南華大學
UNIVERSITY OF SOUTH CHINA

毕业设计(英文翻译)

题　　目　模型验证软件的归纳推理算法的设计与实现

学院名称　南华大学

指导教师　刘杰

职　　称　副教授

班　　级　20 软卓 01 班

学　　号　20200440717

学生姓名　李奕星

2024 年 3 月 24 日

Model checking: recent improvements and applications

Abstract: Model checking (Baier and Katoen in Principles of model checking, MIT Press, Cambridge, 2008; Clarke et al. in Model checking, MIT Press, Cambridge, 2001) is an automatic technique to formally verify that a given specification of a concurrent system meets given functional properties. Its use has been demonstrated many times over the years. Key characteristics that make the method so appealing are its level of automaticity, its ability to determine the absence of errors in the system (contrary to testing techniques) and the fact that it produces counter-examples when errors are detected, that clearly demonstrate not only that an error is present, but also how the error can be produced. The main drawback of model checking is its limited scalability, and for this reason, research on reducing the computational effort has received much attention over the last decades. Besides the verification of qualitative functional properties, the model checking technique can also be applied for other types of analyses, such as planning and the verification of quantitative properties. We briefly discuss several contributions in the model checking field that address both its scalability and its applicability to perform planning and quantitative analysis. In particular, we introduce six papers selected from the 23rd International SPIN Symposium on Model Checking Software (SPIN 2016).

Keywords Model checking; Planning; Strategy synthesis; Probabilistic model checking; Partial-order reduction;

1 Introduction

The current issue of the journal Software Tools for Technology Transfer (STTT) contains six revised and extended versions of papers presented at the 23rd International SPIN Symposium on Model Checking Software (SPIN 2016) [8]. SPIN 2016 was held in Eindhoven, The Netherlands, on 7Ð8 April 2016 collocated with the Joint European Conferences on Theory and Practice of Software (ETAPS).

These six papers were selected by the guest editors out of the sixteen papers presented at the event, based on their ranking given by the peer reviewers.

During the last two decades the SPIN symposiums have established themselves as traditional annual forums for researchers and practitioners for the verification of software systems. The evolution of the SPIN events has to a great extent mirrored the maturing of model checking into a prevailing technology for the formal verification of software systems. The first SPIN workshop was held in Montreal in 1995. The next couple of subsequent editions of SPIN were intended as gatherings for presenting extensions and applications of the model checker Spin [24], to which the series owes its name. Starting with the 2000 edition, the scope of the event clearly broadened to include techniques for formal verification and testing in general. In addition, the SPIN events aim to promote interaction and exchange of ideas across related software engineering areas, like static and dynamic analysis.

This special issue nicely demonstrates the current scope of the SPIN events. First of all, in addition to the Spin model checker, contributions in this issue use the tool TAPAAL [13], the Afra model checking tool [29], the ASSET tool [40], and the CADP toolbox [19].

Second of all, the majority of the papers in this issue are on extending and applying model checking beyond its traditional set-up, i.e. the formal verification of concurrent systems w.r.t. qualitative behavioural properties. Four of the six papers are on the application of model checking to construct a strategy or plan to solve a particular scheduling or control problem constrained by time and/or resource requirements. Another paper is on on-the-by verification of quantitative properties via probabilistic model checking [3]. In that sense, one of the papers is more traditional in its scope, but it addresses the main drawback of model checking, i.e. its limited scalability, by contributing to the topic of partial-order reduction [22,35,39], a very effective technique to mitigate state

space explosion.

The remainder of this preface is organised as follows: Section 2 discusses the use of model checking for the synthesis of strategies and plans. In Sect. 3, the verification of quantitative properties by means of probabilistic model checking is considered. Partial-order reduction to on-the-by reduce state spaces explored by model checkers is discussed in Sect. 4. Finally, in Sect. 5, some concluding remarks are given.

2 Planning and strategy synthesis

The application of model checking to construct a plan or synthesise a strategy is not far-fetched, as model checking and planning have much in common [1,11,37,43,44]: in both cases, a (large) state space has to be explored, looking for interesting behaviour. While in traditional model checking, this behaviour is essentially undesirable, violating some functional properties, in planning the interesting behaviour is desirable and constitutes a successful plan to optimise a system while fulfilling given constraints. When synthesising a strategy, typically the notion of a controller is added to the model, and the question is whether there exists a strategy for that controller such that any possible behaviour under that strategy satisfies the specification.

In the paper Integrating river basin DSSs with model checking by del Mar Gallardo et al. [18], which extends their SPIN 2016 paper [17], it is demonstrated how the Spin model checker can be applied in a decision support system (DSS) that mitigates the effects of floods in river basins. Model checking is used to synthesise management recommendations that meet the constraints given by the dam manager. A set of constraints is added to a PROMELA model that interacts with an external model for the river basin. SPIN exhaustively explores all possible manoeuvres and produces a trace, i.e. a sequence of manoeuvres, that fulfils the given constraints.

