

A injeção de fluxos analógicos em algoritmos: um “pecado” do UNStudio

The injection of analog streams in algorithms: a “sin” of UNStudio

■ Rovenir Bertola Duarte

Universidade Estadual de Londrina, Brasil

rovenir@uel.br

Abstract

After the proliferation of script programs in architecture seems clear the possibility of transformation of the architectural design process and paradigm shift. According Carpo, the approach of writing architecture and code brings us to the “variability of paradigm” (Carpo, 2011b). Although it is premature to talk about paradigm shifts can speculate on the proliferation of a more codified kind of thinking. So, what are the consequences of this thinking more structured and encoded, for architectural design? Is it possible to graft something not coded language in a binary environment such as digital? The UNStudio experience and Deleuze’s ideas seem to reveal some way (the “sin”).

Keywords: UNStudio, Diagrama Deleuziano: Algoritmos

Introduction

Na abertura da revista AD Programming Cultures de 2006, a editora inglesa Helen Castle faz a seguinte pergunta ou provocação: “poderia o scripting tornar-se o novo desenho?” Esta frase está diretamente relacionada com a popularização dos programas de script dentro do ambiente projetual arquitetônico nos últimos anos. Em outras palavras, ela está relacionada com a (re)aproximação dos arquitetos a escrita e a um pensamento mais regrado, codificado e estruturado. Deste modo, pode-se pensar na possibilidade da formação de um tipo de pensamento projetual na arquitetura contemporânea, algo talvez próximo ao que Peters e Kestelner (2013) chamam de “pensamento algorítmico”. Ou ainda que vá em direção ao estudo do neurologista Stanislas Dehaene, para quem o homem moderno pode estar modificando pouco a pouco o modo de pensar devido à força da difusão da escrita em sua vida. De qualquer modo, este pensar próximo à escrita estaria mais próximo a uma linguagem em código (estruturada, convencional, modular e digital) que de uma linguagem analógica (caótica, não semelhante, modulada e diagramática no sentido deleuzeano). Assim, a possível proliferação de um tipo de pensamento mais codificado abre a lacuna para algumas questões: quais seriam as consequências desta mudança do pensar, mais codificado e estruturado, para o processo de criação projetual? O que se poderia estar perdendo com a ausência de uma linguagem menos codificada? Por fim,

o que se poderia fazer para enxertar algo de linguagem não codificada em um ambiente binário como o digital?

Para procurar alargar estas questões e aproximar a algumas respostas, recorreu-se neste artigo às reflexões feitas pelo filósofo francês Gilles Deleuze nos anos 80 sobre linguagem em código e analógica, como também algumas de suas influências na construção do pensar. Nestas reflexões foi encontrada uma “idéia um pouco louca”, a possibilidade de “injetar fluxos analógicos” na linguagem codificada digital. Para discutir esta idéia elegeram-se o trabalho do escritório holandês UNStudio, onde diagramas deleuzeanos e os seus chamados design models parecem buscar esta tal injeção de fluxos. Este artigo objetiva ampliar o debate sobre o impacto dos softwares de script no processo projetual arquitetônico, de modo mais específico sobre as consequências de uma possível mudança de paradigma orientada a um pensamento codificado e estrutural. Por fim, discute-se a possibilidade de inserção de fluxos analógicos neste pensamento estruturado: um “pecado” em prol da emergência da diferença.

Arquitetura e uma Reaproximação a Escrita: o Fortalecimento de um Modo de Pensar

O pesquisador italiano Mario Carpo tem se dedicado nos últimos anos (2001) (2003) (2011a) a estudar a relação da arquitetura com a escrita, como também os possíveis paradigmas relacionados com esta relação. Em 1998 Carpo

detecta uma primeira mudança na arquitetura com a introdução da imprensa mecânica (“Arquitetura na era da Imprensa”), neste período os códigos e algoritmos de herança medieval foram substituídos pelo poder da imagem autográfica dos desenhos impressos: paradigma albertiano (Carpo, 2011a). Contudo para este teórico, nova mudança se avizinharia à arquitetura nos tempos atuais, um movimento “do alfabeto ao algoritmo” (Carpo, 2011b), com o fortalecimento do código escrito nas atividades projetuais arquitetônicas passando de uma imagem autográfica para outra mais alográfica: “paradigma da variabilidade” (Carpo, 2011b).

