Design Pattern Specification Templates for Coordination Methods of Collectives: A Critical Analysis

Tadas Verbyla and Audronė Lupeikienė
Faculty of Mathematics and Informatics, Vilnius University

Introduction

Collectives consist of numerous agents working together to achieve a common objective, relying on decentralised coordination, resulting in self-organization, what is also called emergent behaviour. The coordination method here is highly significant because it forms the foundation for the creation of collectives and collective intelligence. This means that its selection and implementation constitute one of the most important stages in the engineering of collective systems. Various coordination methods have been proposed for different forms of desired self-organization, but they have not been adopted as standard solutions.

Design patterns are the blueprints for solving common problems. Reusing technology by applying patterns helps save time and avoid errors as they have already been tested in previous situations, making them generally reliable solutions. The use of design patterns is a well-established practice in software development [1]. This is ensured by the ability to completely describe design patterns; that is, a sufficient template for their specification is required. While these pattern templates (or meta-models) exist, it is not clear if they are capable to completely describe system solutions beyond their conceptual area, such as collective systems.

Objective

Further advances in the area of collective systems require for existing solutions to be formalized and refined to act as common best-practices, which can be achieved via the use of patterns, creation of which would require a pattern template. This critical analysis paper summarizes the results of the assessment of various patterns (see bellow), with the main objective to answer the question: "Are the proposed design pattern specification templates sufficient to describe the coordination of collective systems?".

Analysed Works

The work critically examines the templates for different, significant patterns. Chirstopher Alexander's architecture patterns [2] were selected as the foundational work for the field of pattern definition. "Gang of Four"'s (GoF) object oriented paradigm patterns [1], as well as Pattern-Oriented Software Architecture (POSA) patterns [3] were selected for their seminal role in applying patterns in the field of informatics and software engineering. Ensamble system patterns of the EDLC project [4] have been additionally analyzed due to their direct relation to the topic of the work. The main aspects of analysed pattern templates have been illustrated bellow in Fig 1.

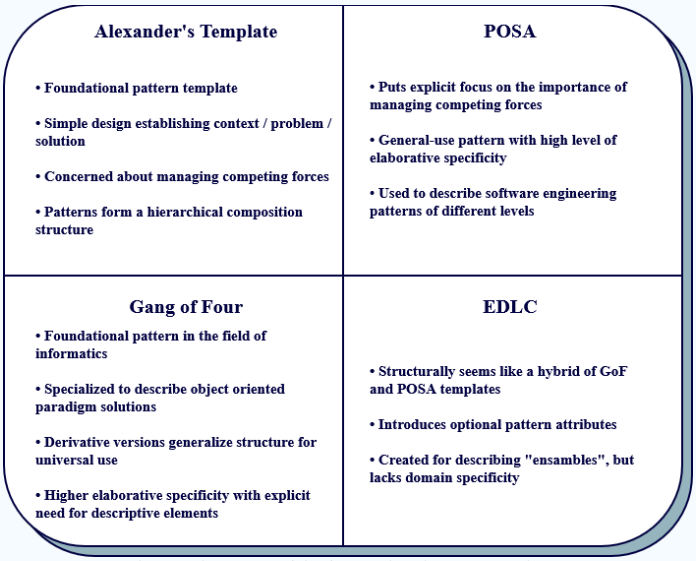

Fig 1. Main aspects of the four analysed pattern templates.

Results

1. None of the analyzed pattern templates proved to be satisfactory to completely describe the solutions of the specifc category of collective coordination methods.
2. GoF template is specifically tailored for OOP paradigm solutions. Similar tailoring would be desired for the sought collective coordination method pattern template.
3. Alexander, POSA, EDLC pattern templates offer general-purpose descriptive structure for patterns, meaning they are not tailored to their specific use-case. While they can be used for collective coordination methods, they couldn't explicitly capture the areas specifics (same applies to commonly encountered "generalized" GoF patterns).
4. Alexander templates implicitly and POSA templates explicitly emphasize the management of competing, contradictory forces as the core of a good pattern. The management of forces should be examined in greater detail when constructing pattern template for collective systems.

Conclusions

The analysis has demonstrated that besides the original “Gang of Four” pattern schema, all analysed pattern schemas (including generalized version of the “Gang of Four” schema) can be used to define coordination methods for collectives. However, all of these are of generally-applicable nature and do not capture the domain specificity of collective coordination (see Fig 2). Thus, building upon the analysed schema, a new, special-purpose collective coordination pattern schema must be constructed.

References

[1] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design patterns: Elements of reusable object-oriented software. Addison-Wesley.

[2] C. Alexander. “The Timeless Way of Building”. Oxford University Press, 1979.

[3] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. " Pattern-Oriented Software Architecture, A System of Patterns.” Wiley. 1996.

[4] Hölzl, M., Koch, N., Puviani, M., Wirsing, M., Zambonelli, F. (2015).

Contacts

Tadas Verbyla: tadas.verbyla@mif.vu.lt
Audronė Lupeikienė: audrone.lupeikiene@mif.vu.lt

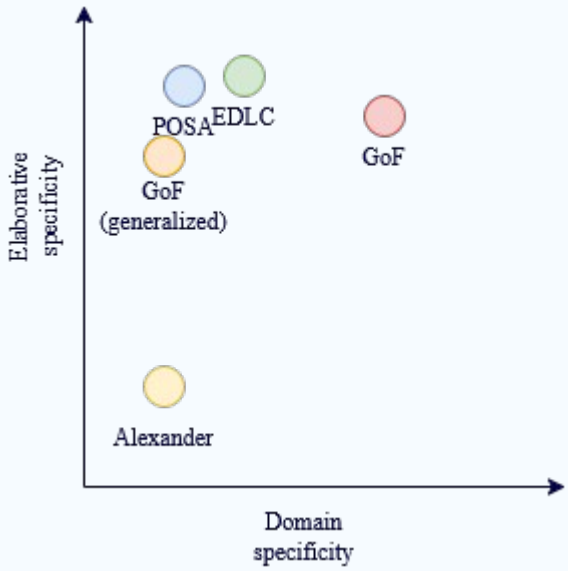

Fig 2. Comparison of pattern template specificity.