The paper A Case Study of Planning for Smart Factories – Model Checking

and Monte-Carlo Search for the Rescue by Edelkamp and Greulich [15], which extends their SPIN 2016 paper [16], proposes to use the Spin model checker to construct plans for multi-agent systems that control the industrial production of goods. Assembling stations use queues to buffer materials, and the core objective is to optimise the throughput of the system. The authors demonstrate that by using branch-and-bound searching, optimised plans consisting of thousands of steps can be produced in reasonable time. For comparison, they also consider using a Monte Carlo search framework and conclude that such an approach is even better in constructing plans. They conjecture that building a model checker that uses Monte Carlo search is an interesting topic to investigate in future work.

Of course, timing is crucial when synthesising strategies to control real-time systems, but its introduction makes the use of model checking more challenging. The previous contribution handles timing by carefully modelling it explicitly such that a model checker unaware of timing could still be used. An alternative is to use model checking techniques that natively support timing. Symbolic continuous-time on-the-by methods, such as those employed in the tools Kronos [9], UPPAAL [5], Tina [6] and Romeo [20], have been employed in on-the-by algorithms for controller synthesis [4,36]. How-ever, for such a task, discrete-time methods turn out to be very competitive [2].

The paper Discrete and Continuous Strategies for Timed-Arc Petri Net Games by Jensen et al. [25], which extends their SPIN 2016 paper [26], addresses this topic and proposes an on-the-by algorithm for the synthesis of timed controllers relative to safety objectives. It turns out that when restricting the context to the use of urgent controllers that act immediately or wait for another occurrence of the same event, then discrete-time methods can be used to determine the existence of a continuous-time safety controller.

Schedulability and resource utilisation of wireless sensor and actuator

network (WSAN) applications are addressed in the paper Modeling and Analyzing Real-Time Wireless Sensor and Actuator Networks Using Actors and Model Checking by Khamespanah et al. [27]. This paper extends their SPIN 2016 paper [28]. Such applications can be modelled by defining a number of concurrent actors, each providing services that can be requested by other actors by sending messages. Schedulability of the operations can be checked using Timed Rebeca, and Timed Computation Tree Logic (TCTL) model checking can be performed to check more complicated properties, such as minimal resource utilisation.

3 Probabilistic model checking

To check quantitative properties of systems, for example referring to time constraints or energy consumption, models can be extended with probabilities associated with behavioural events. The potential behaviour of such systems can then be captured in Markov Chains or probabilistic transition systems (PTSs) [21], which essentially are discrete-time Markov Chains in which transitions are labelled with actions and probabilities, and communication between concurrent processes is modelled. Probabilistic model checkers, such as Prism [30] and Storm [14], can be used to analyse these Markov Chains and deter-mine whether they satisfy given probabilistic properties.

To express these properties, suitable temporal logics need to be defined, such as probabilistic computation tree logic (PCTL) [23].

In the paper On-the-Fly Model Checking for Extended Action-Based Probabilistic Operators by Mateescu and Requeno [32], which extends their SPIN 2016 paper [33], a new regular probabilistic operator is proposed to specify the probability measure of a path described by a generalised regular formula involving computations on data values. This operator subsumes the until operators of PCTL and their action-based counterparts. The authors integrate this operator into MCL (Model Checking Language) and implement an

on-the-by model checking method in the CADP tool-box.

4 Partial-order reduction

The partial-order reduction (POR) technique [22,35,39] is perhaps the most efficient technique to mitigate the state space explosion problem in model checking. In recognition of this fact the founding fathers of POR, Godefroid, Peled, Valmari, and Wolper, received the 2014 CAV award. POR exploits the observation that the state space may contain several paths that are similar, in the sense that their differences are not relevant to the property under consideration. By pruning certain transitions, the size of the state space can be reduced.

The current issue features the paper Fair Testing and Stub-born Sets by Valmari and Vogler [41], which extends their SPIN 2016 paper [42]. Valmari was the first to notice the necessity for the so-called cycle proviso to ensure the correctness of POR when cycles are present in the state space. In the presence of cycles, POR without such a proviso may incorrectly terminate after having investigated a cycle, consistently ignoring behaviour that leaves the cycle. Hence, this problem is known as the ignoring problem. The cycle proviso turned out to be crucial for various adaptations of POR to different search orders of the state space (such as breadth-first search [7]), as well as parallel searches, both for shared memory (in settings using multiple cores [31] and graphics processing units [34]) and distributed architectures [10,38].

In the paper by Valmari and Vogler, it is proven that a partial-order method originally proposed for trace equivalence also preserves fair testing equivalence, in which deadlocks are unified with livelocks that cannot be exited. Thus, it supports a practical fairness assumption. Com-pared to the original SPIN 2016 paper, the extended version presents new observations regarding the ignoring problem in this context, remarking that the preservation of trace and fair testing equivalence does not imply that the ignoring problem is addressed.

