Related Work and Research Summary

Nathan Jarus

April 29, 2015

1 Research Project Concept

My research focuses on automated model building for cyber-physical systems (CPS). Building analytic models (those that analyze performance or dependability attributes) of a CPS requires each functional configuration of the system to be separately modeled. This modeling process must be repeated for each attribute that the user wants to measure. The end result is that comparing different CPS architectures using multiple metrics is a time-consuming process. In addition, errors in models can be difficult or impossible to detect without detailed analysis.

I propose to build a CPS modeling tool that addresses both of these issues. It will provide an ontological model for describing the functional organization of a CPS. This ontology will be represented in a powerful type system, allowing the tool to perform sanity checking on the functional model. The tool will also be capable of converting this functional model into various performance and dependability models, allowing users to easily model various attributes of a CPS. This tool will form a foundation for real-world CPS design, allowing people to compare the performance of various possible CPS designs with multiple metrics.

Developing this tool will involve three main research tasks. The first task will be to create a powerful ontology expression language. This language must be capable of describing complex relationships between components of a CPS as well as annotating them with information that simulation or analysis tools may not be able to deduce. I will represent these ontologies with a type system, allowing me to borrow type theory concepts for model checking and transformation. For instance, this system could verify that the inputs and outputs of components have the same units. It could also express more complex system requirements concerning the type and topology of components.

Secondly, I will develop methods for transforming functional system models described in terms of an ontology into analytical models of those systems. A 'functional model' describes the physical and cyber components of a system and how they are interconnected. On the other hand, an 'analytic model' describes some aspect of the system's performance or dependability, for instance, end to end delay or reliability. The system will be able to extract essential topological information from the functional model and use this to derive analytic models.

Finally, since many analytic models have already been studied in great detail, I will provide methods to integrate model solution software with this modeling software. Users will be able to choose between solution methods for different models, so long as appropriate software is written to convert those models to a format understood by the solver. This will allow for both exact closed-form solutions and approximate solutions to models, depending on model complexity.

Several other teams have attempted similar work. I will examine two closely related projects and describe how their work relates to my project idea.

2 Comparison with Ptolemy

Ptolemy [1] is developed by Ed Lee's research group at Berkeley. It is under active development; recent publications include [2, 3].

It is designed specifically for cyber-physical systems and widely used for CPS modeling [4, 5].

2.1 Type system

Ptolemy features a type system for describing variables and values in models. This type system is very similar to a basic computer programming type system; it consists of relatively low-level types such as int, short, and double. It includes some more complex types, including array, structure, and union types, as well as matrices and complex numbers that are frequently needed in systems modeling. These types are arranged in a lattice, allowing certain types to be automatically converted to others, provided the lattice describes a method for this conversion. Components can have constraints on their inputs and outputs, specifying, for instance, that the output type must be convertible to the input type.

The main application of the type lattice and type constraints is automatic type inference. Ptolemy can, in most cases, infer the types of component inputs and outputs by starting from data sources in a model and propagating those types forward. In certain cases, especially when cycles are present in the model, this method does not apply. Ptolemy also offers backwards type inference, which works backwards from specified output types to deduce other types in the model. This can allow type inference to occur even in the presence of cycles that forward type inference could not deduce a type for.

Ptolemy's type system has a few disadvantages. First, Hindley-Milner type deduction, which is used in Haskell and other languages, is capable of far more complex type deductions than simple forward/back inference. As well, it is possible to create infinitely nested types that cause Ptolemy's type deducers to loop forever.

2.2 Ontology

In addition to its type system, Ptolemy contains tools to construct ontologies for models. The ontology system allows for both ontology deduction and model checking. Unlike the type system, it is built on Hindley-Milner type theory, which allows Ptolemy to deduce ontologies for complex models. As with the typing system, components can set input and output constraints which both assist ontology deduction and provide a means for checking the model.

One example usage for ontologies is matching units between inputs and outputs and tracking unused and constant parameters. The addition of the ontology system is relatively new [6], so features are still being added to it and it is not widely used yet.

3 Comparison with Möbius

Möbius [7, 8] is developed by Bill Sanders' research group at the University of Illinois. It is also under active development [9, 10]. Möbius is designed more for network and information system modeling, but the CRUTIAL project and some others have used it for CPS modeling [11, 12, 13, 14, 15, 16]. As well, its primary modeling formalism, Stochastic Activity Nets (SANs), are a generalization of both Stochastic Petri Nets (SPNs) and Markov chains, which are widely used in CPS modeling.

