X-by-Construction Meets AI

Maurice H. ter Beek[1(✉)], Loek Cleophas[2,3], Clemens Dubslaff[2,4], and Ina Schaefer[5]

[1] CNR–ISTI, Pisa, Italy
m.terbeek@isti.cnr.it
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
{l.g.w.a.cleophas,c.dubslaff}@tue.nl
[3] Stellenbosch University, Stellenbosch, Republic of South Africa
[4] Centre for Tactile Internet with Human-in-the-loop (CeTI), Dresden, Germany
[5] KIT, Karlsruhe, Germany
ina.schaefer@kit.edu

Abstract. During the past decade, researchers have investigated *X-by-Construction* (XbC), encompassing extensions beyond correctness concerns as in the more traditional Correctness-by-Construction (CbC) paradigm. Like CbC, XbC is a refinement approach to engineer systems that by-construction satisfy certain properties (e.g., *non-functional* ones in the case of XbC)—also, and in particular, in the setting of probabilistic systems and properties, and both at design time and at runtime. In line with the need to integrate concepts from artificial intelligence (AI), this track brings together researchers and practitioners to share their views on the many possible synergies between CbC/XbC and AI.

1 Motivation

Correctness-by-Construction (CbC) sees the development of software (systems) as a step-wise refinement process from specification to code, ideally by CbC design tools that automatically generate error-free software (system) implementations from rigorous and unambiguous requirement specifications. Afterwards, testing only serves to validate the CbC process rather than to find bugs.

A lot of progress has been made on CbC, and after a successful track on the combination of CbC with post-hoc verification at ISoLA 2016 [7], at ISoLA 2018 it was time to look further than correctness by investigating a move from CbC to *X-by-Construction* (XbC), i.e., by also considering non-functional properties [6]. XbC is thus concerned with a step-wise refinement process from specification to code that automatically generates software (system) implementations that by construction satisfy specific non-functional properties (i.e., concerning security, dependability, reliability, resource or energy consumption, and the like). In line with the growing attention to fault tolerance and the increasing use of *machine learning* (ML) techniques in modern software systems, which make it hard to establish guaranteed properties [21]—as witnessed in other tracks at ISoLA 2022 [13,18] and at AISoLA 2023 [19]—a third track in this series, at

T. Margaria and B. Steffen (Eds.): ISoLA 2024, LNCS 15222, pp. 155–161, 2025.
https://doi.org/10.1007/978-3-031-75387-9_10

ISoLA 2020/2021, focused on XbC in the setting of probabilistic systems and properties [4]. Finally, a fourth track in this series at ISoLA 2022 [5] focused on the synergies between XbC and *runtime verification* (RV). The current proliferation of systems with data-driven *artificial intelligence* (AI) components makes it difficult—if not impossible—to ensure system correctness already at design time. Therefore, RV is concerned with monitoring and analysing actual software (and hardware) system behaviour at runtime to detect and possibly repair system failures. However, also insurances are needed that the corrected system is indeed better than the previous one, where XbC into play.

2 Aim

Building on the highly successful ISoLA tracks mentioned above, the aim of this track is to bring together researchers and practitioners who are interested in CbC/XbC, and who acknowledge the need to join forces with concepts from AI and explainability. We believe this is important since on the one hand AI-based components employed in a software system give rise to new challenges for correctness arguments, both for functional and for non-functional properties due to their inherent probabilistic and non-deterministic nature; and on the other hand AI-supported software engineering and AI-enhanced verification approaches offer great potential for more efficient development of CbC software.

Given this specific topic, this ISoLA 2024 track fits perfectly as fifth track in the aforementioned highly successful series of ISoLA tracks.

