Semio: A meta-model for formalizing reusable parametric semantic architectural concepts for non-standard collaborative testable designing

Saluz U., Geyer P.
Leibniz University Hannover, Germany
u.saluz@iek.uni-hannover.de

Abstract. Every architectural design is the result of many decisions. The more a design can be organized into loose coupled, robust, and reusable parts, the faster, flexible, and reliable is the design and decision-making process. The earliest organization of a design is an architectural concept. The objects of such are high-level, non-standard and change quickly. Various platforms for parametric definition of objects exist but they are not generally interoperable as they share no interfaces. This paper proposes a meta-model that allows to organize a design into non-standard objects by capturing semantics as a graph and compute it. Thus, enabling descriptive instead of imperative parameterization of architectural concepts. Such formalization offers new possibilities regarding authoring, testing, collaboration, and reuse. Further, computational tools (like version control, unit and integration testing, graph algorithms, graph rewriting, recommender systems, graph neural networks) become applicable. An interoperable prototype has been implemented and a use-case has been provided.

1. Introduction

Early decisions in planning of construction projects have the largest impact on the ecological, economical and social performance (Bragança et al., 2014). Despite the effectiveness of rule of thumbs in early design, exploring different architectural designs (AD) helps to find better solutions (Iano and Allen, 2022). An AD is mostly about organization of virtual objects for initiating processes in reality. It is expressed in a conservative sense through *iconic* models such as physical models, pictures and drawings, but in a wider sense it also includes *analogue* models such as a simulation model and *symbolic* models such as mathematical models (Roberts, P.H. et al., 2019). While finding form is the result of *concrete* modelling, it is intrinsically linked to *cognitive* modelling (Akalin and Sezal, 2009). Such a cognitive model can be both abstract through concepts and virtually concrete in shape. An AD is both geometric and semantic.

Developing an AD, designing, happens in multitude of ways (Smith and Smith, 2014). Yet, the process rarely consists only of individual, unlinked and unstructured decisions. Often architects design with a guiding mental structure: an architectural concept (AC). Despite major architectural theories which provide mostly analytic and proactive methods for design space exploration (DSE) (Johnson, 1994), there is unsurprisingly no general definition of what an AC is, but instead there are many personal definitions (Eilouti, 2018; Vesnic, 2017). The development of ACs is normally a fast, minimal rule based and inaccurabangbbai 1 442.34 270.12 Tm0 g(

Architecture theory

Architectural theories are mostly not intended to be formal enough to be directly translated into a formal structure. Yet, methods for designing complex things, which require high organization, that were developed by architects, have proven to be effective in providing methods to identify actors and to assign them concepts as also identifying relationships between them. This solves one of the major adoption problems. For this reason, some architectural theories have found wide applications outside of the domain of architecture (Molly Wright Steenson, 2022). A famous example is Alexander, with design patterns (Alexander et al., 1977) required an incredible high level of abstraction. L-Systems is a good formalism for modelling

Biology

P

2. Meta-model

2.1 Concepts

Figure 2: Meta-model UML diagram

A *parameter* is a key-value pair with meta-data such as the context of the *parameter*. The *context* helps to build more complex data structures like lists, dictionaries, trees, or graphs which are native to *platforms*. A *parameter*
tex *Parameters* together
with a uniform resource identifier (uri) of a *scri*[illegible]*form a*

room type capsuu.000008873 0 595.44 842088730 Gw0665.44 842.04 reW*nBT/F1 11.88 Tf1 0 0 1 121.03 7

