9-2014

Event analytics

Jin Song DONG

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

Yang LIU

Yuan-Fang LI

Event Analytics

Jin Song Dong[1], Jun Sun[2], Yang Liu[3], and Yuan-Fang Li[4]

[1] National University of Singapore, Singapore
[2] Singapore University of Technology and Design, Singapore
[3] Nanyang Technology University, Singapore
[4] Monash University, Australia

Abstract. The process analysis toolkit (PAT) integrates the expressiveness of state, event, time, and probability-based languages with the power of model checking. PAT is a self-contained reasoning system for system specification, simulation, and verification. PAT currently supports a wide range of 12 different expressive modeling languages with many application domains and has attracted thousands of registered users from hundreds of organizations. In this invited talk, we will present the PAT system and its vision on “Event Analytics” (EA) which is beyond “Data Analytics”. The EA research is based on applying model checking to event planning, scheduling, prediction, strategy analysis and decision making. Various new EA research directions will be discussed.

1 Introduction

Large complex systems that generate intricate patterns of streaming events arise in many domains. These event streams arise from composite system states and control flows across many interacting components. Concurrency, asynchrony, uncertain environments - leading to probabilistic behaviours- and real-time coordination are key features of such systems. Many of the functionalities are realized in these systems by embedded software (and hardware) that must interact with the physical agents and processes. The proper functioning of such systems depends crucially on whether the software-mediated event patterns that are generated fulfill the required criteria. For example in a public transport system such as the Metro railway systems in large cities the control software must ensure that the distances between two trains sharing a track must never fall below a certain threshold and at the same time must optimize the number of trains deployed on track and their speeds to cater for increased demand for service during peak hours.

The key barriers to designing and deploying software-controlled complex systems are capturing system requirements and parameters and verifying important reliability, security and mission critical properties. There are well known methods (the so-called formal methods) for tackling this problem that are based on mathematical modelling and logic. The history of formal methods can be traced back to an early paper “Checking Large Routine” presented by Alan Turing at a conference on High Speed Automatic Calculating Machines at Cambridge University in 1949. More recently one particularly successful technique called Model Checking [CE81] was recognized through the Turing award being awarded to its creators Clarke, Emerson and Sifakis. At a Computer Aided Verification (CAV) conference, a research group from the Intel Corporation reported that the entire Intel i7 CPU core execution cluster was verified

G. Ciobanu and D. Méry (Eds.): ICTAC 2014, pp. 17–24, 2014.

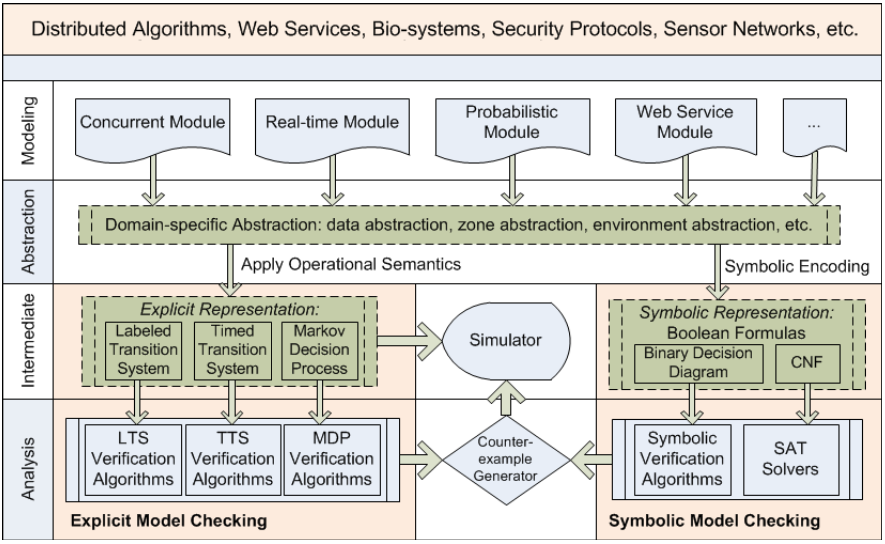
using model checking without a single test case [KGN+09]. The Static Driver Verifier (SDV) [BCLR04] from Microsoft, a model checking system, has been deployed to verify Windows driver software automatically. Bill Gates in 2002 stated:

> "For things like software verification, this has been the Holy Grail of computer science for many decades. But now, in some very key areas for example driver verification we're building tools that can do actual proofs of the software and how it works in order to guarantee the reliability. "

These two industrial case studies are particularly exciting because they show that the exhaustive search techniques that the model checking method is based on can handle systems of large sizes: Intel i7 has eight cores and millions of registers, whereas driver software typically has thousands of lines of code. For a restricted version of model checking called bounded model checking, which often suffices in many practical settings, the scope of applicability is further enhanced by the so called SMT solvers [DeB08]. As a result many industries have started to actively invest in model checking technology [MWC10].

