

Gerenciamento de Subdivisões Planares Hierárquicas

WALDEMAR CELES FILHO¹
MARCELO GATTASS¹
PAULO CEZAR PINTO CARVALHO²

¹ TeCGraf-Grupo de Tecnologia em Computação Gráfica
PUC-Rio-Pontificia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 255
22453-900 Rio de Janeiro, RJ, Brasil
celes, gattass@icad.puc-rio.br

² IMPA-Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110
22460-320 Rio de Janeiro, RJ, Brasil
pcezar@visgrafimpa.br

Abstract. Topological data structures have been used to support applications that work with planar subdivisions. Many of these applications require an hierarchical representation not supported by the usual data structures. This paper proposes a topological data structure to represent hierarchical planar subdivisions. A set of operators needed to model such subdivisions is presented and discussed.

Introdução

Diversas aplicações em Engenharia, Mineração, Cartografia e Geologia trabalham com representações que são essencialmente subdivisões do espaço, sobre as quais atribuem-se propriedades pertinentes. Como exemplos destas aplicações, podemos citar geradores de malhas de elementos finitos, simuladores de escavação subterrânea, simuladores de transformações geológicas, representação de mapas cartográficos, e sistemas de informações geográficas.

Estas representações têm sido implementadas com estruturas de dados topológicas, originadas na área de modelagem de sólidos. Para os casos bidimensionais, as estruturas originalmente elaboradas para a representação da fronteira de sólidos *manifold* vêm sendo usadas com sucesso, interpretando o modelo como o grafo planar associado a um sólido 3D (Wawrzynek, 1987). Nos casos tridimensionais, estruturas mais complexas são necessárias, pois estas devem suportar representações *non-manifold* (Martha, 1989).

Além da característica comum de modelagem através de subdivisões do espaço, observa-se, nestas aplicações, a freqüente necessidade de representação de modelos hierárquicos. Não raro, estes sistemas requerem uma disposição hierárquica das subdivisões: em mapas cartográficos, as fronteiras dos estados são

hierarquicamente inferiores às fronteiras dos países; geradores de malhas de elementos finitos subdividem o modelo em regiões, sobre as quais se aplicam os métodos de discretização do domínio; simuladores de transformações geológicas agrupam diversas camadas litológicas em módulos que viabilizem as simulações.

As implementações de sistemas similares, com estruturas de dados topológicas convencionais, criam artifícios para garantir a representação de hierarquias. Por serem empregados sobre estruturas originalmente elaboradas para suportar modelos não hierárquicos, estes artifícios tornam ainda mais complexas estas implementações, tendo em vista as dificuldades já inerentes ao tratamento de estruturas topológicas. Como consequência, a consistência de modelos hierárquicos recai, invariavelmente, em códigos de difícil entendimento e manutenção.

Neste estudo, apresentamos uma estrutura de dados topológica que suporta representações hierárquicas de subdivisões planares. Com isto, acreditamos obter uma base adequada para o desenvolvimento de sistemas como os citados acima.

Hierarquia com estruturas convencionais

Nesta seção, discute-se a representação de subdivisões hierárquicas com a utilização das estruturas topológicas convencionais. Analisam-se duas propostas, ambas já

empregadas no desenvolvimento de aplicações (com uso industrial) para geração de malhas de elementos finitos.

O emprego do método dos elementos finitos, nas simulações numéricas da área de engenharia, exige que o domínio do problema seja discretizado em pequenos (finitos) elementos (Gattass et al., 1991). Muitos dos métodos empregados para a geração automática desta discretização impõem restrições topológicas às áreas de mapeamento. Mapeamentos transfinitos, por exemplo, só podem ser empregados, nos casos 2D, em regiões topologicamente equivalentes a triângulos ou retângulos.

Estas restrições topológicas, associadas à própria natureza heterogênea dos modelos (representação de diferentes materiais e/ou propriedades físicas, geométricas e analíticas), caracterizam diferentes níveis hierárquicos de subdivisão. O modelo é constituído por diferentes regiões (com suas respectivas propriedades), que são subdivididas em sub-regiões para viabilização dos mapeamentos. Estes mapeamentos discretizam as sub-regiões em elementos finitos.

Nos trabalhos de Martha (1989) e Campos (1991), o modelo para geração da malha de elementos finitos é dividido em quatro níveis (considerando o caso 2D). O primeiro nível contém o modelo geométrico, representando as diferentes regiões do domínio. No próximo nível, encontra-se o modelo de sub-regiões formado pelas subdivisões necessárias para emprego dos mapeamentos. No terceiro nível, encontra-se o nível das subdivisões das arestas das sub-regiões; por fim, o modelo da malha propriamente dita, formando o quarto nível hierárquico (Figura 1).