Essa última transformação de paradigma estaria diretamente relacionada com um revivalismo dos programas de script (código) na última década, como detectado por McCullough (2006) e Silver (2006). A evolução dos programas com interface visual e com design shrink-wrapped possibilitou a popularização destes dentro do ambiente dos arquitetos, de modo que um arquiteto possa até mesmo nem saber quando está ou não produzindo um script (código) (McCullough, 2006) (Picon, 2010). Assim, a escrita e a linguagem codificada podem tornar cada vez mais presentes nos processos projetuais, ainda que nem sempre visíveis. Para Silver (2006) esta aproximação ao scripting permite gerar novos modos de expressão, onde em poucas linhas uma vida nova se abre. Na mesma direção, McCullough (2006) observa que a parte surpreendente do scripting é como ele pode acrescentar um nível extra ao pensamento projetual. Em direção semelhante, Peters e Kestelie (2013) comentam sobre a possibilidade do surgimento de um modo de pensar baseado no algoritmo e no código, como eles observam os computadores poderiam aumentar a inteligência do projetista, aumento sua capacidade para resolver problemas complexos e ainda lhe fornecer inspiração gerando resultados inesperados. Os dois autores definem uma espécie de “pensamento algorítmico” baseado em scripts, onde destacam a necessidade da exploração dos códigos de geração em prol da especulação de novas possibilidades.

Deste modo, esta reaproximação à escrita, como sua possível transformação de paradigma, retoma uma antiga e constante discussão do processo de projeto: a repetição de padrões (o regramento) e a inserção da diferença (a ruptura). Pois, como lembra Wolfram (Apud Silver, 2006), o script no final trata de um código cuja abstração geradora emula a formação de padrões dentro de um universo de variações. No entanto, por outro lado, sempre há a possibilidade de se pensar um modo de tentar jogar com as regras e escapar do padrão, como ressalta McCullough (2006), tudo que o arquiteto precisa é vontade de improvisar. Afinal, ainda que alguém possa argüir que os scripts ainda são fundamentalmente diferentes do modo de pensar humano, Picon (2006) observa como é ilusória a separação entre o impulso criativo e o conjunto de regras que capacita ao arquiteto controlá-lo. Como este investigador francês comenta, uma parte interna do pensamento está constantemente buscando escapar da regulação, enquanto a outra tenta funcionar em modo quase mecânico (Picon, 2006). As regras podem sempre engatilhar

um pensamento de escape e espontaneidade, ao mesmo tempo em que algum princípio regulador escondido possa emergir de um pensamento mais caótico.

Esta questão da diferença e da regra é tratada por Silver (2006), que enxerga nos programas de script a possibilidade da diferença dentro da universalidade, devido à variabilidade de operações existente dentro de um único instrumento. Neste caso a flexibilidade dos comandos e dos procedimentos poderia transformar uma função em outra, e a universalidade da programação suportaria a diferença, ao contrario da esterilidade associada com protocolos rígidos e procedimentos fixos (Silver, 2006). No entanto, a linguagem do pensamento, em face com o computador, permanecerá caracterizada pela estruturação do código e do significante, pois como recordam Peters e Kestelie (2013), o computador necessitará de instruções escritas em linguagem (código) para que este converse com o projetista. Neste caso, seria necessário questionar se Silver (2006) não estaria confundindo “diferença” com “diverso”, pois como lembra Deleuze (1988), não se deve cometer essa confusão, pois o diverso já é dado, enquanto a diferença é a razão do diverso, em outras palavras, aquilo que o gera. Por esta razão, por mais flexível que a linguagem seja, tratará de um tipo de pensamento estruturado e, assim, seria oportuno um “pecado” para tentar corromper a linguagem codificada e emergir realmente a diferença e a intuição, um alargamento da além do já reconhecido.

