



## ***A geração dos sistemas CAD***

*The generation of systems CAD*

GELLY RODRIGUES

Mestre em arquitetura e construção pelo programa de pós-graduação da FEC-UNICAMP.

[gelly@fec.unicamp.br](mailto:gelly@fec.unicamp.br)

### **Resumo**

Este trabalho pretende apresentar e discutir o sentimento adotado por algumas ferramentas CAD no que envolve metodologias de projeto e o uso do computador. Alguns conceitos e métodos computacionais, como exemplo da orientação a objetos, são abordados de maneira a desmistificar e estimular o uso mais ambicioso dessas ferramentas. Nesse contexto, espera-se que o trabalho possa contribuir para uma compreensão positiva quanto à efetiva utilização dos sistemas CAD (*Computer-Aided Design*).

**Palavras-chave:** CAD, metodologia de projeto, algoritmos.

### **Abstract**

the present paper aims at introducing and discussing the adopted movement by some CAD tools in what it involves design methods and the use of the computer. Some computational concepts and methods, for example the object-oriented analysis, are present in way to demystify and stimulate the most ambitious use of the computer. In this context, expects that the work can contribute for a positive understanding of how much is effective the use of CAD tools.

**Keywords:** CAD, design methods, algorithms.

## ***A geração dos sistemas CAD***

Atualmente algumas ferramentas CAD estão convergindo para um movimento chamado *Building Information Models* (BIM), mudança que consiste no desenvolvimento de softwares voltados à indústria da arquitetura, engenharia e construção (AEC) cujo objetivo é permitir descrever projetos 3D com maior qualidade e rapidez, além de evidenciar o uso de objetos

aos seus usuários. Em linhas gerais, os conhecidos BIM são caracterizados como softwares “inteligentes” que permitem a adaptabilidade de objetos, por meio de tecnologias paramétricas, além de possibilitar a interoperabilidade<sup>1</sup> de informações nos sistemas, gerando um trabalho mais dinâmico e integrado com outras ferramentas CAD.

Nesse sentido, fica claro que os segmentos de produtos CAD encontrados atualmente no mercado se classificam a partir de dois sistemas de representação do projeto. Um evidenciado pelo uso de objetos paramétricos como paredes, colunas, vigas, pilares e outros – exemplo do software Revit da Autodesk – e o outro que é reconhecido pelo uso de objetos gráficos primitivos como linhas, arcos, hachuras e demais – a exemplo do AutoCAD também da Autodesk.

Embora essas duas linhas de desenvolvimento possuam um enfoque bastante diferenciado quanto à representação de seus objetos, a reflexão computacional que envolve tais sistemas é a mesma, ou seja, o paradigma da orientação a objetos<sup>2</sup>.

No mundo orientado a objetos, como é o caso dos sistemas de computadores, qualquer coisa pode ser definido como objeto. A representação de uma casa, por exemplo, é um objeto, assim como as coisas que formam essa casa, tais como: paredes, portas ou janelas. Keogh e Giannini (2004) afirmam que um objeto pode ser uma pessoa, um lugar, um utensílio, um conceito ou possivelmente um evento. Nesse contexto, em unidades de representação CAD, qualquer coisa pode constituir um objeto sejam estes em um elevado grau de complexidade como paredes, colunas, vigas ou pilares, ou ainda num grau de complexidade menor como linhas, arcos, hachuras e muitos outros – os chamados objetos primitivos.

O fato é que cada uma dessas linhas de ferramentas CAD possui seus valores, assim como suas limitações no que se refere aos modelos de representação. Entretanto, é importante ressaltar que tais sistemas computacionais nunca serão completos diante de seus usuários exigentes e nem tão pouco sob o ponto de vista do complexo processo de projeto. Razão pela qual algumas técnicas de customização passaram a ser utilizadas como métodos para auxiliar esse processo do projeto. Conceitos como exemplo da orientação a objetos são agora tratados de maneiras explícitas através da implementação de processos lógico-matemáticos para representação de um procedimento do projeto.

