

Abstract/Introduction:

Nowadays, modeling languages for cyber-physical systems play an important role in software engineering. A corresponding example is the automotive branch where safety has a high relevance. Apart from that, the magnitude and quantity of software projects rises constantly, as software development needs spread among scientific and technical disciplines. For this case, Domain Specific Languages (DSLs) are used in order to cope with these specialized tasks and environments, thus there exists a necessity for tools to define these DSLs comfortably. There are two kinds of problems: One of the problem is data-oriented (e.g. Amazon), and the other problem is decision-oriented (e.g. autonomous driving car) which is our use case. Moreover, implementation of the above-mentioned modeling language is still a problem to make them work properly. „Component and Connector“ (C&C) approaches are used as a common method to describe the architecture of DSLs. Testing on cyber-physical systems is still necessary, as testing in „real-life“ would cause higher costs. One difficulty which makes simulation difficult is also that physical laws must hold in such systems. MontiCar is such an environment or respectively a framework you can use to do agile and model-driven software development in which the core language of this is EmbeddedMontiArc [Armin]. In order to cope with the rising complexity and recurring nature of DSLs, parser-generators or software language workbenches can be of great value.

This paper represents a case-study, evaluating the ease of use and re-usability of MontiCore for reactive systems. Our exact task and the topic of our seminar thesis is to analyse the components and behaviour of the Pacman game. Furthermore, it also belongs to our task that we write about our experiences among other things how long we need to learn EMA and which difficulties we face during our project. To that it also belongs that we shall write what was missing in EMA to fulfill our tasks and how long we needed to create a model. Moreover this seminar paper will be based on several research questions which are among other things:

RQ1: Is EmbeddedMontiArc suitable for other systems?

RQ2: Is it possible to integrate other simulators in a recent amount of work?

RQ3: What kind of background knowledge is needed to model C&C in EMA?

RQ4: What features are good and what are not suited?

Content:

1. Abstract
2. Introduction
3. Context
4. Approach
5. Modelling
6. Evaluation
7. Conclusion

References:

1. Modeling Architectures of Cyber-Physical Systems (Kusmenko, Roth, Rumpe, von Wenckstern)
2. Simulation Framework for Executing Component and Connector Models of Self-Driving Vehicles (Grazioli, Kusmenko, Roth, Rumpe, von Wenckstern)
3. Model-Based Development of Self-Adaptive Autonomous Vehicles using the SMARDT Methodology (Hillemacher, Kriebel, Kusmenko, Lorang, Rumpe, Sema, Strobl, von Wenckstern)
4. MontiCAR - 3D Modeling using EmbeddedMontiArcMath (Mokhtarian)
5. Modular and Optimized C++ Code-Generator for the Component and Connector Modeling Language MontiCAR (Schneiders, von Wenckstern, Kusmenko, Rumpe)

6. Component and Connector Views in Practice: An Experience Report (Betram, Maoz, Ringert, Rumpe, von Wenckstern)

7. Guidelines for conducting and reporting case study research in software engineering (Runeson, Höst)