Poster: Model-Based Design of Time-Triggered Real-time Embedded Systems for Digital Manufacturing

Jiang Wan
Department of Electrical Engineering & Computer Science
University of California
Irvine, CA, USA

Arquimedes Canedo
Siemens Corporation
Corporate Technology
Princeton, NJ, USA

M. A. Al Faruque
Department of Electrical Engineering & Computer Science
University of California
Irvine, CA, USA

ABSTRACT

This work presents a novel *Model-Based Design* (MBD) approach and associated tool-chain for the IEC 61131-3 specific *Programmable Logical Controllers* (PLC) [3]. Our tool-chain automatically synthesizes software for manufacturing Time-Triggered Real-time Embedded (TTRE) systems. A *Software-In-the-Loop Simulation* (SILS) framework integrated into our tool-chain helps to reduce the design iterations. Using a manufacturing robot-arm use-case, we validate our tool-chain and demonstrate a 39× improvement in the *Quality-of-Control* (QoC) when compared to the state-of-the-art approach [2]. Our auto-generated scheduler meets all the hard real-time constraints (zero deadline misses) for a given TTRE system when compared to the scheduler (e.g., 145 deadline misses for a CPU utilization of 95%) presented in [14].

1. INTRODUCTION

Today, manufacturing Time-Triggered Real-time Embedded (TTRE) system is experiencing a major paradigm shift thanks to the innovations in the semiconductor and software industries that make the manufacturing faster, more energy efficient, and reliable [4, 3, 7]. Model-Based Design (MBD) [11, 12] is considered to be a promising solution for the design and verification of the TTRE systems because any modification to the system may be rapidly addressed and verified in a higher level of abstraction. Moreover, hardware/software may be quickly and automatically generated from the model using *Electronic Design Automation* (EDA) tools.

The manufacturing TTRE systems are typically implemented in *Programmable Logical Controllers* (PLCs) and programmed with IEC 61131-3 languages [8]. In [10, 13], MBD methods for PLC application development are discussed. Similarly, Simulink [1] is a commercial MBD tool capable of modeling and simulating TTRE systems according to a synchronous reactive *Model of Computation* (MoC). The *Simulink PLC Coder* [2] can be used to automatically generate IEC 61131-3 code for Simulink subsystems. However, all the above mentioned academic and commercial MBD methodologies share the following limitations: 1) the state-of-the-art *Simulink PLC Coder* [2] tool can only generate non-executable *Function Blocks* (FBs)–An FB is an IEC 61131-3 specific semantic for PLCs; 2) minimum support for the design-space exploration of concepts and control strategies [6]; 3) the lack of a tool-chain to perform rapid SILS or *Hardware-In-the-Loop Simulation* (HILS) for the developed TTRE systems on a target Soft-PLC[1]. Although co-simulation solutions have been proposed in [5, 9], these tool-chains do not support simulation in real-time.

To overcome these limitations, this work proposes a complete tool-chain integrating off-the-shelf domain-specific tools and our proposed interfaces and algorithms. Using our tool-chain, designers may automatically generate, from a high-level model, executable IEC 61131-3 programs with timing (*Worst Case Execution Time* (WCET)) and schedulability analysis for a target Soft-PLC platform.

2. MBD OF TTRE FOR SOFT-PLC

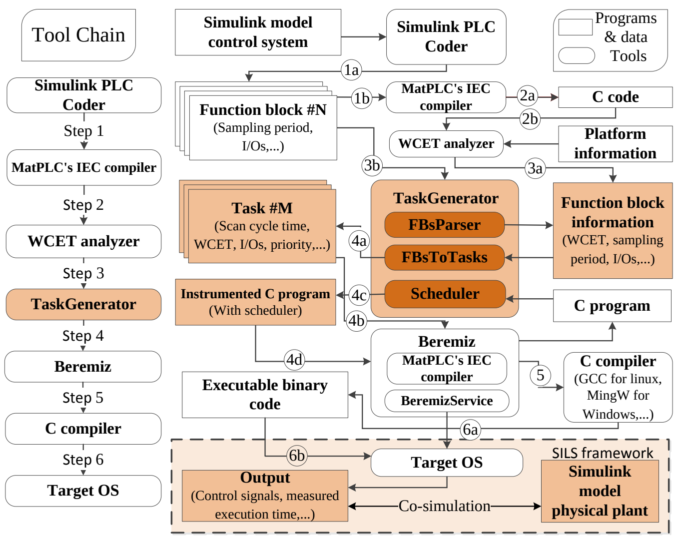

Figure 1: Our proposed MBD approach and the complete tool-chain (the highlighted parts are our contributions)

Figure 1 shows our proposed 6 steps MBD approach and associated tools.

Step 1: The first step of the proposed design approach is to automatically generate non-executable FBs written in the IEC 61131-3 ST language using the *Simulink PLC Coder* [2].

Step 2: Using an IEC 61131-3 compiler [14] to translate the

[1]Soft-PLCs or PC-based controllers are control programs running on personal computers.

FBs into ANSI C-based *Function Codes* (FCs).
Step 3: Calculating the FCs WCET using C language as the intermediate representation.
Step 4: Our *TaskGenerator* generates the executable tasks with the timing information and synthesizes a real-time preemptive scheduler for the run-time system. Moreover, the timing and scheduling information is fed back to the system designers. This will help them to achieve reduced iterations and shorter design-time.
Step 5: When tasks are automatically generated, this step implements the control program in a Soft-PLC.
Step 6: Soft-PLC programs are compiled for a target *Operating System* (OS) and made executable together with the provided Soft-PLC services.

