

Modelagem cognitiva do processo criativo em arquitetura por meio da técnica de programação orientada a objetos / Cognitive modeling of the creative process in architecture by means of the object-oriented programming technique

Gelly Rodrigues / UNICAMP, Brasil / gelly@fec.unicamp.br / Gabriela Celani / UNICAMP, Brasil / celani@fec.unicamp.br, www.fec.unicamp.br/~celani

Abstract *The aim of this research was to investigate the relationship between the object-oriented paradigm and the design process in architecture. The work was inspired by Mitchell's (1990) comparison between architectural types and classes of objects. An analogy was set between the development of classes and the structuring of design problems based on architectural typologies. The method was then compared to Alexander's (1964) in terms of levels of abstraction. Two classes were implemented, illustrating the application of the object-oriented paradigm in architectural design. The method developed is expected to help architects develop a new understanding of the design process.*

Antecedentes Desde o movimento dos métodos da década 60, tem havido várias iniciativas de descrever o processo de projeto de maneira lógica e ordenada, sempre objetivando aumentar a qualidade dos projetos de arquitetura, a eficiência do trabalho colaborativo, além de possibilitar a implementação do processo de projeto no computador. Entretanto, embora a utilização do computador esteja cada vez mais presente nas etapas de esquematizar, modelar e implementar projetos de arquitetura, conceitos e métodos computacionais raramente são explicitados no processo de projeto (CELANI, 2002). Uma possível justificativa para isso deve-se à visão simplificada que muitos arquitetos possuem em relação ao CAD.

Segundo Terzidis (2006) tornou-se muito comum confundir um procedimento de “computerization”, atualmente desempenhado por intermédio das ferramentas CAD, como um trabalho resultante da computação em seu sentido mais exato. Para ele a computação propriamente dita diz respeito à utilização de procedimentos lógico-matemáticos na solução de problemas. Trata-se de um procedimento que envolve o raciocínio, lógica e algoritmos, além de estruturas mentais, cognição, simulação e regras baseadas na inteligência. Em contraste, a

“computerization” é definida pelo autor como o ato de preencher ou armazenar informações no computador ou sistemas de computadores, automação e mecanização que geralmente envolve a digitalização de entidades ou processos que são predeterminados e bem definidos.

Diante desse contexto, fica evidente que o uso da computação como ferramenta de projeto baseada no computador ainda é bastante limitado em arquitetura. Os processos desenvolvidos na mente do projetista continuam sendo ações de preenchimento, manipulação ou armazenamento nos sistemas de computadores, prevalecendo um modelo de utilização dos computadores baseado na “computerization”. Esse problema deve-se ao fato dos projetistas não aproveitarem o poder computacional dos computadores, acreditando muitas vezes que qualquer modelo criado em computador é um produto de computação (TERZIDIS, 2006).

Uma técnica alternativa para o uso efetivo dessa ideia dialeticamente oposta é encontrada no desenvolvimento de algoritmos para arquitetura. De acordo com Terzidis (2006), através do uso de scripting languages o projetista pode ir além do uso do mouse, transcendendo um conjunto de limitações dos softwares atuais, além de



incorporar a complexidade computacional e o uso criativo dos computadores. Nesse sentido, a automação das etapas do projeto que é proposta neste trabalho por meio da técnica de programação orientada a objetos não é apenas importante do ponto de vista da eficácia do trabalho. Sua maior contribuição para os arquitetos é na verdade possibilitar uma compreensão aprofundada do processo de projeto.

O presente trabalho tem o intuito principal de buscar na ciência da computação, métodos que possam contribuir para a maneira de pensar sobre o projeto de arquitetura. Trata-se de um estudo de caráter exploratório que objetiva apresentar um método alternativo aos métodos tradicionais de pensar e estruturar os problemas de projeto.

Em vista disso, a pesquisa se dividiu em duas etapas primordiais. O momento inicial teve como foco a obra *The Logic of Architecture: Design, Computation, and Cognition* (1990) de William J. Mitchell, em que o autor sugere o estabelecimento de uma analogia entre tipologias arquitetônicas e classes de objetos, fazendo clara alusão ao paradigma da orientação a objetos. Essa conexão de idéias proposta por Mitchell (1990), por sua vez, conduziu a presente pesquisa à necessidade de compreender melhor, por um lado, os tipos arquitetônicos – abordando autores como Argan (2006) e Colquhoun (1967) – e por outro os princípios fundamentais do paradigma da orientação a objetos.