5 Conclusions

Recent improvements and applications in the field of model checking have been discussed and associated with six papers selected from SPIN 2016, that have been included in this special issue. Four of the six papers contribute work on the application of model checking techniques to construct schedules and plans for planning problems, and synthesise strategies for control problems. In addition, one paper con-tributes to the verification of quantitative properties, and one contributes to the topic of partial-order reduction. Together, these papers address both the strengthening of the model checking method itself and its applicability to efficiently solve problems outside its traditional scope.

References

1. Abdeddaïm, Y., Maler, O.: Job-shop scheduling using timed automata. In: Proceedings of the 13th International Conference on Computer Aided Verification (CAV 2001), Lecture Notes in Computer Science, vol. 2102, pp. 478–492. Springer, Berlin (2001)

2. Andersen, M., Larsen, H., Srba, J., Sørensen, M., Taankvist, J.: Verification of liveness properties on closed timed-Arc Petri nets. In: Proceedings of the 8th International Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2012), Lecture Notes in Computer Science, vol. 7721, pp. 69–81. Springer, Berlin (2012)

3. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press, Cambridge (2008)

4. Behrmann, G., Cougnard, A., David, A., Fleury, E., Larsen, K., Lime, D.: UPPAAL-Tiga: time for playing games! In: Proceedings of the 19th International Conference on Computer Aided Verification (CAV 2007), Lecture Notes in Computer Science, vol. 4590, pp. 121–125. Springer, Berlin (2007)

5. Behrmann, G., David, A., Larsen, K., Hakansson, J., Petterson, P., Yi, W., Hendriks, M.: UPPAAL 4.0. In: Proceedings of the 3rd International Conference on Quantitative Evaluation of Systems (QEST 2006), pp. 125–126. IEEE Computer

Society, Washington, DC (2006)

6. Berthomieu, B., Vernadat, F.: Time Petri nets analysis with TINA. In: Proceedings of the 3rd International Conference on Quantitative Evaluation of Systems (QEST 2006), pp. 123–124. IEEE Computer Society, Washington, DC (2006)

7. Bošnaˇcki, D., Leue, S., Lluch-Lafuente, A.: Partial-order reduction for general state exploring algorithms. STTT 11(1), 39–51 (2009)

8. Bošnaˇcki, D., Wijs, A. (eds.): Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641. Springer, Berlin (2016)

9. Bozga, M., Daws, C., Maler, O., Olivero, A., Tripakis, S., Yovine, S.: Kronos: a model-checking tool for real-time systems. In: Proceedings of the 10th International Conference on Computer Aided Verification (CAV 1998), Lecture Notes in Computer Science, vol.

1427, pp. 546–550. Springer, Berlin (1998) 10. Brim, L., ˇ Cerná, I., Moravec, P., Šimša, J.: Distributed partial order reduction of state spaces. In: Proceedings of the 3rd International Workshop on Parallel and Distributed Methods in Verification (PDMC 2004), Electronic Notes in Theoretical Computer Science, vol. 128, pp. 63–74. Elsevier, New York (2004)

11. Brinksma, E.,Mader, A., Fehnker, A.: Verification and optimisation of a PLC control schedule. STTT 4(1), 21–33 (2002)

12. Clarke, E.M., Grumberg, O., Peled, D.: Model Checking. MIT Press, Cambridge (2001)

13. David, A., Jacobsen, L., Jacobsen, M., Jørgensen, K., Møller, M., Srba, J.: TAPAAL 2.0: integrated development environment for timed-Arc Petri nets. In: Proceedings of the 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2012), Lecture Notes in Computer Science, vol. 7214, pp. 492–497. Springer, Berlin (2012)

14. Dehnert, C., Junges, S., Katoen, J.P., Volk, M.: A storm is coming: a modern

probabilistic model checker. In: Proceedings of the 29th International Conference on Computer Aided Verification (CAV 2017), Lecture Notes in Computer Science, vol. 10427, pp. 592–600. Springer, Berlin (2017)

15. Edelkamp, S., Greulich, C.: A case study of planning for smart factories-model checking and Monte-Carlo search for the rescue. Int. J. Softw. Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-018-0498-1

16. Edelkamp, S., Greulich, C.: Using SPIN for the optimized scheduling of discrete event systems in manufacturing. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 57–77. Springer, Berlin (2018)

17. Gallardo, M., Merino, P., Panizo, L., Salmerón, A.: River basin management with SPIN. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 78–96. Springer, Berlin (2016)

18. Gallardo, M., Merino, P., Panizo, L., Salmerón, A.: Integrating river basin DSSs with model checking. Int. J. Softw. Tools Technol. Transf. (2017). https://doi.org/10.1007/s10009-017-0478-x

19. Garavel, H., Lang, F., Mateescu, R., Serwe, W.: CADP 2011: a toolbox for the construction and analysis of distributed processes. STTT 15(2), 89–107 (2013)

20. Gardey, G., Lime, D., Magnin, M., Roux, O.: Romeo: a tool for analyzing time Petri nets. In: Proceedings of the 17th International Conference on Computer Aided Verification (CAV 2005), Lecture Notes in Computer Science, vol. 3576, pp. 418–423. Springer, Berlin (2005)

21. van Glabbeek, R., Smolka, S., Steffen, B.: Reactive, generative and stratified models of probabilistic processes. Inf. Comput. 121(1), 59–80 (1995)

22. Godefroid, P., Wolper, P.: A partial approach to model checking. Inf. Comput. 110(2), 305–326 (1994)

23. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Form.