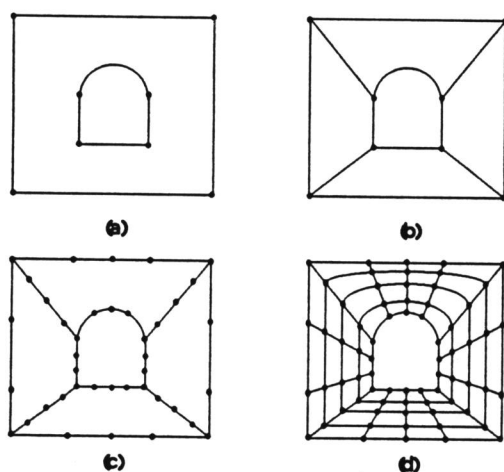


Figura 1: Níveis hierárquicos dos modelos de elementos finitos: (a) geometria; (b) sub-regiões; (c) subdivisões; (d) malha (Carvalho et al., 1990).

Para viabilizar a representação computacional destes modelos, Martha e Campos utilizaram a estrutura de dados topológica *face-edge* (*radial-edge* no caso 3D), inicialmente apresentada por Weiler (1986). A estrutura *face-edge* foi projetada para a representação de sólidos *manifold*. Trabalhando sobre a representação do grafo planar destes sólidos, consegue-se, com esta estrutura e similares, suporte para subdivisões planares e, portanto, para modelos bidimensionais de malhas de elementos finitos (Wawrzynek, 1987). Para cada nível hierárquico proposto por Martha e Campos, existe uma estrutura de dados associada. Como estes quatro níveis guardam entre si uma correspondência hierárquica (todas as entidades topológicas de níveis superiores devem, necessariamente, estar contidos nos níveis inferiores), as quatro estruturas de dados não podem ser independentes. Para garantir esta correspondência, criaram-se ponteiros entre entidades dos níveis hierárquicos adjacentes. Uma aresta do nível das sub-regiões, por exemplo, tem um ponteiro que indica qual a aresta correspondente no nível da geometria, a qual pode ou não existir. Com esta amarração, qualquer evento que ocorre em um determinado nível pode ser reportado aos demais, sendo portanto possível a manutenção da consistência das estruturas.

A evolução deste modelo eliminou o nível das subdivisões, tratando-as apenas como atributos das fronteiras das sub-regiões, tendo em vista a minimização do espaço de memória necessário para seu armazenamento (Vianna, 1992).

A estruturação dos dados nessas propostas trouxe uma significativa contribuição na concepção da arquitetura dos sistemas de geração de malhas, enfatizando as diversas etapas na construção do modelo final (Carvalho et al., 1990). Do ponto de vista da estruturação dos dados, no entanto, essas propostas apresentam dois aspectos negativos. O primeiro diz respeito à duplicação de representações de entidades topológicas idênticas em diversos níveis hierárquicos, tendo em vista que os modelos inferiores englobam, obrigatoriamente, os modelos superiores.

Um segundo aspecto mais importante está no grau de complexidade da manutenção da consistência. Embora cada nível seja representado por uma estrutura de dados já consagrada na área de modelagem e, portanto, bastante conhecida, a interligação dos diversos níveis, via ponteiros cruzados, exige uma implementação complexa, já que não se tem uma única base de dados para a representação do modelo como um todo. Com isto, perde-se o grande benefício do uso dos operadores de Euler para edição das subdivisões planares, pois eles não garantem a consistência entre os níveis hierárquicos.

Baseado nestas experiências, e buscando atender a um pedido do Centro de Pesquisa da PETROBRÁS (CENPES) para o desenvolvimento de um gerador de malhas bidimensionais, propusemos e implementamos o modelo do gerador de malhas *mtool* (MTool, 1992). Neste desenvolvimento, procuramos minimizar as dificuldades do tratamento hierárquico, reduzindo a profundidade da hierarquia. Essa proposta simplifica o modelo e mantém um único nível representando a geometria e as sub-regiões, como se as sub-regiões fizessem parte da geometria. O restante é tratado como atributos das entidades topológicas. Assim, subdivisões das arestas são tratadas como atributos das arestas e as malhas geradas são atributos das regiões. Isto é, cada malha de cada sub-região é representada por uma estrutura de dados independente que é atribuída à face do nível da geometria.

O modelo do *mtool*, entretanto, dificulta a compatibilização de malhas em regiões adjacentes. Para tratar esta compatibilidade, o programa faz uso de buscas geométricas, perdendo mais uma vez a garantia de consistência oferecida pelos operadores topológicos.

