

# Modelação Tridimensional com Restrições Geométricas

José Miguel Losa  
ISEP/INESC  
Pr. Mompilher 22, 4000 Porto, Portugal  
e-mail: losa@asterix.inescn.pt

Vasco Branco  
ISEP/INESC  
e-mail: vab@basinger.inescn.pt

F. Nunes Ferreira  
FEUP/INESC  
e-mail: fnf@porto.inescn.pt

## Sumário

O desenvolvimento de um novo produto é, muitas vezes, realizado a partir de produtos já existentes (*variant design*). Alterações numa peça de um modelo complexo devem ser “inteligentemente” manipuladas pelos sistemas CAD garantindo uma alteração rápida e consistente do modelo.

A modelação baseada em restrições estuda métodos que permitem guardar no modelo informação sobre os seus parâmetros dimensionais assim como restrições geométricas e topológicas entre as várias peças de um conjunto. Deste modo é possível, alterando os seus parâmetros dimensionais, gerar variantes do modelo que mantêm as restrições impostas.

## 1. Introdução

Os sistemas CAD têm ferramentas que permitem restringir entidades, como, por exemplo, traçar uma linha tangente a uma circunferência ou mover um conjunto de objectos. Essas restrições não são explicitamente guardadas na representação do modelo e, por isso, são perdidas quando se altera o modelo. Assim, a manutenção das relações entre as entidades de um modelo são uma tarefa do designer. Por exemplo, se variar o diâmetro de um furo para o alojamento de um parafuso, é da sua responsabilidade alterar o diâmetro do parafuso. Controlar todas as relações entre as partes que formam um modelo é uma actividade morosa, sujeita a erros e pouco interessante.

O conjunto de técnicas conhecido por modelação baseada em restrições tenta oferecer uma solução para este problema. Uma restrição descreve uma relação que deve ser mantida [FrMa 90]. Por exemplo, a restrição de que um objecto deve ser paralelo a



outro ou que uma linha deve ser vertical. Ao impor restrições entre componentes de um modelo complexo é possível gerar, por alterações de alguns parâmetros, e de modo automático, variantes desse modelo, que cumpram essas restrições.

Das várias abordagens estudadas [HyBr, 78], [GoZu 88], [SuAn90], [Roll 91], [Just 92], salientamos o trabalho de van Emmerik [Emme 90], que influenciou o nosso desenvolvimento na aplicação das restrições a modelos 3D.

No entanto o processo de criação de um modelo, no trabalho referido, é um processo pouco natural para um designer, uma vez que este é obrigado a começar por desenhar uma estrutura para o modelo, sem ter os seus componentes como suporte visual. No protótipo implementado, tentámos melhorar o processo de trabalho ao permitir definir o modelo, desenhando os sólidos primitivos e posteriormente estabelecer as relações geométricas e topológicas entre esses sólidos.

O Autocad foi escolhido como base ao desenvolvimento deste protótipo, porque tem um modelador de sólidos com arquitectura aberta, uma forte implantação no mercado e um grande número de aplicações utilizando-o como suporte, o que o tornam quase um “standard”.

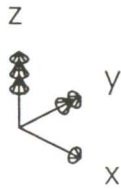
## **2. A utilização do protótipo**

### **2.1 Parametrização de primitivas sólidas**

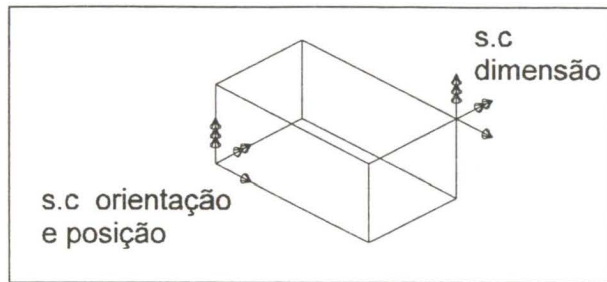
As primitivas sólidas utilizadas são os sólidos elementares (caixa, cilindro, esfera, cone, toro e cunha) e os sólidos de varrimento rotacional e translacional.

A parametrização das primitivas sólidas é realizada associando a cada uma dois sistemas coordenados locais, sendo um para posicionar e orientar o objecto e o outro para dimensionar o objecto (Fig. 1 e 2). Nos sólidos de varrimento é ainda associado um sistema de coordenadas local em todos os pontos de controlo da polilinha usada para efectuar o varrimento, o que permite alterar a forma do objecto. Inicialmente, um sistema coordenado é colocado na origem do sistema de coordenadas universais (scu), nó raíz. Outros sistemas coordenados locais são definidos em relação ao sistema de coordenadas universais, ou em relação a qualquer outro sistema coordenado local já existente, formando uma árvore geométrica de sistemas coordenados.

Cada sistema coordenado local tem seis parâmetros de transformação: três parâmetros de translação ( $t_x$ ,  $t_y$ ,  $t_z$ ) e três parâmetros de rotação ( $r_x$ ,  $r_y$ ,  $r_z$ ). Um sistema coordenado é representado com a simbologia indicada na Figura 1.