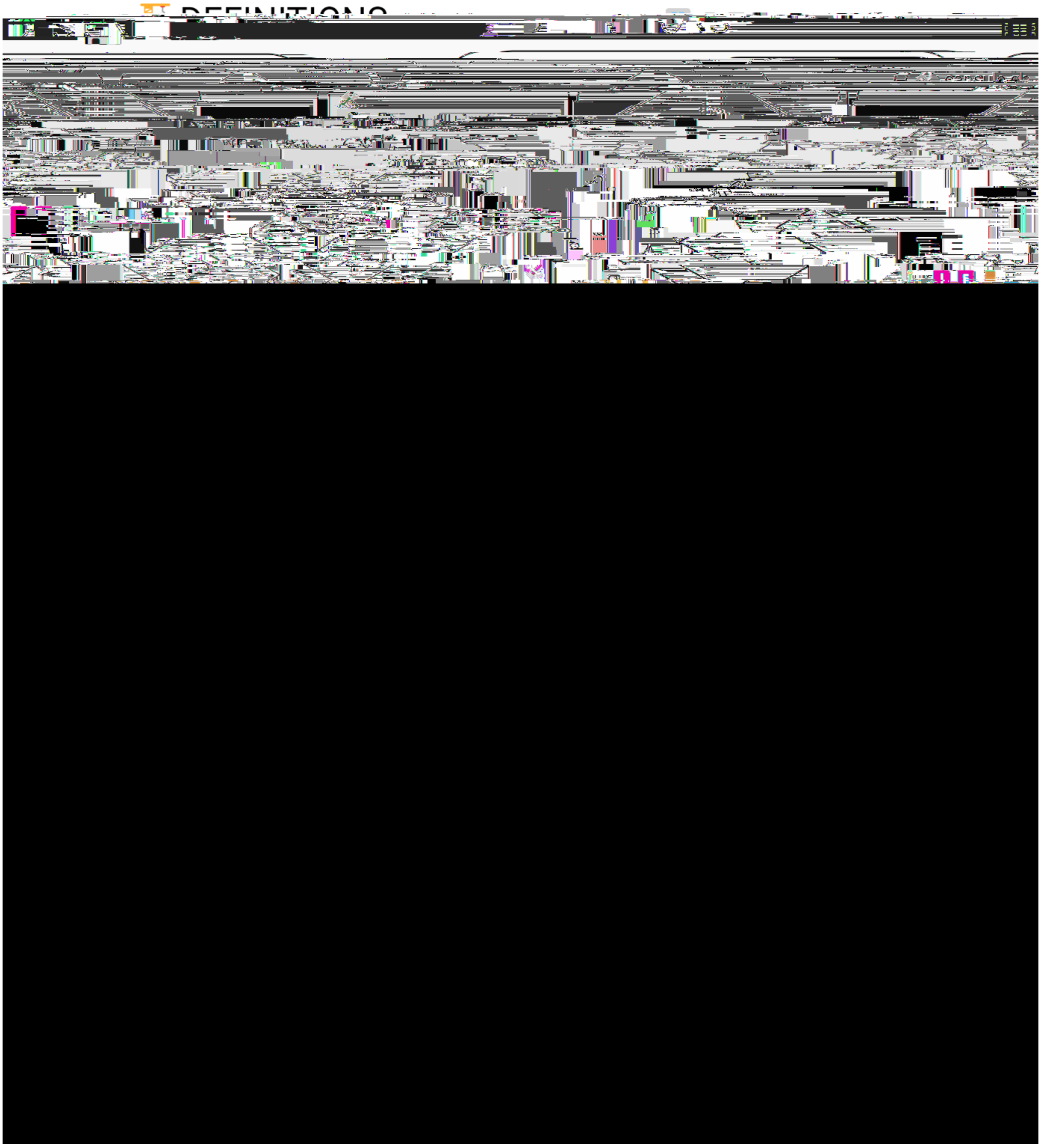

Figure 5: Version one of capsule tower

Whe doubling the number of capsules by expanding the capsule system to have a second type with the entrance in the back. Sho starts to work on expanding the capsule, Ken adds the second door per floor and adjusts the default values of the shaft.

