

Explorando o diagrama de bolhas

Exploring the bubble diagram

► Pedro Luis Alves Veloso
pedroveloso13@hotmail.com

Abstract

This paper presents an interactive bubble diagram developed to support design exploration. The proposed diagram consists of multiple goal-oriented agents, whose interaction with the user stimulates the analysis and exploration of new spatial patterns. Until now, there are two versions implemented. The first was developed combining scripting language and a graphical editor of algorithms embedded in modeling software. The second was developed with event-driven programming techniques emphasizing functions to amplify the interaction between user and agents. Finally, as the result of this exploratory stage of the research, we propose new paths of development for the next versions of the diagram.

Keywords: architectural design; bubble diagram; graph; interaction

Um século de bolhas

Um diagrama de bolha é uma técnica gráfica adotada pelos arquitetos para investigar o edifício como uma rede de elementos funcionais. Usualmente, espaços destinados às atividades previstas no programa são representados como bolhas, cujas medidas podem indicar a sua área e insinuar sua forma preterida. As diversas bolhas são conectadas por linhas que podem indicar propriedades como acessibilidade, conexão visual ou adjacência.

No início do século XX, diversos arquitetos funcionalistas propunham uma revisão dos procedimentos de projeto arquitetônico, a partir da incorporação de princípios e técnicas científicas, como o diagrama de bolhas. Esse diagrama era uma das técnicas gráficas advindas da taxonomia biológica, que consolidava uma visão sobre as entidades orgânicas como uma rede funcional (Emmon, 2006). Sua adoção no projeto era coerente com a doutrina de que a forma arquitetônica seria uma dedução objetiva do programa por métodos científicos (Garcia, 2010; Lueder 2012), resultando em uma expressão funcional análoga a dos organismos e máquinas. Essa doutrina pretendia lidar com a crescente complexidade topológica e funcional da cidade industrial.

Particularmente, nos Estados Unidos e na Inglaterra, essa abordagem objetiva se expandiu dentro de um ambiente intelectual que pretendia compreender a “arquitetura como um ramo autônomo do conhecimento tecno-científico” (Ockman & Sachs, 2012) e que propunha métodos racionais de projeto. Na década de 1950 e 1960, muitos dos métodos computacionais pioneiros incorporavam uma estrutura de dados análoga ao diagrama de bolha, o grafo. O grafo

consiste de um conjunto de vértices que representam os nós e de um conjunto de pares de vértices que representam as conexões. Ele pode ser representado graficamente por um diagrama semelhante ao diagrama de bolhas, mas sem as dimensões das bolhas. Ele foi largamente utilizado para lidar com a análise e síntese de configurações espaciais, dando suporte a técnicas de otimização e automação de projeto (Steadman, 1976; Mitchell, 1977).

Nas décadas de 1960 e 1970, a arquitetura moderna e os métodos racionais de projeto – o que inclui a análise e a síntese espacial a partir de grafos e diagramas de bolhas - sofreram duras críticas. Por exemplo, a vertente formalista da arquitetura norte-americana passou a defender uma leitura do diagrama proveniente da análise formal, da Gestalt e da Semiótica, repudiando o uso de diagramas funcionais (Garcia 2010; Somol, 1999; Lueder 2012).

Além disso, no âmbito das pesquisas computacionais surgiram propostas que, em detrimento da otimização de soluções espaciais, utilizavam grafos e diagramas de bolhas para criticar a estrutura tradicional do projeto e a autoridade do arquiteto. Podemos destacar o Flatwriter, desenvolvido pelo arquiteto Yona Friedman (1980), ou o programa YONA, desenvolvido por Guy Weinzapfel e Nicholas Negroponte (1976) no Architecture Machine Group (MIT, 1967-1980).