**Figura 1.** Sistema coordenado.



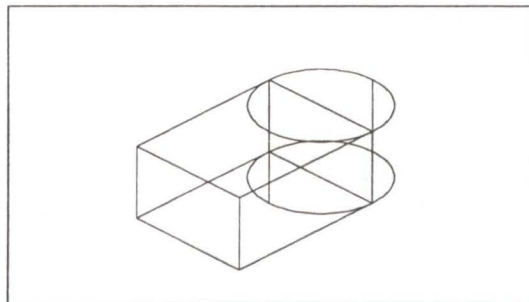
**Figura 2.** Parametrização de uma caixa.

Um sistema coordenado local, ou nó, é representado no ecrã, usando o comando de parametrização que associa um sistema coordenado a uma primitiva sólida, ou com o comando `criar_eixo`. A sua posição e orientação é determinada pelo tipo de objecto seleccionado ou pelo utilizador no comando `criar_eixo`.

## 2.2 Árvore geométrica

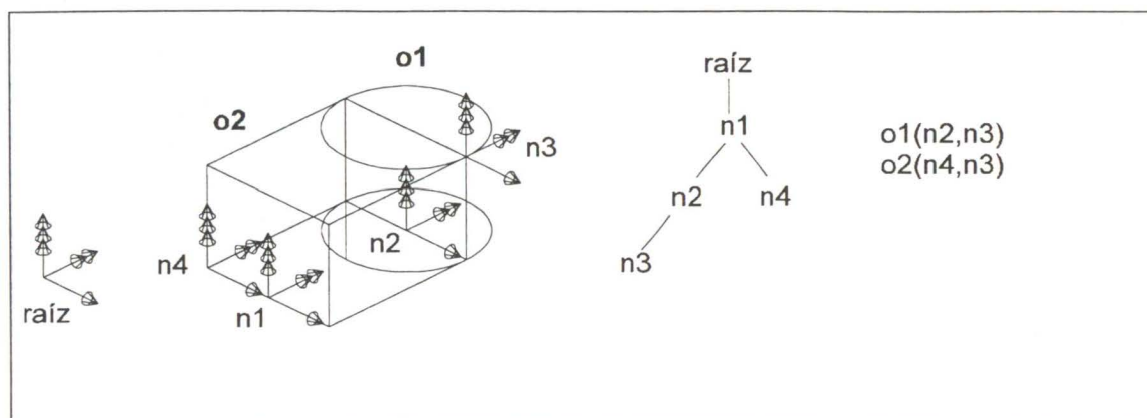
Em [Emme 90], a parametrização de um modelo é realizada definindo primeiro uma árvore geométrica de sistemas coordenados. Esta árvore forma a base para posicionar, orientar e dimensionar sólidos primitivos. A instanciação de um primitivo, é realizada posteriormente seleccionando de um menu um objecto primitivo e seleccionando da árvore um ou mais nós, que definem a geometria do sólido. Este processo não é um modo natural de criação de um modelo, obrigando o utilizador a construir a árvore geométrica sem ter os objectos como suporte visual. Para modelos mais complexos, o utilizador é levado a esboçar os objectos, no papel, para uma primeira análise, do que poderá ser a árvore geométrica.

Neste protótipo, pretendeu-se seguir um processo diferente que permitisse ao utilizador definir o modelo desenhando os sólidos primitivos (Fig.3) e, posteriormente, definir a árvore geométrica que estabelece as relações geométricas e topológicas entre esses sólidos (Fig.4).



**Figura 3.** Modelo desenhado com o AME





**Figura 4.** Modelo parametrizado, árvore geométrica e nós associados a cada sólido primitivo.

O utilizador poderá usar sistemas coordenados já existentes na árvore para parametrização de um sólido primitivo, podendo por isso um nó estar associado a mais de um objecto, caso do nó  $n3$ , que é nó de dimensão quer da caixa, quer do cilindro. Podem ser criados eixos não directamente ligados a sólidos, mas que permitem uma determinada organização da árvore geométrica. Na Figura 4, o nó  $n1$  não está ligado a nenhum sólido, mas permite que a posição e orientação do modelo seja determinada por esse nó.

A árvore geométrica pode ser vista como uma forma elementar de modelação baseada em restrições [Emme 90]. As dimensões dos sólidos são fixadas pela posição de sistemas coordenados locais, e as relações geométricas entre sólidos são especificadas pela estrutura da árvore geométrica e o modo como os sólidos estão ligados aos nós da árvore. A transformação de um nó em coordenadas universais é obtida por concatenação de todas as transformações intermédias até à raiz da árvore. Alterações num nó afectam todos os nós da árvore abaixo dele.

Na Figura 4, o nó  $n3$  é nó de dimensão, quer da caixa, quer do cilindro, por isso uma transformação  $n3.tz=-35$  provoca uma alteração na altura dos dois sólidos (Fig.4b). Uma transformação  $n2.ty=50$  provoca uma alteração no comprimento da caixa e mantém o diâmetro do cilindro na aresta da caixa, dado que  $n2$  é o seu nó de posição e orientação (Fig.4c).

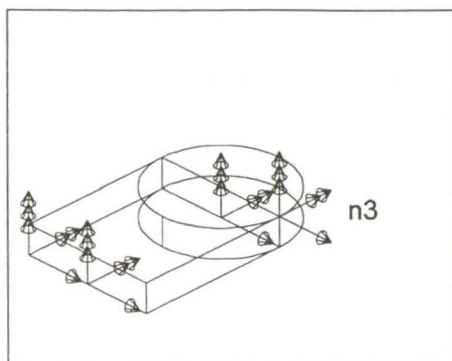


Figura 4b.  $n3.tz=-35$ .

