Choreographic Language Extension for Markov Chain Generation

Master Thesis Proposal

TECHNISCHE UNIVERSITÄT DARMSTADT

Software Engineering Group

Introduction

Choreographic programming [8] is an emerging paradigm designed to address the complexities arising from analysing distributed systems and the interactions between their components. In choreographic programming, a program—called a *choreography*—provides a global view of the communication patterns that define the interactions within a distributed system. Rather than focusing on the internal communications of each individual component, choreographies emphasize the global behavior of the system through decentralized interactions.

PRISM [1] is a probabilistic model checker that offers a powerful framework for the specification and verification of probabilistic concurrent systems. PRISM has been widely applied in fields ranging from randomised distributed algorithms [5, 7] to security [6, 9] and biological systems [3, 4]. Given a distributed system, we can use PRISM to model the behaviour of each of its nodes, and then verify desired properties for the entire system. However, this approach can become difficult to manage as the number of nodes increases.

We developed a choreographic language [2] that is specifically designed for modeling concurrent probabilistic systems. Additionally, we have developed a compiler that translates protocols written in this language into PRISM code, facilitating integration with PRISM's model-checking capabilities. This approach not only simplifies the modeling of complex systems but also ensures that the resulting implementations can be rigorously verified, leveraging PRISM's powerful tools for analyzing probabilistic behaviors and system properties.

Proposal

This thesis proposal has two main objectives. First, to improve our choreographic language by extending its capabilities to better model probabilistic behaviors in distributed systems. Second, to develop a compiler that translates the extended choreographies into Markov Chains, enabling the use of model checking tools like PRISM for formal verification of the system's properties.

In particular, two key limitations of the current language will be addressed:

- **Conditional Statements**: the current language prohibits the use of an "if-then" statement without an accompanying "else" clause. This restriction is designed to prevent potential deadlocks, but it limits expressiveness in scenarios where unbalanced conditional logic is required. The idea is to extend the language to handle these cases while maintaining safety guarantees.
- **Probabilistic Branching**: In certain systems, processes often need to synchronize at specific points, but they also require the ability to make internal choices independently before re-synchronizing. The current language does not support this feature, which limits its use in systems that rely on such random decisions for complex coordination.

In addition to these upgrades, the thesis will explore the translation of the improved choreographic language into PRISM code.

Expected Contributions

1. **Language Extension**: Improved expressiveness for probabilistic systems, supporting unbalanced conditional statements and probabilistic branching.
2. **Compiler Integration**: An implementation of a compiler that translates the extended choreographic language into PRISM-compatible models.
3. **Markov Chain Generation**: Development of a compiler that translates the choreographic language into Markov Chains.

Required Skills

The candidate should possess the following skills:

- **Programming Skills**: Experience with functional programming or a similar paradigm.
- **Compiler Knowledge**: An understanding of compiler construction concepts is highly recommended.
- **Knowledge of Formal Methods**: Familiarity with formal verification techniques like model checking.
- **Familiarity with Probabilistic Systems**: An interest in systems that involve random behavior or decision-making is beneficial, though an expertise is not required.
- **Distributed Systems Knowledge**: A good knowledge of distributed systems and communication protocols is important for understanding the overall context.

Prior knowledge of Markov Chains or choreographies is not required aqg can be acquiring during the course of the thesis.

Work Profile

The thesis work will be structured into the following phases:

1. **Research and Analysis**: Understand the limitations of the existing choreographic language and explore the design space for the new extensions. This phase involves reviewing literature and analyzing case studies that require probabilistic branching and complex conditionals.
2. **Language Design**: Propose language extensions that improve expressiveness without compromising the safety guarantees of the existing system.
3. **Compiler Development**: Implement the extensions within the existing compiler, ensuring that the generated code integrates with PRISM.
4. **Testing and Validation**: Develop test cases and benchmarks to evaluate the correctness and performance of the extended language and compiler.
5. **Documentation and Thesis Writing**: Document the design choices, implementation details, and results of the research. Write the final thesis report.

Approximate Work Distribution

Contact

Adele Veschetti
Software Engineering Group
adele.veschetti@tu-darmstadt.de

References

[1] Prism documentation. `https://www.prismmodelchecker.org/`. Accessed: 2024-10-01.

[2] Marco Carbone and Adele Veschetti. A probabilistic choreography language for PRISM. In *COORDINATION*, volume 14676 of *Lecture Notes in Computer Science*, pages 20–37. Springer, 2024.

[3] Frits Dannenberg, Marta Kwiatkowska, Chris Thachuk, and Andrew Turberfield. DNA walker circuits: computational potential, design, and verification. In D. Soloveichik and B. Yurke, editors, *Proc. 19th International Conference on DNA Computing and Molecular Programming (DNA 19)*, volume 8141 of *LNCS*, pages 31–45. Springer, 2013.

[4] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. In C. Priami, editor, *Proc. Computational Methods in Systems Biology (CMSB'06)*, volume 4210 of *Lecture Notes in Bioinformatics*, pages 32–47. Springer Verlag, 2006.

[5] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic verification of Herman's self-stabilisation algorithm. *Formal Aspects of Computing*, 24(4):661–670, 2012.

[6] M. Kwiatkowska, G. Norman, D. Parker, and M.G. Vigliotti. Probabilistic mobile ambients. *Theoretical Computer Science*, 410(12–13):1272–1303, 2009.

[7] M. Kwiatkowska, G. Norman, and R. Segala. Automated verification of a randomized distributed consensus protocol using Cadence SMV and PRISM. In G. Berry, H. Comon, and A. Finkel, editors, *Proc. 13th International Conference on Computer Aided Verification (CAV'01)*, volume 2102 of *LNCS*, pages 194–206. Springer, 2001.

[8] Fabrizio Montesi. *Introduction to Choreographies*. Cambridge University Press, 2023.

[9] G. Norman and V. Shmatikov. Analysis of probabilistic contract signing. *Journal of Computer Security*, 14(6):561–589, 2006.