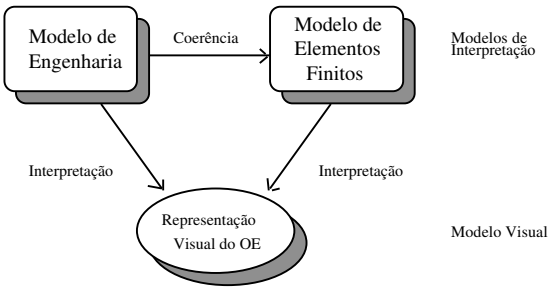


Figura 4. Relacionamento entre a representação visual do OE e os MI

Exatamente sobre esta dicotomia entre MV e MI é que vamos discutir a força de expressão da linguagem de interface. O MV é um modelo sintático, ele orienta a geração da representação visual. O MI é um modelo semântico, ele orienta a captação de dados. Como dados entendemos: tanto a especificação da geometria e da malha, como os atributos a elas associados. Para efeito de esclarecimento a Tabela 2 apresenta os termos interrelacionados que serão utilizados alternadamente na sequência do trabalho.

Característica	Orientação	Modelo
Modelo Visual	Sintático	Modelo de Subdivisão Planar
Modelo de Interpretação	Semântico	Modelo de Engenharia
		Modelo de Elementos Finitos

Tabela 2. Termos interrelacionados

3. Os Modelos Sintático e Semântico nos Sistemas Pré-Processadores

Os processos de representação e construção, conforme foram apresentados, trazem vantagens e desvantagens ao usuário. Conforme for a orientação adotada na interação usuário-sistema, será a capacidade de manipulação direta sobre a representação gráfica.

3.1. Modelos Conceituais Básicos

O modelo conceitual de um sistema é concebido pelo projetista de sistemas a partir do problema real. A maioria dos sistemas tem seu modelo conceitual assentado em um único modelo básico. O termo modelo básico surgiu em nossa pesquisa como uma abreviação de modelo conceitual básico, para distinguir a presença de mais de um modelo de referência no modelo conceitual de um sistema.

Para melhor compreensão sobre a idéia do que é um modelo básico vamos inicialmente exemplificar com um editor de textos. O modelo básico de um editor de textos é bastante simples. O editor trata o texto como uma sequência de caracteres, para o qual ele oferece recursos de formatação e operações, como cópia, apagamento e outras. Porém, ao invés disto, o editor poderia manter um critério gramatical ativo sobre o texto. Neste caso, suponha uma operação de edição em que o usuário desejasse retirar um verbo de uma frase para substituí-lo por outro. Se ele tentasse usar a opção de apagar a palavra para depois incluir outra, o sistema poderia bloquear a operação em função do critério gramatical ativo. Por este critério não poderia ser construída uma frase sem verbo e, portanto, entre uma operação e outra a frase estaria inconsistente. Neste exemplo, o modelo básico do sistema foi alterado, passou de um modelo de manipulação física do texto para um modelo de manipulação gramatical. As operações anteriores passaram a ser condicionadas a um conjunto de regras da língua.

Da mesma forma, nos pré-processadores, podemos ter um caso em que a representação visual sofre manipulações compatíveis com desenhos geométricos (tal como o texto sofre manipulações compatíveis com cadeias de caracteres), ou um outro em que ela sofre manipulações compatíveis com a engenharia do objeto (tal como o texto sofre manipulações compatíveis com a gramática da língua). A Tabela 3 ilustra as formas de manipulação citadas para editores de texto e pré-processadores.

	Editores de Texto	Pré-processadores
Manipulação Orientada à Forma	sequência de caracteres / tratamento plástico	tratamento combinatório / desenho
Manipulação Orientada a Conteúdo	texto na língua / tratamento gramatical	tratamento geométrico/topológico / representação do artefato

Tabela 3. Diferentes formas de manipulação para editores de texto e pré-processadores

No caso do editor de textos o modelo semântico apresentado é opcional. O produto do editor de textos é utilizado diretamente pelo usuário, o qual é responsável por sua validação. A utilização do modelo semântico seria um recurso extra oferecido ao usuário no sentido de auxiliá-lo a produzir um texto com qualidade semântica. No caso do pré-processador, a validação semântica sobre a representação visual não é opcional. Ela pode ser adiada para uma segunda etapa, mas não pode ser abandonada. Isto porque o pré-processador é um captador de dados para um sistema de análise e simulação, para o qual deve ser garantida a validade dos dados de acordo com o modelo semântico.

3.2. A Funcionalidade dos Modelos Sintático e Semântico

Os modelos sintático e semântico comportam primitivas formais e regras de comportamento dessas primitivas, a partir do quais são geradas estruturas válidas no modelo. No modelo sintático, o comportamento definido sobre as primitivas é meramente combinatório e estruturador, sendo que as estruturas geradas atendem a regras de combinação entre as primitivas e englobam uma hierarquia de relacionamento entre elas. O modelo semântico é mais complexo, as regras são mais relacionais e ele incorpora uma atribuição de significado às primitivas e às estruturas geradas, significado este relativo a um modelo conceitualmente específico.

De acordo com um modelo sintático, a decisão sobre a validade e consistência de estado no sistema de representação está ligada à natureza das primitivas formais e de sua combinação. Nele uma estrutura é válida se as primitivas forem válidas e a combinação entre elas for válida. No caso de um modelo semântico, além disso, a decisão sobre validade e consistência de estado está ligada ao significado atribuído às estruturas pelo modelo específico. Uma determinada estrutura pode ser válida do ponto de vista sintático, combinação das primitivas formais, e inconsistente no que se refere ao modelo de significado. Por exemplo, a Figura 5 ilustra a situação de validade de determinados traçados em modelos sintáticos e semânticos.

Sob o enfoque de um modelo sintático, como por exemplo o modelo de sub-divisão planar, as Figura 5(a) a 5(d) são válidas, porque atendem às regras de combinação das primitivas formais no modelo (basicamente o traçado de arestas). Sob o enfoque de um modelo semântico, por exemplo o modelo de engenharia, apenas a Figura (c) será válida, porque só ela atende às regras de interpretação do ME. A mesma Figura (b) que é inválida no modelo de engenharia pode ser válida em outro modelo semântico, por exemplo um modelo de representação gráfica de redes elétricas, porque só ela atende às regras de interpretação desse modelo. A Figura (d) tem uma

aparência semalhante à (c), do ponto de vista sintático ela está correta, aparentemente ela poderia ser válida no modelo de engenharia. No entanto, ela contém uma inconsistência semântica, uma aresta solta no meio de uma face.

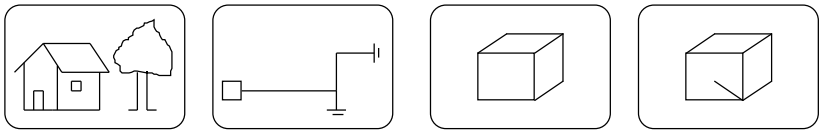


Figura 5. Situações de validade em aspectos sintáticos e semânticos

Estes exemplos, evidenciam a diferença de exigências entre um modelo sintático e um modelo semântico, e principalmente a permissividade do modelo sintático no que se refere à manipulação do traçado. Uma vez que as regras do modelo sintático são apenas combinatórias, elas oferecem mais flexibilidade de manipulação.

3.3. Opção pela Representação ou pela Construção

Se a manipulação da representação visual for executada com uma orientação exclusivamente sintática (representação), a captação de dados poderá estar comprometida. A captação deverá ser feita em uma etapa posterior, após validação da representação. Na maioria dos casos os sistemas têm dificuldades em recuperar a consistência de uma representação visual gerada em um modelo exclusivamente sintático. Neste caso o usuário ganha em flexibilidade na manipulação da representação visual e perde em consistência. A orientação sintática facilita a operação no nível da representação visual.

Se ela for executada com uma orientação exclusivamente semântica (construção), a manipulação poderá estar comprometida. De forma semelhante, o usuário terá dificuldades em manipular a representação visual através de um modelo exclusivamente semântico. A orientação semântica garante a consistência dos dados com relação ao modelo que atribui significado à representação visual. Em ambos os casos, as vantagens que se poderia obter de uma entrada de dados visual se perderiam, ou pelo excesso de flexibilidade, ou pela rigidez dos princípios de manipulação.

A título de exemplo, o sistema ANSYS [ANSYS] opera com uma manipulação sintaticamente orientada, onde primitivas como face e volume não são automaticamente reconhecidas pelo sistema no traçado geométrico. Para que eles sejam reconhecidos o usuário deve ativá-los, o que significa associar explicitamente a semântica de face ou volume ao traçado. Para que a ativação de volume seja permitida, o usuário deve previamente ativar todas as faces que o compõem. A operação inversa é verdadeira; em caso de alteração do traçado é necessária a desativação prévia das primitivas semânticas. Por exemplo, se o usuário deseja apagar uma aresta que delimita um volume, ele deverá desativar o volume e desativar as faces às quais a aresta pertence. Só então ele poderá operar sobre a aresta.