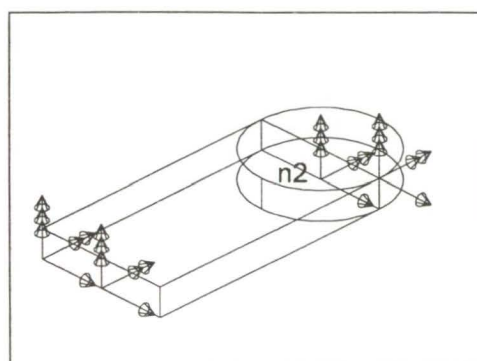


Figura 4c.  $n2.ty=50$ .

Para permitir uma visualização da hierarquia dos nós da árvore geométrica, esta é representada como uma estrutura 3D (Fig.5).

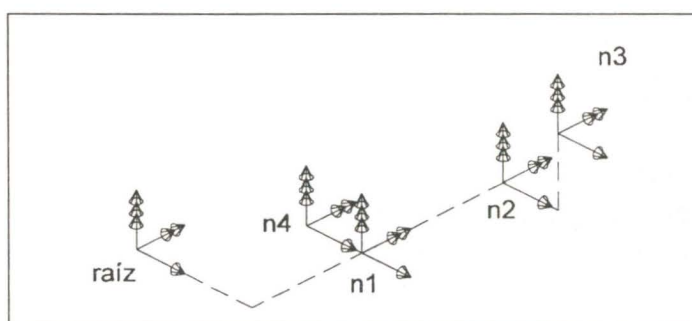


Figura 5. Representação da árvore como uma estrutura 3D.

### 2.3 Restrições

As restrições estendem o conceito de árvore geométrica de modo a poder definir relações geométricas entre nós, que devem ser mantidas pelo sistema, para além das definidas na hierarquia da árvore geométrica.

Uma restrição é aqui definida como uma relação geométrica unidireccional de um dado sistema coordenado ( $\mathbf{n}$ ) relativamente a um sistema coordenado referência ( $\mathbf{r}$ ). Uma restrição fixa um ou mais graus de liberdade de um sistema coordenado, que de outro modo seria especificado por uma transformação rígida em relação ao seu nó pai.

As restrições são especificadas entre sistemas coordenados e não directamente entre objectos o que conduz a uma grande simplificação nas técnicas de cálculo.

Depois de associados os nós a um sólido, o utilizador pode inserir restrições para cada um dos nós ligados a esse sólido, produzindo o sistema uma imediata alteração do sólido que verifique essa restrição. Este comportamento permite utilizar as



restrições como ferramentas de apoio na construção do modelo, tornando possível desenhar os objectos sem grande rigor geométrico, uma vez que as restrições e a consequente reacção imediata do sistema garantem as relações geométricas pretendidas, quer nas dimensões dos objectos, quer na posição relativa entre eles.

O utilizador pode também introduzir mais restrições ao modelo posteriormente à sua parametrização inicial, assim como apagar restrições já introduzidas, tendo apenas como limitação o facto de as restrições só poderem ser referidas a nós criados anteriormente ao nó que se pretende restringir.

As restrições podem ser divididas em dois grupos, que implicam técnicas de cálculo diferentes. Restrições translação, que fixam a posição ( $t_x$ ,  $t_y$ ,  $t_z$ ) do sistema coordenado local do nó ( $\mathbf{n}$ ) em relação ao nó pai ( $\mathbf{p}(\mathbf{n})$ ); restrições rotação, que fixam a orientação ( $r_x$ ,  $r_y$ ,  $r_z$ ) do sistema coordenado local do nó em relação ao nó pai.

- Restrições translação:

$x\_igual$	posição em x de dois nós iguais
$x\_distância\ linear$	distância linear entre dois nós
$x\_razão$	posição em x como uma razão entre dois nós
$xy\_distância\ radial$	distância entre dois nós num plano

- Restrições rotação:

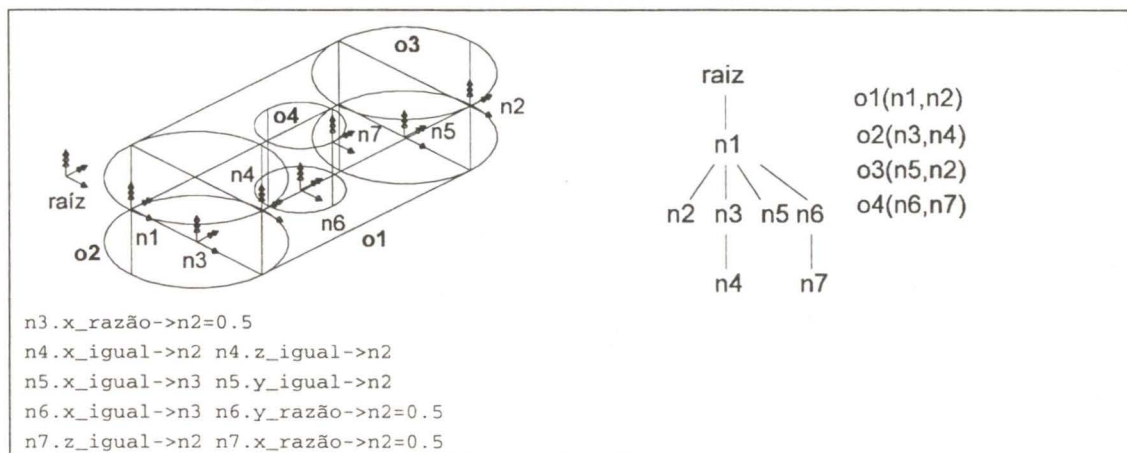
$xy\_paralelo$	eixos dos x e y de ambos os nós paralelos
$x\_aponta$	eixo dos x de um nó aponta para outro nó