Portanto, ambas as propostas, embora funcionais e possíveis, apresentam dificuldades devido à ausência de uma formalização da hierarquia que permita o uso de uma única base de dados para a representação do modelo hierárquico como um todo. Uma proposta para o gerenciamento destes modelos hierárquicos é apresentada nas seções subsequentes.

Estrutura de dados topológica hierárquica

A estrutura de dados proposta neste artigo foi projetada para representar subdivisões planares hierárquicas. Por subdivisão planar, entende-se uma partição do plano induzida por um mapeamento de um grafo planar (Preparata-Shamos, 1990), guardando as seguintes propriedades: (i) a interseção de duas arestas ou é vazia ou é um vértice; (ii) a interseção de duas faces ou é vazia ou é um conjunto de vértices ou é um conjunto de arestas e vértices. A Figura 2 mostra um exemplo e um contra-exemplo de uma subdivisão planar.

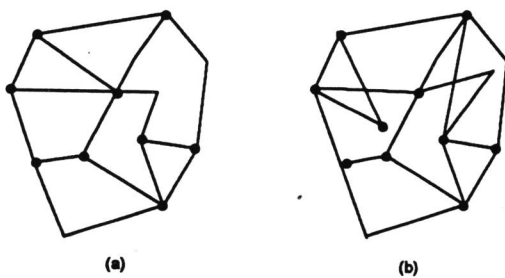


Figura 2: (a) subdivisão planar; (b) partição do plano que não é subdivisão planar.

Múltiplas subdivisões planares podem estar relacionadas hierarquicamente. Nesta hierarquia, toda entidade topológica de um nível inferior está inteiramente contida em uma entidade topológica do nível superior. Assim, o conjunto de faces de um nível inferior que interceptam uma dada face f do nível superior estão contidas em f e induzem uma subdivisão planar de f . As diversas entidades topológicas que compõem um modelo de subdivisões planares hierárquicas podem estar dispostas em diferentes níveis de hierarquia, de tal forma que cada nível deve corresponder a uma subdivisão planar válida, e todas as entidades topológicas de níveis hierárquicos superiores devem estar representadas (subdivididas ou não) nos níveis inferiores.

Seja N um nível na hierarquia de um modelo. Denotamos por N^+ o nível hierárquico imediatamente acima de N e por N^- o nível imediatamente abaixo. Dizemos que uma entidade topológica t é definida no nível N quando t pertence a N e não a N^+ . As entidades topológicas do modelo podem ser definidas em um nível hierárquico qualquer satisfazendo as seguintes propriedades:

- Se f é uma face definida em N , então existe uma face f' pertencente a N^+ que contém f .
- Se f é uma face pertencente a N , então ou f pertence a N^- ou existe um conjunto de faces em N^- cuja união corresponde a f .
- Se e é uma aresta definida em N , então ou existe uma aresta e' pertencente a N^+ que contém e ou existe uma face f' pertencente a N^+ que contém e .
- Se e é uma aresta pertencente a N , então ou e pertence a N^- ou existe um conjunto de arestas em N^- cuja união corresponde a e .
- Se v é um vértice pertencente a N , então v pertence a N^+ .

A Figura 3 ilustra um modelo bastante simples de subdivisões planares com dois níveis hierárquicos. O primeiro nível hierárquico define a fronteira externa do modelo, tendo, portanto, duas faces (a interna, limitada, e a externa, ilimitada), quatro arestas e quatro vértices. O segundo nível subdivide a face interna do primeiro nível em duas outras faces, sendo representado, então, por três faces, sete arestas e seis vértices.

O objetivo da estrutura proposta é armazenar, numa mesma base de dados, o modelo completo, podendo ser identificados os diversos níveis hierárquicos com suas respectivas entidades, sem duplicidade de informações.

Observa-se que, no exemplo, as duas arestas verticais do nível I são mantidas no nível II, enquanto as horizontais são subdivididas. Portanto, para evitar duplicidade, as arestas verticais devem ser as mesmas, tanto no nível I como no nível II. Por outro lado, as

arestas horizontais representadas no nível I não são entidades do nível II, mas aparecem quatro novas arestas inexistentes no nível superior. As entidades armazenadas pela estrutura de dados, para este exemplo, estão identificadas na figura.

