Controlling Software Execution: An Emerging Application Area for Control Engineering

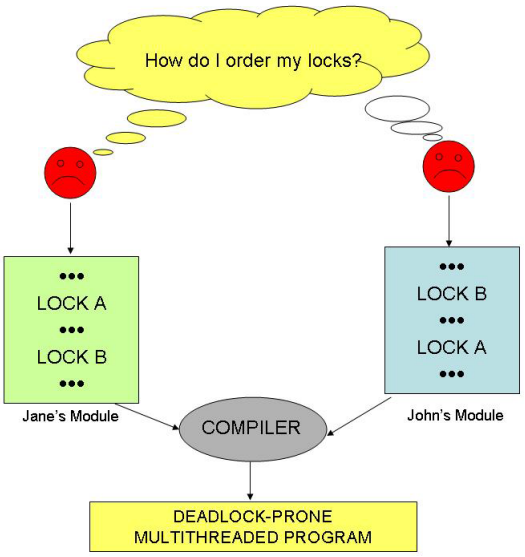
Half-day Workshop at ACC 2012

26 June 2012, 1pm-5pm

Yin Wang (Organizer), Stéphane Lafortune (Co-organizer), and Spyros Reveliotis (Co-organizer)

I. Motivation and Goals

Computer and software engineering is a rich application area for control systems technology. Indeed, classical control theory, based on continuous-variable and time-driven models, has been applied to computer systems problems, such as throughput stabilization, and has achieved commercial success [1]. Many important problems in computer and software engineering, however, are discrete in nature and naturally fall into the realm of discrete-variable and event-driven systems, i.e., Discrete Event Systems (DES). The growth of software defects has paralleled and shadowed the rapid advancement in computer technology. Relevant issues such as safety and consistency are inherently discrete-event problems. Recently, there has been a growing interest in the application of concepts and techniques from DES to software systems and embedded systems; see, e.g., [2], [3], [4], [5], [6], [7], [8], [9].



In particular, the paradigm of *controlling software execution to avoid software defects at runtime* has received significant attention in three communities: control engineering [5], [10], programming languages [11], and operating systems [12]. The objective of this workshop is to present this paradigm to a control audience, with relevant background on software issues and DES theory, using recent work on deadlock avoidance in multithreaded software by the organizers as the guiding case study. More specifically, the goals of the workshop are the following:

- Introduce computer systems problems to the control community, and discuss possible applications of control theory in this domain.
- Educate the attendees about how to apply DES control technology to real computer systems problems, using the circular-mutex-wait deadlock avoidance project as an example.
- Discuss the relevant control theory, its benefits that are especially relevant in computer science applications, and the missing gaps that need to be addressed for successful application.
- Stimulate involvement of control researchers and students in this emerging area of research, both on the applications front and on the theoretical front.

II. Background and Rationale

The paradigm of controlling software execution to avoid software defects at runtime was developed to tackle decades-old thorny issues related to software dependability. In the methodology developed by our team under the so-called "Gadara Project" [13], [12], well-understood DES control techniques such as supervisory control [14] and supervision based on place invariants [15] are employed to synthesize control logic that is instrumented into the source code and enforces safety properties at runtime. Compared with typical solutions previously used by software systems engineers, this new methodology is model-based and guarantees desirable features such as correctness and maximal permissiveness. As a result, this approach has been very well received in the computer

Y. Wang is with HP Labs, Palo Alto. S. Lafortune is with the University of Michigan, Ann Arbor. S. Reveliotis is with the Georgia Institute of Technology, Atlanta.

science community [7]. Moreover, this work led to the development of better control solutions that exploit the special features and address the specific requirements of the given application [16], [17].

The principal focus of the Gadara project so far has been the problem of circular-wait-mutex deadlock in multithreaded software. This is an important problem due to the prevalence of multicore computer architectures. In fact, there are numerous other software problems where we believe control engineering techniques from the field of DES hold great promise. These include other types of deadlocks, such as reader-writer deadlock, condition wait/signal deadlock, inter-process deadlock, and other concurrency issues such as race, atomicity violation, and priority inversion. There are many other application domains as well.