Restrições mais complexas podem ser definidas impondo ao nó várias restrições. O número de graus de liberdade fixados por uma restrição depende do tipo de restrição. Por exemplo, uma restrição que especifica uma distância linear em x fixa a transformação em x ( $t_x$ ), mas não afecta os outros graus de liberdade. Algumas restrições não fixam uma transformação, mas impõem uma restrição em várias transformações. Por exemplo, uma restrição  $xy\_distância$  não impõe uma única solução, pois há um número infinito de posições xy que satisfazem a restrição. Estas restrições são chamadas restrições de segunda ordem.

As restrições translação podem ser definidas em termos de desigualdades (<, >, ...), fixando os parâmetros de transformação apenas quando é excedido um dado valor mínimo ou máximo.

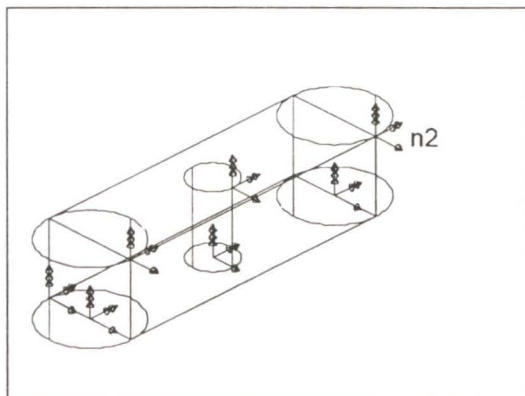
Na Figura 6, a restrição  $n4.x\_igual \rightarrow n2$ , significa que  $n4.x$  terá o mesmo valor que  $n2$  tem em  $x$ , em relação ao sistema de coordenadas do nó pai ( $n3$ ).

A restrição  $n3.x\_razão \rightarrow n2 = 0.5$ , significa que  $n3.x$  será metade do valor que  $n2$  tem em  $x$ , em relação ao sistema de coordenadas do nó pai ( $n1$ ).

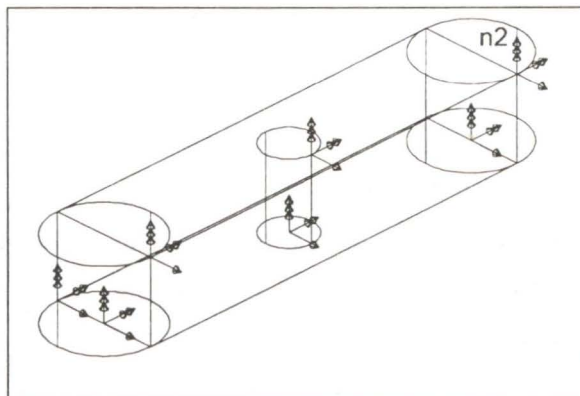


**Figura 6.** Modelo com restrições  $x\_igual$  e  $x\_razão$

Na Figura 6, as restrições garantem que o nó  $n2$  dimensiona todos os objectos do modelo e que as seguintes relações entre os objectos são mantidas. Os diâmetros dos cilindros  $o2$  e  $o3$  permanecem nas arestas da caixa, todos os objectos têm a mesma altura, o cilindro  $o4$  é posicionado no centro da caixa e o seu diâmetro é metade da largura da caixa. Na Figura 7 ilustra-se este comportamento, quando são aplicadas transformações ao nó  $n2$ .



**Figura 7b.**  $n2.tx = -50$



**Figura 7c.**  $n2.ty = 50$

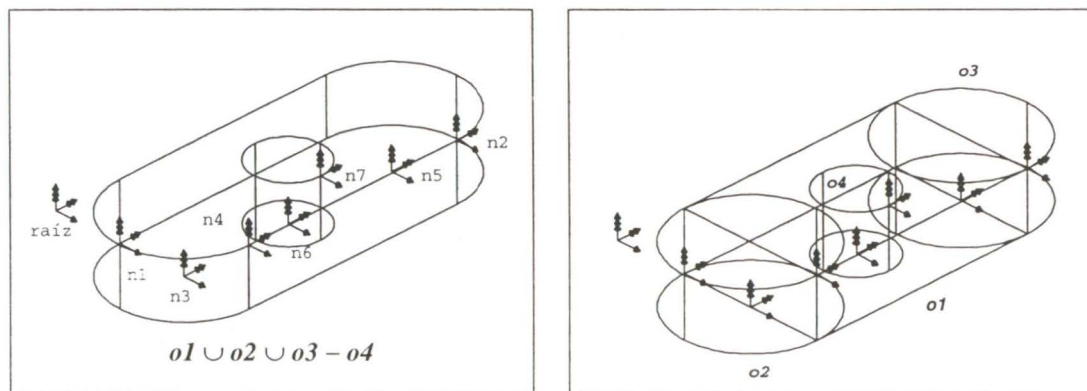
## 2.4 Objectos externos

Os objectos compostos no protótipo são de dois tipos, compostos AME e objectos externos. Compostos AME são sólidos booleanos, a que internamente corresponde uma árvore CSG, criados com os operadores booleanos do AME. Na Figura 8



ilustra-se a aplicação de operadores booleanos ao modelo da Figura 6. No protótipo redefiniram-se estes operadores de modo a só ser possível a sua utilização sobre objectos já parametrizados. Objectos externos são objectos parametrizados, que podem ser inseridos noutros modelos podendo assim desenvolver-se modelos mais complexos.