A expectativa neste modelo de interação é oferecer flexibilidade na representação do OE. Porque permite ao usuário representar todo o objeto, atribuindo semântica à representação, posteriormente, passo a passo, na medida em que as primitivas são ativadas. Com isto o sistema não tem a necessidade de validar toda a representação visual de uma só vez, o que provavelmente estaria fora de seu controle. À medida em que o usuário associa a semântica ao traçado o sistema faz a validação, recusando a associação que gere inconsistência. É evidente que todo este procedimento é extremamente trabalhoso para o usuário, especialmente nos casos de alteração, onde a semântica deve ser desassociada e reassociada após a alteração. Neste caso a flexibilidade de manipulação tem um custo bastante alto para o usuário.

Em sistemas mais atuais, como por exemplo o MTool [MTool], a tentativa dos projetistas tem sido de oferecer flexibilidade e controle semântico simultaneamente. À medida em que se tenta fazer isto, os conflitos afloram, em função das incompatibilidades entre o processo de representação e construção. E é para estes conflitos que buscamos uma solução.

Sobre essas duas orientações vamos argumentar, apresentando situações dos sistemas pré-processadores onde nenhuma delas provê uma boa interação entre usuário e sistema, o que torna a linguagem de comunicação da interface insuficiente. Ao final, será proposta uma solução intermediária que permite um tratamento adequado aos problemas apresentados.

4. Uma Abordagem Teórica à Entrada de Dados Visual

No âmbito teórico vamos definir de que forma se processa a comunicação com o usuário. Serão esclarecida as noções de modelo conceitual de sistema e modelo conceitual de interação, e será

apresentada a forma de tradução do modelo conceitual do sistema para o usuário em sistemas pré-processadores.

4.1. O Modelo Conceitual do Sistema e o Modelo Conceitual da Interação

O modelo conceitual do sistema é aquele concebido pelo projetista a partir do problema real. Entre as perspectivas definidas por Kammersgaard [1988] para a comunicação usuário-sistema, a que nos interessa apresenta o sistema como um *media* de comunicação entre agentes humanos, no caso entre o projetista do sistema e o usuário. Nesta perspectiva, a interação do usuário com o sistema seria o equivalente à interação do usuário com o projetista, tendo o sistema como um veículo de comunicação entre eles. Para que esta perspectiva se concretize é necessário que ao utilizar o sistema o usuário identifique, através da interface, o modelo conceitual do projetista. Portanto, a linguagem de comunicação da interface tem o papel de tradutora do modelo conceitual do sistema para o usuário, e da intenção deste para o sistema.

Essa abordagem deve ser particularizada nos sistemas gráficos por manipulação direta. Nestes, a interface é a representação explícita do domínio do sistema [Hutchins-Hollan-Norman (1986)]. Ao fazer a manipulação da representação gráfica o usuário atua diretamente sobre o domínio do sistema, incluindo, excluindo e alterando dados. Neste contexto é necessário distinguir entre: modelo conceitual do sistema e modelo conceitual da interação.

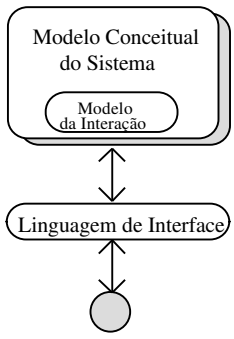


Figura 6. Integração entre Modelo Conceitual do Sistema e Modelo Conceitual da Interação

Em sistemas com entrada de dados através da manipulação direta da representação visual, a interação com o usuário adquire uma dimensão maior. Os problemas que envolvem a captura de dados tornam-se mais complexos. É necessário um tratamento conceitual mais apurado, que

ofereça suporte ao problema de manipulação direta. A Figura 6 ilustra a interação entre os modelos, onde o modelo conceitual da interação é parte do modelo conceitual do sistema.

Enquanto o modelo do sistema define a validade dos dados com que o sistema opera, o modelo de interação, através da forma de atuação adotada sobre a representação gráfica, define de que forma ocorre a captação na interface. Ou seja, o modelo conceitual do sistema modela o dado, o modelo conceitual de interação modela a captura de dados especificamente. A linguagem de interface deve traduzir para o usuário o modelo conceitual do sistema como um todo.

4.2. Tradução do Modelo Conceitual do Sistema para o Usuário

A distância a ser superada pela linguagem da interface na comunicação usuário-sistema, é tratada por Norman [Norman (1986)] como um "golfo de comunicação", dividido em "golfo de execução" e "golfo de avaliação". O primeiro refere-se à distância entre a intenção do usuário, ou sua ação conceitual, e sua transformação em uma operação do sistema. Na medida da adequação da interface, ao fazer a tradução da sua ação conceitual, o usuário deverá estar indo ao encontro da ação conceitual definida pelo projetista. Por exemplo, suponha um caso onde o modelo conceitual do sistema define operações que atuam somente sobre elementos previamente marcados, conceitualizados como um grupo. Em sistemas de manipulação gráfica esta idéia é frequentemente utilizada para dar ao usuário a flexibilidade de executar determinadas operações sobre um ou mais elementos simultaneamente. Por exemplo uma operação de apagamento, onde o sistema não apaga elementos, mas apaga um grupo, que pode ser composto por vários elementos de natureza diferente. Isto significa, por exemplo, que o usuário não encontrará no sistema a operação de apagar aresta ou apagar vértice, apenas a operação conceitual de apagar grupo. Ele terá então que marcar o elemento a ser apagado como pertencente ao grupo e em seguida ativar a operação de apagamento do grupo. Para que o usuário utilize corretamente o sistema é necessário que esse modelo conceitual esteja expresso na interface. Quanto mais claramente ele estiver expresso na interface, menor a distância do golfo de execução.

O golfo de avaliação é o caminho inverso, refere-se à interpretação do usuário sobre o comportamento do sistema. Tomando como exemplo o mesmo caso citado acima, os elementos marcados como pertencendo ao grupo, podem mudar de cor para indicar seu *status.* Essa sinalização visual é a resposta do sistema à operação de marcar elementos. Quanto mais claros para o usuário os indícios da operação executada, mais rapidamente ele poderá avaliar se sua

intenção foi atingida e menor será o golfo de avaliação. A Figura 7 esclarece o conceito de golfo de comunicação.

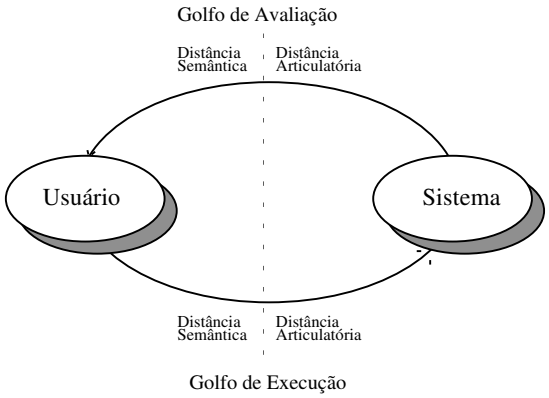


Figura 7. Golfo de comunicação entre usuário e sistema

A ultrapassagem do golfo de comunicação é feita em duas etapas: tradução semântica e tradução articulatória. Na tradução semântica o usuário traduz sua intenção, ou sua operação conceitual, em uma operação do sistema. Na tradução articulatória o usuário traduz a operação do sistema em uma ativação do sistema. O usuário conceitualiza e traduz a operação sobre o feedback que ele recebe do sistema.

Nos sistemas em questão as traduções semântica e articulatória acontecem de forma diferente, conforme ilustrado na Figura 8. Como foi dito, os pré-processadores fazem a captação de dados através da representação visual. A representação visual é o veículo de passagem de dados do usuário para o sistema. Ao entrar em contato com a interface, no golfo de avaliação, o usuário recebe duas informações: uma informação visual e uma interpretação semântica sobre a informação visual. Sobre esse conjunto de informações o usuário conceitualiza a operação a ser executada. Na ultrapassagem do golfo de execução o usuário faz dois níveis de tradução semântica para sua ação conceitual. No primeiro nível ele traduz sua intenção em uma operação do sistema, ou seja, ele traduz sua ação conceitual para uma operação conceitual do sistema. No segundo nível o usuário faz uma tradução da operação conceitual do sistema para a representação visual, através da qual é feita a captação de dados. O usuário traduz de que forma aquela operação conceitual se concretiza na representação visual do OE. Ou seja, ele faz uma primeria tradução semântica para o MI e uma segunda tradução semântica para o MV.

No terceiro momento é feita a tradução articulatória ou ativação do sistema. Neste momento é ativada a manipulação sobre a representação visual do objeto, que vai atender à operação conceitualizada.