Atualmente, poucas pesquisas computacionais tratam de diagramas funcionais. Podemos destacar aqui o trabalho *Floating Bubbles*, de Hua Hao e Jia Ting-Li (2010), que automatiza o funcionamento do diagrama de bolhas por meio de um sistema baseado em agentes, e o *Realtime Physics for Space Planning* (Syp,

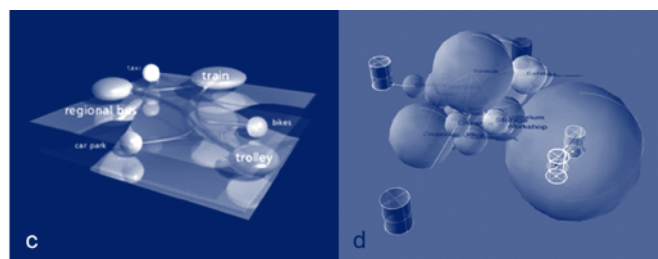
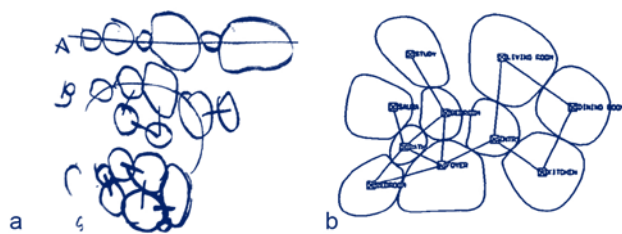


Figura 1: Diagramas de bolhas no último século: a) Le Corbusier, em 1929 (2004, p. 218); b) YONA (Weinzapfel & Negroponete, 1976, p.77); c) UNStudio (Berkel & Bos, 1999, p. 667); d) Realtime Physics for Space Planning (Syp, 2010).

2010), que utiliza um *spring algorithm* para criar um diagrama de bolhas tridimensionais. Outro desdobramento dos diagramas de organização funcional, são as propostas de escritórios contemporâneos como o OMA, o UNStudio ou o SANAA. Em contraste com a pretensa neutralidade dos diagramas funcionais do começo do século, tais escritórios propõem diagramas personalizados e explicitamente adaptados à sua prática.

O diagrama de bolhas como um jogo

Frete à apropriação crítica do diagrama de bolhas na arquitetura contemporânea e à atual domesticação de recursos computacionais, iniciamos uma pesquisa exploratória que visa propiciar uma maior familiaridade com o funcionamento e potencialidades do diagrama. Nossa pesquisa enfatiza a capacidade de o diagrama interagir com o projetista e de ser flexível para representar distintos elementos do projeto.

Nesse *paper*, apresentamos a primeira etapa de desenvolvimento do diagrama de bolhas computacional, fundamentado na interação entre projetistas e agentes artificiais baseados em objetivos. Enquanto as bolhas (agentes) buscam manter relações espaciais determinadas por protocolos internos, o projetista mantém uma visão do todo e busca novas soluções. Essa interação com agentes, não pretende automatizar a geração de layouts ou configurações espaciais “ótimas”. Ao contrário, por meio de configurações sub-ótimas, o diagrama pretende estimular um olhar diferenciado sobre o programa e a investigação de novos padrões espaciais no ciclo de projeto.