A Questão do Pecado e o Código: uma Incursão no Pensamento de Gilles Deleuze

Deleuze é conhecido por ser um dos filósofos dedicados a idéia da diferença, na busca de um pensamento afastado da banalidade cotidiana da reconhecimento (ou reconhecido), um pensamento verdadeiramente criativo que não representasse simplesmente as regras da identidade, da representação ou do clichê. Nesta direção sua filosofia busca meios para desfazer as representações de um mundo preexistente, desfazer semelhanças “... para produzir a ‘imagem presença’ (...) uma imagem sem semelhanças” (Deleuze, 2007, p. 101). Para explicar esta idéia de “imagem sem semelhanças” Deleuze recorre à tradição cristã e ao ensinamento que o homem seria feito à imagem e à semelhança de Deus. Contudo, nesta mesma tradição, o homem após “pecar” conservaria somente a imagem perdendo a semelhança: imagem sem semelhança. Deste modo, um “pecado” seria necessário para abrir lacunas na correspondência da reconhecimento, como por exemplo, na correspondência estabelecida pelos códigos.

Sobre este tema dos códigos e a imagem sem semelhanças, Deleuze em 1981 em seu curso na Universidade de Vincennes dedica-se a explicar os tipos de transporte de semelhança desde uma visão da pintura e do diagrama. De modo resumido, a representação de algo se torna semelhante a este algo através do transporte de semelhanças ou de similitudes de relação. No primeiro caso se repete alguma qualidade como a mesma cor, forma, textura e etc. No segundo caso, funciona

por isomorfismo, ou seja, trata-se de um fenômeno de similitude construído por convenção ou código. Como observa este pensador francês, o código é o princípio básico de uma linguagem, ele não é nada mais que unidades significativas determináveis por uma sucessão de eleições binárias. Por esta razão, Deleuze associa o código com a binarização da linguagem digital, onde é “isso ou aquilo” (0 ou 1), uma eleição binária que implica convenção.

Por outro lado, ainda existiria uma linguagem analógica que não simplesmente transportaria semelhanças, mas as produziria, “...uma analogia que produz uma semelhança por meios completamente distintos que o transporte de similitude” (2007, p.134). Nesta linguagem o destinatário da mensagem participa do processo de decifrar uma mensagem, pois ela trabalha com “isso e/ou aquilo”, onde é possível pensar mais de uma opção que deverá ser eleita e o código se tornaria relativizado. Para Deleuze a linguagem analógica cumpriria uma importante função no pensamento criativo, no entanto, em certo momento esboça uma hipótese de inserção de fluxos analógicos na linguagem codificada através de um diagrama que agiria como um “pecado” que abre lacunas na lógica de correspondência do código, pois desfaria a semelhança e introduziria escapes.

Desse modo, apesar de Deleuze afirmar que “...não concebo códigos que não impliquem ou não produzam fenômenos de similitudes”, em um momento específico de seu curso (5 de maio de 1981), o filósofo imagina “...uma idéia um pouco louca: codificar relações analógicas. Ou seja, uma linguagem que segue sendo analógica, mas que passa por um código...” e comenta que poderia existir um interessante modo de “...injetar fluxos analógicos” (Deleuze, 2007, p.139). Assim, seria necessário pensar por meios que não se resumissem a convenções ou códigos, como a escrita da linguagem computacional, para emergir a diferença e um pensamento realmente novo. Uma espécie de diagrama (pecado) que poderia atravessar a semelhança que o pensamento codificado produz e, assim, faça surgir a diferença. Ainda que não passasse de uma hipótese sem qualquer desenvolvimento por parte de Deleuze, seria interessante produzir alguma reflexão sobre esta “injeção” desde a experiência do escritório holandês UNStudio.

A Experiência do UNStudio

UNStudio trata-se de um escritório de arquitetura que já começou informatizado e como explicou seu sócio fundador Ben van Berkel, desde o começo o computador sempre consistiu em uma maneira de romper radicalmente com a tradição (Berkel, 1995). Nesta direção experimental, logo decidiram dar um passo adiante, os paramétricos surgem no escritório antes do surgimento dos programas conhecidos no mercado. No entanto, como explica um de seus arquitetos-sênior, Nuno Almeida (2013), para eles a parametria parece ter surgido inicialmente mais como um modo de aumentar o controle sobre todo o processo de projeto, um meio de regar

o processo, “...nós não fazemos os paramétricos para sermos surpreendidos, os fazemos para não sermos surpreendidos” (Almeida, 2013). Como observa Sollazo (2011), este escritório muitas vezes cria um objeto virtual que pode se converter em uma verdadeira réplica editável do edifício arquitetônico projetado.