Asp. Comput. 6(5), 512–535 (1994)

24. Holzmann, G.: The SPIN Model Checking: Primer and Reference Manual. Addison-Wesley, Boston (2003)

25. Jensen, P.-G., Larsen, K.G., Srba, J.: Discrete and continuous strategies for timed-Arc Petri net games. Int. J. Softw. Tools Technol. Transf. (2017). https://doi.org/10.1007/s10009-017-0473-2

26. Jensen, P., Larsen, K., Srba, J.: Real-time strategy synthesis for timed-Arc Petri net games via discretization. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 129–146. Springer, Berlin (2018)

27. Khamespanah, E., Sirjani, M., Mechitov, K., Agha, G.: Modeling and analyzing real-time wireless sensor and actuator networks using actors and model checking. Int. J. Softw. Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-017-0480-3

28. Khamespanah, E., Sirjani, M., Mechitov, K., Agha, G.: Schedulability analysis of distributed real-time sensor network applications using actor-based model checking. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 165–181. Springer, Berlin (2018)

29. Khamespanah, E., Sirjani, M., Sabahi-Kaviani, Z., Khosravi, R., Izadi, M.J.: Timed rebeca schedulability and deadlock freedom analysis using bounded floating time transition system. Sci. Comput. Program. 98(P2), 184–204 (2015)

30. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Proceedings of the 23rd International Conference on Computer Aided Verification (CAV 2011), Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer, Berlin (2011)

31. Laarman, A., Wijs, A.: Partial-order reduction for multi-core LTL model checking. In: Proceedings of the 10th Haifa Verification Conference (HVC 2014),

Lecture Notes in Computer Science, vol.8855, pp. 267–283. Springer, Berlin (2014)

32. Mateescu, R., Requeno, J.I.: On-the-fly model checking for extended action-based probabilistic operators. Int. J. Softw. Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-018-0499-0

33. Mateescu, R., Requeno, J.: On-the-fly model checking for extended action-based probabilistic operators. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software,Lecture Notes in Computer Science, vol. 9641, pp. 189–207. Springer, Berlin (2018)

34. Neele, T.,Wijs, A., Bošnaˇcki, D., Pol, J.v.d.: Partial-order reduction for GPU model checking. In: Proceedings of the 14th International Symposium on Automated Technology for Verification and Analysis (ATVA 2016), Lecture Notes in Computer Science, vol. 9938,pp. 357–374. Springer, Berlin (2016)

35. Peled, D.: All from one, one for all: on model checking using representatives. In: CAV 1993, Proceedings, vol. 697, pp. 409–423(1993)

36. Pnueli, A., Asarin, E., Maler, O., Sifakis, J.: Controller synthesis for timed automata. In: Proceedings of the 5th IFAC Conference on System Structure and Control (SSC 1998), IFAC Proceedings Volumes, vol. 31, pp. 447–452. Elsevier, New York (1998)

37. Ruys, T.: Optimal scheduling using branch and bound with SPIN 4.0. In: Proceedings of the 10th International SPIN Workshop on Model Checking Software, Lecture Notes in Computer Science, vol. 2648, pp. 1–17. Springer, Berlin (2003)

38. Simsa, J., Bryant, R., Gibson, G., Hickey, J.: Scalable dynamic partial order reduction. In: Proceedings of the 3rd International Conference on Runtime Verification, Lecture Notes in Computer Science, vol. 7687, pp. 19–34. Springer, Berlin (2012)

39. Valmari, A.: Stubborn sets for reduced state space generation. Adv.Petri Nets

483, 491–515 (1991)

40. Valmari, A.: A state space tool for concurrent system models expressed in C++. In: Proceedings of the 14th Symposium on Programming Languages and Software Tools (SPLST 2015), CEUR Workshop Proceedings, vol. 1525, pp. 91–105. CEUR-WS.org(2015)

41. Valmari, A., Vogler, W.: Fair testing and stubborn sets. Int. J. Softw.Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-017-0481-2

42. Valmari, A., Vogler, W.: Fair testing and stubborn sets. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol.9641, pp. 225–243. Springer, Berlin (2018)

43. Wijs, A.: What to do next? analysing and optimising system behaviour in time. Ph.D. Thesis, Vrije Universiteit Amsterdam(2007)