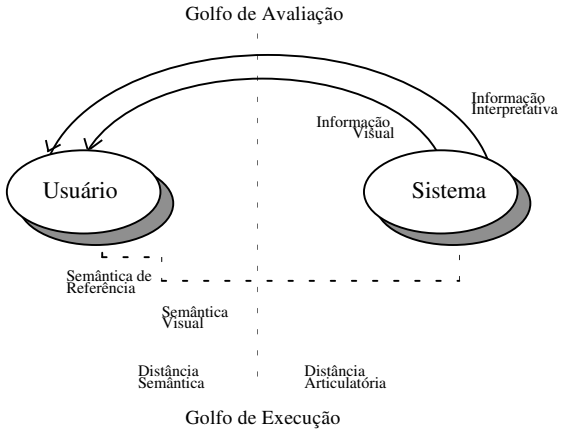


Figura 8. Golfo de Comunicação entre usuário e sistema para Sistemas Pré-Processadores

Por exemplo, o usuário tem como meta definir uma região sobre o traçado geométrico do OE. Para efetivar essa operação conceitual ele faz a tradução semântica para o MI, obtendo a operação "traçar fronteira da região". A seguir ele faz outra tradução semântica para a representação visual, obtendo a operação "traçar uma aresta". Por último, faz a tradução articulatória, quando então é ativado o traçado da aresta. Esta questão será apresentada com detalhes no estudo de caso.

No item anterior foi apontada a existência de um MI e um MV. Aqui verificamos de que forma a tradução semântica se processa entre ambos. De modo geral podemos afirmar que a meta do usuário é sempre semântica, e que a execução desta meta é semântica e sintática. A tradução do nível semântico para o nível da representação visual é um fardo computacional que cabe ao usuário.

Através do enfoque teórico fica esclarecida a dicotomia entre o MV e o MI, no modelo conceitual da interação. O *gap* entre as duas semânticas, representado no golfo de execução, na Figura 8, será menor ou maior em função da capacidade de expressão da linguagem de comunicação da interface. A linguagem de comunicação será mais ou menos expressiva em função da orientação adotada no modelo conceitual de interação.

No estudo de caso a seguir será demonstrado que nem a orientação sintática pura, nem a orientação semântica pura atendem adequadamente ao problema de manipulação. Na medida em que isto ocorre a linguagem de comunicação fica insuficiente, e o ônus de ultrapassagem desse *gap* fica a cargo do usuário.

5. Estudo de Caso

O objetivo deste estudo de caso é demonstrar, através da análise de algumas operações em MSiO e MSeO, as questões discutidas até o momento. Os casos apresentados foram baseado no sistema MTool [1994], mas o objetivo não é discutir a operação implementada no sistema. Os exemplos estão sendo apresentados de forma bastante genérica, sendo analisados em diferentes contextos de manipulação. O objetivo é discutir uma classe de problemas que se mostra através de certos tipos de manipulação. No primeiro estudo, sobre as operações com regiões, serão discutidos os conflitos em função do modelo de interação adotado. O segundo estudo, sobre a representação do furo, é um problema clássico que não tem solução no modelo de visual.

5.1. Um Estudo sobre as Operações de Definir e Desfazer Região

Será discutido o caso de uma operação e sua operação inversa, à quais chamaremos de Definir e Desfazer Região. Nestes exemplos podem ser observados claramente os dois níveis de tradução semântica: da intenção do usuário para o operação conceitual do sistema, e desta para a representação visual. Através dos dois níveis de tradução, diferentes intenções do usuário podem se transformar em operações idênticas sobre o MV. Ou seja, diferentes intenções do usuário podem se concretizar em manipulações idênticas sobre a representação visual. A situação inversa também ocorre, uma mesma intenção do usuário pode se concretizar em diferentes operações sobre a representação visual, dependendo das condições de traçado existentes. Este fato caracteriza fortemente a dicotomia entre MV e MI, e o fardo de tradução que esta dicotomia acarreta ao usuário.

5.1.1. A Operação de Definir Região

É o caso em que o usuário tem a intenção de definir uma região sobre a geometria do OE para, em seguida, associar um determinado material ela. Vamos tratar apenas a operação de definir a região. Por exemplo, a Figura 9(a) mostra uma representação de dois túneis em corte transversal. As representações estão simplificadas para efeito didático. A Figura 9(b) mostra a mesma representação, porém a área que circunda os túneis foi dividida em três regiões, que serão especificadas como três tipos de solo diferentes. Cada região está assinalada por uma textura diferente. De acordo com o modelo semântico do sistema suposto, existe um mapeamento região-face, isto é, uma região no modelo semântico corresponde a uma face na representação visual. Portanto, em 9(a) existem três regiões definidas, e em 9(b) existem 5 regiões definidas.

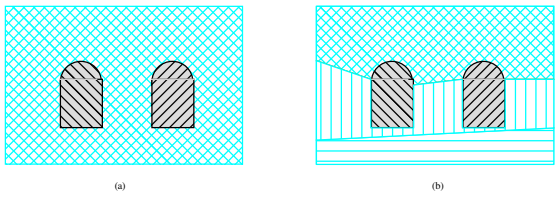


Figura 9. Divisão da geometria do OE em regiões

Na Tabela 4 estão apontados os dois níveis de tradução semântica e a tradução articulatória, para a definição de cada região.

Intenção do usuário:	Definir Região	
		Tradução semântica
Modelo de Interpretação:	Criar fronteira	
		Tradução semântica
Modelo Visual:	Traçar aresta	
		Tradução articulatória
Articulação:	Ativar traçado aresta	

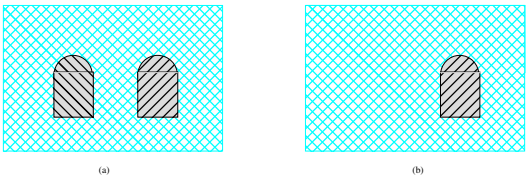
Tabela 4. Traduções semântica e articulatória para a operação de Definir Região

5.1.2. A Operação de Desfazer Região

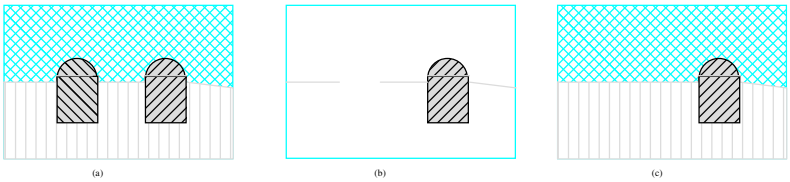
Esta é a operação inversa da operação de Definir Região. É o caso em que o usuário tem a intenção de desfazer uma região já definida. A seguir são apresentadas cinco situações diferentes

para desfazer uma região, identificadas por Situações 1 a 5. A Figura 10 mostra a sequência de estados na manipulação do traçado, para cada uma. A sequência de estados apresentada é a sequência mínima necessária para executar a operação. Por exemplo, em alguns casos, se o usuário tem que apagar três arestas, ele pode apagar uma a uma ou as três simultaneamente. Estamos considerando que ele apaga as três simultaneamente.

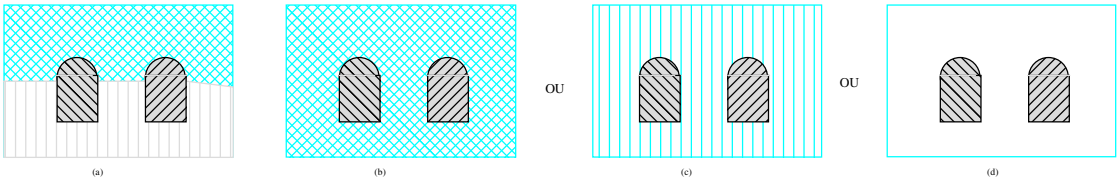
Situação 1



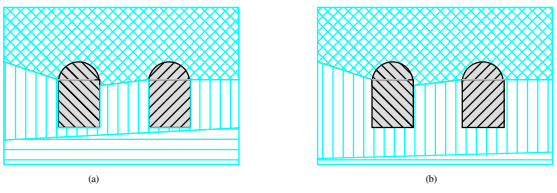
Situação 2



Situação 3



Situação 4



Situação 5

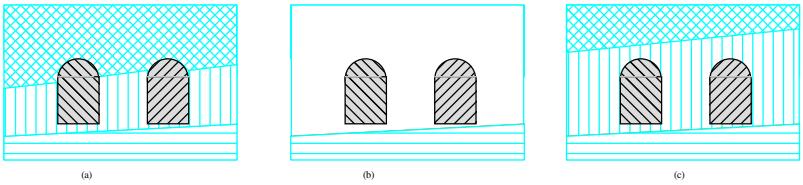


Figura 10. Sequências de estado na manipulação do traçado, durante diferentes operações de Desfazer Região.

Na Tabela 5 são apontados, para cada caso, os dois níveis de tradução semântica e a tradução articulatória, necessários à concretização da operação. É apresentado também o número mínimo de passos necessários a cada operação.

Situação 1 - Foi dito que de acordo com o modelo semântico do sistema, existe um mapeamento região-face. Neste caso, a intenção do usuário é apagar o traçado correspondente a um dos túneis. Em (a) existem três regiões definidas, em (b), após o apagamento do traçado correspondente ao túnel, existem apenas duas regiões. Pode ser verificado na Tabela 5 que esta operação foi executada em 1 passo.

