9-2015

Event and strategy analytics

Jin Song DONG

Jun SUN
Singapore Management University, junsun@smu.edu.sg

Yang LIU

Yuan-Fang LI

Jing SUN

See next page for additional authors

Citation

DONG, Jin Song; SUN, Jun; LIU, Yang; LI, Yuan-Fang; SUN, Jing; and SHI, Ling. Event and strategy analytics. (2015). *Proceedings of the 2015 International Symposium on Theoretical Aspects of Software Engineering (TASE), Nanjing, China, September 12-14*. 1, 4-6. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/4976

Event and Strategy Analytics

Jin Song Dong*, Jun Sun†, Yang Liu‡, Yuan Fang Li§, Jing Sun¶, and Ling Shi*
*National University of Singapore, {dongjs, shiling}@comp.nus.edu.sg
†Singapore University of Technology and Design, sunjun@sutd.edu.sg
‡Nanyang Technology University, yangliu@ntu.edu.sg
§Monash University, yuanfang.li@monash.edu
¶University of Auckland, j.sun@cs.auckland.ac.nz

***Abstract*—Model checking has been pervasive and successful in finding bugs in hardware and software systems, including real-time and probabilistic systems. Applying model checking to decision making is relative new and has an excellent potential to be compliment to data analytics and other Artificial Intelligent (AI) or Operational Research (OR) based decision making techniques. Our last 8 years research has focused on the development of PAT (Process Analysis Toolkit) [18] which supports modelling languages that combine the expressiveness of event, state, time and probability based modeling techniques to which model checking can be directly applied. The next direction for PAT is to move from verification to analytics, we call it "Event Analytics" with a special focus on "Strategy Analytics".**

I. Background on PAT

PAT is a verification framework rather than just a single model checker, and it currently supports many different formalisms and languages ranging from graphical Timed Automata to programming languages for sensor networks. Its core language is called CSP# [17] which is based on Hoare's event based formalism CSP (Communicating Sequential Processes) extended with shared variables and the design of the CSP# with extensions are influenced by the integrated specification techniques (e.g., [12]). The formal semantics of CSP# [13] is defined in Unifying Theories of Programming [7]. The key idea is to treat sequential terminating programs, which may be as complex as C# programs, as events. The resulting modeling language is highly expressive and can cover many application domains such as concurrent data structures [11], web services [20], sensor networks [22], multi-agent systems [6], [14], mobile systems [3] and cyber security systems [1], [2], [5]. The PAT system is designed to facilitate the development of customized model checkers and analysis tools. It has an extensible and modularized architecture to support new languages, reduction, abstraction and new model checking algorithms [4]. PAT has attracted more than 3500 registered users from hundreds organizations around the world.

II. Planning/Scheduling as Model Checking

Recently, we investigated the feasibility of using model checking to solve classic planning problems [8]. Our experimental results indicate that the performance of PAT is comparable to that of state-of-the-art AI planners for certain problem categories. In addition, a successful application of PAT to an intelligent public transportation management system, Transport4You, has won the ICSE 2011 SCORE Competition [9]. In the Transport4You project, PAT model checker was used not only as a verification tool but also as a service that computes optimal travel plans. PAT's new real-time and probabilistic verification modules can reason about real-time properties and calculate min/max probabilistic values for particular events or state [15]. This sets a solid foundation for applying Model Checking technology systematically to various problems on decision making, which is the research direction for the next version of PAT: Event and Strategy Analytics.

III. Event Analytics

Event analytics (EA) duels with timed and probabilistic events that can evolve dynamically. The sophisticated algorithms need to be developed to synthesize timing and probabilistic parameter variables for real-time and probabilistic concurrent systems. Domain specific models with abstraction are critical for the accuracy and efficiency of analytic systems. We note that while "data" is typically static "event" are dynamic and involve causality, communication, timing and probability. We believe EA driven technologies can offer significant advantages that are orthogonal to those based on "data analytics". With EA, we aim to answer the questions like "what is the maximum time delay of a critical event beyond which the overall system reliability will be compromised" and "what is the minimum probability shift (delta) of a specific event that will tip the balance of the winning strategy".