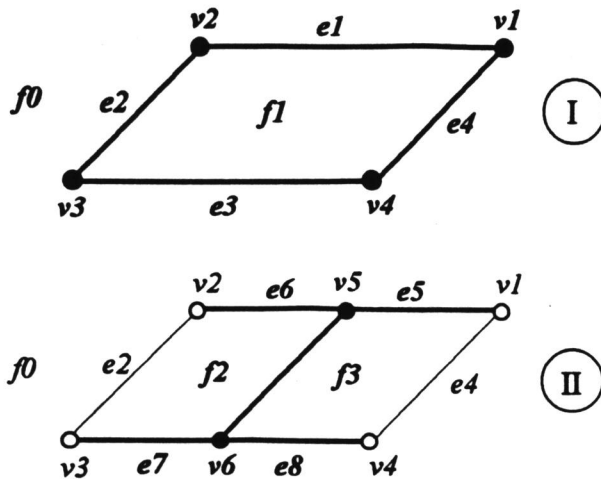


Figura 3: Entidades de um modelo hierárquico.

No nível I, a face $f1$ é delimitada pela seguinte seqüência de vértices e arestas: $v1-e1-v2-e2-v3-e3-v4-e4$. Já no nível II, a face $f2$ é delimitada por $v5-e6-v2-e2-v3-e7-v6-e8$ e a face $f3$ por $v1-e5-v5-e9-v6-e8-v4-e4$.

Além destas informações, a estrutura deve ser capaz, ainda, de identificar que as faces $f2$ e $f3$ são subdivisões da face $f1$, assim como as arestas $e5$, $e6$ e $e7$, $e8$ são subdivisões de $e1$ e $e3$, respectivamente. A representação das hierarquias destas entidades é fundamental e permite implementar mecanismos de herança dentro da estrutura.

Representação interna dos dados

As entidades topológicas presentes nos modelos são dipostas na estrutura de dados seguindo uma estruturação *top-down*. A Figura 4 mostra esta disposição, que é a mesma das estruturas convencionais. No topo, representa-se o modelo, ou uma lista de modelos independentes entre si. Um modelo é constituído por faces, arestas e vértices. As faces são delimitadas por laços (loops) que são definidos por uma seqüência de arestas. Cada aresta é apoiada em dois vértices. Para viabilizar a representação das relações de adjacências, representa-se, explicitamente, os usos de cada aresta (cada aresta tem dois usos) (Mäntylä, 1988).

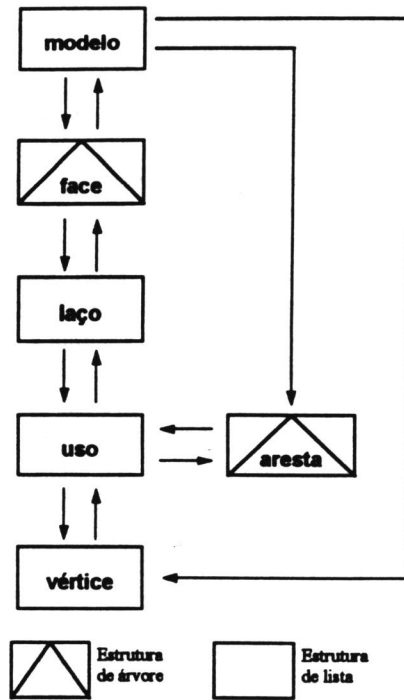


Figura 4: Modelo *top-down* das entidades topológicas.

De modo diferente das estruturas convencionais, as faces e arestas dos modelos são armazenadas em árvores, ao contrário do uso convencional de listas. As estruturas de árvores são necessárias para representar as situações onde faces ou arestas de níveis superiores são subdivididas em níveis inferiores. Para ilustrar, a Figura 5 mostra um esquema da representação interna das entidades do modelo da Figura 3.

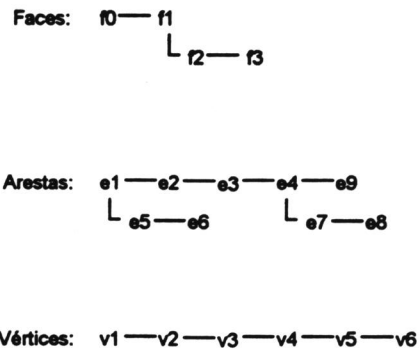


Figura 5: Disposição das entidades em árvores e listas.

Observa-se que arestas de níveis inferiores podem estar representadas de duas formas: quando as entidades surgem devido a subdivisões de arestas do nível superior, são colocadas como “filhas” na árvore; quando não têm correspondência nos níveis superiores (estão contidas em faces superiores), aparecem como

nós primários da árvore do modelo. As faces recebem tratamento análogo, com uma diferença: somente as faces do nível mais superior serão dispostas como nós primários, já que todas as faces inferiores serão, obrigatoriamente, subdivisões de faces já existentes. Os vértices, por serem entidades indivisíveis, aparecem em estruturas de listas convencionais.

