A Design Pattern Language for Hybrid Intelligent Teams

Tajaddini, M.; Brinkman, W.P.; ten Teije, Annette; Neerincx, M.A.

Publication date
2021

Document Version
Final published version

Published in
BNAIC/BeneLearn 2021

Citation (APA)
Tajaddini, M., Brinkman, W. P., ten Teije, A., & Neerincx, M. A. (2021). A Design Pattern Language for Hybrid Intelligent Teams. In E. L. A. Leiva, C. Pruski, R. Markovich, A. Najjar, & C. Schommer (Eds.), *BNAIC/BeneLearn 2021: 33rd Benelux Conference on Artificial Intelligence and 30th Belgian-Dutch Conference on Machine Learning* (pp. 723-725)

A Design Pattern Language for Hybrid Intelligent Teams

Mani Tajaddini[1][0000−0001−9038−3524], Willem-Paul Brinkman[1][0000−0001−8485−7092], Annette ten Teije[2][0000−0002−9771−8822], and Mark Neerincx[1,3][0000−0002−8161−5722]

[1] Interactive Intelligence Group, Delft University of Technology, The Netherlands
{w.p.brinkman,m.a.neerincx,m.tajaddini}@tudelft.nl
[2] Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
annette.ten.teije@vu.nl
[3] TNO Human Factors, Soesterberg, The Netherlands
mark.neerincx@tno.nl

Abstract. The field of Hybrid Intelligence (HI) is like a vast land with many tribes that speak different languages. Our goal is to develop a *lingua franca* to unify the peoples of the HI land. We expect our language to facilitate documentation and communication of research results and thus collaboration among various HI fields by making use of design patterns describing human-AI interactions.

Keywords: Design pattern language · Hybrid intelligence · Human-AI interaction.

1 Introduction

The premise of this project is to create a language whose words are design patterns in Hybrid Intelligence (i.e., HI) design. These patterns describe a configuration of machine and human agents which are designed to carry out a particular task in a particular set of circumstances [1, 4]. The difficulty of such an endeavor lies in the fact that the HI field is like a vast land with many tribes (i.e., groups of scientists and engineers from many different disciplines) working in it. These tribes usually speak different languages and work in diverse contexts and have diverse backgrounds. This causes dispersion in communication between various tribes and through time. Difficulties arise in integrating one tribe's work in another tribe's project, in communicating one tribe's findings in a certain field to another field, in getting two or more tribes to collaborate on a project, and even to deliver one tribe's work through time so that future projects can benefit from it.

We would like to facilitate communication throughout the HI land by creating a *lingua franca* which all the various peoples of the HI land can speak. To do this, a strategy is needed to document the experiences of designers in various HI fields by extracting useful patterns from those experiences. In effect, future designers will not need to reinvent the wheel and will have a vocabulary and a framework that will guide them in their designing efforts.

2 Development and Evaluation

What we need in order to build a language is a vocabulary and a grammar (which is some kind of structure). The vocabulary is the set of design patterns and most design pattern languages stop there. They are just a catalogue or a dictionary of design patterns. However, having the same vocabulary does not necessarily mean mutual intelligibility between two languages. To have a universal language we also need a common grammar. The grammar is a formalism (e.g., context-free grammar, combinators, lambda calculus, category theory, etc.) that structures the vocabulary and dictates how to compose different design patterns within different levels and between different levels of abstraction.

As a starting point, we should identify design patterns through observing how people working in HI think and solve their problems. In representing design patterns we should focus on how these patterns compose so that we can superimpose a structure as the language's grammar. Our effort will be geared toward generating as many useful syntactically possible combinations of our atomic patterns as possible [2, 3]. We expect the pattern language to have both an easy-to-access graphical notation and a formal representation that can be manipulated by computer-tools (e.g., editor, validator, search tool, configurator, etc). One very idealistic end product to imagine is a Domain Specific Language accessible to both humans and machines which could be used to design an Integrated Development Environment.

In evaluating how successful the design pattern language is, we can take into account the following. Firstly, the employed design patterns should be valid, meaning they have to be instantiated in a concrete context to see how far the system behaves according to the pattern. To do this, in certain cases, tests involving simulations to address the size, diversity, and dynamics of the human and artificial cognitive processes can be carried out. Furthermore, the design pattern language should be as complete as possible or viable, meaning it must be able to describe as many as the HI application scenarios which are useful for the users. However, the most important questions to ask in evaluating the language should address researchers' and engineers' performance in reliably instantiating an HI design pattern into a solution for their situated problem. Questions like how easy it is to understand the idea expressed in a design pattern; how easy it is to find a design pattern expressing a certain idea; how well the language expresses the important properties of an idea, such as its scope or impact; and is it possible to compare design patterns or ideas therein using the pattern language. It is also very important that our language be dynamic, meaning designers must be able to add to it and change it as the research field develops.

3 Concluding Remarks

The above paragraphs sketch out a plan for constructing a design pattern language. However, to begin with, we must understand what a design pattern means in the field of HI; which design patterns are available in this field; and what efforts have been made in formalizing design pattern languages. Therefor, we have

set out to write two review papers on design patterns in HI and on methods of design pattern formalization. The future plan consists of incrementally discovering and formalizing extant design patterns in HI and identifying gaps therein.

References

1. Akata, Z., Balliet, D., De Rijke, M., Dignum, F., Dignum, V., Eiben, G., Fokkens, A., Grossi, D., Hindriks, K., Hoos, H., et al.: A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. Computer **53**(8), 18–28 (2020)
2. van Bekkum, M., de Boer, M., van Harmelen, F., Meyer-Vitali, A., Teije, A.t.: Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases. arXiv preprint arXiv:2102.11965 (2021)
3. Van Harmelen, F., Teije, A.t.: A boxology of design patterns for hybrid learning and reasoning systems. arXiv preprint arXiv:1905.12389 (2019)
4. van der Waa, J., van Diggelen, J., Siebert, L.C., Neerincx, M., Jonker, C.: Allocation of moral decision-making in human-agent teams: A pattern approach. In: International Conference on Human-Computer Interaction. pp. 203–220. Springer (2020)