Quando se pretende alterar um modelo, que é um objecto composto sob o ponto de vista do modelador de sólidos, é fundamental que o utilizador tenha uma visualização dos componentes que formam esse composto, principalmente na fase de esboço do modelo. Esses objectos componentes só são visualizados quando o utilizador pretende alterar o modelo, permitindo assim uma clara ligação entre os objectos primitivos e os sistemas coordenados associados, como representado no desenho da direita da Figura 8.

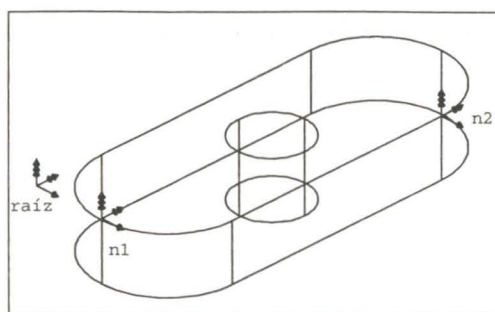


**Figura 8.** Modelo depois de aplicados operadores booleanos (esq.). Representação do objecto quando se pretende alterar o modelo (dir.).

Num modelo parametrizado, definidos os sistemas coordenados e as restrições entre eles, apenas alguns nós são determinantes para posicionar, orientar e dimensionar o modelo. No modelo da Figura 6, dado o modo como foi definida a árvore geométrica e as restrições entre nós, apenas o nó  $n_1$  e nó  $n_2$  são determinantes na alteração do modelo.

O comando *agrupar* permite seleccionar para um dado modelo quais os nós activos, retirando da imagem eixos não importantes na alteração do modelo (Fig.9).

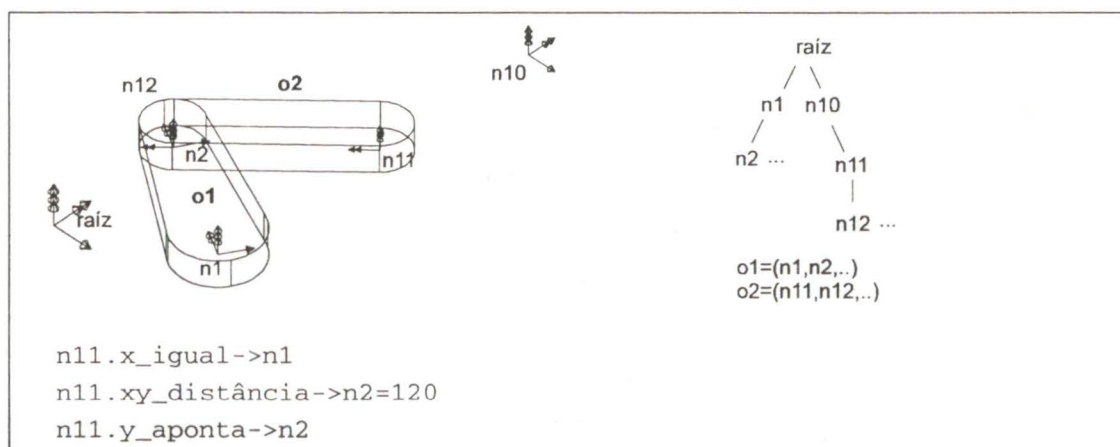




**Figura 9.** Modelo depois de aplicado o comando agrupar: n1 nó pai do grupo e n2 nó activo.

A definição de um objecto externo é semelhante à utilização do comando agrupar, com a diferença de que a aplicação vai gravar o modelo num ficheiro. Estes objectos podem ser inseridos noutra modelo, com o comando `insere-modelo`. Apenas os nós activos do modelo importado serão visualizados, comportando-se o modelo para o utilizador como um objecto primitivo. Na árvore geométrica do modelo os objectos externos são representados abreviadamente, apenas se representam os nós activos. Para o utilizador, a complexidade crescente da árvore não afecta a interacção com o modelo, uma vez que apenas pode manipular os nós activos.

Na Figura 10 o modelo é formado por dois objectos externos e está parametrizado de modo a que a rotação de `n1` provoca uma translação linear de `n11`, sobre a recta que os une. Esta figura ilustra a aplicação do princípio de prorrogação do cálculo às restrições de segunda ordem.



**Figura 10.** Modelo com dois objectos externos

Quando o nó `n1` roda obriga a que `n2` rode, dada a sua posição na árvore, alterando a posição de `n2` no plano `xy`. A restrição `xy_distância` e a restrição `x_igual`, aplicadas a `n11`, obrigam a que só a posição em `y` possa mudar para que ambas as



restrições impostas ao nó sejam satisfeitas. O nó  $n_{10}$  obriga a que a posição em  $y$  tenha uma solução acima de  $n_2$ . A restrição  $y_{aponta}$  orienta o objecto  $o_2$  de modo a apontar para  $n_2$ .