Cada entidade topológica (face, aresta ou vértice) tem, ainda, um indicador do nível a que pertence. Este indicador de nível permite dispor as diversas entidades dos diferentes níveis em estruturas de dados comuns. Outro aspecto importante está no fato de não haver duplicação de entidades idênticas em níveis diferentes. É possível, por exemplo, uma determinada face de um nível, digamos nível I, ser subdividida em diversas faces no nível III. A face superior terá faces "filhas" de dois níveis abaixo, indicando que a face considerada no nível intermediário (no caso, o nível II) é idêntica a face superior e, portanto, não representada. Se ocorrer uma operação que atue diretamente sobre a entidade não representada, há necessidade de ser feita uma cópia da entidade até então idêntica. Uma desvantagem desta estruturação está no fato de que pode ser necessário percorrer todas as entidades do modelo para enumerar as entidades de um determinado nível.

Ainda, para viabilizar a obtenção das relações de adjacências, os usos que formam cada laço são dispostos em listas circulares com elos múltiplos. Para exemplificar, considera-se o uso associado à aresta e_2 . Se estiver sendo percorrido o laço associado à face f_1 (nível superior), o próximo uso do laço corresponde ao associado à aresta e_3 (sentido anti-horário). No entanto, se estiver sendo percorrido o laço associado à face f_2 , o próximo uso está associado à aresta e_7 . Portanto, as entidades que representam os usos das arestas têm tantos elos quanto forem os níveis hierárquicos das arestas.

Observa-se que esta estratégia é adequada pois garante elos únicos no nível hierárquico mais inferior, o que é razoável, pois o número de entidades deste nível tende a ser superior aos demais (o nível inferior representa um refinamento dos demais níveis). A seguir, é apresentada uma possível estruturação dos dados relativos às entidades, utilizando a sintaxe da linguagem C.

```
struct modelo
{
  int          maxnivel;          número de níveis
  struct modelo *prox,*ant;      lista de modelos
  struct face  *faces;           árvore de faces
  struct edge  *edges;           árvore de arestas
  struct vert  *verts;          lista de vértices
};
```

```
struct face
{
  int          nivel;            nível da face
  struct face  *prox,*ant;      elos hor. da árvore
  struct face  *pai,*filho;     elos vert. da árvore
  struct modelo *modelo;       modelo a que pertence
  struct laco  *lext;           laço externo
  struct laco  *lacos;         lista de laços internos
};
struct laco
{
  struct laco *prox,*ant;      lista de laços
  struct face *face;          face a que pertence
  struct uso  *uso;           um dos usos associado
  struct vert *vert;         vértice se for isolado
};
struct aresta
{
  int          nivel;            nível da aresta
  struct aresta *prox,*ant;     elos hor. da árvore
  struct aresta *pai,*filho;    elos vert. da árvore
  struct uso    *usol,*uso2;    usos associados
};
struct uso
{
  union
  {
    struct uso **prox;          vetor de elos
    struct uso *prox0;         elo para aresta de nível 0
  };
  union
  {
    struct uso **ant;          vetor de elos
    struct uso *ant0;         elo para aresta de nível 0
  };
  struct laco  *laco;          laço a que pertence
  struct aresta *aresta;      aresta a que pertence
  struct vert  *vert;         vértice origem
};
struct vert
{
  int          nivel;            nível do vertice
  struct vert  *next,*prev;     elos da lista
  union
  {
    struct uso *uso;           uso associado
    struct laco *laco;        laço isolado
  }
};
```

Relações de adjacências

As estruturas de dados topológicas para representação de subdivisões do plano respondem a nove relações de adjacências, combinando as entidades faces, arestas e vértices. Na estrutura proposta, em cada nível, estas informações podem ser conseguidas com buscas locais, já que cada nível é uma subdivisão planar e está devidamente representada.

Além das relações entre as entidades de um mesmo nível, existem as relações entre diferentes níveis hierárquicos. Neste ponto, a estrutura também garante informações suficientes:

- Dada uma face de um nível intermediário, obtém-se diretamente, através da estrutura de árvore, a face do nível superior (em relação a qual a face dada representa uma subdivisão) e as faces do nível inferior (que representam subdivisões da face dada).

- Dada uma aresta de um nível intermediário, encontram-se duas situações em relação ao nível superior: ou a aresta dada representa uma subdivisão de uma aresta superior (neste caso procede-se analogamente como descrito para as faces); ou não se tem representação da aresta dada no nível superior; neste caso, pode-se determinar, através das relações de adjacências das faces vizinhas, em que face superior a aresta está contida. Com relação ao nível inferior, a situação é análoga ao caso das faces.