ISoLA 2016 Correctness-by-Construction and Post-hoc Verification: Friends or Foes?
ISoLA 2018 X-by-Construction
ISoLA 2020/2021 X-by-Construction: Correctness Meets Probability
ISoLA 2022 X-by-Construction Meets Runtime Verification
ISoLA 2024 X-by-Construction: Correctness Meets AI

We have therefore invited both researchers and practitioners working in the following communities to participate in this track and share their views on the many possible synergies between CbC/XbC and AI:

– People working on systems involving components that employ ML or other AI approaches. In these settings, models and behaviour are typically dependent on what is learned from large data sets, and may change dynamically based on yet more data being processed. As a result, guaranteeing properties (whether correctness or non-functional ones, such as properties concerning security, reliability, resilience, energy consumption, performance, sustainability, and the like) becomes difficult. Probabilistic reasoning can mitigate this challenge by providing guarantees with high probabilities instead w.r.t. such properties for the components employing AI approaches. This may also include specific classes of properties, such as trustworthiness or explainability [2,16], that are only relevant for systems which incorporate AI-based components. As a consequence, people working in this domain may be interested in applying CbC/XbC techniques to provide guarantees for such AI-based systems.

- People working on quantitative modelling and analysis (e.g., through probabilistic/real-time systems and probabilistic/statistical model checking [11]), in particular in the specific setting of systems comprising classical and AI-based components. These people typically focus not only on correctness, but also on non-functional properties concerning safety, security, performance, dependability, and the like, which becomes even more challenging when a system includes AI-based components.
- People working on AI-supported software engineering processes or on AI-enhanced verification approaches, be it for generating specifications from existing program code, for obtaining invariants for a specific loop in a program or for improving state-space exploration, abstraction or proof search. These people may be interested in integrating those AI-enhanced methods and techniques into CbC/XbC engineering in order to obtain stronger and more efficient approaches. One promising line of research may be the AI-supported selection of CbC/XbC-refinement rules to make program construction more automated [1].

3 Contributions

In his keynote contribution, Platzer [17] calls for the study of the new field of *Intersymbolic AI*, intended as the combination of *symbolic AI*, whose building blocks have inherent significance/meaning, with *subsymbolic AI*, whose entirety creates significance/effect despite the fact that individual building blocks escape meaning. The idea is that Intersymbolic AI combines both symbolic and subsymbolic AI to increase the effectiveness of AI compared to either kind of AI alone, in much the same way that the combination of both conscious and subconscious thought increases the effectiveness of human thought compared to either kind of thought alone. Finally, Platzer surveys some successful contributions to the Intersymbolic AI paradigm in this paper, but many more are considered possible by advancing Intersymbolic AI.

In [15], Marques-Silva provides a technical survey of *logic-based Explainable AI* (XAI), its origins, the current topics of research, and emerging future topics of research. XAI is concerned with providing human decision-makers with understandable explanations for the predictions made by machine-learning models, which is a cornerstone of *trustworthy AI*. Despite its strategic importance, given that the operation of the most advanced AI models is often beyond the grasp of human decision makers, most work on XAI lacks rigour, and so its use in high-risk or safety-critical domains serves to foster distrust instead of contributing to build the much needed trust. To this aim, logic-based XAI has recently emerged as a rigorous alternative to those non-rigorous methods of XAI. Marques-Silva also highlights the many myths that pervade non-rigorous approaches for XAI.

In [8], Belmonte et al. present the design of a meta-programming system for *Hybrid AI*, which integrates spatial model checking and machine learning. The proposed system architecture blends different programming languages and execution technologies together using a simplified, declarative meta-language.

The result is a follow up to the spatial model checker VoxLogicA for declarative medical image analysis, aimed at *Explainable-by-Construction AI*.

In [12], Kodetzki et al. discuss the extent to which AI-tools can support CbC engineering. This is a formal methods approach to incrementally developing functionally correct programs on the basis of a formal specification. Using sound refinement rules, the correctness of the constructed program can be guaranteed during the development process. The authors analyse the CbC process with respect to potential AI-tool support in the tool CorC, which implements CbC. They classify their findings in five areas of interest and discuss for each of the areas whether and to what extent AI-tools can support *CbC software development*.