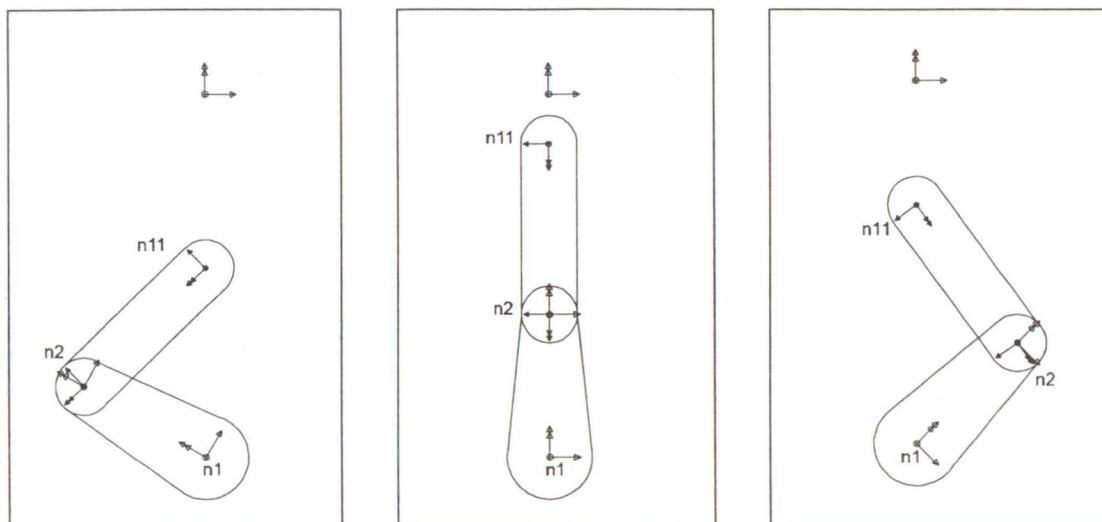


Figura 11. A rotação de  $n_1$  provoca uma translação linear de  $n_{11}$

### 3. Implementação do protótipo

A aplicação cria novos comandos Autocad, através dos quais manipula entidades do Autocad ou do AME. Estas entidades são a representação gráfica dos sistemas coordenados locais e as primitivas sólidas do AME. Os principais comandos da aplicação serão *parametrizar* e *alterar* modelo, cuja estrutura se representa na Figura 12.



Figura 12. Estrutura dos comandos parametrizar e alterar

Na Figura 13 representam-se as estruturas de dados principais da aplicação e as suas relações com a estrutura de dados do modelador de sólidos. Existe uma clara

separação entre a estrutura de dados do modelador de sólidos que é gerida pela aplicação AME e a estrutura de dados da aplicação realizada.

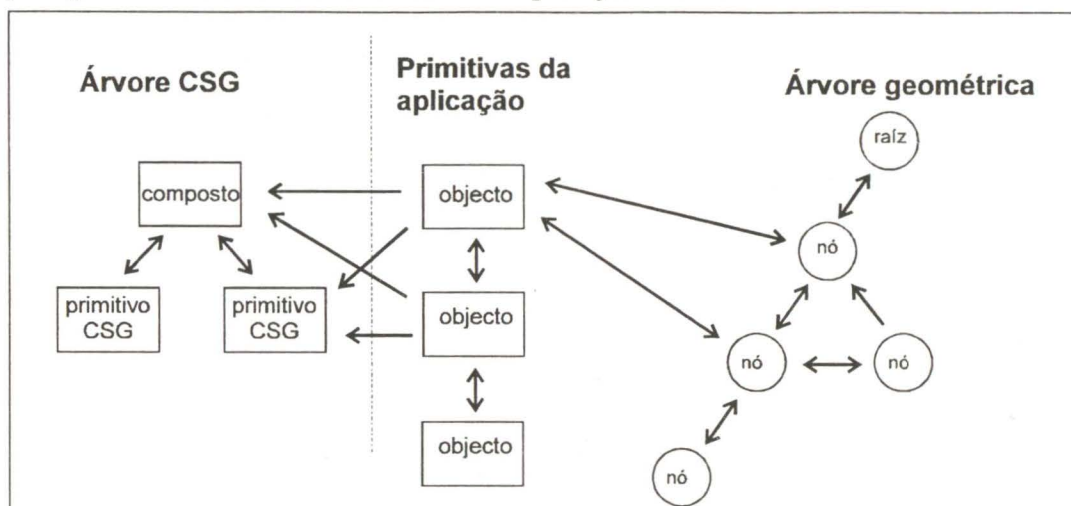


Figura 13. Relação entre as estruturas de dados do AME e as estruturas de dados da aplicação.

### Estrutura de dados dos nós

Os nós estão organizados numa árvore em que cada nó tem um apontador para o nó pai, um apontador para uma lista de filhos, apontadores para a lista de irmãos, e um apontador para o objecto a que está associado (Fig.13). As restrições são guardadas como propriedades dos nós, tendo cada nó um apontador para uma lista de restrições, não representado na Figura 13. Cada nó contém um identificador que é gerido pela aplicação de um modo sequencial o que vai ser fundamental no cálculo das restrições. A cada nó está associada uma matriz de transformação que define a transformação do sistema coordenado local do nó em relação ao sistema coordenado do nó pai.

