Towards an Enhancement of Coarse Design Diagrams in Renew*

Marcel Hansson[1], Clara Dorothea von Bargen[1] and Daniel Moldt[1]

[1]University of Hamburg, Faculty of Mathematics, Informatics and Natural Sciences, Department of Informatics, http://www.paose.de

Keywords
Petri Nets, Agent Interaction Protocols, Coarse Design, Refinement, Coarsening, PAOSE, Renew

1. Introduction

A system can be described by agents that communicate with each other. The Paose[1] approach combines this perspective with Petri nets. Mulan is a toolset that implements the Paose approach [1], extending the Petri net tool Renew[2] with agent functionalities.

There are a number of techniques that allow modeling use cases and interactions, some of them provided in the context of Paose. The goal of this paper is to further extend the capabilities of existing Paose techniques. For this, modeling functionalities are moved from Mulan to Renew, existing diagram types are interconnected, and the concept of refinement is presented.

In the following, we describe the foundations for our work (sec. 2) and set the objectives of this paper (sec. 3). We then describe the realization (sec. 4), examine related work (sec. 5) and discuss our findings (sec. 6). Finally, we provide a summary and present possibilities for future work (sec. 7).

2. Foundations

Paose is an approach that is described in detail by Cabac [1]. It aims to develop executable multi-agent systems based on agent-oriented Petri nets [2]. For designing such a system, modeling techniques like *Coarse Design Diagrams* (Cdds) and *Agent Interaction Protocols* (Aips) are used [1].

Cdds allow to create an overview of a system. They are based on UML Use Case Diagrams, consisting of actors and use cases. While the syntax is maintained, both the semantics and the perspective are adapted. Instead of users, actors represent agent roles as part of the modeled system. Use cases represent interactions within the modeled system, not with it [1].

Aips provide an easily readable technique for modeling interactions between multiple roles that are taken by agents [1]. Each Aip describes exactly one interaction, also called a conversation, that consists of the exchanged messages and is initiated by a trigger [3].

Renew is a tool originally designed to draw and simulate different Petri net formalisms [4]. The system consists of a number of plugins that offer different functionality by extending Renew [1]. Over the years, Renew has evolved from being only a modeling tool to also being a software development tool [4] that provides IDE support for various tasks [1]. Within the toolkit of Renew, Aips can be expressed and transformed into Petri nets [3] based on a predefined mapping [1].

PNSE'25, International Workshop on Petri Nets and Software Engineering, 2025
* Supported by participants of our teaching project classes and student theses.

[1]Petri net-based, Agent- and Organization-Oriented Software Engineering, http://paose.de
[2]The Reference Net Workshop: http://renew.de

3. Objectives

In the followup of redesigning Renew (see [5]), the constant goal is to enhance its modeling capabilities. Options to achieve this are introducing additional modeling techniques, combining existing techniques and introducing abstraction hierarchies.

Based on this, the functionalities of implementing Cdds in Renew, associating Aips with use cases and introducing refinement and coarsening for Cdds were proposed. The current state of our work is presented in section 4.

4. Realization

As a first step, we introduced the Cdd functionality as a new plugin in Renew. For this, we copied the original Mulan plugin and reduced it to its core features. Following the transfer, additional small enhancements were implemented to improve the Cdd functionality, including syntax checking and improvements regarding usability. Adding Cdds in Renew allows to further extend its functionality and use Cdds in a more general modeling context.

While Cdds describe what interactions are taking place[1], Aips describe the messages exchanged between the roles within each of these interactions [1, p.95]. As our second step, we implemented tool support to represent this relationship in the model. Renew now allows users to attach and detach zero to n Aips to each use case within a Cdd diagram. The attached Aips of a use case can be opened as separate drawings with a single click. Every use case with at least one attached Aip is highlighted to show the attachment.

Abstraction and concretion are common strategies for dealing with complexity, describing the same system at different levels of detail. In the context of Petri nets, these terms are also defined as coarsening and refinement. In our third step, we discussed how to apply these concepts to Cdds and their elements. As a first idea on how to describe the elements in more detail, we allow marking an actor or a use case as to be refined. The marked element can then be refined in a separate diagram. For refining one single actor or use case, any number of actors and/or use cases can be used. The same idea can be applied in reverse: a number of elements marked for coarsening can be represented by one abstract actor or use case.

5. Related Work

HERAKLIT is a modeling technique based on Petri nets [6]. It includes composition to create a hierarchical architecture, which served as inspiration for our ideas regarding refinement and coarsening. The HERAKLIT approach was related to the Paose concepts in [3]. Bringing the abstraction mechanics to Cdds is a further step in this direction.