IV. Strategy Analytics

Decision making based on uncertainty has been well researched in AI and OR communities. Probabilistic model checking systems that can handle complex state may offer new ways for strategy analysis based on probability. We have recently conducted an interesting application on sports strategy analysis for tennis using the PAT probability model checking module. We automatically extracted average probability distribution for each tennis stroke/action (event) from online data, then generated Markov decision process (MDP) model (profile) in PAT for each top 100 Association of Tennis Professionals (ATP) and Women's Tennis Association (WTA) players. With the MDP models for each player, we were able to auto-generate prediction outcomes for any two players. Our prediction results are much more accurate in comparison to the best online sports betting sites. What is more interesting is that we can generate sensitivity diagrams and highlight the

978-1-4673-7612-9/15 $31.00 © 2015 IEEE
DOI 10.1109/TASE.2015.28

IEEE computer society

potential small improvements which could lead to significant impact to the winning percentage. This type of sports strategy analysis is in fact a special case of Event Analytics. The ideas behind it can be generalized to apply to many other domains, i.e., financial decision making and military strategy analysis. There are also a number of interesting research observations and directions that we can consider and discuss further.

V. Complimentary to Data Analytics

Big Data and Data Analytics have received much hype in recent years. One significant limitation of current data analysis techniques is the use of machine learning based black-box techniques to generate results that cannot be explained. The ability to extract critical events from Big Data and to synthesize high-level models from such events can allow us to gain insights that are previously unattainable. For instance, better control on analysis that offer guarantees in accuracy or trust, combined with explanation can allow more confident decision making that rely on Big Data analysis.

VI. Event and Model Extraction

Large amounts of data streams can be generated from different sources, such as online websites, social media and sensors. The granularity of such data may be too fine, and the quantity may still be too large for model checking techniques even with various reduction techniques. The data generated from these sources are not random: there are often (implicit or explicit) structures and semantics behind it. In other words, knowledge can be extracted from such data. It is important to investigate the integration of data mining techniques to continually extract patterns from raw data. Such patterns, higher-level summaries, will then be turned into event traces which can be more effectively utilized as inputs to model checking.

Events extracted from Big Data are temporal in nature: they occur sequentially or concurrently, and form concurrent event traces that are interacting in complex ways. An expressive mathematically based model that represents an entire system using states and events will enable deep analyses of interacting event traces on a globally level. For example, the L* algorithm is proposed to learn deterministic finite automata (DFA) from a set of events. It will be interesting to investigate the problem of synthesizing, or generating appropriate models from event traces which may be based on our early synthesis and verification work [16], [10], [21], [19]. Model checking techniques have traditionally been applied to the analysis and verification of software and hardware systems, where complete knowledge of the system and its environment is usually assumed. However, such an assumption is often too strong for open scenarios such as emergency response and infectious disease management. It is important to investigate novel model checking techniques that are capable of handling such organic systems.

VII. Wide Applications and Linking to AI and OR

EA systems can certainly be deployed to assist the decision making and risk analysis in financial systems, and they can also provide context based activity/service planning for cyber-physical systems. For future research it will be interesting to investigate the potential integration of AI uncertainty reasoning techniques, OR optimisation techniques and Data Analytics into EA systems. The first step in this integration will be to identify a complex decision problem which sub-problems could be solved by different reasoning systems and their input/output can be linked and evolved together to solve the overall complex problem.

Acknowledgment

We would like to thank E. Clarke, D. Rosenblum, A.P. Sheth, P.S. Thiagarajan and many others for insightful discussions on these topics.

References

[1] G. Bai, J. Hao, J. Wu, Y. Liu, Z. Liang, and A. Martin. TrustFound: Towards a Formal Foundation for Model Checking Trusted Computing Platforms. In *FM'14*, pages 110–126, 2014.

[2] G. Bai, J. Lei, G. Meng, S. Venkatraman, P. Saxena, J. Sun, Y. Liu, and J. S. Dong. AUTHSCAN: Automatic Extraction of Web Authentication Protocols from Implementations. In *NDSS 2013*, 2013.

[3] G. Ciobanu and M. Zheng. Automatic Analysis of TiMo Systems in PAT. In *ICECCS 2013*, pages 121–124, 2013.

