HUB eventos 2020
SIIMI/2020
#19.ART
RETINA
dat DESIGN, ART AND TECHNOLOGY

Padrões emergentes criados a partir de algoritmos de funções randômicas

Emergent patterns generated from random functions algorithms

Rui Alão[142]

Resumo
Este artigo explicita como experimentos feito através de código com funções randômicas pode lançar luz sobre o processo de emergência em sistemas complexos adaptativos. Através destes algoritmos o pesquisador é capaz de simular comportamentos que podem gerar padrões emergentes relevantes para o campo do design.
Palavras-chave: complexidade, algoritmos, padrões, emergência.

Abstract/resumen/resumé
This article explains how the experiment done through code with random functions can shed light on the emergence process in complex adaptive systems. Through these algorithms the researcher is able to simulate behaviors that can generate emerging patterns relevant to the design field.
***Keywords**: complexity, algorithms, patterns, emergence.*

Introdução

Nossa pesquisa desenvolvida nos últimos anos é ligada a métodos projetuais em design que possam dar conta de problemas de grande complexidade. A percepção de que a área de design está mal equipada para lidar com problemas realmente complexos está se tornando consensual e tem dado origem a algumas pesquisas na área desde a década de 80. O assunto complexidade no contexto da teoria dos sistemas é muito pouco abordado nos cursos de design em geral, mesmo quando falamos nos programas de pós graduação, tanto dentro quanto fora do país.

Por complexidade aqui me refiro a contextos em que se lida com um número muito grande de agentes ou autômatos celulares com comportamento próprios e que disparam eventos próprios dentro de sistemas complexos adaptativos, como eventos em cascata e o surgimento de padrões emergentes. O estudo da complexidade pode ser de grande interessa para a área de design uma vez que há possibilidade de algum tipo de analogia

[142] Rui Alão é designer digital, sócio do estúdio Carambola Digital, arquiteto pela FAU USP, especialista e mestre em Design pela Universidade Anhembi Morumbi e doutor na área de Design e Arquitetura na FAU USP. Professor de graduação e pós graduação do Centro Universitário Belas Artes de São Paulo e na Universidade Anhembi Morumbi. Participa como pesquisador do MediaLab (UAM) e do Grupo de Pesquisa Design e Convergência (Belas Artes).

entre os fenômenos de auto-organização e os métodos projetuais, como será discutido adiante.

Fenômenos dos sistemas complexos

Os eventos em cascata, também chamados de "grandes eventos", são típicos dos sistemas complexos adaptativos. Eles ocorrem quando um evento dispara uma série de outros eventos e estes outros, gerando o chamado efeito borboleta. Os efeitos borboleta tendem a desestabilizar os sistemas e a deixá-los mais suscetíveis a estímulos de qualquer natureza. Existem eventos em cascata em vários domínio, desde as avalanches até as crises econômicas de grande alcance, como a de 2008, que deixa seus rastros até os dias de hoje.

O extremo oposto dos eventos em cascata são as demonstrações de resiliência. Estes fenômenos ocorrem quando um sistema é atingido por um estímulo muito poderoso e, logo que este desaparece, desaparecem também os efeitos destes estímulos. É assim quando jogamos uma pedra num lago e sua superfície se reconstrói logo após o evento. Outro exemplo é o atentado contra as Torres Gêmeas em Nova Iorque em 2001. Neste último caso, a cidade ficou muito abalada por um ou dois dias, mas logo depois voltou ao seu cotidiano, reconstruindo suas relações e demonstrando sua resiliência. Este fenômeno é característico dos sistemas complexos muito interconectados que se reconstruam rapidamente.

Um outro fenômeno típico dos sistemas complexos adaptativos é a emergência, isto é, o surgimento de um padrão a partir das interações dos seus elementos constitutivos. John Holland é um estudioso da complexidade em geral e é um pioneiro dos algoritmos generativos. Nas palavras dele,

> *Emergência é, sobretudo, produto de interações relacionadas e dependentes de contexto. Tecnicamente estas interações, e o sistemas resultante, são não lineares: o comportamento do sistema maior não pode ser obtido pela soma dos comportamentos de suas partes constituintes... o todo é, de fato, mais do que a soma de suas partes. Não podemos mais realmente entender estratégias num jogo de tabuleiro através da compilação de estatísticas de movimentos de suas peças mais do que podemos entender o comportamento da uma colônia de formigas em termos de médias. Sob estas condições, o todo é realmente mais do a soma de suas partes. (HOLLAND, 2009, p. 121)* [143]

[143] Tradução do autor: Emergence is above all a product of coupled, context-dependent interactions. Technically these interactions, and the resulting system, are nonlinear: The behavior of the overall system cannot be obtained by summing the behaviors of its constituent parts… the whole is indeed more than the sum of its parts. We can no more truly understand strategies in a board game by compiling statistics of the

Os fenômenos emergentes são especialmente interessantes para a área de design pois eles são uma forma de auto-organização, ou seja, uma forma espontânea de organização que é capaz de propor padrões metodológicos para a área de design, ou seja, padrões de interação colaborativa que são capazes de enfrentar problemas complexos da área que não são capazes de ser enfrentados pelos métodos tradicionais de projeto.