Uma tendência atual é o uso de *scripts language* para transformar programas de CAD comuns em softwares paramétricos. Programas como Rhinoceros, 3DMax (fig.1) e Maya

<sup>1</sup> Capacidade de um sistema de se comunicar com outro sistema semelhante ou não.

<sup>2</sup> Metodologia muito utilizada na ciência da computação para o desenvolvimento de sistemas baseados em unidades que se relacionam entre si, os chamados objetos.

possuem suas próprias linguagens, enquanto o AutoCAD utiliza a linguagem VBA, comum a outros aplicativos do Microsoft Windows. Mesmo o CATIA, um software que já apresenta funcionalidades paramétricas, permite sua customização e ampliação de funções por meio da programação com *VB script*, outra linguagem comumente usada em diversos aplicativos do Windows.

```

terreno = copy S
convertToMesh terreno
vertices = terreno.vertices
escala = 10
for i = 1 to getNumVerts terreno do
(
point = vertices[i].pos
x = point.x
y = point.y
z = point.z
x=ceil(x/(escala*2))*escala*2
y=ceil(y/escala)*escala
z=ceil(z/(escala/2))*(escala/2)
vertices[i].pos = [x,y,z]
)
update terreno
select terreno

```

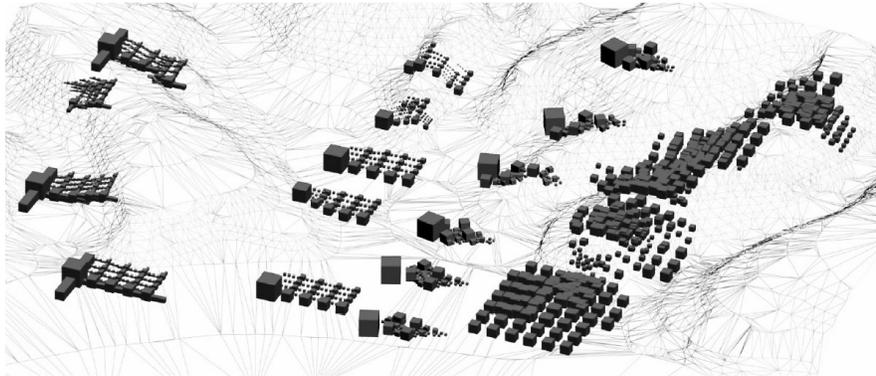


Fig. 1: Script para 3DMax que permite adaptar uma matriz de cubos a superfícies irregulares. Fonte: Culagovski; Guevara, 2006, p.270.

Fica claro que o movimento dos sistemas CAD está substancialmente nutrindo idéias que anteriormente eram apenas utilizados por áreas como a ciência da computação. Autores como Terzidis (2006) e Celani (2003), mostram que é perfeitamente possível chegar a soluções mais interessantes por intermédio da automatização e algoritmização de processo de projetos. Enquanto o primeiro autor utiliza conceitos provenientes da inteligência artificial, como os autômatos celulares e os algoritmos genéticos, o segundo utiliza conceitos lógico-matemáticos como recursão, parametrização, simetria, composições aleatórias, composições regradas e formas emergentes para auxiliar esse processo.

Essa busca pelo uso do CAD no processo de geração da forma, entretanto não é uma novidade; Mitchell já na década de 70, época em que o computador era pouco usado pelos arquitetos e designers, afirmava:

*O uso menos ambicioso da máquina é dar-lhe apenas funções de representação, deixando a responsabilidade pela geração e avaliação de soluções com o projetista. Numa versão mais ambiciosa, a máquina pode receber a função de avaliar soluções produzidas por projetistas humanos (...) O mais ambicioso uso da máquina consiste na tentativa de utilizá-la na solução inteligente de problemas mal definidos. (MITCHELL, 1975, p.149).*



Assim, com o intuito de ilustrar o conceito envolvido por determinadas ferramentas CAD, uma síntese dessas propostas é apresentada a seguir:

**Autodesk Revit:** é caracterizado por possuir famílias de objetos já definidos tais como parede, janela, coluna, piso, telhado, escada, rampa e outros. Seu ambiente de representação é dito “inteligente” por possibilitar uma atualização automática dos cenários, permitindo uma maior liberdade nas mudanças de parâmetros dentro de um mesmo projeto, seja no todo ou em exemplos individuais.