44. Wijs, A., Fokkink, W.: From χt to μCRL: Combining Performance and Functional Analysis. In: Proceedings of the 10th

模型检查：最近的改进和应用

摘要： 模型检查（Baier and Katoen in Principles of model checking， MIT Press， Cambridge， 2008;Clarke et al. in Model checking， MIT Press， Cambridge， 2001）是一种自动技术，用于正式验证并发系统的给定规范是否满足给定的功能属性。多年来，它的使用已被多次证明。使该方法如此吸引人的关键特征是其自动性水平，确定系统中是否存在错误的能力（与测试技术相反），以及在检测到错误时生成反例的事实，这不仅清楚地表明存在错误，而且还清楚地表明了如何产生错误。模型检查的主要缺点是其有限的可扩展性，因此，在过去几十年中，关于减少计算工作量的研究受到了广泛关注。除了定性功能特性的验证外，模型检查技术还可以应用于其他类型的分析，例如定量特性的规划和验证。我们简要讨论了模型检查领域的一些贡献，这些贡献解决了其可扩展性及其在执行规划和定量分析方面的适用性。特别值得一提的是，我们介绍了从第 23 届国际 SPIN 模型检查软件研讨会（SPIN 2016）中选出的六篇论文。

关键词：模型检查;规划;策略综合;概率模型检查;部分阶减少;

1 引言

最新一期的《技术转让软件工具》（STTT）杂志收录了第 23 届国际 SPIN 模型检查软件研讨会（SPIN 2016）上发表的六篇论文的修订版和扩展版[8]。SPIN 2016 于 2016 年 4 月 7 日至 8 日在荷兰埃因霍温举行，与欧洲软件理论与实践联合会议 （ETAPS） 同期举行。这六篇论文是由客座编辑根据同行评审员给出的排名，从出席活动的 16 篇论文中选出的。

在过去的二十年中，SPIN 研讨会已成为研究人员和从业人员验证软件系统的传统年度论坛。SPIN 事件的演变在很大程度上反映了模型检查的成熟，成为软件系统形式化验证的流行技术。第一届 SPIN 研讨会于 1995 年在蒙特利尔举行。接下来的几个版本是 SPIN 的聚会，用于展示模型检查器 Spin 的扩展和应用[24]，该系列的名字来源于此。从 2000 年版开始，该活动的范围明显扩大到包括一般形式验证和测试技术。此外，SPIN 活动旨在促进相关软件工程领域（如静态和动态分析）之间的互动和思想交流。

本期特刊很好地展示了 SPIN 事件的当前范围。首先，除了 Spin 模型检查器之外，本期的贡献还使用了 TAPAAL [13]、Afra 模型检查工具 [29]、ASSET 工具 [40] 和 CADP 工具箱 [19]。

其次，本期的大多数论文都是关于扩展和应用模型检查，超越其传统设置，即对具有定性行为属性的并发系统进行形式验证。六篇论文中有四篇是关于应用模型检查来构建策略或计划，以解决受时间和/或资源要求限制的特定调度或控制问题。另一篇论文是关于通过概率模型检查对定量性质进行实时验证[3]。从这个意义上说，其中一篇论文在范围上更为传统，但它通过对部分阶约简这一主题[22,35,39]提出了模型检查的主要缺点，即其有限的可扩展性，这是一种非常有效的缓解状态空间爆炸的技术。

本序言的其余部分组织如下：第 2 节讨论了使用模型检查来综合战略和计划。在第 3 节中，考虑了通过概率模型检查来验证定量性质。第 4 节讨论了模型检查器探索的部分阶约简到逐次约简状态空间。最后，在第 5 节中，作了一些结论性发言。

2 综合规划与战略

应用模型检查来构建计划或综合策略并不牵强，因为模型检查和计划有很多共同点[1,11,37,43,44]：在这两种情况下，都必须探索（大）状态空间，寻找有意义的行为。虽然在传统的模型检查中，这种行为本质上是不可取的，违反了某些功能属性，但在规划中，有意义的行为是可取的，并且构成了一个成功的计划，以优化系统，同时满足给定的约束条件。在合成策略时，通常会将控制器的概念添加到模型中，问题是该控制器是否存在策略，以便该策略下的任何可能行为都满足规范。

在 del Mar Gallardo 等[18]的论文《将流域 DSS 与模型检查相结合》中，该论文扩展了他们的 SPIN 2016 论文[17]，展示了如何将 Spin 模型检查器应用于减轻流域洪水影响的决策支持系统（DSS）。 模型检查用于综合满足大坝管理者给出的约束条件的管理建议。将一组约束添加到与流域外部模型交互的 P ROMELA 模型中。SPIN 详尽地探索了所有可能的操作，并生成了一条跟踪，即一系列操作，以满足给定的约束条件。

Edelkamp 和 Greulich [15] 的论文 A Case Study of Planning for Smart Factories – Model Checking and Monte-Carlo Search for the Rescue 扩展了他们的 SPIN 2016 论文 [16]，建议使用 Spin 模型检查器来构建控制商品工业生产的多智能体系统的计划。装配站使用队列来缓冲材料，其核心目标是优化系统的吞吐量。作者证明，通过使用分支和绑定搜索，可以在合理的时间内生成由数千个步骤组成的优化计划。为了进行比较，他们还考虑使用蒙特卡罗搜索框架，并得出结论，这种方法 在构建计划时甚至更好。他们推测，构建一个使用蒙特卡洛搜索的模型检查器是未来工作中一个有趣的主题。

