cgen.dlang: *An Evolutionary Approach to Design Collaboration*

Erick O. Oduniyi
University of Kansas
Lawrence, KS 66045, USA
eeoduniyi@gmail.com

Hamza T. Chaudhry
Harvard University
Cambridge, MA 02138, USA

Kristin N. Abbey
Shawnee Mission School District
Shawnee, KS 66204, USA

ACM.
AVI'20,, September 28– October 3, 2020, Island of Ischia, Italy

ACM 978-1-4503-6819-3/20/04.
https://doi.org/10.1145/3334480.XXXXXXX

Abstract
This paper presents a system for building community-generated designs (communication systems) through The Community Generated Design Language application (cgen.dlang). cgen.dlang is a design system developed for small communities (5 – 50 people) by where participants of these communities use the cgen.dlang application from which they can select structures (designs) and modify their properties (i.e., shape, color, texture, typography) to design the expected (traditional) see (or forsee) the unexpected (revolutionary), and utilize everything in between. Concretely, the cgen.dlang application provides a foundation and environment for solo design generation (seed design), community voting (democratic design), and evolutionary problem-solving (evolutionary design). As designs are made, saved, and voted on, participants collaborate and contribute to an ever-evolving design collection (language system). As a result, these human-created and computer-generated language systems are optimal (theoretically) reflections of a community's aesthetic values. Throughout this paper, we present the fruits of cgen.dlang: outline of cgen.dlang infrastructure, system architecture, and cgen.dlang prototype.

Author Keywords
Collaborative design, cultural aesthetics, evolutionary algorithms, operative design.

CCS Concepts
•**Human-centered computing** → **Human computer interaction (HCI); Ethnographic studies;** User studies;

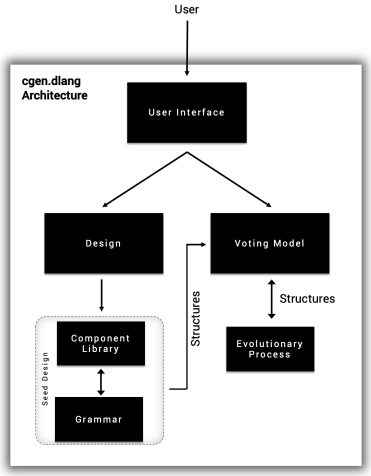

Figure 1: Overview of cgen.dlang's system architecture. Users interface with the cgen.dlang interface to design structures (seed design) that are functions of the underlying component library and associated grammar. Users also have the option of voting on other user-generated structures and algorithmically generated. structures.

Introduction

In general, using an arbitrary combination of alphabets, grammar, vocabularies, corpora of standard expressions (e.g., templates), and any other class of linguistic structures, communities create complex language systems [5, 8, 7]. Understanding how such complex language systems evolve and how the reflecting conventions, identities, protocols, and communities emerge is at the core of this project. Openly, this goal guides the design, implementation, and presentation of a system for building community-generated designs. The Community Generated Design Language application (cgen.dlang) is an evolutionary strategy for generating design compositions from community constraints (individual preferences and voting). In doing so, cgen.dlang aims to bridge hetero and homogeneous design teams via an ongoing exercise of assuming (bias), reflecting (understanding), and transcending the values of team members of different levels of expertise and field scope.

Design Language

Today, companies use design language systems (DLS) across product design (e.g., IBM Design Language, Airbnb design Language), industrial design (e.g., Snow White, Kinetic Design), software architecture (i.e., software design patterns), and graphical interfaces (e.g., Flat design, Material design, Human Interface Guidelines) [4, 2]. We theorize that studying and then engineering DLS (or patterns) using the linguistic theory of generative grammar [7, 8] it is possible to create complex language systems design research, design thinking, and design prototyping application (all in one!). The analysis of data (design) to the synthesis of the design form of product, cgen.dlang supports the design of robust (diverse) communication systems.

Cultural Evolution

Cognitive and social scientists use cultural evolution — *a mechanistic theory developed to conceptualize the forces of evolution underlying change in culture over time, where the mechanisms of change are in symbiosis with human cognition* [6, 5]. In this domain, scientists define broadly —- *anything that replicates by passing through a cognitive system*. To tease at these cognitive mechanisms (e.g., bias) involved in replication, cultural scientists use linear transmission chains to examine how people prefer to encode aspects of content, where language games and chain drawing exercises are the primary stimuli for participants [6, 3]. Research using cultural transmission chains has shown how participants tend to strip out moral choice in narrative content and show content that is counter-intuitive or emotional is better cumulatively recalled in contract to their non-emotional counterparts [3]. These cognitive mechanisms and the resulting human artifacts are then capable of shaping subsequent generations of ecological and industrial systems.