0.1

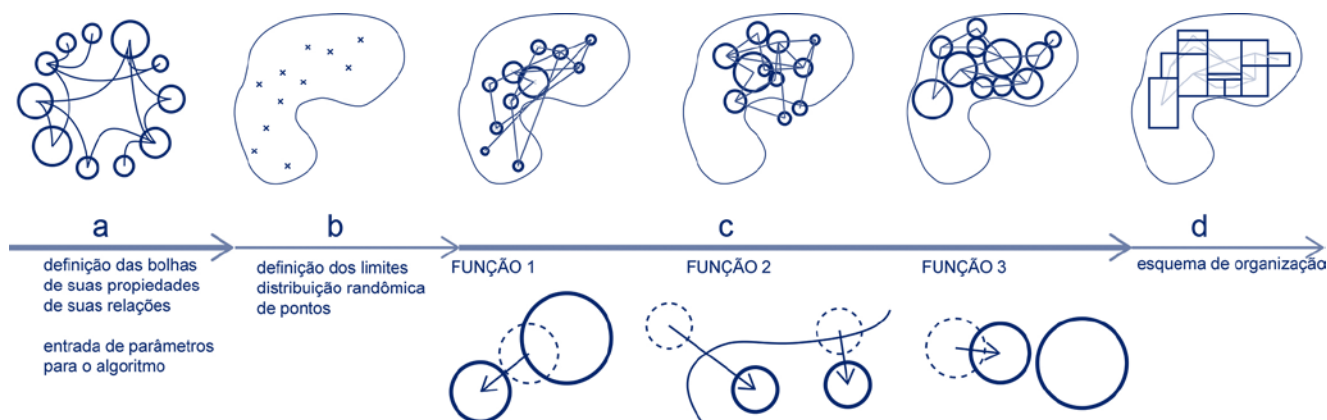
Na primeira versão do diagrama, utilizamos os recursos de modelagem disponibilizados pelo *Rhinceros* e o seu editor gráfico de algoritmos (*Grasshopper*) para desenvolver os elementos geométricos do diagrama. Utilizamos também a linguagem Python para criar o algoritmo de atração que, além de armazenar as propriedades das diversas bolhas, estabelece sua interação e movimento.

Nessa versão, o comportamento do diagrama de bolhas é controlado por um segundo diagrama em uma área isolada do modelo. Nesse diagrama de controle, o projetista pode desenhar círculos que representam as bolhas, e curvas para realizar as conexões (figura 2a e figura 3a). A qualquer momento, o projetista pode alterar esses círculos e curvas, modificando o número, a área das bolhas e seus elementos de conexão. Nesse sentido, o diagrama controlador permite atualizar o *input* do algoritmo de atração e, conseqüentemente, a geração de variações sobre o diagrama de bolhas.

Outro recurso de interação disponibilizado é o controle das próprias bolhas no diagrama principal. O projetista escolhe algumas bolhas que serão controladas por ele e não pelo algoritmo. Essas bolhas podem ser utilizadas para controlar elementos do projeto ou do entorno que tenham uma posição pré-definida, preferencial ou mesmo elementos que o projetista pretenda alterar intuitivamente.

O algoritmo de atração é encapsulado em um componente gráfico do plug-in *Grasshopper* e recebe os parâmetros de entrada estipulados pelo diagrama de controle e por outros recursos do *Grasshopper*.

Figura 2: princípio de funcionamento do diagrama.