2 PAT Model Checking Systems

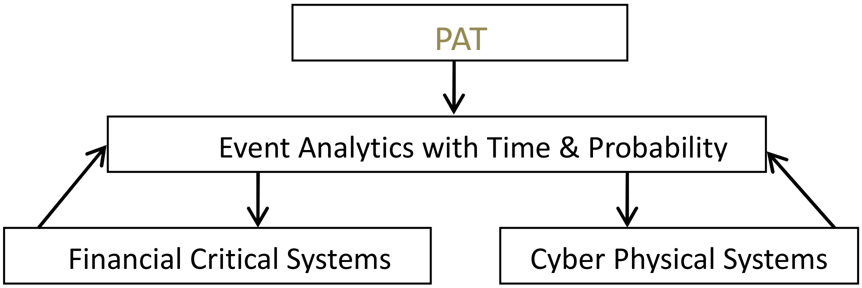
Many model checkers have been developed and successfully applied to practical systems, among which the most noticeable ones include SPIN, NuSMV, FDR, UPPAAL, PRISM and the Java Pathfinder. However those tools are designed for specialized domains and are based on restrictive modeling languages. Users of such systems usually need to manually translate models from the user's domain to the target language. In contrast, the Process Analysis Toolkit (PAT) [SLDP09] support modelling languages that combine the expressiveness of event, state, time and probability based modeling techniques to which model checking can be directly applied. PAT currently supports 12 different formalisms and languages ranging from graphical Timed Automata to programming languages for sensor networks. PAT is a self-contained system for system specification, simulation, and verification. Its core language is called CSP# which is based on Hoare's event based formalism CSP (Communicating Sequential Processes) [Hoare85] and the design of the CSP# is influenced by the integrated specification techniques (e.g. [MD00, TDC04]). The formal semantics of CSP# [SZL+13] is defined in Unified Theory of Programming [HH98]. 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 [LCL+13], web services [TAS+13], sensor networks [ZSS+13], multi-agent systems [HSL+12], mobile systems [CZ13] and cyber security systems [BLM+13]. 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 [DSL13]. PAT has attracted more than 3,000 registered users from 800+ organizations in 71 countries, including companies such as Microsoft, HP, Sony, Hitachi, Canon and many others. Many universities use PAT for teaching advanced courses. The following diagram illustrates the PAT architecture.



Recently, we have successfully applied PAT to event planning and scheduling problems [LSD+14] and have developed verification modules for real-time and probabilistic systems [SSLD12]. These research results provide a solid foundation for a new future research direction, namely, to develop model checking based approaches to event analytics that can serve the needs of planning and prediction as well as decision making. We note that while "data" is typically static "event" are dynamic and involve causality, communication, timing and probability. We believe event analytics (EA) driven technologies can offer significant advantages that are orthogonal to those based on "data analytics".

3 Event Analytics: A New Proposal

The development of novel parameterised real-time and parameterised probabilistic model checking algorithms and systems can support complex event analytics, i.e., to automatically 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". While we believe there are wide applications for event analytics, we have started to work on the mission critical aspects of two major application domains: financial transactions systems and cyber-physical systems. The EA methodology and the accompanying tools can be validated in (applied to and get feedback from) those two major application domains:



3.1 Event Analytics

One of the goals of event analytics is to construct event streams that lead from the initial state to the desired goal states. Recently, we investigated the feasibility of using model checking to solve classic planning problems [LSD+14]. 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, called Transport4You, won the ICSE 2011 SCORE Competition [LYW11]. In the Transport4You project, PAT model checker is 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 a particular events or states (the methodology and some preliminary results are reported in [SSLD12] with fixed value for timing and probability parameters). This sets a solid foundation for the proposed EA research. For EA systems to work with timed and probabilistic events that can evolve dynamically, one must develop sophisticated algorithms that can synthesizes timing and probabilistic parameter variables for real-time and probabilistic concurrent systems. It is important to conduct research to make the techniques scalable by developing new abstraction and reduction techniques, and apply multi-core and many-core verification to improve the performance. EA systems can then 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. It will be interesting to investigate the potential integration of optimisation techniques from Operations Research into EA systems.