### Estrutura de dados das Restrições

Cada nó na árvore geométrica tem associada uma lista de restrições. Para cada restrição é definido um apontador para o nó que fixa a restrição, o tipo de restrição, um operador lógico e o valor da restrição.

### Estrutura de dados dos Objectos Primitivos da Aplicação

Os objectos primitivos, depois de parametrizados, são inseridos numa lista. Cada objecto tem apontadores para os nós associados, que determinam a sua geometria. Cada objecto tem um identificador que é gerado pelo AME. Sempre que o objecto intervém numa operação booleana, é criada uma cópia, desse objecto e guardam-se os



identificadores para o correspondente primitivo CSG e para a raiz da árvore CSG (Fig.14).

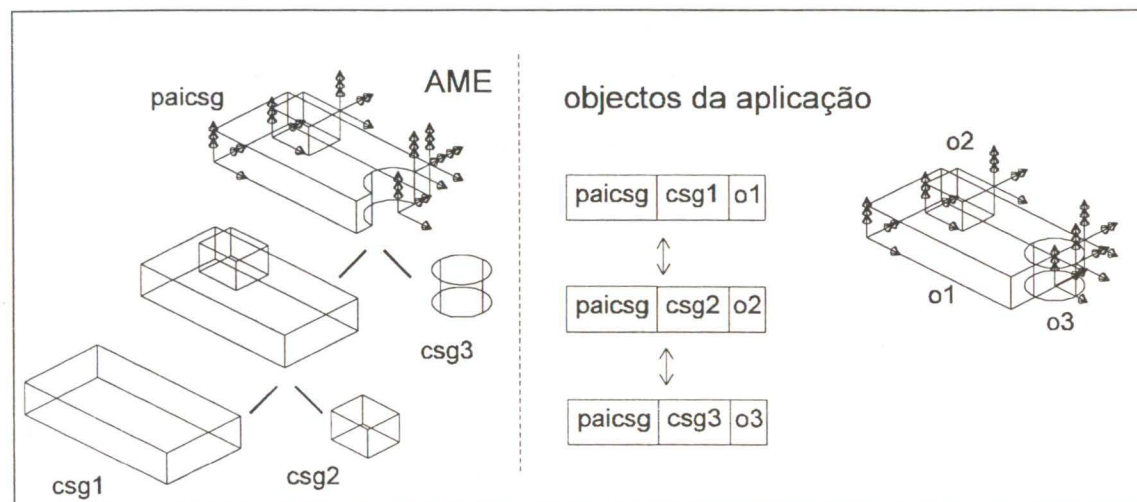


Figura 14 Ligações entre os objectos da aplicação e as primitivas sólidas do AME

### 3.1 Descrição do ambiente de desenvolvimento

O protótipo realizado foi implementado usando como suporte o Advanced Modeling Extension (AME) do Autocad e as suas ferramentas de desenvolvimento, Autocad Development System (ADS) e Application Programming Interface (API). O ADS é a interface entre uma aplicação externa desenvolvida em C e o Autocad. O desenvolvimento de aplicações sobre o AME é realizado utilizando a biblioteca de funções API, que implementa funções correspondentes aos comandos interactivos do AME e permite aceder às suas estruturas de dados.

O AME é um modelador baseado num esquema de representação de sólidos do tipo CSG (Constructive Solid Geometry), sob um ponto de vista operativo, sendo no entanto um modelo híbrido, pois contém informação, ainda que limitada, sob a fronteira dos sólidos (B-rep). O AME possui uma biblioteca de primitivas sólidas e um conjunto de operações booleanas (união, subtração, intersecção) que permitem a construção do modelo segundo o método construtivo. Internamente, o modelo é descrito por uma árvore CSG.

A alteração de parâmetros de uma primitiva sólida dum modelo é realizada pelo comando interactivo SOLCHP, que decompõe a árvore nos seus primitivos e a recalcula, após alteração dos parâmetros da primitiva. A alteração de parâmetros de um sólido feita no AME pelo comando SOLCHP, não é possível usando o API. Para

alterar os parâmetros de um sólido é necessário apagá-lo e criar outro com os novos parâmetros. Esta limitação do API obrigou a criar uma cópia de todas as primitivas sólidas envolvidas em operações booleanas.

### 3.2 Cálculo e validação das restrições

As restrições são calculadas utilizando a técnica de cálculo sequencial. O processo de cálculo começa por uma entidade que é independente de todas as outras, o nó raiz, colocado na origem do sistema de coordenadas universais .

Começando no nó raiz, todos os nós são calculados segundo a ordem sequencial dos identificadores dos nós. Este processo de cálculo exige que, ao serem calculadas as restrições de um nó, todos os nós que o restringem já tenham sido calculados. É este processo de cálculo que impõe a limitação de um nó não poder ter restrições referentes a nós criados posteriormente; a restrição `n3.z_igual->n5` é ilegal. A ordem sequencial dos identificadores dos nós permite impedir estas situações, avisando o utilizador que a restrição não pode ser criada.