In [3], Beckert et al. discuss and categorise different use cases and application scenarios on combining *specification synthesis via Large Language Models (LLMs) and deductive program verification*. They perform preliminary quantitative experiments on the capabilities of LLMs to generate correct specifications by evaluating a prototypical integration of GPT with the deductive program verifier KeY and the bounded model checker JJBMC on a set of Java programs that are partially annotated with specifications written in the Java Modeling Language (JML). They conclude with a vision of how LLMs may support rigorous formal verification of software systems in the future and describe the necessary next steps in this direction.

In [20], Wenzel et al. present a significant step towards achieving *trustworthy AI* decisions by introducing a novel framework for enhancing *traceability and accountability by construction*. Their approach encompasses the entire decision-making pathway—from the raw datasets used to train the AI system, through the algorithms and programs employed, to the involved parties and the final decisions made. A so-called *Decision Bill of Materials* (DBOM) is at the core of their methodology. It documents in detail all elements contributing to a decision, while ensuring accountability and traceability through cryptographic signatures. By leveraging results from logic programming, they are able to verify that the system meets specific certification standards and that individual decisions can be qualified as trustworthy. As such, their framework not only advances the construction of reliable AI systems but also aligns technological developments with ethical imperatives and regulatory expectations.

In [14], Maderbacher et al. show how to achieve the advantages of Generalised Reactivity(1) (GR(1))—a type of propositional *reactive synthesis* used to automatically generate circuits or programs from temporal logic specifications, which offers a good compromise between expressiveness and performance—in infinite-state reactive synthesis. They show how to use an SMT solver to solve the synthesis problem and present techniques to efficiently compute the enforceable predecessor and optimise the fixpoint computation. They also show how to generate efficient programs as the result of the synthesis procedure, using techniques that are different from the circuit generation methods used in the propositional domain. The method is implemented as a prototype and its efficiency is shown on several benchmarks, both existing and new.

In [9], Bloem et al. study *threat model repair*, a method to automatically suggest structural changes to the design that mitigate threats discovered by the analysis. This helps find a secure design early in the process by allowing a user to quickly iterate over different design variants. This is a *Security-by-Construction* approach to system development, according to which security considerations are integrated into the design process from the very beginning. Threat modelling in particular helps to identify potential threats and vulnerabilities early in the system development process, assess the risk associated with each threat, and design appropriate mitigation actions.

In [10], Bozzano et al. report on ongoing research, funded by the Italian Space Agency (ASI) under the "Innovative Space Technologies" initiative, that addresses the formal design, development and validation of FDIR (Fault Detection, Identification and Recovery) integrating rule-based components with components based on *Machine Learning* (ML) and *Deep Learning* (DL). The development of accurate, reliable and effective FDIR components is essential in several application domains, to meet the dependability constraints and to accomplish the higher degree of autonomy required in future missions. The authors show that the integration of symbolic and AI techniques can substantially improve the effectiveness and efficiency of FDIR management functions, while formal tool-supported verification and validation can provide a formal guarantee of the quality of the FDIR systems before they are implemented and deployed.

References

1. Ahrendt, W., Gurov, D., Johansson, M., Rümmer, P.: TriCo—triple co-piloting of implementation, specification and tests. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 11th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Verification Principles (ISoLA'22). LNCS, vol. 13701, pp. 174–187. Springer (2022). https://doi.org/10.1007/978-3-031-19849-6_11
2. Baier, C., et al.: From verification to causality-based explications. In: Bansal, N., Merelli, E., Worrell, J. (eds.) Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP'21). LIPIcs, vol. 198, pp. 1:1–1:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPICS.ICALP.2021.1
3. Beckert, B., Klamroth, J., Pfeifer, W., Röper, P., Teuber, S.: Towards combining the cognitive abilities of large language models with the rigor of deductive progam verification. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
4. ter Beek, M.H., Cleophas, L., Legay, A., Schaefer, I., Watson, B.W.: X-by-construction: correctness meets probability. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Verification Principles (ISoLA'20). LNCS, vol. 12476, pp. 211–215. Springer (2020). https://doi.org/10.1007/978-3-030-61362-4_11