Em trabalhos anteriores tentamos lidar com estes cenários, ora trabalhando especificamente dentro do contexto dos fenômenos emergentes (ALÃO, 2008) ora apontando para possibilidade metodológicas de enfrentamento da complexidade no âmbito do design (ALÃO, 2015).

Nos dias de hoje é muito comum que problemas de alta complexidade sejam tratados a partir de uma abordagem colaborativa e, neste processo, os agentes frequentemente se auto-organizam de forma análoga aos fenômenos emergentes em sistemas complexos. É o caso das iniciativas Open Source, por exemplo, ou de projetos como a Wikipedia, no qual usuários do mundo inteiro geram um repositório coletivo e cooperativo sem um controle central.

> *Esta é uma característica diferenciadora do movimento Open Source: a sua exploração de forma inteligente do poder da produção colaborativa. O que torna este poder extraordinário é a capacidade de melhorar com o tempo, curando-se organicamente, como se o enorme exército de colaboradores de um projeto como o Linux fosse um sistema imunológico, sempre vigilante e ágil na reação a qualquer coisa que ameace o organismo. E apesar dos receios naturais causados por um modelo de desenvolvimento inovador para os padrões tradicionais, os projetos de Open Source não descambam para a anarquia, mas mantém uma coesão impressionante. (TAURION, online)*

Este caráter de "cura", como coloca Taurion, é de certa forma análogo ao da resiliência mencionado pouco acima: o sistema retorna a estados mais estáveis, como que se preparando para outros estímulos na medida em que se reconstrói. E os padrões de auto-organização dos usuários que contribuem para seu conteúdo têm caráter emergente. Assim, estudar a criação destes padrões em sistemas complexos — ou em simulações computadorizadas — pode gerar *insights* sobre como se comportam.

---

movement of its pieces than we can understand the behavior of an ant colony in terms of averages. Under these conditions, the whole is indeed more than the sum of its parts.

Novidade e resiliência

Os sistemas complexos tendem a adquirir o que se pode chamar de "vida própria", isto é, tornam-se tão complexos e criam tantas interdependências que exibem algumas características de seres vivos. Os sistemas aprendem, evoluem, aumentam e eventualmente se reproduzem, mesmo quando são apenas entidades baseadas em código. As hierarquias que surgem dentro de um sistema complexo adaptativo são de natureza diversa das hierarquias tradicionais. Estamos acostumados a lidar com hierarquias de cima para baixo, de caráter impositivo, como numa empresa, onde um diretor tem poder sobre um chefe e este sobre analistas etc. A hierarquia que predomina nos sistemas complexos é de baixo para cima, isto é, estruturas superiores (em escala) são geradas a partir de padrões de interação e comportamento de agentes em níveis inferiores. É como se uma empresa fosse ágil o suficiente para autoprojetar sua própria estrutura de poder e funcionamento conforme sua função se modificasse.

Alguns teóricos do campo do design enxergaram o potencial inovador dos processos emergentes e hoje tentam implementá-lo tanto em modelos teóricos quanto no mundo prático. É o caso de Gregory Van Alstyne e Robert Logan, professores da Universidade OCAD, no Canadá.

> *Nós propomos que o design possa cultivar o processo de emergência; pois é somente através de desdobramentos emergentes bottom-up, interativos massivos que produtos e serviços novos podem ser melhorados com sucesso, e introduzidos e difundidos no mercado. (LOGAN e VAN ALSTYNE, online)*[144]

Neste mesmo sentido e com esta mesma expectativa, tentamos criar processos emergentes através da modelagem de software, gerando agentes randômicos e projetando comportamentos experimentais para que, soltos no ambiente, possamos observá-los e verifica-los. Geralmente este processo envolve uma infinidade de ajustes e alterações no código até que algum traço emergente se manifeste.