Dado que as restrições são definidas entre sistemas coordenados o seu cálculo é facilmente realizado utilizando as técnicas usuais da álgebra linear [Losa 94]. No cálculo das restrições de segunda ordem são utilizados dois princípios, prorrogação do cálculo e perturbação mínima. A prorrogação do cálculo implica que uma restrição de segunda ordem não é calculada, antes de todas as restrições de ordem mais baixa serem calculadas. Por exemplo, para uma restrição `xy_distância`, se existir restrição em `x` , o sistema modifica apenas a posição em `y` de modo a verificar as duas restrições. Este procedimento não garante uma única solução, a selecção de uma solução final é baseada no segundo princípio de perturbação mínima. O sistema calcula a solução que produz uma menor transformação em relação ao nó pai.

Depois de todos os nós da árvore serem calculados, são em seguida calculados todos os sólidos associados aos nós da árvore.

Dado o modo como são criados os nós não existem situações sub-restringidas. Se um nó não tem restrições aplicadas, os seus graus de liberdade têm sempre um valor, definido pela sua posição na árvore geométrica.

As restrições que é possível impor a um nó dependem das restrições já impostas, de modo a não criar uma situação de sobre-restricção. Esta situação pode ocorrer quando várias restrições fixam o mesmo grau de liberdade, por exemplo, quando se aplica a



um nó uma restrição  $x_{igual}$  e  $x_{razão}$ . O processo mais simples para resolver esta situação [Emme 90] é desligar as opções de restrições que já não podem ser especificadas para esse nó. Sempre que se especifica uma nova restrição, o sistema verifica quais os graus de liberdade já fixados, desligando todas as opções que fixam os mesmos graus de liberdade.

O utilizador pode criar situações em que as restrições podem não ser satisfeitas. Para resolver estas situações, sempre que se insere uma nova restrição, todas as restantes são calculadas e sempre que uma restrição não pode ser satisfeita, o sistema avisa o utilizador de que ela é impossível e regressa à situação inicial.

As restrições também podem não ser possíveis por transformações aplicadas ao nó pelo utilizador. Sempre que existe uma alteração de um nó todas as restrições do modelo são calculadas de modo a verificar se essa transformação é válida.

### 3.3 Leitura e escrita de modelos

Informação adicional (*extended entity data*) é um mecanismo criado pelo Autocad para gravar informação das aplicações, criadas com o ADS ou com o Autolisp, em entidades Autocad, mecanismo que é utilizado pelo AME para manter a sua informação nas entidades. Este mecanismo permite que a informação de uma aplicação fique gravada no ficheiro de desenho não sendo necessário recorrer a ficheiros de informação separados. Quando uma aplicação é carregada constrói a sua própria base de dados por leitura da informação adicional nas entidades do desenho.

A aplicação desenvolvida segue este processo, escrevendo a informação nas entidades que manipula, ou seja, os sistemas de eixos e os objectos primitivos da aplicação. Quando um modelo já parametrizado é editado, a aplicação constrói as suas estruturas de dados por leitura desta informação adicional gravada nas entidades.

## 4. Conclusões

O processo de criação de um modelo no ambiente AME e a sua posterior alteração segundo as restrições impostas, ficou muito mais facilitado com as ferramentas que desenvolvemos neste protótipo. No entanto, dadas algumas limitações deste modelador o processo de alteração de um modelo é muito lento.

As técnicas utilizadas, baseadas nas matrizes de transformação associadas às primitivas sólidas, permite a sua aplicação a qualquer modelador de sólidos com uma arquitectura aberta, permitindo assim ultrapassar as limitações do modelador de sólidos utilizado.

A manipulação dos sistemas coordenados para interacção com o modelo, apesar de se ter mostrado eficaz, quer na especificação das restrições, quer no cálculo e manutenção destas, ainda não é um modo natural e intuitivo de interacção, constituindo este aspecto o tópico principal dos nossos desenvolvimentos futuros.

## 5. Referências Bibliográficas

- [Emme 90] Emmerik M.J.G.M. van: Interactive Design of Parameterized 3D Models by Direct Manipulation, Delft University Press, Delft, The Netherlands, 1990
- [Emme 90a] Emmerik M.J.G.M. van: A direct manipulation technique for specifying 3D object transformations with a 2D input device, Computer Graphics Forum, Vol 9, N°4, 1990
- [FrMa 90] Freeman-Benson B.N., Maloney J., Borning A.: An Incremental Constraint Solver, Communications of the ACM, Vol 33, Jan 1990
- [GoZu 88]. Gossard D.C., Zuffante R.P., Sakurai H.: Representing Dimensions, Tolerances, and Features in MCAE Systems, IEEE Computer Graphics & Applications, Vol 8, N°4, Mar 1988
- [HyBr, 78] Hilliard R.C., Braid, I.C.: Analysis of dimensions and tolerances in computer-aided mechanical design, Computer-Aided Design, Vol 10, Jun 1978
- [Just 92] Juster N.P.: Modelling and Representations of dimensions and tolerances: a survey, Computer-Aided Design, Vol 24, N°1, Jan 1992
- [Losa 94] Losa J.M.: Modelação Tridimensional com Restrições Geométricas, Tese de Mestrado, FEUP, 1994.
- [Roll 91] Roller D.: An approach to computer-aided parametric design, Computer-Aided Design, Vol 23, N°5, Jun 1991
- [SuAn 90] Suzuki H. , Ando H. , Kimura F: Geometric constraints and reasoning for geometrical CAD systems, Computer & Graphics, Vol 14, N°2, 1990.

Nota: Este trabalho é parcialmente financiado pela JNICT (Junta Nacional de Investigação Científica e Tecnológica), projecto nº PCMT/C/TIT/444/90.