No entanto, ao mesmo tempo em que os modelos paramétricos os ajudaram a aumentar o controle, eles também se revelaram como importantes mecanismos para auxiliar o reaproveitamento de algumas de suas experiências e perpetuar o espírito de “studio” do escritório. A partir de 2001, após passar pela primeira vez por um processo de certificação de qualidade, o escritório sentiu a necessidade de resgatar sua própria história e os seus processos de trabalho, “para detectar as linhas que conectavam um projeto a outro” (ALMEIDA, 2013). Nesta época foi necessário pensar meios para dividir o conhecimento projetual gerado internamente (“sharing knowledge”), pois estava difícil intercambiar informação em um escritório com mais de 100 arquitetos e ainda manter um perfil mais experimental. Deste modo, surgiu uma rede interna para troca de experiências entre todos os integrantes, um espaço onde se podia arquivar estratégias e táticas para serem repetidas. Como resultado de muitas discussões internas surge algo que logo será chamado de design models.

Pouco a pouco, o design models foi sendo usado e reusado através de diagramas, pois foram “...encontrando certos diagramas que poderiam ser manipulados e aplicados de diferentes modos” (Berkel y Bos, 2006, p.16). Pois, em “uma tentativa inconexa de continuar apresentando o projeto paramétrico como uma vanguarda estimulante” (2006, p.14), os design models se aproximam à parametria. Mas, como explica Berkel (2012), os design models não devem ser pensados formalmente, como um tipo de organização que poderia permear todo o processo de projeto, porque a “forma tem que estar informada!” (2012). Os diagramas do escritório parecem penetrar nas formas parametrizadas, no entanto não se tratavam de qualquer diagrama, para o escritório era necessário pensar em diagramas não representacionais, uma imagem em principio a-significativa que poderia superpor às idéias estruturais arquitetônicas reconhecíveis e transformá-las (Duarte, 2015). Em outras palavras, o emprego do diagrama por UNStudio se aproxima claramente à idéia de Deleuze e seu emprego analógico, “...nós usamos extensivamente os escritos de Deleuze, ainda (...) que nossa leitura seja especificamente arquitetônica” (Berkel y Bos, 2006b, p. 111).

Deste modo, este diagrama teria a função de manter o caráter experimental e buscar a inovação, como lembram os arquitetos do UNStudio a “inovação existe! Mas você tem que entender que hoje a inovação é impossível apenas por sua conta própria” (Berkel y Bos, 2006, p.16). Os arquitetos entendem a necessidade de empregar algum esforço em busca de fazer emergir a “diferença” por meio de diagramas, afinal “...quanto mais diferenças tenha no nosso trabalho melhor” (Berkel, 2012). Seus diagramas procuram romper com o reconhecimento imediato, como já observava Berkel

em 1998 “um diagrama produz novos significados” (p.71). Estes diagramas possuíam uma clara intenção de aproximação à linguagem analógica, de tal modo que procuram uma atuação sobre a linguagem codificada dos modelos paramétricos do escritório, para produzir escapes no pensamento codificado. Através dos diagramas injetados nos design models “... significados são formados e transformados interativamente” (Berkel y Bos, 2006b, p.113).

A linguagem alográfica do analógico abria escapes para a criação dentro do escritório, como o diagrama de garrafa de Klein no projeto da estação de Arnhem (1996-2014). Este projeto [fig.1], tendo seus fluxos trabalhados por diagramas convencionais por um ano, foi sobreposto a um diagrama da garrafa de Klein com objetivo de produzir novos modos de conexão de seus diferentes níveis. A introdução de um diagrama que capturasse esta organização técnica/espacial (Berkel y Bos, 2006b), como observa Sollazzo (2011), o escritório UNStudio começou a entender que a percepção da linha em um estrutura convencional é limitada, necessitando expandir seus limites, de modo que começaram a usar diagramas que “(re) definem estruturas” (Berkel y Bos, 2006, p.15). Todo o trabalho parametrizado dos fluxos funcionais era sobrecarregado por uma construção a-significante arquitetonicamente. A idéia seria entender as forças que parecem descodificar os processos conhecidos de continuidade e conectividade na arquitetura.