O segundo momento da pesquisa partiu para a análise de outros estudos que se propuseram a utilizar conceitos e métodos computacionais relacionados ao processo de projeto e ao desenvolvimento de implementações em CAD. Para isso realizou-se uma sistematização das idéias do arquiteto e matemático Christopher Alexander apresentadas em *Notes on the synthesis of form* (1964), referência na área exatamente por apresentar métodos de projeto semelhantes aos métodos utilizados na ciência da computação, e também um delineamento de suas idéias apresentadas em *A Pattern language: towns, buildings, construction* (1977), obra em que o autor defende a utilização de padrões comprovadamente eficientes no desenvolvimento de projetos arquitetônicos. Uma vez estabelecida essa estruturação de idéias, passou-se ao

desenvolvimento de implementações, como forma de ilustrar o processo de projeto arquitetônico assistido por computador, com uma abordagem verdadeiramente computacional.

Materiais e métodos A implementação computacional de programas que ilustram os conceitos estudados foi realizada por meio do uso da linguagem de programação Visual Basic for Applications (VBA) dentro do ambiente do AutoCAD. Apesar da linguagem VBA não apresentar todas as características de uma linguagem orientada a objetos tradicional, ela possui os principais conceitos de orientação a objetos, como a possibilidade de criação e instanciamento de novas classes com propriedades e métodos, além do encapsulamento. O uso da linguagem VBA no ambiente AutoCAD permitiu que se utilizasse o editor gráfico e a estrutura de objetos já existente no AutoCAD, facilitando o trabalho.

No que diz respeito à implementação dos aplicativos buscou-se utilizar um método bastante popular na área da ciência da computação, conhecido como desenvolvimento incremental. Através desse processo foi possível desenvolver inicialmente programas bastante simples, que se tornaram gradativamente mais complexos, à medida em que foram sendo introduzidas novas variáveis, controles e funções. Outro método empregado da ciência da computação foi a utilização de uma linguagem de modelagem gráfica para a representar as classes de objetos desenvolvidas, isto é, a Unified Modeling Language (UML). O uso dessa convenção permitiu estruturar as classes arquitetônicas de maneira visual e organizada, facilitando o trabalho de implementação dos programas. Através do uso de conceitos e métodos da ciência da computação, foi possível implementar programas de grande complexidade, porém baseados em uma estrutura clara e inteligível.

No que se refere à pesquisa teórica, o desenvolvimento do método de projeto aqui proposto tomou como ponto de partida as idéias de dois autores que são referência na área de CAD: William Mitchell e Christopher Alexander. O primeiro estabeleceu uma analogia entre tipos arquitetônicos e classes de objetos (Mitchell, 1990), enquanto o segundo propôs a modelagem formal



do problema de projeto por meio de um exercício mental que vai da realidade do ambiente à abstração do projeto, e em seguida volta da abstração do projeto à realidade da obra construída (Alexander, 1964). Procurou-se, neste trabalho, inter-relacionar e estender os conceitos desses dois autores, propondo uma estrutura de pensamento com um nível a mais de abstração que o modelo de Alexander, no qual o projeto de arquitetura é definido como uma classe de objetos. O caminho de volta à realidade passa pelo instanciamento da classe e pelo estabelecimento das propriedades acidentais específicas da instância, até chegar à obra construída como objeto único, ainda que esta guarde semelhanças com a classe a partir da qual foi gerada.

O método de projeto desenvolvido nesta pesquisa sugere que a estrutura do pensamento do projeto seja dividida nas seguintes etapas: Identificação do problema; formação de uma imagem mental; estruturação de uma imagem formal; definição das propriedades essenciais e acidentais do objeto; estudo da classe de objetos; implementação da classe, instanciamento do objeto e objeto real. Essa estrutura apresentava um nível a mais de aprofundamento que aquela apresentada por Alexander. O nível acrescentado permite inserir as idéias de Mitchell (1990) no processo de projeto, com a especificação das propriedades essenciais e acidentais do objeto arquitetônico e a definição de uma classe de objetos, que corresponde à tipologia arquitetônica procurada.

Implementações Para ilustrar o método de projeto sugerido na pesquisa, foram desenvolvidos inicialmente dois protótipos preliminares, de acordo com os exemplos propostos por Mitchell (1990) em *The Logic of Architecture*. Os tipos “cadeira” e “templo dórico” foram definidos como classes de objetos e implementados por meio da linguagem de programação orientada a objetos VBA, em ambiente AutoCAD, conforme apresentado a seguir.