[4] J. S. Dong, J. Sun, and Y. Liu. Build your own model checker in one month. In *ICSE'13, tutorial*, pages 1481–1483, 2013.

[5] J. Hao, Y. Liu, W. Cai, G. Bai, and J. Sun. vTRUST: A Formal Modeling and Verification Framework for Virtualization Systems. In *ICFEM 2013*, pages 329–346, 2013.

[6] J. Hao, S. Song, Y. Liu, J. Sun, L. Gui, J. S. Dong, and H-F. Leung. Probabilistic Model Checking Multi-agent Behaviors in Dispersion Games Using Counter Abstraction. In *PRIMA 2012*, pages 16–30, 2012.

[7] C.A.R. Hoare and J. He. *Unifying Theories of Programming*. Prentice-Hall, 1998.

[8] Y. Li, J. S. Dong, J. Sun, Y. Liu, and J. Sun. Model checking approach to automated planning. *Formal Methods in System Design*, 44(2):176–202, 2014.

[9] Y. Li, H. Yang, and H. Wu. Pat approach to transport4you, an intelligent public transportation manager. (Tutors: J. S. Dong, Y. Liu, J. Sun), ICSE 2011 SCORE winner. 94 teams from 22 countries registered the competition where 56 teams completed the projects; two teams were selected as winners.

[10] S-W. Lin, É. André, Y. Liu, J. Sun, and J. S. Dong. Learning Assumptions for CompositionalVerification of Timed Systems. *IEEE Transactions on Software Engineering*, 40(2):137–153, 2014.

[11] Y. Liu, W. Chen, Y. A. Liu, J. Sun, S. Zhang, and J. S. Dong. Verifying Linearizability via Optimized Refinement Checking. *IEEE Transactions on Software Engineering*, 39(7):1018–1039, 2013.

[12] B. Mahony and J. S. Dong. Timed Communicating Object Z. *IEEE Transactions on Software Engineering*, 26(2):150–177, 2000.

[13] L. Shi, Y. Zhao, Y. Liu, J. Sun, J. S. Dong, and S. Qin. A UTP Semantics for Communicating Processes with Shared Variables. In *ICFEM 2013*, pages 215–230, 2013.

[14] S. Song, J. Hao, Y. Liu, J. Sun, H-F. Leung, and J. Zhang. Improved EGT-based Robustness Analysis of Negotiation Strategies in Multi-agent Systems via Model Checking. *IEEE Transactions on Human-Machine Systems (THMS)*, (Accepted).

[15] S. Song, J. Sun, Y. Liu, and J. S. Dong. A Model Checker for Hierarchical Probabilistic Real-Time Systems. In *CAV 2012*, pages 705–711, 2012.

[16] J. Sun and J. S. Dong. Design Synthesis from Interaction and State-Based Specifications. *IEEE Transactions on Software Engineering*, 32(6):349–364, 2006.

[17] J. Sun, Y. Liu, J. S. Dong, and C. Chen. Integrating Specification and Programs for System Modeling and Verification. In *TASE'09*, pages 127–135, 2009.

[18] J. Sun, Y. Liu, J. S. Dong, and J. Pang. PAT: Towards Flexible Verification under Fairness. In *CAV'09*, pages 709–714, 2009.

[19] J. Sun, H. Xiao, Yang Liu, S-W. Lin, and S. Qin. TLV: Abstraction through Testing, Learning and Validation. In *FSE 2015. (Accepted)*.
[20] T. H. Tan, É. André, J. Sun, Y. Liu, J. S. Dong, and M. Chen. Dynamic synthesis of local time requirement for service composition. In *ICSE'13*, pages 542–551, 2013.
[21] Y. Xue, J. Wang, Y. Liu, H. Xiao, J. Sun, and M. Chandramohan. JS*: Detection and Classification of Malicious JavaScript via Attack Behavior Modelling. In *ISSTA 2015. (Accepted)*.
[22] M. Zheng, D. Sanán, J. Sun, Y. Liu, J. S. Dong, and Y. Gu. State Space Reduction for Sensor Networks Using Two-Level Partial Order Reduction. In *VMCAI 2013*, pages 515–535, 2013.