

Chapter 20

Iterative Knowledge Based Embedded Systems Development Framework

Goh Kiah Mok

Singapore Institute of Manufacturing Technology, Singapore

Benny Tjahjono

Cranfield University, UK

Ding Wei

Penn State University, USA

ABSTRACT

Developing an embedded software solution can be time consuming and challenging especially for non-software trained engineers. This is because traditionally, embedded software is programmed manually in proprietary computer languages such as C, C++, Java and assembly languages, meaning that the developers have to be familiar with at least one of these languages. In addition, most of the embedded software design environments do not cater for both microprocessors-based and Field Programmable Gate Array (FPGA) based embedded computing environments, making the development process even more difficult without the assistance of a common method. This chapter proposes a design of a new embedded system code generator framework which is based on the International Electrotechnical Commission (IEC) 61499 Function Block, XML and EBNF. Along with this code generator, an Iterative Knowledge Based Code Generator (IKBCG) is presented to improve the accuracy of the target codes.

1. INTRODUCTION

An embedded system is a special-purpose, small size hardware device and software systems com-

monly used in standalone products or added on to other equipment to enhance their functionalities. Examples include consumer products and industrial control equipment often found in manufacturing or aerospace sectors. By using embedded systems, both complexity and flexibility of equipment can

DOI: 10.4018/978-1-60960-086-0.ch020

be fulfilled mainly because embedded systems developers need to focus only on specific functions requested in order to optimize the solution.

Developing an embedded software solution is a challenging task and can be time consuming especially for non-software trained engineers such as automation and control engineers. Most of the non-software trained engineers or stakeholders cannot participate in the design and development of embedded system. Traditionally, software that runs on an embedded system is programmed manually in various proprietary computer languages such as C, C++, Java and assembly languages and the executable file is downloaded onto the hardware platform. This means the developers of embedded systems have to be familiar with many proprietary computer languages that support multiple embedded platforms. However, these proprietary languages are not compatible with each other, making it difficult to reuse the source codes. For example, although C and C++ are similar, they are not fully compatible with some of the subset embedded computer languages like SystemC, HandelC, and SpecC. In order to facilitate a more effective and efficient development of embedded systems solutions, a common embedded systems development framework for various embedded systems platforms is therefore needed.

The International Electrotechnical Commission (IEC) has proposed the IEC 61499 standard (IEC, 2000) which provides a modelling method aimed at manufacturing and automation domain experts. The standard allows encapsulation of control logic into function blocks that represent sequence of execution, based on process state, communication, control, event and dataflow using both networks and composite function blocks. The IEC61499 also encapsulates the low level source codes and makes low level coding independent from design. Although the IEC 61499, to a large extent, has been endorsed by automation and manufacturing engineers (Vyatkin et. Al., 2005), it does not address the lower level embedded system software issues such as the limited defi-

nition in handling the low level code generation. The overall purpose of the work described in this paper is therefore to explore the use of the IEC 61499 as a system design method for embedded systems and to extend its capability to address the aforementioned issues within the embedded systems design framework.

The paper is structured as follows. Having briefly introduced the topic of the paper, Section 2 describes the background of the research and its related work. Section 3 discusses the proposed framework for embedded system development. Section 4 presents the development environment and the chain of tools which leads to Section 5 that illustrates the validation test of the methodology using simple light switches. Section 6 discusses the code generation of SystemC using XML, and the IKBCG before the conclusions and future work.

2. BACKGROUND AND RELATED RESEARCH

This section presents the current embedded system environments and recent research on standard-based methodologies for embedded systems code generator.

2.1 Embedded System Environments

Embedded systems are typically developed using Microprocessor, FPGA and Application Specific Integrated Circuits (ASIC) hardware platforms. Figure 1 shows the different types of embedded systems development processes, design methodologies, tools and approaches for microprocessor, FPGA and ASIC. ASIC design methodologies involve a wide variety of complex tasks in the placement and routing step, such as placement of circuit and physical optimization, clock tree synthesis, signal integrity and routing. In the placement and routing step, both FPGA and ASIC development processes consist of defining the design specifications such as input/output

23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/iterative-knowledge-based-embedded-systems/50442

Related Content

A Formal Language for XML Authorisations Based on Answer Set Programming and Temporal Interval Logic Constraints

Sean Policarpio and Yan Zhang (2013). *Developing and Evaluating Security-Aware Software Systems* (pp. 138-160).

www.irma-international.org/chapter/formal-language-xml-authorisations-based/72203

Enhancing ERP System with RFID: Logistic Process Integration and Exception Handling

Dickson K. W. Chiu, Kai-Pan Mark, Eleanna Kafeza and Tat-Pui Wong (2011). *International Journal of Systems and Service-Oriented Engineering* (pp. 63-79).

www.irma-international.org/article/enhancing-erp-system-rfid/58513

A Novel Spatial Data Pipeline for Orchestrating Apache NiFi/MiNiFi

Chase D. Carthen, Araam Zaremehri, Vinh Le, Carlos Cardillo, Scotty Strachan, Alireza Tavakkoli, Frederick C. Harris Jr. and Sergiu M. Dascalu (2024). *International Journal of Software Innovation* (pp. 1-14).

www.irma-international.org/article/a-novel-spatial-data-pipeline-for-orchestrating-apache-nifiminifi/333164

Proposals of a Method Detecting Learners' Difficult Points in Object Modeling Exercises and a Tool to Support the Method

Takafumi Tanaka, Kazuki Mori, Hiroaki Hashiura, Atsuo Hazeyama and Seiichi Komiya (2015). *International Journal of Software Innovation* (pp. 63-74).

www.irma-international.org/article/proposals-of-a-method-detecting-learners-difficult-points-in-object-modeling-exercises-and-a-tool-to-support-the-method/121548

Enhancing Software Maintainability by Unifying and Integrating Standards

William C. Chu, Chih-Hung Chang, Chih-Wei Lu, Yeh-Ching Chung, Hongji Yang, Bing Qiao and Hewijin Christine Jiau (2003). *Advances in Software Maintenance Management: Technologies and Solutions* (pp. 114-150).

www.irma-international.org/chapter/enhancing-software-maintainability-unifying-integrating/4901