5. ter Beek, M.H., Cleophas, L., Leucker, M., Schaefer, I.: X-by-construction meets runtime verification. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 11th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Verification Principles (ISoLA'22). LNCS, vol. 13701, pp. 141–148. Springer (2022). https://doi.org/10.1007/978-3-031-19849-6_9
6. ter Beek, M.H., Cleophas, L., Schaefer, I., Watson, B.W.: X-by-construction. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Modeling (ISoLA'18). LNCS, vol. 11244, pp. 359–364. Springer (2018). https://doi.org/10.1007/978-3-030-03418-4_21
7. ter Beek, M.H., Hähnle, R., Schaefer, I.: Correctness-by-construction and post-hoc verification: friends or foes? In: Margaria, T., Steffen, B. (eds.) Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques (ISoLA'16). LNCS, vol. 9952, pp. 723–729. Springer (2016). https://doi.org/10.1007/978-3-319-47166-2_51
8. Belmonte, G., Bussi, L., Ciancia, V., Latella, D., Massink, M.: Towards hybrid-AI in imaging using VoxLogicA. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
9. Bloem, R., Chlup, S., Ničković, D., Schmittner, C.: On threat model repair. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
10. Bozzano, M., Cimatti, A., Cristoforetti, M., Griggio, A., Svaizer, P., Tonetta, S.: Towards formal design of FDIR components with AI. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
11. Katoen, J.: The probabilistic model checking landscape. In: Grohe, M., Koskinen, E., Shankar, N. (eds.) Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'16), pp. 31–45. ACM (2016). https://doi.org/10.1145/2933575.2934574
12. Kodetzki, M., Bordis, T., Kirsten, M., Schaefer, I.: Towards AI-assisted correctness-by-construction software development. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
13. Larsen, K.G., Legay, A., Nolte, G., Schlüter, M., Stoelinga, M., Steffen, B.: Formal methods meet machine learning (F3ML). In: Margaria, T., Steffen, B. (eds.) Proceedings of the 11th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation: Adaptation and Learning (ISoLA'22). LNCS, vol. 13703, pp. 393–405. Springer (2022). https://doi.org/10.1007/978-3-031-19759-8_24
14. Maderbacher, B., Windisch, F., Bloem, R.: Synthesis from infinite-state generalized reactivity(1) specifications. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS vol. 15222,, Springer (2024)
15. Marques-Silva, J.: Logic-based explainability: past, present and future. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium

on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
16. Marques-Silva, J., Ignatiev, A.: Delivering trustworthy AI through formal XAI. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI'22), pp. 12342–12350. AAAI Press (2022). https://doi.org/10.1609/AAAI.V36I11.21499
17. Platzer, A.: Intersymbolic AI: Interlinking symbolic AI and subsymbolic AI. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
18. Seisenberger, M., ter Beek, M.H., Fan, X., Ferrari, A., Haxthausen, A., James, P., Lawrence, A., Luttik, B., van de Pol, J., Wimmer, S.: Safe and Secure Future AI-Driven Railway Technologies: Challenges for Formal Methods in Railway. In: Margaria, T., Steffen, B. (eds.) 11th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation: Practice (ISoLA'22). LNCS, vol. 13704, pp. 246–268. Springer (2022). https://doi.org/10.1007/978-3-031-19762-8_20
19. Steffen, B. (ed.): Proceedings of the 1st international conference on bridging the gap between AI and reality (AISoLA'23), LNCS, vol. 14380. Springer (2023). https://doi.org/10.1007/978-3-031-46002-9
20. Wenzel, J., Köhl, M.A., Sterz, S., Schmidt, A., Fetzer, C., Hermanns, H.: Traceability and accountability by construction. In: Margaria, T., Steffen, B. (eds.) Proceedings of the 12th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'24). LNCS, vol. 15222, Springer (2024)
21. Wing, J.M.: Trustworthy AI. Commun. ACM **64**(10), 64–71 (2021). https://doi.org/10.1145/3448248