3.1 Model Conversion

One unique attribute of Möbius is that it does not limit the user to the modeling tools it contains. Users can construct SANs with a built-in tool, or interface other modeling tools using an API. Models are hierarchical, so each component can be modeled in detail; these detailed models are then composed together into a system-wide model. Being able to compose different types of models like this gives users the ability to model systems with components from several different domains, using the best tool for each component.

Models can be solved using simulation techniques. Möbius has several techniques for solving co-dependent models or models with circular dependencies. It can also apply analytic techniques to derive exact solutions for models that have the appropriate annotations (mostly to specify event probability distributions). Hierarchical model solving is an ongoing research area in Möbius [17, 12].

However, Möbius does not have the strong ontology system that Ptolemy does. As well, model verification for heterogeneous models is a very difficult problem that currently isn't done well [17].

4 Conclusion

Overall, there is a lot of room for improvement in the CPS modeling tool realm. Both Ptolemy and Möbius are solid tools, but both lack features that the other has. As well, there are some larger unsolved problems in this area, mostly related to model checking and model transformation. In particular, converting functional models to analytic models is not a very automated process at the moment, requiring user annotations and only working on certain systems.

One potential avenue for investigating CPS model transformation is the field of software engineering model transformation. Sofware engineering has used model transformation to convert UML models of programs to both program implementations and analytic models of that program. Some examples of this include [18, 19, 20, 21, 22]. Other non-UML methods have investigated deriving nonfunctional attribute models from specifications [23, 24, 25].

Conclusion: While there is definitely work to be done in this area, I'm not sure it would be a good phd project. - Lots of reimplementation or have to build on others' work - Competition with other students working on those projects directly

References

[1] E. Lee, "Ptolemy project home page," 2015.

[2] D. Broman, L. Greenberg, E. A. Lee, M. Masin, S. Tripakis, and M. Wetter, "Requirements for hybrid cosimulation standards," 2015.

[3] C. Ptolemaeus, ed., *System design, modeling, and simulation: using Ptolemy II.* Berkeley, Calif: UC Berkeley EECS Dept, 1. ed., version 1.02 ed., 2014.

[4] I. Akkaya, Y. Liu, and E. A. Lee, "Modeling and simulation of network aspects for distributed cyber-physical energy systems," in *Cyber Physical Systems Approach to Smart Electric Power Grid* (S. K. Khaitan, J. D. McCalley, and C. C. Liu, eds.), Power Systems, pp. 1–23, Springer Berlin Heidelberg, 2015.

[5] I. Akkaya, S. Emoto, and E. A. Lee, "PILOT: An actor-oriented learning and optimization toolkit for robotic swarm applications," 2015.

[6] B. Lickly, C. Shelton, E. Latronico, and E. A. Lee, "A practical ontology framework for static model analysis," in *Proceedings of the Ninth ACM International Conference on Embedded Software*, EMSOFT '11, (New York, NY, USA), pp. 23–32, ACM, 2011.

[7] B. Sanders, "The möbius tool :: Home," 2015.

[8] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J. Doyle, W. Sanders, and P. Webster, "The mobius modeling tool," in *9th International Workshop on Petri Nets and Performance Models, 2001. Proceedings*, pp. 241–250, 2001.

[9] C. Buchanan and K. Keefe, "Simulation debugging and visualization in the möbius modeling framework," in *Quantitative Evaluation of Systems* (G. Norman and W. Sanders, eds.), no. 8657 in Lecture Notes in Computer Science, pp. 226–240, Springer International Publishing, Sept. 2014.

[10] S. Gaonkar, K. Keefe, R. Lamprecht, E. Rozier, P. Kemper, and W. H. Sanders, “Performance and dependability modeling with möbius,” *SIGMETRICS Perform. Eval. Rev.*, vol. 36, pp. 16–21, Mar. 2009.

[11] G. Dondossola, “Deliverables of the project - crutial - CRitical utility InfrastructurAL resilience,” 2015.

[12] J. Fitzgerald, K. Pierce, and C. Gamble, “A rigorous approach to the design of resilient cyber-physical systems through co-simulation,” in *2012 IEEE/IFIP 42nd International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 1–6, June 2012.

[13] S. Chiaradonna, F. D. Giandomenico, and P. Lollini, “Evaluation of critical infrastructures: Challenges and viable approaches,” in *Architecting Dependable Systems V* (R. d. Lemos, F. D. Giandomenico, C. Gacek, H. Muccini, and M. Vieira, eds.), no. 5135 in Lecture Notes in Computer Science, pp. 52–77, Springer Berlin Heidelberg, 2008.