Situação 2 - Neste exemplo a intenção do usuário é a mesma do caso anterior, apenas as condições de traçado são diferentes. Existe uma fronteira de região dividindo a área de contorno do túnel. A operação tem que ser executada, obrigatoriamente, em dois passos: um passo de apagamento do traçado do túnel e um passo de recomposição da fronteira. Entre o passo de apagamento e o de recomposição da fronteira existe um momento em que o traçado fica incompleto, conforme representado no quadro (b).

Situação 3 - Neste exemplo a intenção do usuário pode ser a de transformar duas regiões em uma região única, ou simplesmente eliminar as regiões. Na primeira suposição, partindo de (a) ele poderia obter os resultados em (b) ou (c), na segunda obteria o resultado em (d).

Situação 4 - Neste caso a intenção do usuário é alterar as dimensões da região. Para tanto ele necessita simplesmente mover a fronteira de região, conforme quadro (b).

Situação 5 - Neste caso a intenção do usuário é a mesma do caso anterior. Porém as condições de traçado são diferentes. O traçado da fronteira a ser alterada está dividido em três arestas. Por isto, de forma semelhante à Situação 2, será necessário executar a operação em dois passos, um passo de apagamento e um passo de traçado.

	Situação 1	Situação 2	Situação 3	Situação 4	Situação 5
Intenção	Eliminar 1 túnel	Eliminar 1 túnel	Transformar em região única	Alterar região	Alterar região
Tr.Mod.Interp.	Apagar fronteira	Apagar fronteira	Apagar fronteira	Mover fronteira	Apagar fronteira
Tr.Mod.Visual	Apagar aresta	Apagar aresta	Apagar aresta	Mover aresta	Apagar aresta
Tr.Artic.	Ativar apagamento	Ativar apagamento	Ativar apagamento	Ativar movimento	Ativar apagamento
Tr.Mod.Interp.		Recompor fronteira			Criar fronteira
Tr.Mod.Visual		Criar aresta			Criar aresta
Tr.Artic.		Ativar criação			Ativar criação

Tabela 5. Traduções semântica e articulatória para operações de Desfazer Região

5.1.3. Análise das Operações de Desfazer Região em um MSiO

Se as operações acima detalhadas fossem executadas com base em um modelo sintaticamente orientado, apenas a representação visual seria tratada pelo sistema durante a manipulação. O sistema não trataria, durante a manipulação, as informações de materiais associados às regiões. No entanto, ao fazer a análise temos que supor a possibilidade de haver uma associação de material à região, porque durante a manipulação do traçado esta informação deve ser preservada para tratamento posterior.

Vamos inicialmente supor que não haja material associado a nenhuma das regiões. Neste caso, se o usuário chegar ao final das operações propostas o modelo se manterá consistente. Porém se, por algum motivo, o usuário interromper a operação em uma fase intermediária, o modelo perde a consistência. Porque se perde a informação topológica que é fundamental para a base de traçado da malha de elementos finitos. Por exemplo a Situação 2(b).

A inconsistência por interrupção da operação também pode ocorrer nos casos em que o usuário executa em mais de um passo uma operação que poderia ser executada em apenas um. Por exemplo a Situação 1, se ao invés de apagar de uma só vez toda a fronteira que delimita o túnel, o usuário apagasse apenas a parte correspondente a uma única aresta, ficaria com uma situação semelhante à apresentada na Figura 11. Se a operação fosse interrompida neste momento, o modelo também ficaria inconsistente por perda da informação topológica.

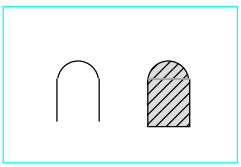


Figura 11. Estado intermediário na manipulação do traçado, durante operação de Desfazer Região

Supondo que houvesse material associado às regiões, mesmo que o usuário chegasse ao final da alteração da representação visual, nas Situações 2,3 e 5, desfeitas as regiões, seria gerada uma inconsistência na informação de material associado.

5.1.4. Análise das Operações de Desfazer Região em um MSeO

Inicialmente faremos uma análise das operações somente sobre a questão da geometria/topologia, considerando que não existem materiais associados às regiões. Numa interação semanticamente orientada, no momento em que uma face é delimitada na representação visual ela é automaticamente reconhecida como uma região. A relação inversa também é verdadeira. Se o sistema trabalha com a informação semântica, ele não permite o apagamento de uma região com adjacências, por exemplo a Situação 1. Porque semanticamente, ao apagar uma região destrói-se a região adjacente.

Na Situação 2 a questão é mais grave, porque além da operação agredir a regra semântica pelo fato de destruir a região adjacente, existe um passo intermediário onde de fato a representação visual está semanticamente inconsistente, quadro (b). Neste caso, mesmo supondo que o sistema permitisse o apagamento da região, o modelo semântico ativo acusaria um erro entre o inicio e o fim da operação total. Existe um intervalo entre o início e o fim da operação no qual a representação visual fica inconsistente. Para que a operação fosse executada seria necessário que o modelo semântico fosse desativado por um certo intervalo de tempo, até que o usuário pudesse dar consistência à representação.

Na Situação 3 pode ocorrer o mesmo problema se o usuário, ao invés de executar a operação de apagamento da fronteira em um único passo, executá-la em vários passos, apagando uma a uma as arestas que compõem a fronteira. Observe que o intervalo de tempo necessário para que a representação ganhe consistência não é relativo à operação conceitual. Ele é completamente aleatório, uma vez que depende apenas do número de arestas a serem apagadas, ou do número de passos em que o usuário se disponha a executar a operação. A Figura 12 exemplifica a Situação 3 executada em três passos ao invés de um. Observe-se que na Situação 1 a idéia de eliminar uma região está associada à idéia de eliminar uma face, é possível que o usuário execute uma tradução semântica de "eliminar região" para "eliminar face". Na Situação 3 isto não ocorre, a idéia de eliminar região está associada à idéia de eliminar parte da fronteira, obrigatoriamente o usuário

deverá fazer a tradução semântica para "eliminar aresta", o que de certa forma facilita a possibilidade do usuário executar o apagamento da fronteira aresta por aresta.

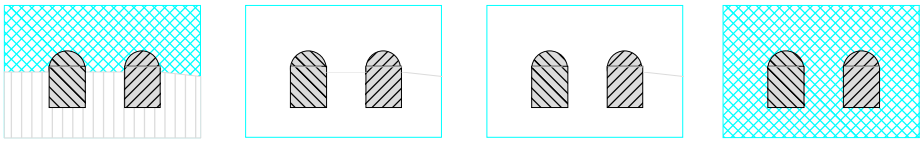


Figura 12. Estados intermediários na manipulação do traçado, durante operação de Desfazer Região

Se supusermos agora que existem diferentes materiais atribuídos a essas regiões o problema semântico fica mais claro. Se uma região é destruída o que acontece, semanticamente, com a região adjacente? Ela pode invadir a primeira, como também pode ser destruída. Na Situação 1 a região adjacente invadiu a região eliminada. Neste caso parece não haver muitas dúvidas uma vez que a região eliminada correspondia a uma face circundada por uma mesma região adjacente. Na Situação 3, por exemplo, esta solução já não fica clara. Seria necessária uma interferência do usuário explicitando se a solução correta seria a equivalente a (b), onde a região superior prevalesce, a equivalente a (c), onde a região inferior prevalece, ou a equivalente a (d), onde ambas as regiões são destruídas.

Com relação às Situações 4 e 5 ocorre que a tradução para o modelo visual depende do traçado. Em ambos os casos o usuário deseja apenas alterar as dimensões da região, ele pretende recolocar a fronteira que delimita a região. Na Situação 4(a) a fronteira que delimita as regiões de solo está abaixo do traçado dos túneis, sendo equivalente a uma única aresta. Para recolocá-la basta movimentar a aresta. Embora essa operação de movimentação altere as dimensões geométricas, ela não altera as informações topológicas. Por isto ela não gera inconsistências para o sistema, mesmo havendo material associado à região. Na Situação 5(a) a fronteira que delimita as regiões de solo está colocada entre o traçado dos túneis. Do ponto de vista da representação visual isto significa três arestas. Para recolocá-la é necessário apagar as três arestas e traçar uma única nova aresta na representação visual. Neste caso, recairemos no problema já apresentado, a operação será bloqueada pelo sistema porque, ao apagar a aresta para recriá-la em outra posição, o usuário estará destruíndo as regiões. Portanto, nestes casos, a mesma operação conceitual pode ser aceita ou bloqueada pelo sistema, dependendo das condições de traçado no modelo visual.

Observe que a movimentação da aresta discutida na Situação 4 é a operação intuitiva do usuário. Não estão sendo discutidas as condições de implementação. Está sendo discutido apenas quais seriam os efeitos dessa operação em função do modelo conceitual da operação.

5.1.5. Conclusões sobre as Operações sobre Região

Comparando-se a operação de Definir Região e sua inversa Desfazer Região observamos que:

- As operações não atendem a um mesmo padrão conceitual. Para definir uma região o usuário não encontra dificuldade em traçar novas arestas, mas para desfazer regiões traçadas ele esbarra nas restrições do modelo semântico.