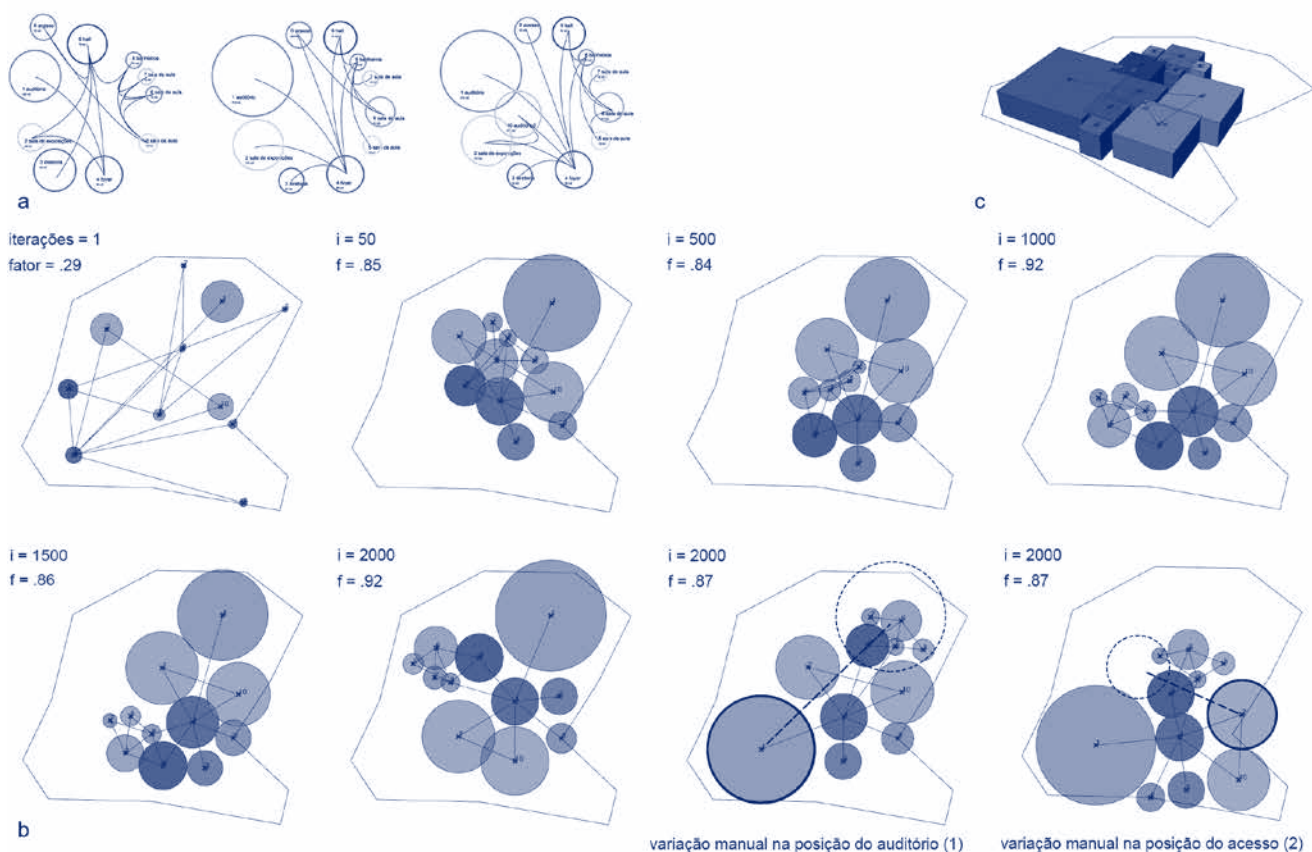


Figura 3: a) diagrama de controle; b) exemplo de funcionamento do diagrama de bolhas 0.1; c) estudo de massa com resultado de “b”.

Por exemplo, o parâmetro “iter” controla o número de iterações do algoritmo, o “crv” recebe uma curva com o limite da área do diagrama de bolha, o “initial_points” recebe a posição dos pontos iniciais das bolhas, o “radius” recebe os raios respectivos de cada bolha e o “rel_tree” recebe uma árvore de dados com as relações entre as bolhas definidas no diagrama de controle.

O algoritmo funciona de uma maneira simples. As bolhas são organizadas como agentes compostos por uma lista de pontos (“initial_points”) e também por uma lista de raios (“radius”), que definem o desenho de sua circunferência. Cada bolha age conforme seus objetivos próprios, disputa e coopera com as demais bolhas para buscar suas posições dentro da área delimitada (“Crv”), respeitando a área das bolhas vizinhas. Para cada bolha, o algoritmo analisa a vizinhança, executando três funções, que geram três vetores de movimento (figura 2c): a função 1 gera um vetor de repulsão a partir da bolha sobreposta mais próxima; a função 2 gera um vetor de atração da bolha para dentro da área limite; a função 3 gera um vetor de atração para cada bolha conectada.

Com o intuito de possibilitar que as ações individuais não restrinjam o comportamento global, foi desenvolvido um fator objetivo (dentro do intervalo de 0 a 1). Quando o fator está com um valor reduzido, o algoritmo está em fase de exploração e as bolhas se reduzem e podem se mover livremente pela vizinhança ou fora da

área limite em busca da melhor posição. Quando o fator aumenta, as bolhas entram na etapa de consolidação e começam a crescer e se tornar menos permeáveis às demais bolhas e à curva limite. O fator pode estar associado ao sucesso individual da bolha, à média do sucesso das diversas bolhas ou a um intervalo de exploração definido pelo usuário.