3.2 Financial Critical Systems Verification

Financial software systems are critical and thus subject to strict requirements on functional correctness, security and reliability. With the recent development of online banking through mobile devices and computerized stock trading, the potential damage which could be the result of software vulnerability (e.g., credit card information leakage, financial loss due to high frequency trading) is high. This is why often strong regulations, e.g., Internet Banking and Technology Risk Management Guidelines from the Monetary Authority of Singapore, are imposed on the financial software design/development/testing process. While the regulations provide a checklist that could contribute to the quality of the software system, there are hardly any formal guarantees. The Mondex project on smart cards has been developed by NatWest bank in collaboration with Oxford

University using formal methods. As a result, Mondex became the first financial product to achieve certification at the highest level, namely, ITSEC level E6 [WSC+08]. While the Mondex project is a success story, demonstrating that applying formal methods does provide a guarantee in terms of system correctness/security, the techniques used in the project are rather limited, i.e., primarily formal modeling and manual theorem proving, which requires a considerable amount of expertise and time.

Recently, we have developed a method that combines hypothesis testing and probabilistic reachability analysis to study real-time trading systems. We identified the weak components inside the system so that the system designer can improve these components to improve the reliability of the whole system (some initial results were reported in [GSL+13]). It is important to investigate this further with EA based techniques, along with reliability predictions of other financial critical systems. It will be also interesting to investigate event based risk analytics for financial decision making which can have potential benefits for e.g., Monetary Authority of Singapore (MAS).

3.3 Cyber-Physical Systems Verification

Cyber-physical systems will play important roles in building smart cities. Such systems are fully automatic and they are 'aware' of their environment and self-adaptive to the environment changes. Many successes have been achieved in research laboratories especially for activity monitoring. However, such systems are not widely deployed due to, not only scalability and a lack of guarantees for correctness and reliability, but also the fact that those system are designed for demonstration purpose with well controlled scenarios in a lab environment. It is important to apply EA technology to analysis real environment and deliver highly reliable systems and reduce the prototyping time and cost. We also plan to apply event analytics to provide automatic intelligent assistive services in the smart city context (initial ideas have been applied to smart transportation systems [LYW11]). We have gained substantial experience will be also based on a recent successful application of the PAT system to "Activity Monitoring and UI Plasticity for supporting Ageing with Mild Dementia at Home" (AMUPADH 2010-12) [BMD10], a joint project with Alexandra Hospital in Singapore. After the Ethics Approval granted in 2011, a smart reminding system with cyber connected sensors has been successfully deployed in Peacehaven nursing home in 2012. The experience and techniques gained in AMUPADH project can certainly be generalized and applied in a wider context. It will be interesting to develop domain specific EA techniques that can automatically analysis the probabilistic and real-time services rules for smart city systems. Furthermore, parameterized probabilistic based verification techniques can be applied to estimate overall reliability of the smart city systems based on component subsystem reliabilities. EA techniques can also be developed to facilitate service compositions.

4 Event Analytics vs Data Analytics

Big Data and Data Analytics have received much hype in recent years. However, we surpassed capacity to store all the data we produce while growth in data creation has continued at an exponential rate [BSH13]. A recent study suggests we are able to analyze only 0.5% of all the data. Another significant limitation of current data analysis technique is the use of black box techniques (which is what Machine Learning techniques are) 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 believability or trust, combined with explanation can allow more confident decision making that rely on Big Data analysis. Furthermore, reducing the reliance on prior training data as is the case with majority of current approaches, possibly substituted or complemented by use of prior domain knowledge, would make Big Data analysis more scalable and robust. One future research can aim to combine model checking, machine learning and knowledge representation techniques and create an event analytics-based decision making engine for Big Data.

4.1 Event Extraction from Big Data

Model checking has been applied to large systems with more than 10^20 states [BCM+92]. However Big Data may still pose challenges to state-of-the-art model checking techniques. It is important to investigate techniques that further improve the scalability of model checking-based Event Analytics techniques. Large amounts of data streams can be generated from different sources such as 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 is often (implicit or explicit) structure 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, continually from raw data. Such patterns, higher-level summaries, will then be turned into event traces which can then be more effectively utilized as inputs to model checking.