While the above-mentioned opportunities can often be solved by existing DES control theory, better customized solutions are often desirable. A crucial issue is scalability, which often necessitates the development of customized algorithms that exploit problem structure. Another crucial issue is the requirement on runtime overhead of the control logic in software applications, which is much more stringent than in other application areas such as manufacturing systems or process control, for instance. This leads to numerous opportunities to advance the state-of-the-art of DES control theory. For example, seeking to establish non-blocking behavior through the SBPI technique mentioned above involves an iterative procedure that, in the general case, may not converge [15]. Yet, it is possible to show that in the context of the circular-wait-mutex deadlock avoidance problem, the SBPI method can be implemented with convergence guarantees. Furthermore, for this type of deadlock, we have also developed a new methodology that is able to synthesize the maximally permissive deadlock avoidance policy in a non-iterative manner, while minimizing the structure of the derived controller (and therefore, the computational overhead that is incurred by its embedding in the undlerying application) [17].

Thus, it can be argued that the application of the control engineering paradigm and of DES techniques to computer systems opens up new avenues of research that cover both theory and implementation. We are frequently asked by computer science graduate students and faculty members regarding possible applications of control theory to their problems. Some of these inquiries can be addressed by existing theories (yet, often, software packages are much more desirable), some require customization of existing solution methodologies (open-source tools can help in that regard), some need new developments in the control theory of DES, yet the others are fundamental open issues in control theory. We believe this emerging research area is very promising and we hope that this workshop will encourage more students and researchers to work on the applications of DES in computer and software engineering.

III. Intended Audience

The workshop is principally aimed at control faculty and graduate students with an interest in DES and/or in computer system applications. The workshop should also be of interest to industrial attendees with a control background and whose work bring them in contact with software defect issues. Prior in-depth knowledge of DES control theory or of computer systems problems will not be required.

IV. Schedule and Speakers

The workshop will be in the afternoon of June 26 and will last approximately four hours. The coverage and speakers are as follows.

- (half an hour) Stéphane Lafortune: Software failures, concurrency bugs and their typical computer science solutions.
- (half an hour) Stéphane Lafortune: Necessary DES background: automata and Petri net models, basic supervisory control theory, and supervision based on place invariants.
- (one hour) Yin Wang: A tutorial on the Gadara methodology for applying DES control theory to dynamically avoid deadlock bugs in concurrent software. The tutorial will employ software tools developed in the Gadara project, to give a more concrete idea of how DES works in practice.
- (1/4 of an hour) Break.
- (one hour) Spyros Reveliotis: Limitations of existing control theory to address above challenges; recent theoretical developments to address these challenges; open problems of interest.
- (half an hour) Yin Wang: Other possible applications of DES in software systems; challenges in modeling and control; new research directions motivated by real computer systems issues. Concluding remarks.
- (remainder) Open discussion.

V. Bio-Sketches of Speakers

Yin Wang is a research scientist at Hewlett-Packard Labs, Palo Alto. He received Bachelor's (2000) and Master's (2003) degrees from the Shanghai Jiao Tong University, Department of Automation. He earned his Ph.D. at the University of Michigan Electrical Engineering and Computer Science department and joined HP Labs in early 2009. While a graduate student, Wang interned at Microsoft Shanghai, IBM Almaden Research Center, and HP Labs. His research interests are in the application of Discrete Event Systems theory to computer systems.

Stéphane Lafortune is a Professor of EECS at the University of Michigan. He received the B. Eng degree from Ecole Polytechnique de Montréal in 1980, the M. Eng. degree from McGill University in 1982, and the Ph.D. degree from the University of California at Berkeley in 1986, all in electrical engineering. He joined the University of Michigan in 1986. Dr. Lafortune is a Fellow of the IEEE (1999). His research interests are in Discrete Event Systems and include multiple problem domains: modeling, diagnosis, control, optimization, and applications to computer systems. He is the lead developer of the software package UMDES. He co-authored, with C. Cassandras, the textbook *Introduction to Discrete Event Systems - Second Edition* (Springer, 2008).