当然，在合成控制实时系统的策略时，时序至关重要，但它的引入使得模型检查的使用更具挑战性。前面的贡献通过仔细地显式建模来处理计时，以便仍然可以使用不知道计时的模型检查器。另一种方法是使用原生支持计时的模型检查技术。符号连续时间在旁方法，如 Kronos [9]、UPPAAL [5]、Tina[6]和 Romeo[20]工具中使用的方法，已被用于控制器合成的在旁算法中[4,36]。无论如何，对于这样的任务，离散时间方法被证明是非常有竞争力的[2]。

Jensen 等[25]的论文 Discrete and Continuous Strategies for Timed-Arc Petri Net Games 扩展了他们的 SPIN 2016 论文[26]，讨论了这一主题，并提出了一种用于合成与安全目标相关的定时控制器的 on-the-by 算法。事实证明，当将上下文限制为使用立即采取行动或等待同一事件再次发生的紧急控制器时，可以使用离散时间方法来确定连续时间安全控制器的存在。

Khamespanah 等[27]的论文《使用参与者和模型检查对实时无线传感器和执行器网络进行建模和分析》讨论了无线传感器和执行器网络（WSAN）应用的可调度性和资源利用率。本文扩展了他们的 SPIN 2016 论文[28]。可以通过定义许多并发参与者来对此类应用程序进行建模，每个参与者都提供服务，其他参与者可以通过发送消息来请求这些服务。可以使用定时 Rebeca 检查操作的可调度性，并且可以执行定时计算树逻辑 （TCTL） 模型检查来检查更复杂的属性，例如最小资源利用率。

3 概率模型检查

为了检查系统的定量属性，例如参考时间限制或能源消耗，可以使用与行为

事件相关的概率来扩展模型。然后，可以在马尔可夫链或概率转移系统（PTS）[21]中捕获这些系统的潜在行为，这些系统本质上是离散时间马尔可夫链，其中转换用动作和概率标记，并对并发过程之间的通信进行建模。概率模型检查器，如 Prism [30] 和 Storm [14]，可用于分析这些马尔可夫链，并阻止挖掘它们是否满足给定的概率属性。

为了表达这些属性，需要定义合适的时间逻辑，例如概率计算树逻辑（PCTL）[23]。

在Mateescu和Requeno[32]的论文On-the-Fly Model Checking for Extended Action-Based Probabilistic Operator 中，扩展了他们的 SPIN 2016 论文[33]，提出了一种新的正则概率算子，以指定由涉及数据值计算的广义正则公式描述的路径的概率度量。该运算符包含 PCTL 的 until 运算符及其基于操作的对应物。作者将该操作器集成到 MCL（模型检查语言）中，并在 CADP 工具箱中实现一种即时模型检查方法。

4 部分阶减少

偏阶约简（POR）技术[22,35,39]可能是缓解模型检查中状态空间爆炸问题的最有效技术。为了表彰这一事实，POR 的创始人 Godefroid、Peled、Valmari 和 Wolper 获得了 2014 年 CAV 奖。POR 利用了状态空间可能包含多个相似路径的观察结果，从某种意义上说，它们的差异与所考虑的属性无关。通过修剪某些转换，可以减小状态空间的大小。

本期的特色是Valmari和Vogler的论文Fair Testing and Stub-born Sets [41]，该论文扩展了他们 SPIN 2016 的论文[42]。Valmari 是第一个注意到所谓的循环但书的必要性，以确保当状态空间中存在循环时 POR 的正确性。在存在周期的情况下，没有这种限制条件的 POR 可能会在调查周期后错误地终止，始终忽略离开周期的行为。因此，这个问题被称为忽略问题。事实证明，循环但书对于POR对状态空间的不同搜索顺序的各种调整（例如广度优先 搜索[7]）以及并行搜索至关重要，无论是共享内存（在使用多内核[31]和图形处理单元[34]的设置中）还是分布式架构[10,38]。

在 Valmari 和 Vogler 的论文中，证明了最初为跟踪等价提出的部分顺序方

法也保留了公平的测试等价性，其中死锁与 无法退出的活锁是统一的。因此，它支持实际的公平假设。与 SPIN 2016 的原始论文相比，扩展版本提出了关于在这种情况下忽略问题的新观察结果，并指出保留跟踪和公平测试等效性并不意味着忽略了问题得到解决。

5 结论

模型检查领域的最新改进和应用已经讨论过，并与 SPIN 2016 中选出的六篇论文相关联，这些论文已收录在本期特刊中。六篇论文中有四篇致力于应用模型检查技术来构建计划问题的时间表和计划，并综合控制问题的策略。此外，一篇论文对定量性质的验证做出了贡献，一篇论文对部分阶约简这一主题做出了贡献。总之，这些论文既讨论了模型检查方法本身的加强，也讨论了它有效解决传统范围之外的问题的适用性。