Comparando-se as operações de Desfazer Região observamos que:

- A distância entre a meta do usuário e a articulação do sistema, e a ausência de uma gramática unívoca de tradução de um para outro. Comparando-se as traduções semânticas e articulatórias, verifica-se que nas Situações 1 e 3 a intenção do usuário é diferente e a articulação sobre a representação visual é igual. Nas Situações 1-2, e 4-5 a intenção do usuário é igual e a articulação sobre a representação visual é diferente. O sistema não "capta" a meta semântica do usuário, porque já recebe uma ativação sintática sobre a representação visual. A operação do sistema fica desvinculada da intenção do usuário.

- No caso de uma manipulação com orientação sintática, o controle de consistência fica a cargo do usuário.

De modo geral, foi possível observar que dependendo da orientação sintática ou semântica, a tradução, ou ultrapassagem do golfo de comunicação, não ocorre ou ocorre de forma insatisfatória. Num MSiO o usuário consegue executar as manipulações desejadas, com grande risco de inconsistência da informação geométrica e topológica do OE. Num MSeO o usuário tem dificuldade de executar certas operações, mas tem a consistência garantida.

5.2. Um Estudo sobre o Problema do Furo

Este é um problema clássico, onde o modelo conceitual que sustenta a manipulação direta não oferece suporte à operação conceitual do usuário. Uma dificuldade que ocorre com os sistemas baseados em subdivisão planar em geral, é a idéia de representar um furo na geometria do objeto. Considerando o problema sob a ótica do ME, criar um furo remete à idéia de apagar a face, uma vez que a face representa a região correspondente ao furo, conforme Figura 13(a). Porém apagar a face significa apagar os elementos que a delimitam, destruindo a região. Sob a ótica do modelo de representação visual, o apagamento da face corresponderia ao apagamento das arestas que a delimitam, levando a situação idêntica. Em ambos os casos o efeito desejado é aquele representado na Figura 13(b), porém o efeito obtido é aquele da Figura 13(c).

Em vários pré-processadores a solução adotada é levar para o nível semântico a informação que deveria ser tratada a nível visual. Esta informação passa a ser registrada como atributo, a região representada pela face recebe o atributo de furo.

Neste caso o golfo de comunicação a ser superado pela interface é grande, por dois motivos. Primeiro porque a operação disponível não corresponde à articulação natural da intenção do usuário, ele tem de fazer uma adaptação da sua operação para a operação do sistema. Segundo porque a solução a nível semântico deteriora o conceito de atributo, podendo gerar dúvidas em outras situações.

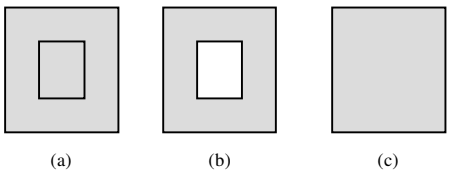


Figura 13. Representação do furo no plano

6. Proposta para um Modelo Sintático Estendido

A proposta deste trabalho é um novo modelo conceitual de interação que busque o equilíbrio nessa negociação entre sintático e semântico. Por ser um modelo intermediário entre um MSiO e um MSeO, foi chamado de Modelo Sintático Extendido. Trata-se de um modelo mais restritivo

do que o modelo sintático e mais flexível do que o modelo semântico. Isto significa estender o modelo sintático de modo que ele passe a dar conta de certos aspectos semânticos, sem a rigidez do modelo semântico. A vantagem dessa extensão é que ela própria pode ser orientada para diferentes modelos semânticos. Isto é, num mesmo sistema é possível manter mais de uma extensão, por exemplo para um ME ou para um MEF.

6.1. Critérios de Extensão do Modelo Sintático

A proposta é que a esse modelo de manipulação sejam acrescentadas heurísticas com base na modelagem de tarefa do usuário, que atendam classes de problemas identificados. É fundamental que as heurísticas atendam a classes de problemas e que mantenham um padrão de referência nesse atendimento, para que não se transformem elas próprias em soluções localizadas. O objetivo da heurística é aproximar a operação executada no modelo visual, da operação conceitual do usuário. Indiretamente a heurística tenta recuperar a intenção do usuário, que ele não expressa quando está ativando uma operação sobre representação visual. Por exemplo o caso de apagamento de aresta, com a intenção de juntar duas regiões. Com isso busca-se diminuir o ônus apontado para o usuário no *gap* entre as traduções semânticas e a tradução articulatória. O resultado é uma extensão da esfera de representação.

A modelagem de tarefas do usuário será feita através de uma gramática de tarefas, a TAG - Task Action Grammar [Payne (1986)]. Durante a manipulação a interface tentará recolher indícios que a orientem em direção a uma determinada tarefa do usuário, a partir da qual a interface poderá propor soluções que agilizem o trabalho do usuário. Por exemplo, no caso de apagamento de uma aresta onde a intenção do usuário é eliminar uma determinada região, que será agregada a outra. Se houver material associado a essa região, a interface, ao invés de bloquear a operação, poderá antecipar alguma solução com relação aos materiais associados. É importante ressaltar que qualquer solução adotada pela interface em função de uma heurística deverá ser claramente sinalizada ao usuário, estando sujeita a cancelamento no caso de falha.

6.2. Especificação dos Modelos de Interação

A especificação dos modelos de interação será feita a partir de suas primitivas formais e as regras de comportamento dessas primitivas. Serão especificados os três modelos básicos assinalados

para os pré-processadores: um modelo sintático - modelo de subdivisão planar, e dois modelos semânticos - modelos de engenharia e elementos finitos.

6.2.1. Especificação das Primitivas dos Modelos

A Tabela 1 mostra a relação de paralelismo entre as primitivas dos três modelos. Apesar dessa relação de paralelismo, dentro de cada modelo elas obedecem uma relação hierárquica diferente, ilustradas na Tabela 6. Parte do comportamento das primitivas dentro do modelo é determinado pela relação hierárquica entre elas. Em função disto, uma mesma primitiva visual obedece a um comportamento diferente dependendo do modelo ativo.

No modelo de subdivisão planar a primitiva de mais alta hierarquia é a aresta, isto significa que a definição de vértices e faces é consequência do traçado de arestas. Uma aresta traçada é delimitada por dois vértices, uma face é delimitada por várias arestas. Quando uma aresta é apagada automaticamente são apagados os vértices que a delimitavam, e desfeita a face, se existir. O comportamento de vértice e face é sempre subordinado ao comportamento de aresta. Neste modelo as primitivas vértice e face não são manipuláveis pelo usuário.

Primitivas do Modelo de Subdivisão Planar	Primitivas do Modelo de Engenharia	Primitivas do Modelo Elementos Finitos
Aresta ↙ ↘ Vértice Face	Região ↓ Fronteira ↓ Nó	Elemento ↓ Nó

Tabela 6. Relação hierárquica entre as primitivas dos modelos básicos

No modelo de engenharia as primitivas obedecem uma correspondência com o modelo matemático. Matematicamente falando, um vértice tem dimensão zero, uma aresta tem dimensão um e uma face tem dimensão dois. Existe uma hierarquia entre as primitivas, onde uma face é delimitada por n arestas, uma aresta é definida por dois vértices. Essa hierarquia faz parte do modelo e deve ser respeitada no traçado do objeto, quando o modelo é utilizado. Por exemplo, o apagamento de um vértice com adjacências deverá ser bloqueado ou deverá causar o apagamento das arestas adjacentes. O mesmo ocorrendo se estas pertencerem a fronteiras de faces. Softwares desenhadores, como por exemplo o AUTOCAD da SoftDesk, não trabalham com este modelo

matemático. Neles não existe hierarquia entre essas primitivas, a movimentação de um vértice não alteraria em nada arestas que estivessem geometricamente posicionadas como adjacentes.

Observe-se a incoerência que ocorre na manipulação do traçado, respeitando-se o comportamento do modelo de engenharia. Pelo modelo devem ser respeitados elementos de hierarquia superior e elementos adjacentes de mesma hierarquia. Isto é, uma aresta não pode ser apagada se estiver delimitando uma face, e uma face não pode ser apagada se houverem faces adjacentes, porque serão destruídas. No entanto, a face é definida pelo traçado de arestas e desfeita pelo apagamento de arestas. Para se alterar uma face traçada é necessário alterar a aresta. Se o modelo semântico fosse seguido à risca, não seria permitido mexer na aresta porque estaria sendo alterado um elemento de hierarquia superior. Trata-se de uma situação de *deadlock*. A situação só seria rompida se toda a face fosse apagada, o que também não seria permitido se houvesse faces adjacentes.

Para que essa situação seja rompida é necessário que o modelo de interação seja flexibilizado. Como veremos mais à frente, nem toda manipulação executável obedece a hierarquia do modelo, em função disto surgem os conflitos, que se não forem tratados de forma metódica, terminam por gerar uma série de soluções totalmente circunstanciais.