O Revit ainda permite que novas famílias de objetos, sub-tipos e parâmetros sejam criados a partir de seu editor de famílias e *templates*. Tipos de parâmetros, por exemplo, que podem ser escolhido a partir de texto, número, área, volume, ângulo, URL, material e da lógica booleana, característica que permite a criação de objetos com ricas informações interligadas. (TSE; WONG; WONG, 2005).

**GraphiSoft ArchiCAD:** No ArchiCAD a representação de objetos são divididos em elementos construtivos e em GDL (*Geometric Description Language*). Os elementos construtivos são os objetos básicos para a construção do edifício tais como paredes, colunas, vigas, lajes, telhados; grupo de objetos que residem no sistema e não podem ser omitidos. Os outros objetos do edifício, tais como portas e janelas, são objetos GDL que residem em algumas bibliotecas externas. O GDL como linguagem de *script* permite criar novos objetos com ricas informações paramétricas, além de possibilitar adicionar um número ilimitado de objetos BIM no ArchiCAD.

Uma importante vantagem do ArchiCAD deve-se ao fato da ferramenta ser um IFC (*Industry Foundation Classes*), ou seja, um padrão industrial que permite que todas as ferramentas CAD reconheçam seus arquivos devido ao seu formato de interoperabilidade.

**ADT:** *Autodesk Architectural Desktop's* é uma ferramenta semelhante ao AutoCAD. Entretanto seu conceito utiliza a “tecnologia dos objetos associados, facilitando a criação de elementos arquitetônicos inteligente que se ajustam a qualquer alteração feita no projeto em tempo real, extinguindo, conseqüentemente as indesejáveis revisões” (PEDRO; CARLO, 2002). O ADT é um programa que além de possibilitar adição de módulos no projeto é compatível com o IFC (*Industry Foundation Classes*).

**Autodesk AutoCAD:** trata dos objetos a um nível de complexidade mais baixo em relação a programas como Revit e ArchiCAR por exemplo. A interação com o usuário que é dado por uma representação de classes de objetos como linhas, arcos, esferas, hachuras e outros, o que não possibilita a adaptabilidade de cenários nem tão pouco reduz características como tempo e atividade de trabalho. Entretanto, através do AutoCAD é permitido automatizar e customizar passos do projeto que serão desenvolvido por meio da linguagem de programação *Visual Basic for Application* (VBA). Ambiente de programação que acompanha a instalação padrão do aplicativo e que possibilita um contato direto com definições do paradigma de orientação a objetos.

**MicroStation:** basicamente é um gerador de objetos em gráficos vetoriais 2D/3D desenvolvido pela Bentley Systems, entretanto, possibilita o usuário expandir ou customizar operações através do ambiente de programação *Visual Basic for Applications* (VBA) da Microsoft embutido em seu formato padrão. Além de contar também com o *MicroStation Basic*, MDL (C) e Java como opções para programação.

Torna-se cada vez mais perceptível a abordagem da orientação a objetos no uso de ferramental gráfico, seja em um nível de programação de objetos ou em um nível conceitual do assunto. Características que são, também, encontradas em softwares de modelagem e animação como é o caso do conhecido Blender 3D, um programa de código aberto que além de permitir o acesso ao código fonte do programa possibilita a customização do aplicativo através da linguagem de programação orientada a objetos conhecida como *Phyton* (<http://www.python.org/>).