参考文献

1. Abdeddaïm, Y., Maler, O.: Job-shop scheduling using timed automata. In: Proceedings of the 13th International Conference on Computer Aided Verification (CAV 2001), Lecture Notes in Computer Science, vol. 2102, pp. 478–492. Springer, Berlin (2001)

2. Andersen, M., Larsen, H., Srba, J., Sørensen, M., Taankvist, J.: Verification of liveness properties on closed timed-Arc Petri nets. In: Proceedings of the 8th International Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2012), Lecture Notes in Computer Science, vol. 7721, pp. 69–81. Springer, Berlin (2012)

3. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press, Cambridge (2008)

4. Behrmann, G., Cougnard, A., David, A., Fleury, E., Larsen, K., Lime, D.: UPPAAL-Tiga: time for playing games! In: Proceedings of the 19th International Conference on Computer Aided Verification (CAV 2007), Lecture Notes in Computer Science, vol. 4590, pp. 121–125. Springer, Berlin (2007)

5. Behrmann, G., David, A., Larsen, K., Hakansson, J., Petterson, P., Yi, W., Hendriks,

M.: UPPAAL 4.0. In: Proceedings of the 3rd International Conference on Quantitative Evaluation of Systems (QEST 2006), pp. 125–126. IEEE Computer Society, Washington, DC (2006)

6. Berthomieu, B., Vernadat, F.: Time Petri nets analysis with TINA. In: Proceedings of the 3rd International Conference on Quantitative Evaluation of Systems (QEST 2006), pp. 123–124. IEEE Computer Society, Washington, DC (2006)

7. Bošnaˇcki, D., Leue, S., Lluch-Lafuente, A.: Partial-order reduction for general state exploring algorithms. STTT 11(1), 39–51 (2009)

8. Bošnaˇcki, D., Wijs, A. (eds.): Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641. Springer, Berlin (2016)

9. Bozga, M., Daws, C., Maler, O., Olivero, A., Tripakis, S., Yovine, S.: Kronos: a model-checking tool for real-time systems. In: Proceedings of the 10th International Conference on Computer Aided Verification (CAV 1998), Lecture Notes in Computer Science, vol.

1427, pp. 546–550. Springer, Berlin (1998) 10. Brim, L., ˇ Cerná, I., Moravec, P., Šimša, J.: Distributed partial order reduction of state spaces. In: Proceedings of the 3rd International Workshop on Parallel and Distributed Methods in Verification (PDMC 2004), Electronic Notes in Theoretical Computer Science, vol. 128, pp. 63–74. Elsevier, New York (2004)

11. Brinksma, E.,Mader, A., Fehnker, A.: Verification and optimisation of a PLC control schedule. STTT 4(1), 21–33 (2002)

12. Clarke, E.M., Grumberg, O., Peled, D.: Model Checking. MIT Press, Cambridge (2001)

13. David, A., Jacobsen, L., Jacobsen, M., Jørgensen, K., Møller, M., Srba, J.: TAPAAL 2.0: integrated development environment for timed-Arc Petri nets. In: Proceedings of the 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2012), Lecture Notes in

Computer Science, vol. 7214, pp. 492–497. Springer, Berlin (2012)
14. Dehnert, C., Junges, S., Katoen, J.P., Volk, M.: A storm is coming: a modern probabilistic model checker. In: Proceedings of the 29th International Conference on Computer Aided Verification (CAV 2017), Lecture Notes in Computer Science, vol. 10427, pp. 592–600. Springer, Berlin (2017)
15. Edelkamp, S., Greulich, C.: A case study of planning for smart factories-model checking and Monte-Carlo search for the rescue. Int. J. Softw. Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-018-0498-1
16. Edelkamp, S., Greulich, C.: Using SPIN for the optimized scheduling of discrete event systems in manufacturing. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 57–77. Springer, Berlin (2018)
17. Gallardo, M., Merino, P., Panizo, L., Salmerón, A.: River basin management with SPIN. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 78–96. Springer, Berlin (2016)
18. Gallardo, M., Merino, P., Panizo, L., Salmerón, A.: Integrating river basin DSSs with model checking. Int. J. Softw. Tools Technol. Transf. (2017). https://doi.org/10.1007/s10009-017-0478-x
19. Garavel, H., Lang, F., Mateescu, R., Serwe, W.: CADP 2011: a toolbox for the construction and analysis of distributed processes. STTT 15(2), 89–107 (2013)
20. Gardey, G., Lime, D., Magnin, M., Roux, O.: Romeo: a tool for analyzing time Petri nets. In: Proceedings of the 17th International Conference on Computer Aided Verification (CAV 2005), Lecture Notes in Computer Science, vol. 3576, pp. 418–423. Springer, Berlin (2005)
21. van Glabbeek, R., Smolka, S., Steffen, B.: Reactive, generative and stratified models of probabilistic processes. Inf. Comput. 121(1), 59–80 (1995)
22. Godefroid, P., Wolper, P.: A partial approach to model checking. Inf. Comput.