Devido às limitações em relação ao controle do usuário e ao custo de memória para rodar iterações mais longas, a pesquisa optou por desenvolver um segundo modelo em um ambiente de programação orientada a eventos que disponibilizasse uma interface gráfica do utilizador.

0.2

Essa segunda versão do diagrama de bolhas foi desenvolvida em *Processing*, enfatizando a interação entre o projetista e os agentes. Os agentes se comportam a partir das mesmas regras definidas na versão anterior, pautada na soma dos três vetores de movimento e, por sua vez, o fator objetivo ficou restrito ao sucesso individual.

O diagrama se inicia com uma tela vazia e o usuário pode interagir com as bolhas por meio de diversas funções: criar bolha; criar bolha randômica; remover bolha; mover uma bolha; (des)conectar um par de bolhas; (des)travar uma ou todas as bolhas; criar uma área limite para a interação entre bolhas; criar um sistema de circulação para a interação entre as bolhas (figura 4).

Devido ao uso de uma *loop* principal, os agentes realizam uma disputa ininterrupta, sem sobrecarregar a memória, enquanto

eventos específicos (como clicar em uma bolha ou ativar uma função) interferem no funcionamento do diagrama. Nesse sentido, o usuário e os agentes agem em tempo real, propiciando a exploração conjunta de alternativas.

Em detrimento de um diagrama de controle separado para organizar as bolhas, o usuário interage com a própria representação gráfica do diagrama de bolhas, promovendo uma compreensão mais intuitiva das consequências de suas decisões. Por exemplo, o usuário pode criar um conjunto de bolhas que representem as funções do edifício, o limite do terreno, além de travar uma determinada bolha que pode representar o acesso. Uma vez travada, essa bolha fica estável em uma posição, mas mantém sua influência no comportamento das demais. O usuário pode testar diversas opções de acesso rapidamente, verificando em tempo real as consequências para a organização do programa. É possível também alterar as conexões e a posição de qualquer bolha em tempo real, de modo que novas hipóteses sobre o programa possam ser colocadas e testadas.

Nota-se que foram definidas algumas estratégias *top-down*, como, a definição de um polígono limite (figura 4g-i) ou da circulação. Outra estratégia *top-down* disponível é o fator de permeabilidade. Em complemento ao fator objetivo, que é calculado no protocolo de cada bolha, o usuário também pode utilizar o mouse para interferir na capacidade das bolhas se cruzarem e saírem do terreno.

Em relação primeira versão do diagrama, notamos um considerável aumento de interatividade e versatilidade na produção e customização de um diagrama.

0.3 e adiante

Apresentamos como desdobramento do trabalho até aqui realizado, o delineamento de novas estratégias para ampliar a capacidade de interação e flexibilidade do diagrama. Tais desdobramentos da pesquisa exploratória pretendem contribuir para delinear