Para definir claramente o processo utilizado na definição e implementação computacional do protótipo “cadeira”, as etapas de seu desenvolvimento foram estruturadas da seguinte maneira:

Em primeiro lugar foi analisada a descrição de uma classe de objetos do tipo cadeira proposta por Mitchell (1990). Em seguida, buscou-se refletir sobre o problema “cadeira” de uma maneira geral. Assim, algumas classes distintas de objetos do tipo cadeira foram consideradas, tais como a cadeira da sala de jantar, a cadeira escolar, a cadeira do escritório, a cadeira de balanço, a cadeira giratória, a cadeira fixa, dentre outros possíveis resultados que atendem de maneira suficientemente expressiva a questão do problema.

Através dessa abstração inicial do problema, tornou-se claro que o tipo cadeira é na verdade um item geral que se refere a muitos objetos, cada qual com suas características e necessidades particulares. Para diferenciar os exemplos de objetos pertencentes a uma classe genérica como o item “cadeira” passou-se a analisar e refletir sobre o tipo em termos de itens gerais e tipos mais específicos desse item, o que permitiu que a classe cadeira fosse analisada e refletida sob uma visão de especificação/generalização de tipos de objetos.

A partir de então, buscou-se definir e estruturar as propriedades essenciais e acidentais para a construção do tipo cadeira. Para esse processo foram realizados dois passos importantes. A primeira ação consistiu na definição dos objetos comuns a todos os tipos cadeira – isto é assento, encosto e pernas – e na definição dos objetos que podem variar entre os mesmos exemplos da tipologia estudada como, por exemplo – braços, estofados, cores e rodinhas – ambas seguindo as definições teórico-filosóficas propostas por Mitchell (1990). Uma vez definida as propriedades essenciais e acidentais do tipo cadeira, buscou-se através do relacionamento da orientação a objetos conhecido como agregação, estruturar e determinar o nível de dependência que a tipologia analisada possui com os objetos envolvidos no relacionamento, conforme ilustra *Figura 1*

Dessa forma, a dependência existente entre o tipo cadeira e os seus objetos essenciais e acidentais de composição foram claramente estabelecidos, incluindo o número mínimo e máximo de elementos (min. máx).



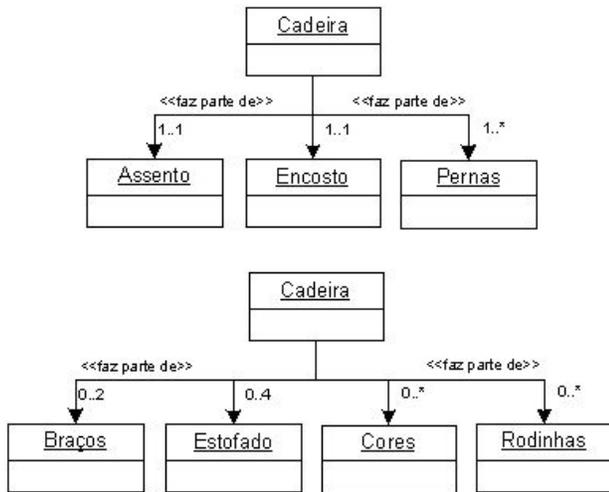


Figura 1 Propriedades essenciais (1..x) e acidentais (0..x).

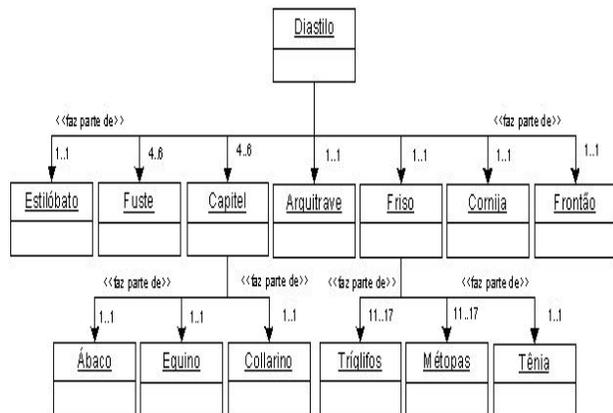


Figura 2 Propriedades essenciais do subtipo templo diástilo.

O programa resultante possui uma interface que permite instanciar objetos do tipo cadeira com características bastante variadas, porém resultando sempre em objetos que se enquadram na classificação do tipo.

Tendo em vista o método de desenvolvimento incremental, buscou-se através do segundo protótipo abordar um número maior de variáveis no processo de resolução do problema de projeto. O desenvolvimento do protótipo “templo dórico” começou com uma refle-

xão sobre o problema dos templos da ordem dórica de uma maneira geral. Por meio de um estudo sobre essa tipologia arquitetônica, tornou-se possível compreender que mesmo um tipo aparentemente bem definido como o templo dórico pode se apresentar diferentes versões. Esses diversos tipos podem variar em relação ao número de colunas na fachada, à distribuição das colunas, ao espaçamento entre as colunas (intercolúmnio) e à disposição dos tríglifos e métopas.