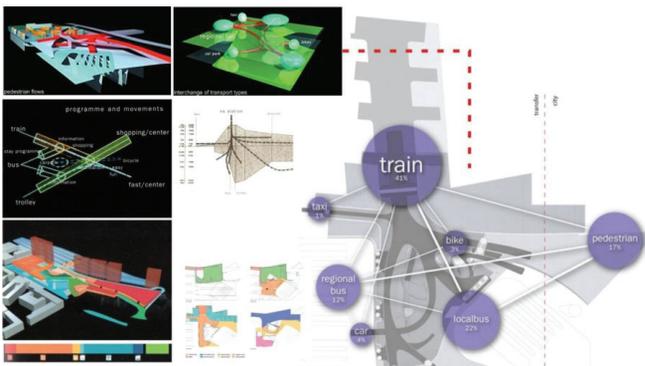


Fig.1 O diagrama dos diversos trajetos do projeto da estação de Arnhem (1996-2014). Fonte: Duarte, 2015.

Em outro exemplo, no museu Mercedes Benz (200), seu diagrama parecia funcionar para ajudar a retardar a implacável intrusão do “tipo”, ou seja, da imagem dos signos e do reconhecimento. Para Berkel y Bos “os tipos já não oferecem soluções adequadas para as demandas e situações contemporâneas” (2006, p.15). Por esta razão, ainda que os tipos sejam parametrizados nas fases iniciais do trabalho (Almeida, 2013), em algum momento do processo o diagrama trabalharia para escapar da estrutura reconhecível e dos modos cotidianos de viver, tentando fazer emergir a “diferença”. Ao final, parece que o diagrama se mescla ao processo produzindo objetos híbridos. Aliás, este conceito parece caracterizar a produção do UNStudio, pois esta idéia de híbrido estimularia um tipo de sensação irreconhecível e incomum, algo que os signos não

podem alcançar por gerar reconhecimento, um gatilho que ativaria o processo criativo, “são híbridos que conhecem não sua história (...) uma nova noção de identidade” (Berkel y Bos, 1999a, p. 383).

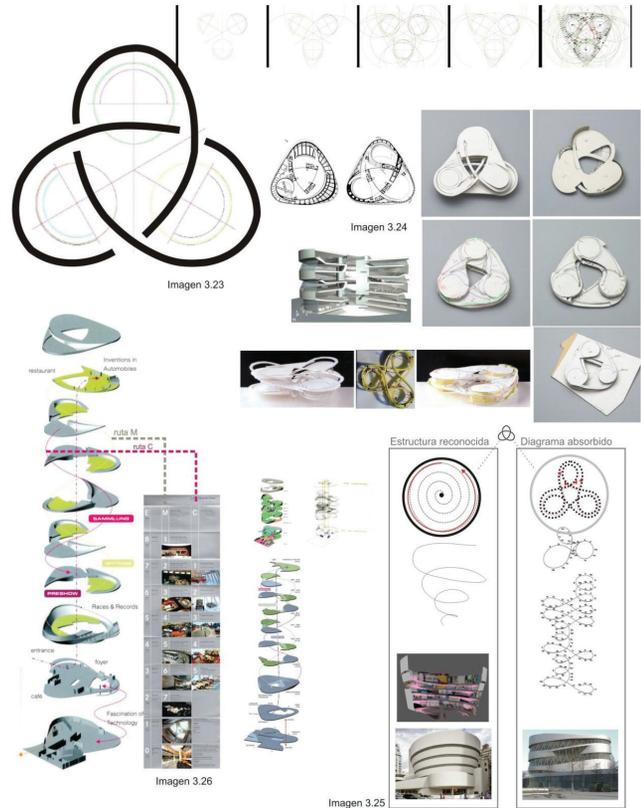


Fig. 2: Tipos e diagramas no Museu Mercedes-Benz. Fonte: Duarte, 2015.