Operadores básicos

Com a finalidade de garantir a manutenção da consistência durante a edição do modelo, definimos um conjunto de operadores básicos responsáveis pelas alterações na estrutura de dados. Nas estruturas topológicas não hierárquicas, este conjunto de operadores é constituído pelos operadores de Euler (Baumgart, 1972).

Aqui, também continuam aplicáveis os operadores de Euler em cada nível hierárquico. A diferença em relação às estruturas convencionais está no fato de que estes operadores podem atuar em diversos níveis ao mesmo tempo. Por exemplo, um vértice inserido num nível superior também deve pertencer aos níveis inferiores. Observa-se que uma mesma e única entidade é adicionada em diversos níveis (daí a necessidade de haver múltiplos elos nas entidades superiores).

Além dos operadores de Euler, faz-se necessário a utilização de operadores adicionais que viabilizam a troca de nível de atuação destes operadores.

A questão que surge diz respeito à determinação dos níveis em que cada operador deve atuar. Como será mostrado a seguir, os níveis de atuação dos operadores de Euler são implicitamente determinados.

Os operadores de Euler escolhidos para implementação da estrutura foram:

- **mvfm** (*make vertex, face and model*): necessário para a inserção do primeiro vértice do modelo;
- **mvr** (*make vertex and ring*): necessário para inserir um vértice no interior de uma face;
- **mvse** (*make vertex, split edge*): necessário para inserir um vértice sobre uma aresta;
- **mekr** (*make edge, kill ring*): necessário para inserir uma aresta que destrói um anel interno de uma face;
- **mesf** (*make edge, split face*): necessário para inserir uma aresta que divide uma face existente.

A estes operadores aliam-se os respectivos operadores inversos. Neste conjunto de operadores, três estão associados a inserção de vértices no modelo e dois estão associados a inserção de arestas.

Com exceção do operador primário (**mvfm**) que atua em todos os níveis do modelo (e obrigatoriamente é o primeiro operador a ser aplicado), os níveis hierárquicos de atuação de cada um dos demais operadores são determinados pela entidade sobre a qual a inserção será feita. Assim, os operadores **mvr**, **mekr** e **mesf** têm seus níveis de atuação determinados pelo nível da face sobre a qual eles atuam: o nível superior de atuação corresponde ao nível da face (e as entidades criadas terão este nível); o nível inferior de atuação é o nível imediatamente superior ao nível das faces “filhas” da face em questão, ou o nível mais inferior, no caso da face não ter “filhas”.

Analogamente, o operador **mvse** tem seus níveis de atuação determinados pela aresta sobre a qual o vértice será inserido.

Observa-se, ainda, que o fato de forçar a aplicação do operador primário (**mvfm**) em todos os níveis (sendo que as entidades criadas nesta operação são do nível mais superior) garante que todas as faces de níveis inferiores serão obrigatoriamente “filhas” de faces superiores, já que a face inicialmente criada cobre todo o plano.

Para possibilitar a mudança de atuação de níveis, adicionam-se dois operadores: **cf** (*copy face*) e **ce** (*copy edge*). Estes operadores efetuam cópias de entidades superiores em níveis inferiores. Eles são necessários para viabilizar operações em níveis inferiores. Por exemplo, para subdividir uma face no nível inferior, faz-se, inicialmente, uma cópia da face no nível que se deseja atuar e, sobre esta cópia, aplica-se o operador **mesf**. Observa-se que estes operadores (**cf** e **ce**) são os dois únicos operadores para os quais deve-se especificar explicitamente o nível de atuação, isto é, em que nível a cópia será realizada. Paralelamente, existem os operadores inversos que permitem desfazer cópias de entidades idênticas.

Por fim, para possibilitar a inserção de arestas em níveis superiores que cobrem arestas já existentes em níveis inferiores, criou-se o operador **cep** (*change edge parent*) que possibilita redefinir de que aresta uma outra será “filha”. Verifica-se que, antes da operação, as arestas que foram cobertas por uma superior não eram “filhas” de nenhuma outra aresta, e, após a inserção, devem ser consideradas “filhas” da aresta inserida.

Para exemplificar a atuação dos operadores, consideremos o modelo de três níveis hierárquicos ilustrado na Figura 6. Suponhamos que se deseja inserir uma aresta no nível superior, definida pelos vértices v_1 e v_2 . A inserção desta aresta deve ser feita através do operador **mesf** (*make edge, split face*). Este operador atua nos níveis superior e intermediário, pois a face que

será subdividida pertence ao nível superior e tem como “filhas” faces do nível inferior (a mesma face do nível superior pertence também ao nível intermediário). Observa-se que esta inserção de aresta é suportada pelos operadores pois a aresta inserida é totalmente coberta por arestas definidas no nível inferior. Através do operador *cep* (*change edge parent*), atualiza-se a hierarquia das arestas, pois as arestas e_a , e_b e e_c passam a ser “filhas” da nova aresta inserida.