6.2.2. Especificação do Comportamento dos Modelos

As regras de comportamento dos modelos de interação são especificadas através da gramática de tarefas. A seguir estão especificados três dicionários de tarefas simples. O primeiro para um MSiO de interação, com base no modelo de subdivisão planar, os outros dois para um MSeO de interação, com base nos ME e o MEF. Para efeito de análise, a dicionarização de tarefas simples sobre a primitiva aresta está resumida às tarefas básicas. Por exemplo, para a operação de criar aresta o sistema pode oferecer a possibilidade de criar uma linha, uma polilinha, uma curva de Bezier, e outras. Estamos considerando apenas a criação de linhas. Também com relação às operações de movimentação de arestas, pode ser feita uma movimentação ao longo de uma linha, em torno de um eixo, e outras. Estamos considerando apenas uma movimentação simples.

Na especificação da TAG também não constam os parâmetros necessários às operações. considerou-se que não eram importantes para o objetivo da análise. Seriam um excesso de informação desnecessária. Portanto, foram dicionarizadas apenas as tarefas simples essenciais e foram definidas apenas as regras de comportamento relacionadas às operações em estudo.

O símbolo #, no dicionário, indica que uma tarefa simples só pode ser ativada pelo próprio sistema, não pelo usuário. Por exemplo, a tarefa de definir face em uma interação com orientação semântica é executada automaticamente pelo sistema. O usuário não define face, ele traça arestas e o sistema reconhece a criação de uma face quando uma determinada superfície no plano é delimitada por um conjunto de arestas. O mesmo ocorre com a operação de destruir face, quando uma aresta é apagada o sistema detecta a destruição da face.

O símbolo <> indica que o comportamento especificado é condicional, a operação só pode ser executada se determinada condição for satisfeita. Por exemplo, uma operação de apagar aresta só pode ser executada se a aresta não tiver adjacências. Se a aresta estiver delimitando uma região, seu apagamento irá destruir a região, e com isto estará ferindo sua classe de comportamento.

O símbolo ^ equivale a um conectivo "e", e significa que os dois procedimentos são executados em sequência. O operando n, assinalado após a tarefa, reflete o número de iterações da operação, onde 0 < n <= m. Por exemplo, uma operação de criar aresta ou apagar aresta pode ser executada n vezes para que uma face seja definida ou destruída. O valor de m corresponde ao número de arestas do polígono.

Dicionário de Tarefas Simples

MSiO	MSeO para um ME	MSeO para MEF
	#(definir, face)	
	#(destruir, face)	
	#(deformar, face)	#(deformar, face)
(criar, aresta)	(criar, aresta)	
(apagar, aresta)	<>(apagar, aresta)	
(mover, aresta)	<>(mover, aresta)	
	(criar, vértice)	
	<>(apagar, vértice)	
	<>(mover, vértice)	<>(mover, vértice)

Regras da Gramática

Criar região ---> (criar, aresta)n
Eliminar região ---> (apagar, aresta)n
Desfazer região ---> (apagar, aresta)n
Transformar em região única ---> (apagar, aresta)n
Alterar região ---> (apagar, aresta)n ^ (criar, aresta)n

Alterar região ---> (mover, aresta)n
Vazar face ---> não tem tradução

Uma rápida análise comparativa das tarefas simples dicionarizadas para os três modelos evidencia a flexibilidade do MSiO contra a rigidez do MSeO.

6.3. Análise sobre a Gramática Especificada

Em termos de tarefa conceitual do usuário, ele raciocina tanto no MV como no MI. Se ele está na etapa de construção da geometria do OE, sem definição das propriedades físicas, é natural que ele raciocine em termos de faces e arestas. A partir do momento que começa a associar atributos à geometria, é natural que ele passe a raciocinar em termos de regiões e fronteiras. Portanto, como a TAG reflete a tarefa conceitual do usuário, nas regras da gramática deveria ser especificada uma regra para cada modelo, por exemplo "apagar face" e "apagar região", mesmo que elas impliquem na mesma operação básica. Por questão de simplificação especificamos apenas uma.

A análise sobre a gramática confirma algumas coisas já observadas no estudo de caso e aponta outras.

- Não existe uma gramática padronizada. Uma tradução um para um entre a intenção do usuário e a operação do sistema. Várias operações conceituais são traduzidas na mesma operação do sistema. Em sistemas sem manipulação direta é possível oferecer ao usuário uma ativação da operação correspondente à sua operação conceitual, assim a TAG oferece de fato um mapeamento entre a operação conceitual e a tarefa simples dicionarizada, e o sistema tem a informação da intenção do usuário. No sistema por manipulação direta, em alguns casos, quando o usuário ativa uma operação diretamente sobre a representação gráfica, ele já está ativando a tarefa básica. O sistema não tem referência sobre a intenção do usuário. A Tabela 7 mostra as diferentes tarefas conceituais que são mapeadas numa mesma operação básica.

Intenção do Usuário	Operação Simples
Criar aresta Criar região	criar-aresta
Apagar aresta Eliminar região Desfazer região Transformar região Alterar região	apagar-aresta
Alterar região Mover região	mover-aresta

Tabela 7. Mapeamento entre as intenções do usuário e as operações simples dicionarizadas

- Uma mesma operação conceitual pode ser executada de formas diferentes. Por exemplo, para eliminar uma face sem adjacências o usuário tanto pode raciocinar em termos de apagar a face, como pode apagar as arestas, uma vez que a face é delimitada por arestas. As operações sobre faces transformam-se em operações sobre arestas no sistema. A Tabela 8 ilustra essa relação.

Intenção do Usuário	Operação Simples
Alterar região	apagar-aresta mover-aresta

Tabela 8. Mapeamento entre as intenções do usuário e as operações simples dicionarizadas

- No MSiO todas as operações conceituais encontram tradução livre.
- No ME todas as operações de apagar e mover são condicionadas à presença de primitivas de hierarquia superior ou de mesmo nível adjacentes.
- No MEF só a operação de mover aresta encontra tradução.
- A operação de vazar face não encontra tradução em nenhum modelo. Esta questão já foi discutida no estudo de caso.

6.4. Desenvolvimento das Heurísticas

O objetivo da heurística é flexibilizar a manipulação nos casos em que a operação agride as classes de comportamento do modelo, mas podem ser executadas desde que atendendo a determinadas condições. Portanto a idéia do modelo estendido é que as tarefas simples dicionarizadas sujeitas a condicões, ao invés de bloquearem a operação nos casos em que

agridem a hierarquia, passem o controle para a heurística, de modo que o sistema possa tentar um monitoramento da operação. Em função disto, as heurísticas se concentram em torno das operações apagar e mover. O monitoramento feito pela heurística terá por base as possíveis intenções do usuário, em cada caso, detectadas através da TAG.

6.4.1. Problemas com Relação à Operação de Apagamento

Com relação à operação de apagamento, as restrições existentes são ligados ao comportamento hierárquico das primitivas, ou seja, a possibilidade de apagar a hierarquia superior. No caso de apagamento de face, o problema com relação ao modelo puro é que teoricamente destruir uma face destrói as faces adjacentes. Mas não há problema em que isto ocorra desde que o traçado seja recuperado nos casos de inconsistência. A Figura 14 ilustra diferentes situações de apagamento de face.

No caso (a) a representação do OE não perde a consistência. Nos casos (b) e (c) o usuário precisa um certo número de passos para executar a operação, após o apagamento da face ele precisa de tempo para recuperar a consistência da representação do OE. O caso (b) é um exemplo de apagamento de face interna, onde o usuário deve recuperar as faces adjacentes. O caso (c) é um apagamento de face externa (no limite da geometria).

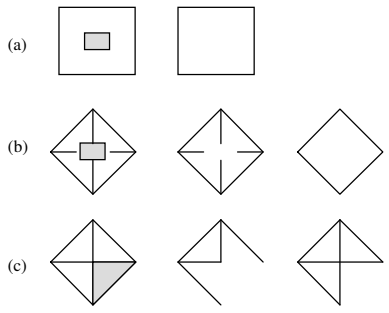


Figura 14. Sequências de estado na manipulação do traçado, em operações de apagamento de faces

Havendo atributos associados à região (material) o sistema deverá questionar o usuário. Em (a) o sistema pode inferir que a área externa invade a área apagada, em (b) é necessário que o usuário informe a situação dos atributos uma vez que cinco regiões foram transformadas em uma.

Nos exemplos apresentados a operação teria sido bloqueada se executada com base no modelo puro, mas pode ser aceita, nas seguintes condições:

- se houver somente uma face externa;
- se o usuário tiver tempo de recuperar o traçado;
- se o sistema puder recuperar a informação semântica (atributo associado);
- se o sistema questionar o usuário quanto à informação semântica.

A heurística sugerida para operações de apagamento seria a seguinte.

Apagamento de face.

Executa apagamento da face.
Se há apenas uma face adjacente
    Recupera informação topológica
    Recupera informação semântica
senão
    Aguarda *n* passos
    Se geometria/topologia foi recuperada
        Questiona usuário sobre informação semântica
        Recupera informação semântica
    senão
        Acusa inconsistência
        Retorna à situação inicial.

O valor de *n* é relativo à capacidade da estrutura de dados na qual é armazenada a sequência de operações. O objetivo é permitir uma operação de *Undo* caso, no limite de armazenamento dessa estrutura de dados, a consistência do traçado não tenha sido recuperada. O valor *n* também pode ser customizado, sendo definido abaixo da capacidade da estrutura de armazenamento das operações, como o número mínimo de passos necessários à reavaliação da semântica.