Evolutionary Design Algorithms

Evolutionary algorithms (or commonly genetic algorithms) are a class of computational approaches modeled on evolutionary dynamics. These algorithms utilize heuristics inspired by Darwinian natural selection to find optimal design solutions through continuous improvement [1]. The components of a

computational evolutionary problem-solving approach include:

- The solution space
- A random population of genotypes (possible solutions)
- Fitness function(s)
- Virtual variational operators: compare, mutate, cross

Ultimately, these continuous operations on random populations subject to some fitness pressure produce a set of “superior” quality solutions [1].

Methodology

cgen.dlang is a collaborative design system developed for small communities (5 – 50 people) who want to generate a collection of the team created and computer mutated visual structures that are likely to be optimal reflections of a team’s aesthetic values. These visual structures capture the full transmission history of aesthetics and are used to form design language systems. In this section, the system is presented in detail to guide the reader into understanding the infrastructure (and structure) of cgen.dlang (see figure 1):

- cgen.dlang’s infrastructure relies on a cocktail of programming languages for front-end and back-end specification.
- The structure (system architecture) of cgen.dlang is composed of a component library, grammar collections, voting model, and an evolutionary process (see figure 1):
  - *component library: collection of pre-defined assets*
  - *component grammar: set of rules that define acceptable operations on and between components to form structures.*
  - *voting model: schema outlining the method by which breeding candidates are selected*
  - *evolutionary processes*

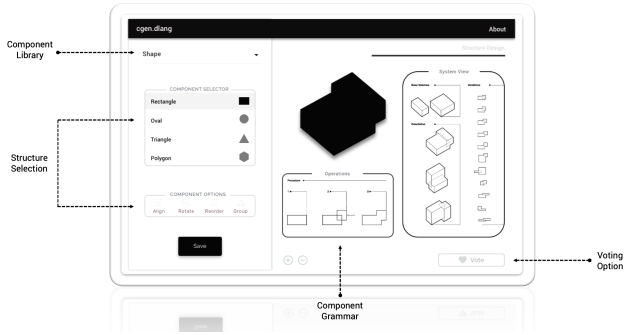

Figure 2: cgen.dlang’s annotated user interface. Here depicted is a user selecting rectangular prism structure from the shape component library. In the Structure Design view they can see the operations performed and constituting views of the structure. The structure was adapted from Anthony Di Mari’s and Nora Yoo’s Operative Design spatial verb catalog.

Results

The formative result presented in this paper is the cgen dlang system: supporting small teams to start with abstract components and end with an ever-evolving understanding of said teams’ collective values. In figure 2, we present the cgen.dlang interface, where an individual (single user) is initially specifying components (shape) and then performing subsequent operations (merge) on and between components (two rectangular prisms). In this stage of the cgen.dlang process, the user is generating a volumetric structure through well-defined operations. From there, the user can then save their structure so that it may be added to the population of designs (solutions), voted on, and used to generate structures.

Conclusion

As stated in the introduction, guiding the cgen.dlang is a deep interest in understanding the intersection between cultural evolution, design language systems, and community. More importantly, how this understanding can be used for small teams

that consist of individuals with varying backgrounds to collaborate on design projects. Through cgen.dlang's shared component library, preset grammars, community voting model, and evolutionary processes, humans and machines exchange design ideas with themselves and between each other for the community's collective good.

REFERENCES

[1] BACK, T. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[2] BOLLINI, L. From skeuomorphism to material design and back. the language of colours in the 2nd generation of mobile interface design. *Colour and colorimetry. Multidisciplinary contributions 12* (2016), 309–320.

[3] BREITHAUPT, F., GARDNER, K. M., KRUSCHKE, J. K., LIDDELL, T. M., AND ZOROWITZ, S. The disappearance of moral choice in serially reproduced narratives. In *2013 Workshop on Computational Models of Narrative* (2013), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[4] BRUNNER, R., EMERY, S., AND HALL, R. *Do you matter?: how great design will make people love your company*. FT Press, 2008.

[5] DOR, D. *The instruction of imagination: Language as a social communication technology*. Foundations of Human Interacti, 2015.

[6] FERDINAND, V., KIRBY, S., AND SMITH, K. The cognitive roots of regularization in language. *Cognition 184* (2019), 53–68.

[7] NICHOLS, J. Functional theories of grammar. *Annual review of Anthropology 13*, 1 (1984), 97–117.

[8] THOMASON, S. G., AND KAUFMAN, T. *Language contact, creolization, and genetic linguistics*. Univ of California Press, 1992.