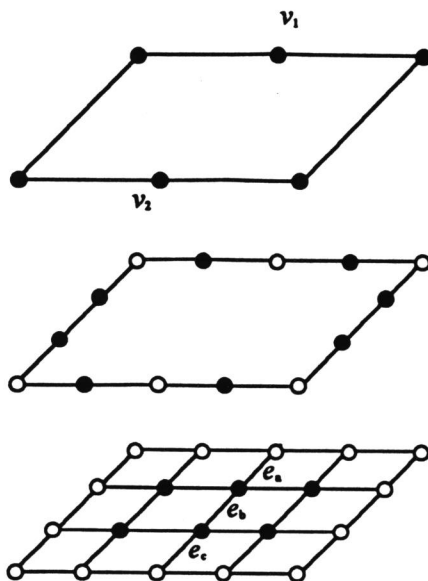


Figura 6: Modelo com três níveis hierárquicos.

Manutenção da estrutura

As estruturas de dados topológicas são próprias para representação de modelos dinâmicos, isto é, modelos que, por agentes externos, sofram alterações. A cada ação externa, deve-se garantir a consistência do modelo, o que pode ser feito através da utilização dos operadores básicos, que têm escopo local devido à disponibilidade das informações de adjacências.

No caso de subdivisões hierárquicas, deve-se definir normas que regulamentam as possíveis modificações. Operações em níveis hierárquicos superiores podem gerar eventos adicionais em níveis inferiores. Por outro lado, determinados eventos em níveis inferiores são inibidos devido a restrições presentes nos níveis superiores.

Estes modelos têm uma forma natural de modelagem que é através da inserção de vértices e arestas que formam a subdivisão planar (Cavalcanti et al., 1991). A disponibilidade destas operações, aliadas às funções destrutivas (operação inversas), propicia um adequado ambiente para construção destas

representações, e a estrutura de dados deve prover suporte para estas implementações.

A regulamentação das normas que regem as operações de modelagem podem seguir estratégias distintas. O conjunto de operadores básicos disponíveis não permite, por exemplo, que um vértice seja criado no interior de uma face que tenha “filhas”, a menos que este vértice já exista numa das faces “filhas”. Situação análoga ocorre com relação à inserção de arestas: uma aresta só pode ser inserida sobre uma face com “filhas” se existir nestas faces “filhas” um conjunto de arestas que cobrem a nova aresta.

Neste ponto, pode-se optar por três estratégias diferentes: na primeira, considera-se que a inserção de uma entidade num nível superior que não respeite as subdivisões inferiores induz a uma destruição das subdivisões inferiores violadas.

Uma segunda estratégia possível admite a inserção de entidades em níveis superiores gerando eventos de inserções análogas nos diversos níveis inferiores.

A terceira respeita as restrições dos operadores e simplesmente impede a inserção.

Para cada uma das estratégias, pode-se enumerar vantagens e desvantagens que dependem da forma de interação da aplicação. Portanto, o gerenciador da estrutura de dados deve prover as três formas, deixando a decisão sobre qual estratégia ser utilizada para a aplicação.

Comparativo de memória

A estrutura proposta permite a representação de subdivisões hierárquicas numa mesma base de dados. Resta-nos fazer uma análise do espaço de memória necessário para esta representação.

Uma primeira análise que deve ser feita consiste na comparação entre a utilização da estrutura hierárquica para representação de uma subdivisão planar simples e a utilização de uma estrutura convencional (sem hierarquia). Neste caso, o espaço adicional gasto pela estrutura proposta consiste nos indicadores de níveis presentes nas entidades face, aresta e vértice, e no armazenamento de arestas e faces em árvores (seria necessário apenas listas). Os elos múltiplos não existem neste caso, já que o único nível existente é também o nível mais inferior no qual são utilizados elos simples.

Uma segunda comparação de gasto de memória possível pode ser feita com relação a simulação de subdivisões hierárquicas através de ponteiros cruzados de subdivisões independentes. Neste caso, o gasto com o armazenamento em estruturas de árvore é comparável ao espaço gasto com os ponteiros cruzados. A diferença existe na representação de elos múltiplos contra a

duplicidade de entidades idênticas. Neste aspecto, pode-se afirmar que o espaço de memória adicional para a representação interna, se existente, está relacionado com o maior ou menor número de entidades idênticas entre as diversas subdivisões.

