Abstract

The Model-Driven Software Development Systems (MDSDS) were initially developed as an attempt to increase software development productivity and quality. This is because focusing on the logical solution abstract is more important than focusing on the pure infrastructure technicalities. Developers discovered the abstracted modelling technique that includes both programming and platform tools in the same time, which is now referred to as MDSDS.

Nowadays, there are plenty of modeling software applications that almost achieve the same work, yet, the user might not be aware of the detailed nuances between them. This paper aims to discover the distinguishing features between four of the most commonly used MDSDS including; YAKINDU, Papyrus-RT, Rhapsody, and The State Machine Compiler (SMC). Analysis of the suitability of those platforms for modeling structural and behavioral domain specific software will be investigated.

The same model will be built using the four MDSDSs. Then, main differences, obstacles,
observations, and overall experience quality using those four environments will be discussed. Some of the common distinguishing features to be explored is GUI intuitivism, user friendliness, clarity of commands and tools, tool learning time needed and learning curve, model building time consumed, etc.

References

1. Sendall, S., & Kozaczynski, W. (2003). Model transformation: The heart and soul of model-driven software development. IEEE software, 20(5), 42-45.
2. Schmidt, D. C. (2006). Model-driven engineering. COMPUTER-IEEE COMPUTER SOCIETY-, 39(2), 25.
3. Home of SMC, The State Machine Compiler, http://smc.sourceforge.net/, accessed on April 12th, 2018.
4. Selic, B. (1998). Using UML for modeling complex real-time systems. In Languages, compilers, and tools for embedded systems (pp. 250-260). Springer, Berlin, Heidelberg.
5. Papyrus-RT/User/User Guide/Getting Started, Getting Started with Papyrus for Real Time v1.0, https://wiki.eclipse.org/Papyrus-RT/User/User_Guide/Getting Started#Our_project:_PingPong, last accessed April 15th, 2018.
6. Kahani, N., Hili, N., Cordy, J. R., & Dingel, J. (2017, May). Evaluation of UML-RT and Papyrus-RT for modelling self-adaptive systems. In Modelling in Software Engineering (MiSE), 2017 IEEE/ACM 9th International Workshop on (pp. 12-18). IEEE.
7. Gery, E., Harel, D., & Palachi, E. (2002, May). Rhapsody: A complete life-cycle model-based development system. In International Conference on Integrated Formal Methods (pp. 1-10). Springer, Berlin, Heidelberg.
8. Khalil, A., & Dingel, J. (2017). Optimizing the Symbolic Execution of Evolving Rhapsody Statecharts. Advances in Computers.
9. Zurowska, K., & Dingel, J. (2011, November). SAUML: A tool for symbolic analysis of UML-RT models. In Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering (pp. 604-607). IEEE Computer Society.
10. Das, T. K., & Dingel, J. (2016). Model development guidelines for UML-RT: conventions, patterns and antipatterns. Software & Systems Modeling, 1-36.
11. Das, T. K., & Dingel, J. (2015, September). State machine antipatterns for UML-RT. In Model Driven Engineering Languages and Systems (MODELS), 2015 ACM/IEEE 18th International Conference on (pp. 54-63). IEEE.
12. Das, T. K., & Dingel, J. (2016). Model Development Guidelines for UML-RT.
13. Dingel, J., Paen, E., Posse, E., Rahman, R. R., & Zurowska, K. (2010, June). Definition and implementation of a semantic mapping for UML-RT using a timed pi-calculus. In Proceedings of the Second International Workshop on Behaviour Modelling: Foundation and Applications (p. 1). ACM.
14. Crane, M. L., & Dingel, J. (2005, October). UML vs. classical vs. Rhapsody statecharts: Not all models are created equal. In International Conference on Model Driven Engineering Languages and Systems (pp. 97-112). Springer, Berlin, Heidelberg.
15. Bagherzadeh, M., Hili, N., & Dingel, J. (2017, August). Model-level, platform-independent debugging in the context of the model-driven development of real-time systems. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software
Engineering (pp. 419-430). ACM.
16. Bagherzadeh, M., Bordeleau, F., Bruel, J. M., Dingel, J., Gérard, S., Hili, N., & Voss, S. Summary of Workshop on Model-Driven Engineering Tools (MDETools’ 17).
17. Liang, H., Dingel, J., & Diskin, Z. (2006, May). A comparative survey of scenario-based to state-based model synthesis approaches. In Proceedings of the 2006 international workshop on Scenarios and state machines: models, algorithms, and tools (pp. 5-12). ACM.
18. Dingel, J., & Filkorn, T. (1995, July). Model checking for infinite state systems using data abstraction, assumption-commitment style reasoning and theorem proving. In International Conference on Computer Aided Verification (pp. 54-69). Springer, Berlin, Heidelberg.
19. Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., & Heldal, R. (2013, September). Industrial adoption of model-driven engineering: Are the tools really the problem? In International Conference on Model Driven Engineering Languages and Systems (pp. 1-17). Springer, Berlin, Heidelberg.
20. Kahani, N., Bagherzadeh, M., Dingel, J., & Cordy, J. R. (2016, October). The problems with eclipse modeling tools: a topic analysis of eclipse forums. In Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (pp. 227-237). ACM.
21. SMC Home Page, The State Machine Compiler, http://smc.sourceforge.net/, last accessed April 14th, 2018.
22. Graics, B., & Molnár, V. (2017). Formal Compositional Semantics for Yakindu Statecharts.
23. Yakindu Statechart Tools, http://statecharts.org/, last accessed April 15th, 2018.
24. Hili, N., Dingel, J., & Beaulieu, A. (2017, May). Modelling and code generation for real-time embedded systems with UML-RT and papyrus-RT. In Software Engineering Companion (ICSE-C), 2017 IEEE/ACM 39th International Conference on (pp. 509-510). IEEE.
25. Kahani, N., Hili, N., Cordy, J. R., & Dingel, J. (2017, May). Evaluation of UML-RT and Papyrus-RT for modelling self-adaptive systems. In Modelling in Software Engineering (MiSE), 2017 IEEE/ACM 9th International Workshop on (pp. 12-18). IEEE.
26. Papyrus for real time (Papyrus-RT), https://www.eclipse.org/papyrus-rt, accessed: 2018-03-25
27. Posse, E. (2015). PapyrusRT: modelling and code generation. In Workshop on Open Source for Model Driven Engineering (OSS4MDE’15).
28. IBM Rational. Rational Rhapsody Developer, http://www03.ibm.com/software/products/en/ratirhap/.

Index Terms

Computer Science
Software Engineering

Keywords

Software Development Systems, Model-Driven Platforms, State Machine Engineering,
YAKINDU, Papyrus-RT, Rhapsody, State Machine Compiler-SMC.