Conclusão

Como lembra Picon (2006), não se deve esquecer que o processo criativo na arquitetura funciona de modo dialético tanto no regramento e no código, como no caos criativo e no a-significativo. Como também não se deve esquecer da nossa proximidade ao modo de pensar regrado da máquina. Talvez por esta razão, essa dialética parece cada vez mais presente na relação entre a inteligência da máquina e a do projetista, como por exemplo, na dialética entre o script (funcionando como um código genótipo) e variações paramétricas (funcionando como adaptações fenotípicas). No entanto, como em qualquer mudança de paradigma, deve-se refletir sobre “o que” e “como” se está transformando o modo de pensar. É necessário entender que a aproximação ao pensamento codificado não deve afastar processos mais sistematizados ou dedicados ao pensamento analógico, em prol de encontrar meios conscientes de romper a visão codificada e alcançar a “diferença”, como parecem buscar os arquitetos do UNStudio. Também se faz necessário não confundir a “diversidade” existente nos objetos paramétricos (variabilidade) com a “diferença” que a faz possível. Por fim,

ausente de qualquer sentimento saudosista ou romântico, é necessário pensar modos de aproximar ao pensamento alográfico e a-significante ao analógico para apontar futuras e novas miscigenações e oxigenações para o conhecimento projetual arquitetônico digital.

Referências

Carmo, Mario. Architecture in the age of printing: orality, writing, typography, and printed images in the history of architectural theory. Cambridge (Mass.): MIT Press, 2001.

_____. Del alfabeto al algoritmo: Sobre la autoría digital y el diseño paramétrico. Arquitectura Viva. África Esencial. Modernidad y tradición: un viaje por seis geografías. Madrid, n. 140, 2011b, p.112.

_____. The alphabet and the algorithm. Cambridge/Massachusetts: MIT Press, 2011a.

Deleuze, Gilles e Guattari, Felix. Mil Mesetas: Capitalismo y Esquizofrenia. Valencia: Pre-textos, 2004.

Deleuze, Gilles. Diferença e Repetição. Rio de Janeiro: Graal, 1988.

_____. Pintura: el concepto de diagrama. Buenos Aires: Cactus, 2007.

Duarte, Rovenir. El diagrama arquitectónico despues de Deleuze: estudio de casos holandeses. Tese de doutorado, ETSAB-UPC, Barcelona, 2015.

Sollazzo, Andrea. Digital Van Berkel: Diagrams, Processes, Models of UNStudio. Milan: IT Revolution in Architecture, 2011.

UNStudio. Buy me a Mercedes-Benz: the book of the museum. Barcelona: Actar, 2006.

Berkel, Ben van. Between ideogram and image-diagram (entrevista con Like Bijlsma, Wouter Deen en Udo Garritzmann). En: OASE: Diagrams (Rotterdam), n.48, 1998, p. 63-71.

_____. Conversación vía modem con Ben van Berkel (por Greg Lynn). El Croquis: Ben van Berkel 1990-1995, n.72[I],

Madrid, 1995, p.6-15.

_____. Interview Ben van Berkel/ UNStudio (por Florian Heilmeyer). En Crystal Talk. BauNetz, 2012. Disponible en: <http://www.baunetz.de/talk/crystal/index.php?lang=en&cat=Interview&nr=19>.

Berkel, Ben van y Bos, Caroline. Diagrams: Interactive Instruments in Operation. In: Lotus Internacional: Diagrams (Milan), n. 127, 2006b, p.106-113.

_____. Move. Amsterdam: UN Studio & Goose, Vol.1, 1999.

_____. Move. Amsterdam: UN Studio & Goose, Vol.2, 1999b.

_____. Move. Amsterdam: UN Studio & Goose, Vol.3, 1999c.

_____. UN Studio: design models, architecture, urbanism, infrastructure. London: Thames & Hudson, 2006.

Entrevista com Nuno Almeida (Senior Architect of UNStudio) realizada em 20 de junho de 2013, em Rotterdam, pelo autor deste artigo.

Peters, Brady e Kestelie, Xavier de. Editorial. In: Architectural Design: Computation Works: The Building of Algorithmic Thought. Brady Peters e Xavier de Kestelie (Eds.), London, Abril, 2013, 152pág.

Mccullough, Malcolm. 20 years of Scripted Space. In: Architectural Design: Programming Cultures: Architecture, Art and Science in the Age of Software Development, July/August. London, 2006.

Picon, Antoine. Forward to Algorithmic Architecture. In: Terzidis, Kostas. Algorithmic Architecture. Press, Oxford, 2006.

Silver, Mike. Towards a Programming Cultures in the design arts. In: Architectural Design: Programming Cultures: Architecture, Art and Science in the Age of Software Development, July/August. London, 2006.