A partir dessa abstração inicial do problema, ficou explícita a possibilidade de um templo dórico variar consideravelmente dentro da mesma tipologia. As possibilidades de variação encontradas nessa tipologia permitem ao arquiteto enveredar por diversas linhas de raciocínio, em que diferentes propriedades da tipologia são avaliadas, mas nenhuma delas é considerada única e universalmente correta.

Na implementação deste protótipo, optou-se por analisar e refletir sobre o tipo templo dórico apenas em termos de seu intercolumnio, desdobrando o tipo ordem dórica nos subtipos diástilo e sistilo. No que diz respeito aos elementos de composição da tipologia estudada, buscou-se definir e as propriedades essenciais para a construção do tipo “templo dórico”.

Por motivos de simplificação, optou-se por analisar apenas as propriedades essenciais de um dos subtipos identificados, o diástilo.

As propriedades essenciais do subtipo diástilo foram definidas e estabeleceu-se a relação entre essa sub-classe e os objetos que a compõem de maneira hierárquica, conforme ilustra a Figura 2

Ao analisar a dependência entre a tipologia estudada e seus objetos essenciais de composição, a seguinte configuração foi estabelecida:

O tipo diástilo possui:

- no mínimo um e no máximo um estilóbato;
- no mínimo quatro e no máximo seis fustes;
- no mínimo quatro e no máximo seis capitéis;
- no mínimo uma e no máximo uma arquitrave;
- no mínimo um e no máximo um friso;



- no mínimo uma e no máximo uma cornija;
- no mínimo um e no máximo um frontão.

O programa resultante permite instanciar objetos do tipo templo diástilo com características bastante variadas, porém resultando sempre em templos que se enquadram na classificação do tipo. A Figura 3 mostra alguns exemplos de templos geradas automaticamente com o programa desenvolvido. *Figura 3*

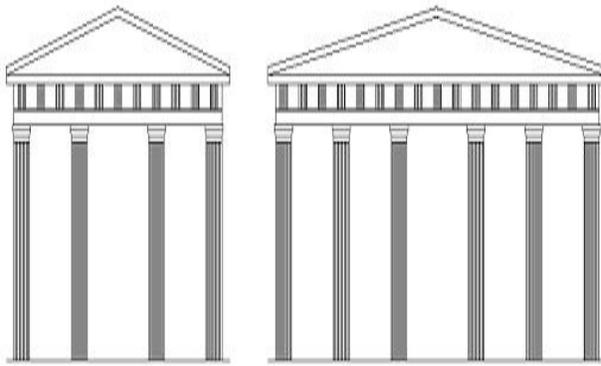


Figura 2 Propriedades essenciais do subtipo templo diástilo.

Considerações finais e trabalhos futuros A próxima etapa desta pesquisa consistirá na implementação de uma nova classe de objetos arquitetônicos, uma tipologia residencial. Para esse exercício será necessário repensar essa tipologia arquitetônica em ter-

mos de propriedades essenciais e acidentais, bem como pesquisar as soluções existentes dentro dessa categoria, ou seja, as diferentes tipologias residenciais consolidadas através da história da arquitetura. Espera-se dessa maneira que o trabalho possa contribuir para uma nova forma de raciocínio sobre o projeto, aplicável ao ensino da arquitetura, por meio de cursos que permitam analisar a eficiência da analogia proposta.

Agradecimentos Ao CNPq e à FAPESP pela concessão de auxílio para esta pesquisa.

Referências Alexander, Christopher, 1964, *Notes on the Synthesis of Form*. London: Harvard University Press. / Alexander, Christopher; et al: 1977. *A Pattern Language: Towns, Buildings, Construction*. 1ª ed. Nova Iorque: Oxford University Press. / Argan, Giulio Carlo: 2006, *Sobre a Tipologia em Arquitetura*. In: NESBITT, Kate (org.). Uma nova agenda para a arquitetura: Antologia teórica (1965-1995). 1ª Edição, São Paulo: Cosac Naify, p. 268-272. / Celani, Gabriela. *Beyond analysis and representation* in CAD: 2002, Ph.D. thesis. Cambridge, MA, MIT. / Colquhoun, Alan. 1967 *Typology and design method*, Essays in Architectural Criticism: Modern Architecture and Historical Change, Estados Unidos, v. 83, p. 43-50. / Mitchell, J: 1990, *The Logic of Architecture: Design, Computation, and Cognition*. Cambridge, MA: MIT Press. / Terzidis, Kostas: 2006, *Algorithmic Architecture*. 1ª Edição, Londres: architectural press.

Keywords: *Design process; design method; object-oriented programming; CAD.*