Salienta-se, no entanto, que o espaço para representação das entidades e seus elos de ligação não constituem a representação da subdivisão como um todo. A cada entidade, associam-se atributos, nem sempre desprezíveis do ponto de vista de espaço de memória, que representam a semântica das aplicações. Neste ponto, o fato da estrutura assegurar a não duplicidade de entidades idênticas pode representar uma economia de memória significativa na representação completa da subdivisão, além de simplificar o gerenciamento dos atributos por parte das aplicações.

Conclusão

As estruturas topológicas têm sido empregadas com sucesso para representação de subdivisões do espaço nas mais diversas áreas. Muitas aplicações apresentam uma natureza hierárquica e a representação desta hierarquia tem sido feita através de artifícios sobre as estruturas topológicas convencionais.

Apresentamos, neste artigo, uma estrutura topológica capaz de representar subdivisões planares hierárquicas, utilizando uma única base de dados. A estrutura proposta busca minimizar a representação do modelo como um todo e impede a duplicidade de informações em níveis hierárquicos distintos.

A disposição das entidades faces e arestas em estruturas de árvores, associada à utilização de listas com múltiplos elos nos encadeamentos dos usos das arestas, possibilita a obtenção das convencionais relações de adjacência em cada nível hierárquico e as relações de adjacência entre entidades de níveis diferentes. Esta base assegura a execução das operações para alteração do modelo utilizando apenas buscas locais.

A proposta apresentada pode ser estendida para subdivisões espaciais, tendo como base estruturas de dados para a representação do contorno (*b-rep*), como a *radial edge*.

Agradecimentos

Este trabalho foi desenvolvido no ICAD - Laboratório de CAD Inteligente da PUC-Rio. O ICAD é apoiado financeiramente pelo TeCGraf que obtém seus recursos através de contratos com a indústria, notadamente o CENPES/PETROBRÁS e o CEPTEL/ELETOBRÁS. Agradecemos a Luiz Henrique de Figueiredo pelas valiosas sugestões na formalização da hierarquia e

organização deste artigo, e a Luiz Fernando Martha que introduziu as idéias que deram origem a este trabalho e muito contribuiu para a difusão delas na PUC-Rio.

Os autores são parcialmente financiados com bolsas de formação e pesquisa da CAPES e do CNPq.

Referências

- B. Baumgart, "Winged-Edge Polyhedron Representation", Stanford Artificial Intelligence, report no. CS-320, 1972.
- J. A. P. de Campos, "Geração de malhas de elementos finitos bidimensionais baseada em uma estrutura de dados topológica", Dissertação de Tese de Mestrado, Departamento de Eng. Civil, PUC-Rio, 1991.
- P. C. P. Carvalho, M. Gattass and L. F. Martha, "A software tool which allows interactive creation planar subdivisions and applications to educational", CATS 90, E. Oñate et al. (Eds.), Barcelona, 1990.
- P. R. Cavalcanti, P. C. P. Carvalho e L. F. Martha, "Criação e Manutenção de Subdivisões Planares", Anais do SIBGRAPI IV, 13-24, 1991.
- M. Gattass, W. Celes Filho e G. L. Fonseca, "Computação gráfica aplicada ao método dos elementos finitos", XIV Congresso Nacional de Matemática Aplicada e Computacional - CNMAC, SBMAC, Nova Friburgo, 1991.
- M. Mäntylä, *An Introduction to Solid Modeling Computer*, Science Press, Rockville, Maryland, 1988.
- L. F. Martha, "Topological and geometrical modeling approach to numerical discretization and arbitrary fracture simulation in three-dimensions", Ph. D. Dissertation, Cornell University, Ithaca, N.Y., 1989.
- MTool - *Bidimensional Mesh Tool*, Manual do Usuário, Grupo de Tecnologia em Computação Gráfica, PUC-Rio, 1992.
- F. P. Preparata and M. Ian Shamos, *Computational Geometry - an introduction*, Springer-Verlag, New York, 1990.
- A. C. Vianna, "Modelagem geométrica para modelos bidimensionais de elementos finitos", Dissertação de Tese de Mestrado, Departamento de Eng. Civil, PUC-Rio, 1991.
- P. A. Wawrzynek, "Interactive finite element of fracture process: an integrated approach", Department of Structural Engineering Report, Cornell University, Ithaca, N. Y., 1987.
- K. Weiler, "Topological structure for geometric modeling", Ph. D. Dissertation, Rensselaer Polytechnic Institute, Troy, N.Y., 1986.