Ontologies have been widely applied as a median for sharing data and knowledge within and across domains. Long-standing research on ontologies and knowledge representation has developed ontology-based data integration techniques [Len02], which are especially suitable for this purpose. It will be interesting to investigate the problems of knowledge representation and learning to automatically induce event ontologies from raw data. Event ontologies can be supplied to model checking techniques to alleviate the data heterogeneity and scalability problems. Further research on ontology-based data/event integration, optimisation of ontology reasoning and more accurate ontology learning that involves more expressive ontology languages is thus required.

4.2 Model Synthesis from Events

Events extracted from Big Data are temporal in nature: they occur sequentially or concurrently, and form concurrent event traces that are interact 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 [Ang87] 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 base on our early synthesis and verification work [SD06, LAL+14]. 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.

Acknowledgement. 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

[Ang87] Angluin, D.: Learning Regular Sets from Queries and Counterexamples. Information and Computation 75(2), 87–106 (1987)

[BCLR04] Ball, T., Cook, B., Levin, V., Rajamani, S.K.: SLAM and Static Driver Verifier: Technology Transfer of Formal Methods inside Microsoft. In: Boiten, E.A., Derrick, J., Smith, G.P. (eds.) IFM 2004. LNCS, vol. 2999, pp. 1–20. Springer, Heidelberg (2004)

[BLM+13] Bai, G., Lei, J., Meng, G., Venkatraman, S., Saxena, P., Sun, J., Liu, Y., Dong, J.S.: AUTHSCAN: Automatic Extraction of Web Authentication Protocols from Implementations. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2013, San Diego, USA (2013)

[BMD10] Biswas, J., Mokhtari, M., Dong, J.S., Yap, P.: Mild dementia care at home – integrating activity monitoring, user interface plasticity and scenario verification. In: Lee, Y., Bien, Z.Z., Mokhtari, M., Kim, J.T., Park, M., Kim, J., Lee, H., Khalil, I. (eds.) ICOST 2010. LNCS, vol. 6159, pp. 160–170. Springer, Heidelberg (2010)

[BCM+92] Burch, J.R., Clarke, E.M., McMillan, K.L., Dill, D.L., Hwang, L.J.: Symbolic model checking: 1020 states and beyond. Information and computation 98(2), 142–170 (1992)

[BSH13] Barnaghi, P., Sheth, A., Henson, C.: From Data to Actionable Knowledge: Big Data Challenges in the Web of Things. IEEE Intelligent Systems (November/December 2013)

[CZ13] Ciobanu, G., Zheng, M.: Automatic Analysis of TiMo Systems in PAT. In: Engineering of Complex Computer Systems, ICECCS (2013)

[CE81] Clarke, E.M., Emerson, E.A.: Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. Logic of Programs, 52–71 (1981)

[DeB] de Moura, L., Bjørner, N.S.: Z3: An efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008)

[DSL13] Dong, J.S., Sun, J., Liu, Y.: Build Your Own Model Checker in One Month. In: 35th International Conference on Software Engineering (ICSE 2013, tutorial), San Francisco, CA, USA, May18-26 (2013)

[GSL+13] Gui, L., Sun, J., Liu, Y., Si, Y., Dong, J.S., Wang, X.: Combining Model Checking and Testing with an Application to Reliability Prediction and Distribution. In: ISSTA 2013, Lugano, Switzerland, July 15-20 (2013)

[HSL+12] Hao, J., Song, S., Liu, Y., Sun, J., Gui, L., Dong, J.S., Leung, H.-F.: Probabilistic Model Checking Multi-agent Behaviors in Dispersion Games Using Counter Abstraction. In: Rahwan, I., Wobcke, W., Sen, S., Sugawara, T. (eds.) PRIMA 2012. LNCS, vol. 7455, pp. 16–30. Springer, Heidelberg (2012)

[Hoare85] Hoare, C.A.R.: Communicating Sequential Process. Prentice Hall (1985)

[HH98] Hoare, C.A.R., He, J.: Unifying Theory of Programming. Pretice Hall (1998)

