An Architecture for Self-Adaptive Abstraction and Approximation in Digital Twins with Real-time Requirements

RAHELEH BIGLARI and JOACHIM DENIL, University of Antwerp, Belgium

Review Article

Problem statement

A Digital twin [1, 2, 6] is a virtual representation of a physical or digital object, process, or system, serving as a bridge between the physical and digital worlds [5]. DTs revolutionise our interactions and management of the system under study, the actual systems, fostering innovation and efficiency. The design and testing of DT systems need simulation models, which provide engineers with the means to analyse system behavior and ensure compliance with performance requirements. However, the complexity of Cyber-Physical Systems (CPS) [3] and DTs translates into high-fidelity simulation models that are computationally expensive, posing significant challenges when real-time constraints necessitate in time decision-making with strict requirements for timeliness and accuracy.

Methods

To address these challenges, this paper introduces an architecture based on MAPE-K from IBM [4], shown in Figure 1. This architecture helps the managed system to do the adaptation and switch to the most appropriate model at run time while reducing complexity.

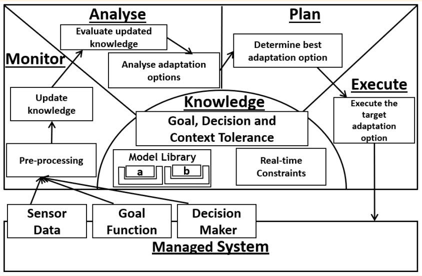

Fig. 1. Self-adaptive abstraction and approximation architecture.

This architecture consists of the following phases:

- **Monitor-phase** gathers all existing sensor data and adds the goal function and the relevant data. Moreover, this phase updated the knowledge. The update is based on the goal, context, and decision model, the quantified tolerances, and the available models with uncertainty quantification.
- **Analyse-phase** begins by evaluating the updated knowledge. Based on the updated knowledge, it creates adaptation options to see which model it can choose.
- **Plan-phase** determinses best adaptation option. In this phase, the most appropriate model is chosen.
- **Execute-phase** where the managed system's scheduler is updated.

Keywords: Digital Twins, Real-Time Constraints, Simulation Models, Adaptive Abstraction, Approximation, Validity Region Map, Decision-Centric Technique

Authors' Contact Information: Raheleh Biglari, raheleh.biglari@uantwerp.be; Joachim Denil, joachim.denil@uantwerp.be, University of Antwerp, Antwerp, Antwerp, Belgium.

https://doi.org/XXXXXXX.XXXXXXX

Results

Our case study is an automated Highway Lane Change Maneuver (LCM) system implemented using MATLAB/SIMULINK that allows a vehicle to move automatically from one lane to another. The actual system is the lane following controller, while the digital twin functions as a lane change planner that monitors the system and initiates lane changes when necessary. In this experiment, we use three different models, listed from the most detailed model to the most possible approximated model: First, the high-validity Model, which uses a controller in each vehicle, is computationally the most expensive model in this experiment. High-validity model is a model that is widely valid in their application domain. Second, the Constant Acceleration model (C.A model) is the kinematic equation $x(t) = 1/2 * a * t^2 + v * t + x_0$. In this formula, a is the vehicle's acceleration in m/s^2, v, the velocity of the vehicle in m/s, and x, the vehicle's position in meters. Third, the Constant Velocity model (C.V model) with the following kinematic equation: $x(t) = v * t + x_0$.

As a result, each car can select its own appropriate model based on its current position, velocity, and acceleration; otherwise, it needs to use the high-validity model. In this experiment, creating knowledge starts at design time by building knowledge.

Conclusion

In this paper, we presented an architecture of self-adaptive abstraction and approximation for DTs with real-time requirements. In the future, we aim to implement a framework with a supporting architecture, methods, and techniques to reason over the use of self-adaptive approximations and abstractions at runtime.

References

[1] Zeeshan Ali, Raheleh Biglari, Joachim Denil, Joost Mertens, Milad Poursoltan, and Mamadou Kaba Traoré. 2024. From modeling and simulation to Digital Twin: evolution or revolution? *Simulation* (2024), 00375497241234680.

[2] Francis Bordeleau, Benoit Combemale, Romina Eramo, Mark van den Brand, and Manuel Wimmer. 2020. Towards model-driven digital twin engineering: Current opportunities and future challenges. In *Systems Modelling and Management: First International Conference, ICSMM 2020, Bergen, Norway, June 25–26, 2020, Proceedings 1*. Springer, 43–54.

[3] Paulo Carreira, Vasco Amaral, and Hans Vangheluwe. 2020. Multi-paradigm modelling for cyber-physical systems: Foundations. *Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems* (2020), 1–14.

[4] Autonomic Computing et al. 2006. An architectural blueprint for autonomic computing. *IBM White Paper* 31, 2006 (2006), 1–6.

[5] Romina Eramo, Francis Bordeleau, Benoit Combemale, Mark van den Brand, Manuel Wimmer, and Andreas Wortmann. 2022. Conceptualizing Digital Twins. *IEEE Software* 39, 2 (2022), 39–46. doi:10.1109/MS.2021.3130755

[6] Andreas Wortmann. 2023. *Digital Twin Definitions.* https://awortmann.github.io/research/digital-twin-definitions/