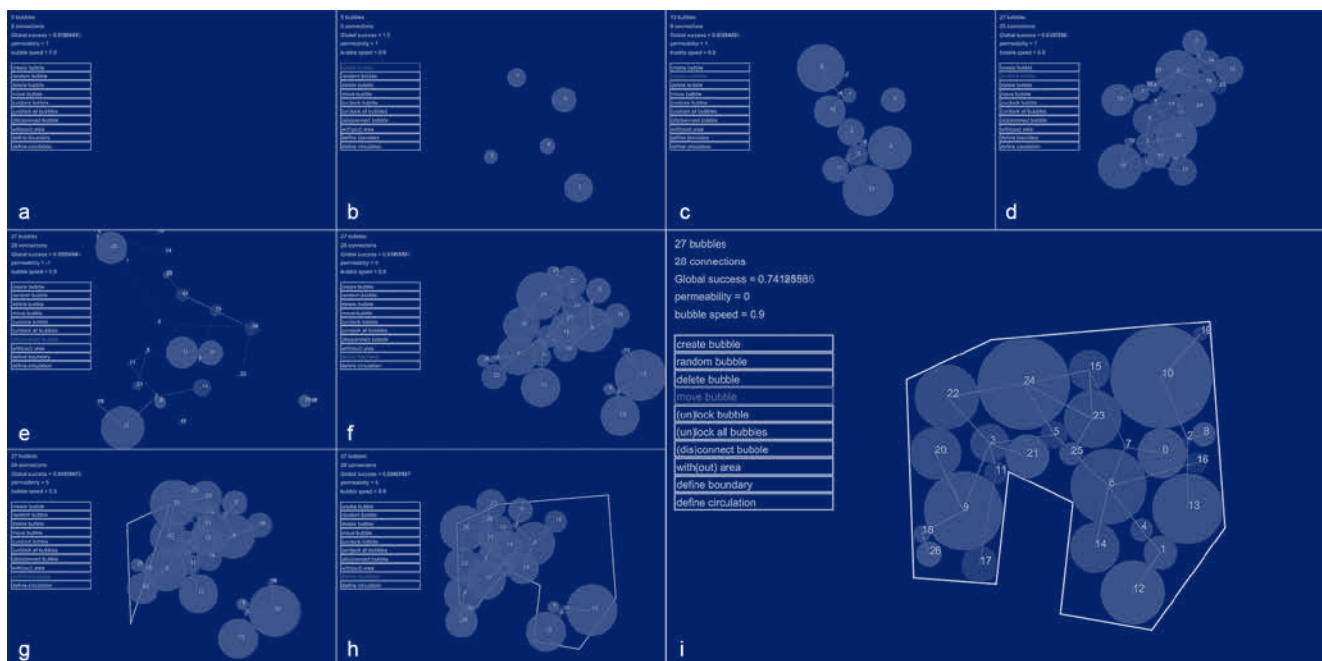
as variáveis de nossa hipótese, tornando possível, no futuro, controlá-las e testá-las em condições experimentais adequadas.

Um primeiro aspecto a ser desenvolvido é a possibilidade de organização das bolhas. Pretendemos propiciar maior controle sobre as funções que definem o vetor de movimento dos agentes, propiciando a movimentação das bolhas em relação a critérios distintos, como a exposição solar ou o fluxo de pedestres. Além disso, programaremos algumas estratégias *top-down* como a configuração geral aleatória ou o desenrolamento do diagrama.

Pretendemos extrapolar a relação entre um elemento funcional e o domínio de uma bolha, sem perder a eficiente lógica isotrópica de organização dos agentes em torno do círculo. Nesse sentido, duas estratégias são propostas: (a) múltiplas bolhas e (b) incorporação. Uma vez que um elemento pode ser representado por mais de uma bolha (a), o campo de possibilidades cresce com o número de bolhas e com os padrões de conexões entre elas. Cada elemento funcional pode ser composto, então, por uma bolha isolada, uma fileira de bolhas, um padrão poligonal fechado ou mesmo um aglomerado com conexões completas. Por sua vez, a incorporação (b) permite que uma bolha faça parte do espaço interno de outra, de modo que o diagrama de bolhas possa representar elementos funcionais em distintas escalas, grupos de afinidade, ou mesmo a fusão de dois elementos funcionais em uma mesma bolha.

Junto a essa estratégia de organização, pretendemos ampliar a representação visual do diagrama para estimular a percepção de distintas espacialidades. Começamos a desenvolver duas estratégias visuais alternativas: *metaballs* e o diagrama de Voronoi. A representação da bolha como objeto *metaball* tem por intuito desmanchar os limites das bolhas e estabelecer visualmente manchas de influência entre elas. Ao contrário, a representação de diagrama de Voronoi delimita o espaço entre as bolhas, transformando os círculos e os

Figura 4: exemplo de funcionamento do diagrama de bolhas 0.2.



espaços adjacentes em polígonos que ocupam completamente a área de interação dos agentes, permitindo avaliar com precisão a área disponível para cada elemento.