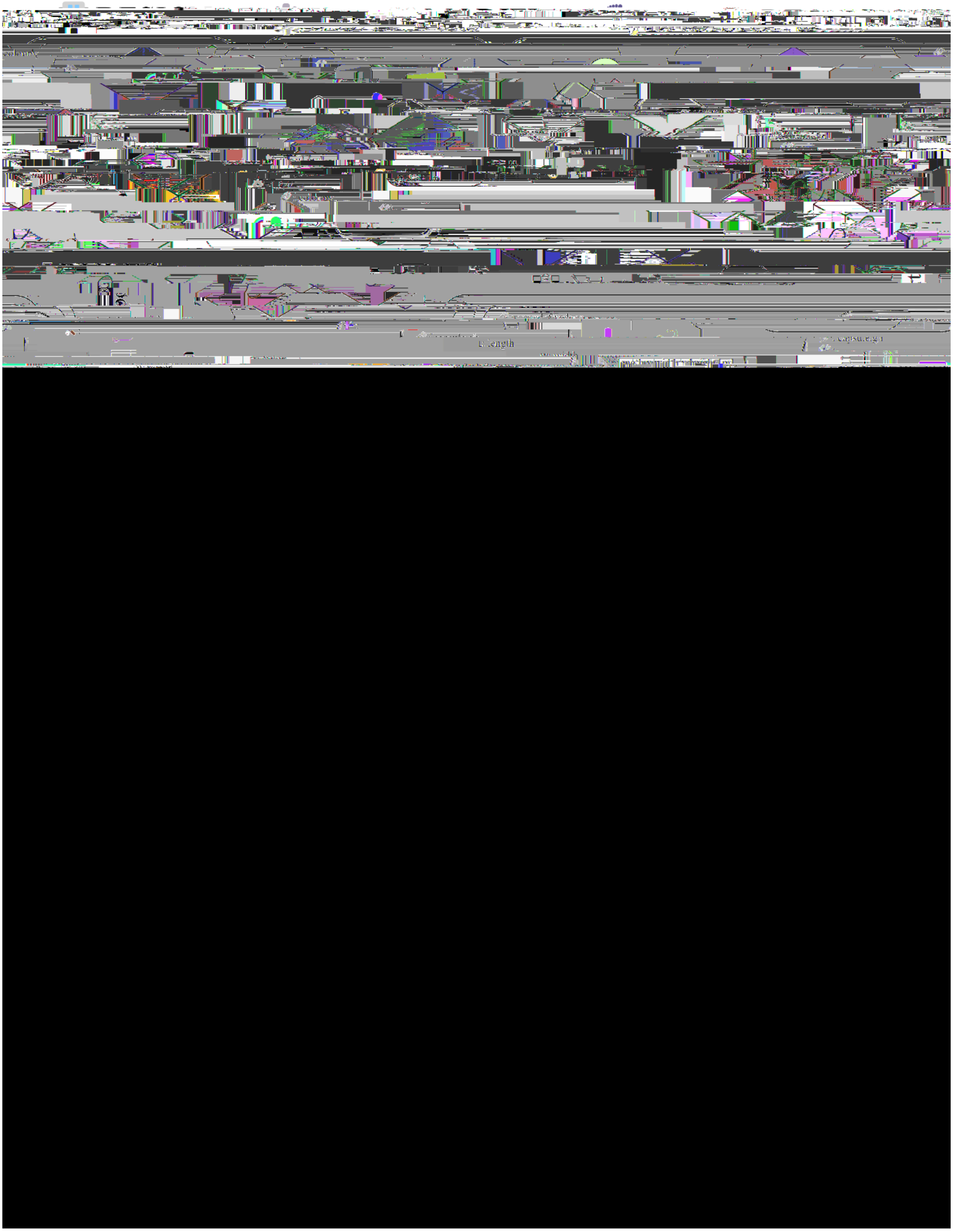

Figure 6: Version with updated capsules

Figure 7: Version three

Hirschberg, U., Hovestadt, L., Fritz, O. (Eds.), 2020. Atlas of digital architecture: terminology, concepts, methods, tools, examples, phenomena. Birkhauser, Boston.

Parish, Y.I.H., Müller, P., 2001. Procedural modeling of cities, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 308.
https://doi.org/10.1145/383259.383292

Purcell, A.T., Gero, J.S., 1996. Design and other types of fixation. Design studies 17, 363 383.

Radermacher, A., 2000. Support for Design Patterns through Graph Transformation Tools, in: Nagl, M., Schürr, A., Münch

Wortmann, T., Stouffs, R., 2018. Algorithmic complexity of shape grammar implementation. AI EDAM 32, 138 146. https://doi.org/10.1017/S0890060417000440