Spiridon (Spyros) Reveliotis is a Professor at the School of Industrial & Systems Engineering at the Georgia Institute of Technology. He holds a Bachelor's degree from the National Technical University of Athens, Greece, a Master's degree in Computer Systems Engineering from Northeastern University, Boston, and a Ph.D. degree in Industrial Engineering from the University of Illinois at Urbana-Champaign. His research interests evolve in the area of Discrete Event Systems and their applications, with special emphasis on the problem of the design and real-time management of the resource allocation function that takes place in many contemporary technological applications. A significant part of his work is epitomized in the monograph *Real-Time Management of Resource Allocation Systems: A Discrete Event Systems Approach*.

References

[1] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback Control of Computing Systems*. Wiley, 2004.

[2] C. Liu, A. Kondratyev, Y. Watanabe, J. Desel, and A. Sangiovanni-Vincentelli, "Schedulability analysis of Petri nets based on structural properties," in *Proc. International Conference on Application of Concurrency to System Design*, 2006.

[3] C. Dragert, J. Dingel, and K. Rudie, "Generation of concurrency control code using discrete-event systems theory," in *Proc. ACM International Symposium on Foundations of Software Engineering*, 2008.

[4] A. Auer, J. Dingel, and K. Rudie, "Concurrency control generation for dynamic threads using discrete-event systems," in *Proc. Allerton Conference on Communication, Control and Computing*, 2009.

[5] M. V. Iordache and P. J. Antsaklis, "Petri nets and programming: A survey," in *American Control Conference*, 2009.

[6] A. Gamatie, H. Yu, G. Delaval, and E. Rutten, "A case study on controller synthesis for data-intensive embedded system," in *Proc. International Conference on Embedded Software and Systems*, 2009.

[7] T. Kelly, Y. Wang, S. Lafortune, and S. Mahlke, "Eliminating concurrency bugs with control engineering," *Computer*, vol. 42, no. 12, pp. 52–60, 2009.

[8] M. V. Iordache and P. J. Antsaklis, "Concurrent program synthesis based on supervisory control," in *Proc. 2010 American Control Conference*, 2010, pp. 3378–3383.

[9] G. Delaval, H. Marchand, and E. Rutten, "Contracts for modular discrete controller synthesis," in *Proc. ACM Conference on Languages, Compilers and Tools for Embedded Systems*, 2010.

[10] J. Dingel, K. Rudie, and C. Dragert, "Bridging the gap: Discrete-event systems for software engineering (short position paper)," in *Proceedings of C* Conference on Computer Science and Software Engineering (C3S2E'09)*, May 2009, pp. 67–71.

[11] Y. Wang, S. Lafortune, T. Kelly, M. Kudlur, and S. Mahlke, "The theory of deadlock avoidance via discrete control," in *POPL*, 2009.

[12] Y. Wang, T. Kelly, M. Kudlur, S. Lafortune, and S. A. Mahlke, "Gadara: Dynamic deadlock avoidance for multithreaded programs," in *OSDI*, 2008.

[13] Y. Wang, T. Kelly, and S. Lafortune, "Discrete control for safe execution of IT automation workflows," in *ACM EuroSys Conference*, 2007.

[14] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, 1987.

[15] J. O. Moody and P. J. Antsaklis, *Supervisory Control of Discrete Event Systems Using Petri Nets*. Kluwer, 1998.

[16] H. Liao, S. Lafortune, S. A. Reveliotis, Y. Wang, and S. A. Mahlke, "Synthesis of maximally-permissive liveness-enforcing control policies for gadara petri nets," in *IEEE Conference on Decision and Control*, 2010.

[17] A. Nazeem, S. Reveliotis, Y. Wang, and S. Lafortune, "Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: The linear case," *Automatic Control, IEEE Transactions on*, vol. 56, no. 8, pp. 1818 –1833, aug. 2011.