No apagamento de arestas, o problema com relação ao modelo puro são os casos em que a aresta delimita uma face. No entanto, nem todo apagamento de aresta causa inconsistência. A Figura 15 ilustra situações de apagamento de arestas.

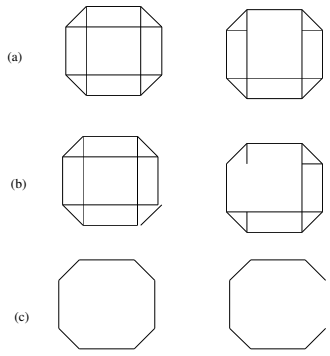


Figura 15. Sequência de estado na manipulação do traçado, em operações de apagamento de arestas

Em (a) a informação topológica pode ser recuperada porque as arestas en torno daquela apagada formam uma nova face. Quando isto acontece, segundo a modelagem de tarefas, o sistema pode inferir que o usuário está tentando alguma operação como alterar região ou eliminar região. Em (b) e (c) será necessário aguardar alguns passos para que o usuário complete a operação desejada, devolvendo a consistência ao traçado. Nestes casos não é possível ter nenhuma inferência sobre a operação desejada.

Observe-se que para executar qualquer uma das operações na Figura 14 o usuário pode ativar apagamento de face ou apagamento de aresta, assinalando as quatro arestas. Portanto, para toda operação sobre aresta é necessário que o sistema faça a verificação em termos de faces e faces adjacentes.

Havendo atributos associados o sistema deverá questionar o usuário. Nos exemplos apresentados a operação teria sido bloqueada se executada com base no modelo puro, mas pode ser aceita nas seguintes condições:

- se o sistema puder recuperar a informação topológica
- se o usuário tiver tempo de recuperar o traçado;
- se o sistema questionar o usuário quanto à informação semântica.

A heurística sugerida para a operação de apagamento de arestas é:

Apagamento de aresta.

Executa apagamento da aresta.
Se aresta não delimita região
    Fim
senão

```
Se está apagando toda a fronteira da região
        Executa heurística de apagamento de face
    senão
        Se as arestas em torno formam face
            Recupera informação topológica
            Questiona usuário sobre informação semântica
            Recupera informação semântica
        senão
            Aguarda n número de passos
            Se consistência recuperada
                Questiona usuário sobre informação semântica
                Recupera informação semântica
            senão
                Acusa inconsistência
                Retorna à situação inicial.
```

6.4.2. Problemas com Relação à Operação de Movimentação

Com relação à operação de mover aresta surgem duas questões: carregar a hierarquia inferior, agredir a hierarquia superior. Sendo que veremos mais à frente que a questão de fato problemática é aquela ligada à hierarquia. Com relação à primeira, significa que ao mover uma aresta o sistema pode carregar ou não as arestas adjacentes. Da mesma forma, ao mover a face o sistema pode carregar ou não as faces adjacentes. Se as faces adjacentes não forem carregadas deverá ser feita uma duplicação das arestas comuns.

Por exemplo, observe-se as operações de movimentação da aresta w, na Figura 16. Se a aresta é o elemento de manipulação é possível movê-la criando novos vértices de apoio na posição desejada, destruindo ou gerando novas faces em função deste movimento, conforme Figura 16(b). Observe que se o usuário quiser "cortar as pontinhas" das arestas adjacentes àquela movimentada, ele terá de selecioná-las e apagá-las. Algumas vezes o usuário deseja esse tipo de efeito para, numa operação subsequente, traçar uma nova aresta complementando a geometria do objeto, conforme Figura 16(c). Porém suponha-se o movimento da aresta no sentido contrário, obedecendo o mesmo critério, ver Figura 16(d). Esse tipo de manipulação parece desnecessária no contexto da aplicação.

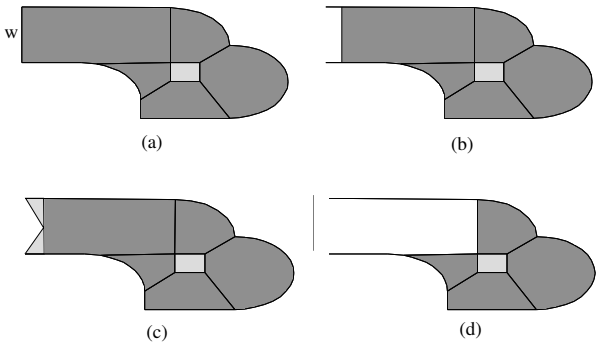


Figura 16. Movimentação de arestas

Na prática a solução anterior não permite ao usuário um efeito bastante utilizado, o de alongar e encurtar a representação gráfica do objeto através do movimento de arestas. Para que este efeito aconteça, a movimentação deverá carregar consigo vértices e arestas adjacentes. Quando a hierarquia inferior é carrregada não se altera a informação topológica. Ambas as soluções estão corretas, nehuma delas fere explicitamente o modelo conceitual, e ambas são úteis. A solução adotada deverá ser aquela que faz sentido com relação à intenção do usuário, portanto, com base nessa utilidade o projetista pode optar por oferecer as duas soluções.

Na Figura 17 são apresentados alguns exemplos de movimentação de arestas e faces, onde são carregados os níveis inferiores de hierarquia.

As Figuras (a) e (b) são exemplos de movimentação de aresta onde o carregamento dos níveis inferiores é bastante útil. Na Figura (a) a aresta inferior que delimita o túnel é movimentada carregando as arestas adjacentes, causando um efeito de alongamento na face que representa o túnel. Na Figura (b) a fronteira que delimita a região foi movimentada carregando as arestas adjacentes. O que faz com que as dimensões da região e da região adjacente sejam redefinidas.

No caso de movimentação de face a utilidade da operação pode ser questionada, uma vez que não causa nenhuma alteração na face movimentada, causando alteração em todas as faces adjacentes.

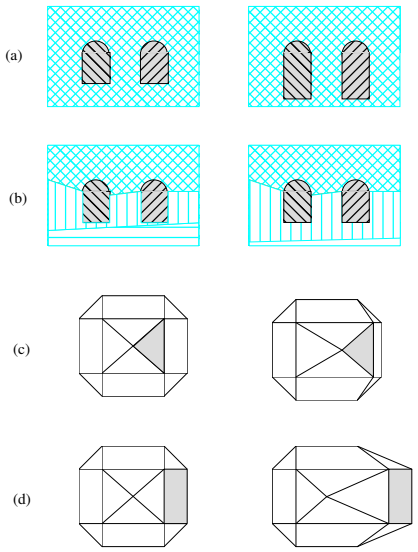


Figura 17. Sequências de estado na manipulação do traçado, em operações de movimentação, com carregamento de hierarquia inferior

A Figura 18 apresenta os mesmos exemplos sendo que não são carregados os níveis inferiores da hierarquia.

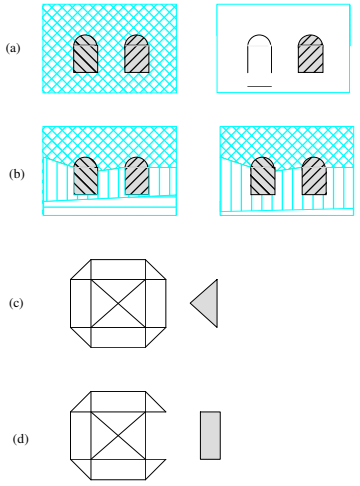


Figura 18. Sequências de estado na manipulação do traçado, em operações de movimentação, sem carregamento de hierarquia inferior

No caso da movimentação de arestas identifica-se pela TAG que o usuário pode estar querendo mover aresta ou alterar a região. Neste caso as duas soluções devem ser preservadas. No caso da

movimentação de faces, não existe nenhum indício de que uma ou outra solução seja mais útil ao usuário.

O problema maior com relação à movimentação de arestas ou faces é ferir a hierarquia, e este é um problema de certa forma incontrolável. No caso de apagamento o efeito era controlável porque o apagamento de uma primitiva atingia somente as primitivas adjacentes. No caso da movimentação atinge qualquer ponto da represntação do OE. A Figura 19 ilustra o problema através de exemplos semelhantes aos já apresentados. Em todos os exemplos o carregamento da aresta invadiu a região adjacente. Nas Figuras (c) e (d) o efeito é incontrolável, até mesmo visualmente.