110(2), 305–326 (1994)

23. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Form. Asp. Comput. 6(5), 512–535 (1994)

24. Holzmann, G.: The SPIN Model Checking: Primer and Reference Manual. Addison-Wesley, Boston (2003)

25. Jensen, P.-G., Larsen, K.G., Srba, J.: Discrete and continuous strategies for timed-Arc Petri net games. Int. J. Softw. Tools Technol. Transf. (2017). https://doi.org/10.1007/s10009-017-0473-2

26. Jensen, P., Larsen, K., Srba, J.: Real-time strategy synthesis for timed-Arc Petri net games via discretization. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 129–146. Springer, Berlin (2018)

27. Khamespanah, E., Sirjani, M., Mechitov, K., Agha, G.: Modeling and analyzing real-time wireless sensor and actuator networks using actors and model checking. Int. J. Softw. Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-017-0480-3

28. Khamespanah, E., Sirjani, M., Mechitov, K., Agha, G.: Schedulability analysis of distributed real-time sensor network applications using actor-based model checking. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol. 9641, pp. 165–181. Springer, Berlin (2018)

29. Khamespanah, E., Sirjani, M., Sabahi-Kaviani, Z., Khosravi, R., Izadi, M.J.: Timed rebeca schedulability and deadlock freedom analysis using bounded floating time transition system. Sci. Comput. Program. 98(P2), 184–204 (2015)

30. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Proceedings of the 23rd International Conference on Computer Aided Verification (CAV 2011), Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer, Berlin (2011)

31. Laarman, A., Wijs, A.: Partial-order reduction for multi-core LTL model checking. In: Proceedings of the 10th Haifa Verification Conference (HVC 2014), Lecture Notes in Computer Science, vol.8855, pp. 267 – 283. Springer, Berlin (2014)

32. Mateescu, R., Requeno, J.I.: On-the-fly model checking for extended action-based probabilistic operators. Int. J. Softw. Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-018-0499-0

33. Mateescu, R., Requeno, J.: On-the-fly model checking for extended action-based probabilistic operators. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software,Lecture Notes in Computer Science, vol. 9641, pp. 189–207. Springer, Berlin (2018)

34. Neele, T.,Wijs, A., Bošnaˇcki, D., Pol, J.v.d.: Partial-order reduction for GPU model checking. In: Proceedings of the 14th International Symposium on Automated Technology for Verification and Analysis (ATVA 2016), Lecture Notes in Computer Science, vol. 9938,pp. 357–374. Springer, Berlin (2016)

35. Peled, D.: All from one, one for all: on model checking using representatives. In: CAV 1993, Proceedings, vol. 697, pp. 409–423(1993)

36. Pnueli, A., Asarin, E., Maler, O., Sifakis, J.: Controller synthesis for timed automata. In: Proceedings of the 5th IFAC Conference on System Structure and Control (SSC 1998), IFAC Proceedings Volumes, vol. 31, pp. 447 – 452. Elsevier, New York (1998)

37. Ruys, T.: Optimal scheduling using branch and bound with SPIN 4.0. In: Proceedings of the 10th International SPIN Workshop on Model Checking Software, Lecture Notes in Computer Science, vol. 2648, pp. 1 – 17. Springer, Berlin (2003)

38. Simsa, J., Bryant, R., Gibson, G., Hickey, J.: Scalable dynamic partial order reduction. In: Proceedings of the 3rd International Conference on Runtime Verification, Lecture Notes in Computer Science, vol. 7687, pp. 19–34. Springer,

Berlin (2012)

39. Valmari, A.: Stubborn sets for reduced state space generation. Adv.Petri Nets 483, 491–515 (1991)

40. Valmari, A.: A state space tool for concurrent system models expressed in C++. In: Proceedings of the 14th Symposium on Programming Languages and Software Tools (SPLST 2015), CEUR Workshop Proceedings, vol. 1525, pp. 91–105. CEUR-WS.org(2015)

41. Valmari, A., Vogler, W.: Fair testing and stubborn sets. Int. J. Softw.Tools Technol. Transf. (2018). https://doi.org/10.1007/s10009-017-0481-2

42. Valmari, A., Vogler, W.: Fair testing and stubborn sets. In: Proceedings of the 23rd International SPIN Symposium on Model Checking of Software, Lecture Notes in Computer Science, vol.9641, pp. 225–243. Springer, Berlin (2018)

43. Wijs, A.: What to do next? analysing and optimising system behaviour in time. Ph.D. Thesis, Vrije Universiteit Amsterdam(2007)

44. Wijs, A., Fokkink, W.: From χ t to μ CRL: Combining Performance and Functional Analysis. In: Proceedings of the 10th