In a more general context, the idea of abstraction and concretion was applied to use case diagrams in [7] and [8]. However, these approaches do not apply the concepts to actors, and do not allow for the use case replacements to include actors.

In [9], the authors use sequence diagrams to describe use cases, which is similar to connecting Aips and Cdds; however, their goal is to describe relationships between use cases.

6. Discussion

With the new features, the modeling capabilities of Renew are extended. While associating Aips with use cases allows combining multiple types of diagrams, refinement and coarsening allows combining multiple levels of abstraction within one model. Although refinement and coarsening have not yet been implemented, the described concept provides a clear path for implementation. Both features can also be combined. Once implemented, they allow for a model that provides navigation through different levels of abstraction and diagram types.

7. Conclusion

We moved the CDDs from MULAN to RENEW and enhanced them with the option to interactively attach AIPs. Additionally, we introduced the concept of refining use cases as well as actors by allowing each element of a CDD to be refined with an additional CDD detailing it. Since each element can be refined by both additional actors and use cases, internal orchestrations can be modeled. This permits visualization of larger systems at various levels of abstraction as required.

So far, we are refining single use cases or actors, but it should be possible to refine multiple actors and use cases together. The guiding idea is to follow the associative composition of Fettke and Reisig with their HERAKLIT approach [6], combined with the HERAKLIT AGENT proposal in [3].

While this work mostly describes the refinement within use case diagrams and expresses coarsening as a simple reverse operation, starting with a complex diagram and wanting to abstract or coarsen parts of it is a different matter. It remains to be seen on how this can be implemented and how the context of Petri nets can be used.

Another point of interest is whether additional functionality can be transferred from MULAN to RENEW, taking domain driven design (DDD) into consideration regarding the modularization of functionality.

As RENEW already contains general modeling techniques like BPMN, and now allows to create use case diagrams, further extending RENEW with additional general diagram types could be considered.

Declaration on Generative AI

During the preparation of this work, the authors used DeepL and LanguageTool in order to: Grammar and spelling check, Paraphrase and reword. After using these tool(s)/service(s), the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] L. Cabac, Modeling Petri Net-Based Multi-Agent Applications, Dissertation, University of Hamburg, Germany, 2010. URL: https://ediss.sub.uni-hamburg.de/handle/ediss/3691.

[2] D. Moldt, F. Wienberg, Multi-agent-systems based on coloured Petri nets, in: P. Azéma, G. Balbo (Eds.), Application and Theory of Petri Nets 1997, number 1248 in Lecture Notes in Computer Science, Springer Verlag, Berlin Heidelberg New York, 1997, pp. 82–101.

[3] D. Moldt, M. Hansson, L. Seifert, K. Ihlenfeldt, L. Clasen, K. Ehlers, M. Feldmann, Enriching heraklit modules by agent interaction diagrams, in: L. Gomes, R. Lorenz (Eds.), Application and Theory of Petri Nets and Concurrency, volume 13929 of *Lecture Notes in Computer Science*, Springer Nature Switzerland AG, 2023, pp. 440–463. URL: https://doi.org/10.1007/978-3-031-33620-1_23.

[4] D. Mosteller, Metamodellierung für den PAOSE-Softwareentwicklungsansatz, Master thesis, University of Hamburg, 2014.

[5] D. Moldt, J. Johnsen, R. Streckenbach, L. Clasen, M. Haustermann, A. Heinze, M. Hansson, M. Feldmann, K. Ihlenfeldt, RENEW: modularized architecture and new features, in: L. Gomes, R. Lorenz (Eds.), Application and Theory of Petri Nets and Concurrency, volume 13929 of *Lecture Notes in Computer Science*, Springer Nature Switzerland, 2023, pp. 217–228. URL: https://doi.org/10.1007/978-3-031-33620-1_12.

[6] P. Fettke, W. Reisig, Understanding the Digital World – Modeling with HERAKLIT, Springer, 2024. URL: https://doi.org/10.1007/978-3-031-61898-7. doi:10.1007/978-3-031-61898-7.

[7] D. Faitelson, S. Tyszberowicz, UML diagram refinement (focusing on class-and use case diagrams), 2017. doi:10.1109/ICSE.2017.73.

[8] E. Cruz, R.-J. Machado, M. Santos, On the decomposition of use case diagrams for the refinement of requirements, 2014. doi:10.1109/ICCSA.2014.54.

[9] J. M. Almendros-Jiménez, L. Iribarne, Describing use-case relationships with sequence diagrams, The Computer Journal 50 (2006) 116–128. URL: https://doi.org/10.1093/comjnl/bxl053.