É gigantesca a quantidade de *scripts* voltados para texturização, modelagem, jogos, animação, renderização, dentre outros encontrados na comunidade do software livre; fator que possibilita sempre a melhoria de determinadas necessidades dos usuários além de gerar e contribuir por inúmeras discussões em fóruns especializados no problema. A ferramenta de modelagem 3D – Blender – ainda aborda conceitos como o de hierarquia para representação de uma estrutura entre objetos; característica que permite determinar um relacionamento de “pai” e “filhos” como uso explícito de conceito da orientação a objeto, sendo utilizado sem qualquer nível avançado de programação apenas através da interação do usuário com a interface gráfica do programa.

Logo, embora alguns CAD sejam ditos menos “inteligentes” do que outros, sempre existirão propostas alternativas aos usuários de modo a utilizar esses softwares da melhor maneira possível, seja pela parametrização de objetos ou mesmo no contato direto como o respectivo conceito já detalhado anteriormente. Além do AutoCAD, Microstation e Blender, existem outros vários programas que possuem linguagens de *scripts* embutidos em suas aplicações padrão. Como exemplos, citam-se ainda:

**Autodesk Maya:** é um software de animação e modelagem que recentemente teve a linguagem de script *Phyton* embutida em seu ambiente de modelagem como uma alternativa para a nativa linguagem de programação do Maya, o MEL. O chamado *Maya Python Scripting* tem o objetivo de aumentar ainda mais a criatividade de seus usuários além de oferecer aos seus programadores a capacidade de manipular, adaptar e automatizar o software eficientemente.

**Vectorworks:** é uma plataforma de software que permite a definição de projetos em 2D e 3D com precisão, além de também possuir uma linguagem de programação integrada chamada de *VectorScript*. O *VectorScript* é um ambiente de desenvolvimento completo e totalmente integrado ao *VectorWorks*, o que permite criar *scripts* que automatizem o processo de criação desde a construção de ferramentas, comandos até objetos paramétricos.

**Autodesk 3D studio Max:** é um software bastante usado em diversas áreas da animação gráfica, tais como produção de filmes de animação, maquetas eletrônicas e criação de qualquer mundo virtual. O software permite customização de certas preferências através do chamado *MaxScript*, inclusive, o desenvolvimento de novos *plug-ins* em um módulo interno. Quase tudo é feito através dos *plug-ins*, inclusive os objetos básicos como esfera, cubo, prismas e outros.

Diante do exposto em todos os softwares aqui listados, nota-se que a principal característica entre essas ferramentas se encontra em sua maior ou menor capacidade quanto à resolução de problemas e de sua adaptação a novas situações. O fato é que tais ferramentas computacionais nunca serão completas, o que é comprovado pela maioria dos programas apresentarem poderosos recursos de customização. É preciso apenas que os usuários

entendam e definam suas reais necessidades para então, poder adaptar tais sistemas de maneira mais eficiente e específica. Ferramentas não faltam, escolha a sua!

### **Agradecimentos**

A autora agradece ao CNPq pela concessão de auxílio para esta pesquisa.

### **Referências**

CELANI, Gabriela. *CAD Criativo*. Rio de Janeiro: Campus-Elsevier, 2003.

CULAGOVSKI, R.; GUEVARA, S. *Arquitectura, datos y forma: una primera aproximaciona instrumental*. In Soza, P. (org.) *Anais do X Sigradi*. Santiago, Universidad de Chile, 2006, p.268-272.

MITCHELL, W. *The theoretical foundation of computer-aided architectural design*. In: *Environment and Planning B*, 1975, v.2, p.127-150.

PEDRO, Aparecido Henriques; CARLO, Marco Luiz Del. *Architectural Desktop 3.3: Prático e Fácil*. São Paulo: Érica, 2002.

KEOGH, Jim; GIANNINI, Mario. *POO Desmistificado: Programação Orientada a Objetos*. Rio de Janeiro: Alta Books, 2005.

TSE, Tao-chiu Kenny; WONG, Kam-din Andy; WONG, Kwan-wah. *The utilisation of building information models in nd modelling: a study of data interfacing and adoption barriers*. *ITCOM*, v. 10, p. 85-102, abr. 2005.

TERSIDIS, K. *Algorithmic architecture*. Cambridge, MA, Architectural Press, 2006.