Outra proposta essencial é a distinção entre espaço de movimentação dos agentes e o espaço de conexão ou de busca entre eles. Essa distinção surgiu quando iniciamos o desenvolvimento uma estratégia *top-down* do sistema de circulação. Nessa função, o usuário pode definir uma *polyline* que representa a circulação principal do edifício, de modo que as bolhas realizarão sua busca no espaço dessa circulação, estimulando certas configurações gerais ou tipos pré-definidos. Nesse caso, as bolhas continuam a se movimentar no espaço bidimensional, mas também consideram a lógica espacial do sistema de circulação. Essa distinção entre espaço de movimentação e espaço de busca possibilita associar um diagrama tridimensional de bolhas não apenas ao “espaço” de uma *polyline*, como também o de um modelo geométrico parametrizado composto por pisos, rampas e outros elementos verticais de circulação. Nesse sentido, a bolha se moveria livremente no espaço, mas sua avaliação da distância em relação às demais bolhas ocorreria no espaço delimitado por essa estrutura, com os diversos custos de movimentação associado ao piso e aos elementos de circulação vertical. O diagrama funcionaria como o proposto por Syp (2010), mas sem as restrições do *spring algorithm* e com um espaço de busca adequado ao arranjo de um edifício.

Essa possibilidade pode se desdobrar em um uso ainda mais radical do diagrama: funcionar como um “*design pattern*” (Woodburry, 2010) para personalizar uma estrutura tridimensional parametrizada. A partir da interação entre as bolhas e o projetista, um modelo geométrico se adaptaria às relações topológicas estabelecidas no diagrama, dobrando, reduzindo ou alterando pisos e modificando os elementos de circulação. Nesse horizonte, o diagrama de bolhas funcionaria como um jogo para a investigação de complexos multifuncionais personalizados.

Agradecimentos

À professora Gabriela Celani e ao professor David Sperling.

Referências

- Berkel, B. V., & Bos, C. (1999). *Move*. Netherland: Goose Press.
- Emmons, P. (2006, Set.). *Embodying networks: bubble diagrams and the image of modern organicism*. *Journal of Architecture*, 11(4), p.441-461.
- Friedman, Y. (1980). *Towards a Scientific Architecture*. Cambridge, MA: The MIT Press.
- Garcia, M. (2010). *The Diagrams of Architecture*. Chichester, John Wiley & Son.
- Hua, H., & JIA, T. *Floating Bubbles: An agent-based system for layout planning*. *Anais do 15º congresso CAADRIA, 2010*, p.175-183.
- LE CORBUSIER. (2004) *Precisões sobre o estado presente da arquitetura e do urbanismo*. São Paulo: Cosac & Naify.
- Lueder, C. (2012). *Diagram Ecologies: Diagrams as Science and Game Board*. *Diagrams. Lecture Notes in Computer Science*, v. 7352, 214–232.
- Mitchell, W. J. (1977). *Computer-aided Architectural Design*. New York: Petrocelli Charter.
- Ockman, J., & Sachs, A. (2012). *Modernism Takes Command*. In J. Ockman (Ed.) *Architecture School: Three Centuries of Educating Architects in North America* (pp. 120-159). Cambridge, The MIT Press.
- Somol, R. E. (1999). *Dummy Text or the Diagrammatic base of Contemporary Architecture*. In P. Eisenman, *Diagrams Diaries* (pp. 6-25). New York, Universe Publishing.
- Steadman, P. (1976). *Graph-theoretic representation of architectural arrangement*. In L. March (Ed.), *The Architecture of Form* (pp. 94-115). New York, Cambridge University Press.
- Syp, Marc. (2010) *Architecture: Realtime Physics for Space Planning*. Resumo da proposta e vídeo. Disponível em: <<http://vimeo.com/15563685>>. Acesso em: abr. 2014.
- Weinzapfel, G., & Negroponte, N. (1976). *Architecture-by-yourself: an experiment with computer graphics for house design*. *Proceedings of the 3rd annual conference on Computer graphics and interactive techniques*, 74-78.
- Woodburry, R. (2010). *Elements of Parametric Design*. Oxon: Routledge.