[14] S. Chiaradonna, F. Di Giandomenico, and N. Nostro, “Model-based assessment of multi-region electric power systems showing heterogeneous characteristics,” in *Computer Safety, Reliability, and Security*, pp. 328–339, Springer, 2012.

[15] D. Jin, *Network-simulation-based evaluation of smart grid applications.* PhD thesis, University of Illinois at Urbana-Champaign, 2013.

[16] R. Bloomfield, L. Buzna, P. Popov, K. Salako, and D. Wright, “Stochastic modelling of the effects of interdependencies between critical infrastructure,” in *Critical Information Infrastructures Security*, pp. 201–212, Springer, 2010.

[17] E. Ábrahám, A. Avritzer, A. Remke, and W. H. Sanders, “Randomized timed and hybrid models for critical infrastructures (dagstuhl seminar 14031),” 2014.

[18] P. King and R. Pooley, “Using UML to derive stochastic petri net models,” in *Proceedings of the 15th UK Performance Engineering Workshop*, pp. 45–56, 1999.

[19] S. Balsamo, A. Di Marco, P. Inverardi, and M. Simeoni, “Model-based performance prediction in software development: A survey,” *IEEE Trans. Softw. Eng.*, vol. 30, pp. 295–310, May 2004.

[20] V. Cortellessa, K. Goseva-Popstojanova, K. Appukkutty, A. R. Guedem, A. Hassan, R. Elnaggar, W. Abdelmoez, and H. H. Ammar, “Model-based performance risk analysis,” *IEEE Trans. Softw. Eng.*, vol. 31, pp. 3–20, Jan. 2005.

[21] C. Eichner, H. Fleischhack, R. Meyer, U. Schrimpf, and C. Stehno, "Compositional semantics for UML 2.0 sequence diagrams using petri nets," in *SDL 2005: Model Driven* (A. Prinz, R. Reed, and J. Reed, eds.), no. 3530 in Lecture Notes in Computer Science, pp. 133–148, Springer Berlin Heidelberg, 2005.

[22] M. Woodside, "From annotated software designs (UML SPT/MARTE) to model formalisms," in *Formal Methods for Performance Evaluation* (M. Bernardo and J. Hillston, eds.), no. 4486 in Lecture Notes in Computer Science, pp. 429–467, Springer Berlin Heidelberg, 2007.

[23] A. Sabetta, D. C. Petriu, V. Grassi, and R. Mirandola, "Abstraction-raising transformation for generating analysis models," in *Satellite Events at the MoDELS 2005 Conference* (J.-M. Bruel, ed.), no. 3844 in Lecture Notes in Computer Science, pp. 217–226, Springer Berlin Heidelberg, 2006.

[24] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *Formal Methods for Performance Evaluation* (M. Bernardo and J. Hillston, eds.), no. 4486 in Lecture Notes in Computer Science, pp. 220–270, Springer Berlin Heidelberg, 2007.

[25] S. Gilmore, L. Gönczy, N. Koch, P. Mayer, M. Tribastone, and D. Varró, "Non-functional properties in the model-driven development of service-oriented systems," *Software & Systems Modeling*, vol. 10, pp. 287–311, Mar. 2010.

[26] H. Feng, *Model transformation with hierarchical discrete-event control*. PhD thesis, Citeseer, 2009.

[27] D. Jin, D. M. Nicol, and G. Yan, "An event buffer flooding attack in DNP3 controlled SCADA systems," in *Proceedings of the Winter Simulation Conference*, WSC '11, (Phoenix, Arizona), pp. 2619–2631, Winter Simulation Conference, 2011.

[28] P. Verissimo, A. Bessani, M. Correia, N. Neves, and P. Sousa, "Designing modular and redundant cyber architectures for process control: Lessons learned," in *42nd Hawaii International Conference on System Sciences, 2009. HICSS '09*, pp. 1–8, Jan. 2009.

[29] P. R. D'Argenio, H. Hermanns, J.-P. Katoen, and R. Klaren, "MoDeST — a modelling and description language for stochastic timed systems," in *Process Algebra and Probabilistic Methods. Performance Modelling and Verification* (L. d. Alfaro and S. Gilmore, eds.), no. 2165 in Lecture Notes in Computer Science, pp. 87–104, Springer Berlin Heidelberg, 2001.

[30] H. Bohnenkamp, T. Courtney, D. Daly, S. Derisavi, H. Hermanns, J.-P. Katoen, R. Klaren, V. V. Lam, and W. H. Sanders, "On integrating the möbius and MoDeST modeling tools," in *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 671–671, IEEE Computer Society, 2003.