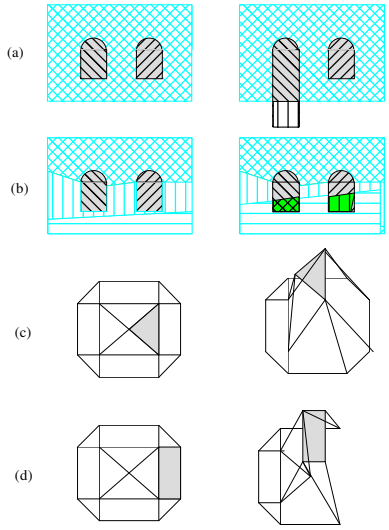


Figura 19. Sequências de estado na manipulação do traçado, em operações de movimentação, com carregamento de hierarquia inferior

A operação de movimentação é excessivamente flexível, principalmente se considerarmos todas as possibilidades de movimentação que podem ser oferecidas ao usuário: ao longo de uma linha, em torno de um eixo, espelhamento, e outras. O modelo sintático como é implementado hoje, já é por si só um modelo estendido. A representação gráfica do OE é um traçado de sub-divisão planar. As operações básicas que atendem à metáfora pura do modelo de subdivisão planar são criar arestas e apagar arestas. Através dessas duas operações toda a sub-divisão planar poderia ser executada. Se um desenhista estivesse traçando uma sub-divisão do plano em papel, ele não teria como movimentar um traço, ele teria que apagá-lo e traçar outro. Portanto, quando o sistema incorpora operações como mover, copiar e deformar arestas, ele já está acrescentando ao seu

modelo sintático operações típicas do recurso computacional, e não do modelo de sub-divisão planar. Essa operações causam um excesso de flexibilidade na manipulação do traçado, que quando utilizadas podem causar problemas.

Neste caso a heurística poderia bloquear as operações com carregamento que invadissem primitivas adjacentes ou hierarquia superior. E para as operações de movimento sem carregamento poderia oferecer um tempo ao usuário para recuperação do traçado.

No caso de inclusão da operação de carregamento a TAG deverá receber os seguintes acréscimos:

Dicionário de Tarefas Simples

<>(carregar,aresta)

Regras da Gramática

Mover aresta com alongamento ---> (carregar,aresta)

Mover região com alongamento ---> (carregar, aresta)

6.4.3. Problemas com Relação à Operação de Vazar Face

Como foi dito o modelo de sub-divisão planar não oferece suporte ao traçado do furo. Em geral, o artifício utilizado é associar à região um atributo de furo. A proposta neste caso é que o modelo de interação mascare esse artifício para o usuário. Para isto basta acrescentar à gramática a operação de vazar-face, remetendo-a a uma operação de associação de atributo. Do ponto de vista do retorno visual a situação não se altera. Independente da forma de ativação o retorno visual é o mesmo, a parte interna da face muda de cor para indicar que está vazada.

No caso de inclusão da operação de vazar face a TAG deverá receber os seguintes acréscimos:

Dicionário de Tarefas Simples

(associar,"furo",face)

Regras da Gramática

Vazar face ---> (associar,"furo",face)

7. Conclusões

Do ponto de vista teórico, através desta solução aumentou-se o espectro de comunicação do usuário com o sistema. No ambiente real de utilização há situações em que o usuário representa e há situações em que ele constrói a representação gráfica do OE. A representação é um procedimento sintaticamente orientado, que oferece flexibilidade na manipulação da representação visual; a construção é um procedimento semanticamente orientado, que garante a consistência dos dados capturados. O Modelo Sintático Estendido não prende o usuário a uma determinada orientação, ao contrário, lhe permite operar em uma faixa de manipulação entre sintático e semântico, usufruindo das vantagens que cada orientação oferece.

O Modelo Sintático Estendido é especificado com base em heurísticas, que por sua vez baseiam-se na modelagem de tarefa do usuário. A modelagem de tarefa é uma referência segura, na medida em que encaminha o comportamento do sistema em direção à expectativa do usuário. O Modelo Estendido aumenta a força de expressão da linguagem de comunicação, resolvendo os conflitos decorrentes de uma orientação puramente sintática ou puramente semântica, e criando regras de comportamento para um modelo intermediário. As heurísticas não substituem a aplicação do modelo, mas, ao contrário, a complementam. Por exemplo, através delas se viabiliza a utilização plena do modelo semântico, contornando as situações de conflito.

No estudo de caso não foram especificados todos os conflitos existentes nestas interfaces. Porém, através da especificação da TAG e das heurísticas foi apresentada uma metodologia para avaliação do problema e especificação dos modelos de interação estendidos para interfaces. Obviamente, deve ser especificada uma TAG completa para o sistema e as heurísticas devem ser expandidas para dar conta de situações mais detalhadas.

Quanto ao aspecto metodológico de desenvolvimento da interface, a utilização da TAG por si só já é lucrativa, pois gera uma especificação formal da interface projetada. Através da TAG evidenciam-se classes de problemas, que propiciam a busca de soluções globais.

Quanto ao aspecto de qualidade da interação usuário-sistema, a heurística contorna os problemas que no modelo puro simplesmente causariam o bloqueio da operação, chegando a soluções satisfatórias na grande maioria dos casos. A regra de comportamento gerada através da heurística oferece ao usuário um sentido de coerência nas soluções adotadas, que lhe permite inferir o comportamento do sistema em situações em que não domina. Esta capacidade de inferência agiliza o trabalho do usuário experiente e permite o aprendizado mais rápido e eficiente para o usuário leigo. A interação com o sistema torna-se mais confortável e mais eficiente.

Como foi dito no início, o problema avaliado para a classe de sistemas pré-processadores pode ser estendido para outros sistemas que atendam às mesmas premissas. Em termos de continuidade do trabalho, o Modelo Sintático Estendido é a base para a geração de uma linguagem visual de comunicação com mais alto poder de expressão.

Bibliografia

ANSYS - User's Manual - Versão 5.0 - Swanson Analysis Systems, Dez/1992.

B-Ed - Building Editor Versão 1.1 - Manual do Usuário. TeCGraf - Grupo de Tecnologia em Computação Gráfica, Fundação Padre Leonel Franca, PUC-Rio, Convênio CENPES/PETROBRÁS, 1992.

Chen, Michael & Mountford, S.Joy & Sellen, Abigail **A Study of Interactive 3-D Rotation Using2-D Control Devices** in *Proceedings of SIGGRAPH'88*, Atlanta, Georgia, 1-5 Ago, 1988. Computer Grafics. Ed. ACM SIGGRAPH. Vol 22, No 4, Ago/1988.

Cordovez, J.C.Gortaire **Consistência Geométrica e Recurso de "Undo" no Pré-Processamento de Edifícios.** Dissertação de Mestrado, Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro, Nov/91.

Figueiredo, L.H. & Souza, C.S. & Gattass, M. & Coelho, L.C.G. **Geração de Interfaces para Captura de Dados sobre Desenhos.** Anais do SIBGRAPI'92. SBC-INPE. Pag 167-175. Águas de Lindóia, SP, 10-12 Nov/1992.

George, P.L. **Automatic Mesh Generation - Application to Finite Element Methods**. John Wiley & Sons, 1991.

Houde, Stephanie **Iterative Design of an Interface for Easy 3-D Direct Manipulation**. *Proceedings of CHI'92*. Pag 135-142. 3-7 May/1992.

Irby, Bergsteinsson, Moran, Newman, Tesler. **A Methodology for User Interface Design**. Xerox 1977.

Gattass, Marcelo & Celes, Waldemar C. Filho & Fonseca, Guilherme Leite **Computação Gráfica Aplicada aos Métodos de Elementos Finitos**. XIV Congresso Nacional de Matemática Aplicada e Computacional - CNMAC - SBMAC - Minicurso. Nova Friburgo, RJ. Set/1991.

Hutchins, E.L. & Hollan, J.D. & Norman, D.A. **Direct Manipulation Interfaces** in [Norman & Draper (1986)], 87-124.

Kammersgaard, J. **Four Different Perspectives on Human-Computer Interaction**. *International Journal of Man-Machine Studies* (1988) 28, pp 343-362.

MTool - Bidimensional Mesh-Tool - Versão 1.0 - Manual do Usuário. Grupo de Tecnologia em Computação Gráfica - TeCGraf/PUC-Rio, Grupo de Geotecnia do SEDEM/DIPREX, Convênio CENPES-PETROBRÁS/PUC-Rio, Maio/1992.

Norman, D.A. **Cognitive Engineering.** in [Norman & Drapper (1986)], 31-61.

Norman, D.A. & Draper, S.W. **User Centered System Design**. Hillsdale, NJ. Lawrence Erlbaum Associates. 1986.

Paulino,G.H.P. e Gattass, M. **Pré-Processamento de Estruturas Espaciais com Reordenação Nodal Usando Computação Gráfica Interativa.** RI 04/89, Departamento de Engenharia Civil. PUC-Rio, 04/89.

Payne, Stephen J. & Green, T.R.G. **Task-Action Grammars: A Model of the Mental Representation of Task Languages.** *Human-computer Interaction*, Volume 2, pp 93-133, 1986.

Requicha, Aristides A. G. **Representations for Rigid Solids: Theory, Methods, and Systems** in *Computing Surveys*, Vol 12, No. 4, pag 437-465. Dez/1980.

Shneiderman, B. **Direct Manipulation: a Step Beyond programming Languages** *IEEE Computer* 16(8), 57-69, 1983.

Souza, C.S. **The Semiotic Engineering of User Interface Languages**. in *International Journal of Man Machine Studies*. 92017, revised version,. ed. B.R. Gaines & D.R. Hill, 1992.