3. RESULTS AND EVALUATION

In our experiments, we study a real-world factory robot-arm application to validate our approach. In this case study, four "robot-arm joints" are controlled by the Soft-PLC. For our experimental purposes, we implemented the "Soft-PLC" and "Robot-Arm Model" in two separated PCs communicated through ethernet to mimic the real scenario in manufacturing factory.
To compare the improvements of QoC using our tool, we use the *Sum of Absolute Differences* (SAD) between MILS and SILS results as the KPI. Figure 2 shows that compared to the Simulink PLC Coder, our tool decreases SAD exponentially to 2.58%. This indicates a 39x improvement.

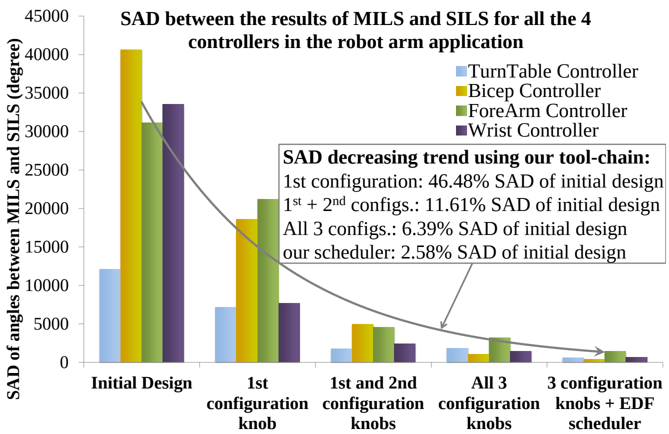

Figure 2: QoC analysis for the robot arm application

4. CONCLUSION

This paper presents a MBD approach and associated tool-chain for the design of TTRE systems for digital manufacturing. Our system is integrated to a SILS framework for the real-time validation of the generated control software using a Soft-PLC. Experiments using our tool chain show a 39× improvement of QoC (2.58% SAD compared to [2]) with zero deadline misses that meets the hard real-time requirement (compared to [14]). Using our configuration knobs, we have demonstrated that a real world manufacturing application can be redesigned in a single iteration. Compared to traditional approaches [1], our approach and tool chain compresses the overall design time.

5. REFERENCES

[1] "Simulink". http://www.mathworks.com/products/simulink/.

[2] "Simulink PLC Coder". http://www.mathworks.com/products/simulink-coder/.

[3] A. Canedo and M. A. A. Faruque. "Towards parallel execution of IEC 61131 industrial cyber-physical systems applications". *in IEEE/ACM Design Automation and Test in Europe (DATE'12)*, pages 554–557, 2012.

[4] A. Canedo, H. Ludwig, and M. A. A. Faruque. "High Communication Throughput and Low Scan Cycle Time with Multi-/Many-Core Programmable Logic Controllers". *in IEEE Embedded Systems Letter (ESL)*, 2014.

[5] H. Carlsson, B. Svensson, F. Danielsson, and B. Lennartson. "Methods for Reliable Simulation-Based PLC Code Verification". *Industrial Informatics, IEEE Transactions on*, pages 267–278, 2012.

[6] D. Dumbacher and S. R. Davis. "Building operations efficiencies into NASAâĂŹs Ares I crew launch vehicle design". *In 54th Joint JANNAF Propulsion Conference*, 2007.

[7] M. A. A. Faruque and A. M. Canedo. "Intelligent and Collaborative Embedded Computing in Automation Engineering". *in IEEE/ACM Design Automation and Test in Europe (DATE'12)*, pages 344–355, 2012.

[8] G. Frey and L. Litz. "Formal methods in PLC programming". *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, pages 2431–2436, 2000.

[9] U. Gangoiti, M. Marcos, and E. Estevez. "Using cyclic executives for achieving closed loop co-simulation". *Decision and Control and European Control Conference (CDC-ECC '05)*, pages 4785–4790, 2005.

[10] J. C. Jensen, D. Chang, and E. A. Lee. "A Model-Based Design Methodology for Cyber-Physical Systems". *Wireless Communications and Mobile Computing Conference (IWCMC'11)*, pages 1666–1671, 2011.

[11] M. Marcos, E. Estevez, N. Iriondo, and D. Orive. "Analysis and validation of IEC 61131-3 applications using a MDE approach". *Emerging Technologies and Factory Automation (ETFA'10)*, pages 1–8, 2010.

[12] D. Mendez-Fernandez, B. Penzenstadler, M. Kuhrmann, and M. Broy. "A Meta Model for Artefact-Orientation: Fundamentals and Lessons Learned in Requirements Engineering". *In Proceedings of the 13th International Conference on Model Driven Engineering Languages and Systems (MODELS'10)*, pages 183 – 197, 2010.

[13] K. Thramboulidis and G. Frey. "Towards a Model-Driven IEC 61131-Based Development Process in Industrial Automation". *Journal of Software Engineering and Applications*, pages 217–226, 2011.

[14] E. Tisserant, L. Bessard, and M. de Sousa. "An Open Source IEC 61131-3 Integrated Development Environment". *IEEE Transactions on Industrial Informatics*, pages 183–187, 2007.