[KGN+09] Kaivola, R., Ghughal, R., Narasimhan, N., Telfer, A., Whittemore, J., Pandav, S., Slobodová, A., Taylor, C., Frolov, V., Reeber, E., Naik, A.: Replacing Testing with Formal Verification in Intel® Core™ i7 Processor Execution Engine Validation. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 414–429. Springer, Heidelberg (2009)

[LAL+14] Lin, S.-W., André, É., Liu, Y., Sun, J., Dong, J.S.: Learning Assumptions for Compositional Verification of Timed Systems. IEEE Trans. Software Eng. 40(2), 137–153 (2014)

[LCL+13] Liu, Y., Chen, W., Liu, Y.A., Sun, J., Zhang, S., Dong, J.S.: Verifying Linearizability via Optimized Refinement Checking. IEEE Transactions on Software Engineering 39(7), 1018–1039 (2013)

[LLX02] Lee, V.Y., Liu, Y., Zhang, X., Phua, C., Sim, K., Zhu, J., Biswas, J., Dong, J.S., Mokhtari, M.: ACARP: Auto correct activity recognition rules using process analysis toolkit (PAT). In: Donnelly, M., Paggetti, C., Nugent, C., Mokhtari, M. (eds.) ICOST 2012. LNCS, vol. 7251, pp. 182–189. Springer, Heidelberg (2012)

[LDS+14] Li, Y., Dong, J.S., Sun, J., Liu, Y., Sun, J.: Model Checking Approach to Automated Planning. Formal Methods in System Design 44(2), 176–202 (2014)

[Len02] Lenzerini, M.: Data integration: A theoretical perspective. In: Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS). ACM (2002)

[LYW11] Li, Y., Yang, H., Wu, H.: 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.

[MD00] Mahony, B., Dong, J.S.: Timed Communicating Object Z. IEEE Transactions on Software Engineering 26(2), 150–177 (2000)

[MWC10] Miller, S.P., Whalen, M.W., Cofer, D.D.: Software model checking takes off. Commun. ACM 53(2), 58–64 (2010)

[SZL+13] Shi, L., Zhao, Y., Liu, Y., Sun, J., Dong, J.S., Qin, S.: A UTP Semantics for Communicating Processes with Shared Variables. In: Groves, L., Sun, J. (eds.) ICFEM 2013. LNCS, vol. 8144, pp. 215–230. Springer, Heidelberg (2013)

[SD06] Sun, J., Dong, J.S.: Design Synthesis from Interaction and State-based Specifications. IEEE Transactions on Software Engineering (TSE) 32(6), 349–364 (2006)

[SLDP09] Sun, J., Liu, Y., Dong, J.S., Pang, J.: PAT: Towards flexible verification under fairness. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 709–714. Springer, Heidelberg (2009)

[SSLD12] Song, S., Sun, J., Liu, Y., Dong, J.S.: A Model Checker for Hierarchical Probabilistic Real-Time Systems. In: Madhusudan, P., Seshia, S.A. (eds.) CAV 2012. LNCS, vol. 7358, pp. 705–711. Springer, Heidelberg (2012)

[SYD+13] Sun, J., Liu, Y., Dong, J.S., Liu, Y., Shi, L., André, É.: Modeling and Verifying Hierarchical Real-time Systems using Stateful Timed CSP. The ACM Transactions on Software Engineering and Methodology (TOSEM) 22(1), 3:1-3:29 (2013)

[TDC04] Taguchi, K., Dong, J.S., Ciobanu, G.: Relating Pi-calculus to Object-Z. In: The 9th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2004), Florence, Italy, pp. 97–106. IEEE Press (2004)

[TAS+13] Tan, T.H., Andre, E., Sun, J., Liu, Y., Dong, J.S., Chen, M.: Dynamic Synthesis of Local Time Requirement for Service Composition. In: ICSE 2013 (2013)

[WD96] Woodcock, J., Davies, J.: Using Z: specification, refinement, and proof. Prentice-Hall (1996)

[WSC+08] Woodcock, J., Stepney, S., Cooper, D., Clark, J.A., Jacob, J.: The certification of the Mondex electronic purse to ITSEC Level E6. Formal Asp. Comput. 20(1), 5–19 (2008)

[ZSS+13] Zheng, M., Sanán, D., Sun, J., Liu, Y., Dong, J.S., Gu, Y.: State Space Reduction for Sensor Networks using Two-level Partial Order Reduction. In: The 14th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2013), Rome, Italy, pp. 515–535 (January 2013)