Algoritmos como ferramenta

[144] We propose that design must harness the process of emergence; for it is only through the bottom-up and massively iterative unfolding of emergence that new and improved products and services are successfully refined, introduced and diffused into the marketplace

Os experimentos trazidos para o Hub 2020 foram criados a partir de algoritmos que usam funções randômicas. A intenção destes experimentos é a de perceber como se formam padrões emergentes a partir de *inputs* aleatórios. A ideia é sempre a de gerar comportamentos emergentes a partir de agentes interdependentes (que se influenciam mutuamente) e algumas poucas regras impostas pelo próprio algoritmo. Frequentemente os padrões que emergem destas experiências são um tanto inesperados e surpreendem seus criadores, como foi o caso do algoritmo tratado neste artigo especificamente, cujo código se encontra anexo a este artigo.

O trabalho exposto[145] foi criado para a linguagem Processing[146], mas o código apresentado de fato no evento foi transposto para outra biblioteca javascript, p5.js[147], para que fosse possível que outras pessoas pudessem, no decorrer da exposição, ver o resultado numa página web, pois enquanto o código Processing não pode ser apresentando diretamente numa página web, o p5.js pode. O algoritmo foi concebido para descobrir qual o padrão resultante de uma distribuição de vários pontos que se perseguem mutuamente a partir de posições aleatórias num grid bidimensional. Assim, define-se logo nas primeiras linhas do código o número de pontos a ser criado. O padrão definido para cada ponto é que ele deve perseguir o ponto anterior, diminuindo o espaço entre eles a cada ciclo. Para isso, divide-se a distância entre o ponto corrente e o anterior em um certo número de segmentos e fazemos o ponto se movimentar a distância de um segmento por ciclo.

O padrão resultante é o de uma elipse. Mesmo quando usamos números mais altos, como 50 ou 100, existe uma clara tendência para que a matriz de pontos venha a encontrar estabilidade em torno de uma forma elíptica. De forma análoga, outros experimentos podem ser idealizados para que se incentive a ocorrência de fenômenos emergentes para que, a partir de posterior análise, se descubra as interações que deram forma ao que se observa.

---

[145] Acesso à simulação apresentada e ao código fonte através da URL https://www.caramboladigital.com.br/expohub/

[146] www.processing.org
A linguagem Processing é voltada para o aprendizado de programação a partir de uma visão artística. Ela foi criada por Ben Fry e Casey Reas com o auxílio do MIT MediaLab e do Instituto Ivrea de Design.

[147] www.p5js.org
O p5.js é uma biblioteca javascript que guarda muitas semelhanças com o Processign. Ela foi criada para tornar a programação acessível a artistas e designers. Atualmente o projeto é liderado por Moira Turner e foi criado por Lauren McCarthy.

Referências

ALÃO, Rui S. D. **Design e emergência: concepção de projeto no design contemporâneo**. Dissertação de mestrado. São Paulo: UAM. 2008.
ALÃO, Rui S. D. **Projeto e complexidade. Reflexões sobre um design colaborativo.** Tese de doutorado. São Paulo: FAU USP. 2015.
HOLLAND, John. **Emergence: from chaos to order**. New York: Basic Books. 1999.
LOGAN, Robert. VAN ALSTYNE, Greg. **Designing for Emergence and Innovation: Redesigning Design**. Disponível em ‹http://www.bealinstitute.org/blog/system/files/VanAlstyneLogan_DesEme_Artifact.pdf›. Acessado em março de 2015.
TAURION, Cezar. **Open Source Software: evolução e tendências**. Disponível em ‹http://www.smashwords.com/books/download/48905/1/latest/0/0/open-source-evolucao-e-tendencias.pdf›. Acessado em janeiro de 2014.

Anexo 1

Código do sketch em p5.js

```
let n=20; // número de agentes
let x = [];
let y = [];
let d = 10; // diametro das bolas
let passos = 50;// passos para diminuir distancia

function setup() {
  createCanvas(960, 600);
  fill(255,0,0, 188);
  noStroke();
  for (let i = 0; i <n; i++) {
    x[i]=random(960);
    y[i]=random(600);
  }
}

function draw() {
  background(0);
  for (let i=0; i<n; i++) {
    if (i==(n-1)) {
      x[i] = x[i] - (x[i]-x[0])/passos;
      y[i] = y[i] - (y[i]-y[0])/passos;
    } else {
      x[i] = x[i] - (x[i]-x[i+1])/passos;
```

```
      y[i] = y[i] - (y[i]-y[i+1])/passos;
    }
    ellipse(x[i], y[i], d, d);
  }
}

function mousePressed() {
  // Ações do mouse
  for (let i = 0; i <n; i++) {
    x[i]=random(1200);
    y[i]=random(900);
  }
}
```