통합된 Esterel/C++시뮬레이션을 위한 GUI 코드자동생성

리수주안 *, 임기욱**,이재호***, 한태숙***
* 중경우전대학 자동화학과
** 선문대학교 전자계산학과
*** 한국과학기술원 전산학과
e-mail : sujuanliu@gmail.com

A GUI Module Generator for Integrated Esterel/C++ simulation

Sujuan Liu*, Kee-Wook Rim**, Jaeho Lee ***, Taisook Han ***
* Dept. of Automation, Chongqing University of Posts and Telecommunication
** Dept. of Computer Science, Sun Moon University
*** Dept. of Computer Science, Korea Advanced Institute of Science and Technology

Abstract

Nowadays, as the increasing functionality and scales of embedded systems, system design grows more complex than before. So verification and simulation of systems become an important facet in hardware-software co-design issues. But it is almost impossible to simulate an embedded system without real hardware implementation or environment communication, especially for control-dominated reactive systems. Therefore, in this paper, we will introduce a GUI environment module generator for integrated Esterel\C++ simulation. By generating the GUI modeling environment, we can simulate and verify the whole embedded system conveniently.

1. Introduction

Hardware-software co-design has brought lots of benefits in the development of embedded system design. With traditional embedded system design, system verification and simulation is done after the prototyping hardware is available. But with hardware-software co-design, it is more necessary to verify correct system functionality before the hardware is built, it is the reason why there are so many researchers try to develop environment for system simulation [2, 4, 5, and 6]. The goal of this work is to generate an automated GUI simulation environment from system abstract specification and simulate the whole system.

Modern computerized systems can be divided into three categories: transformational systems, interactive systems and reactive systems. As it is impossible to generate simulation environment for all types of systems, we only focus on control-dominated reactive systems in this paper. Control-dominated reactive systems are typically embedded systems continuously react to the environment. Generally, an embedded system can be viewed as three modules parts: hardware (HW), software (SW) and communication environment. For hardware part, we use a synchronous description language Esterel to implement the whole system specification. Esterel [1] is an imperative concurrent language which is developed for reactive systems. Its program execution progressed a cycle at a time, which provides programmer precise time control. For software part, we use high level programming language (e.g. C or C++), C\C++ can provide high usability power. Communication component is to transfer data information between hardware and software part. By using CEC (Columbia Esterel Compiler) [3], we can compile Esterel specification and generate APIs for simulation in C language.

In this paper, we consider the case where system partition has already finished. In order to generate simulation GUI environment, besides Esterel main control module to control system processing, we also need another specification language (e.g. XML) to define GUI modules. In GUI component specifications, we define four signal types corresponding to Esterel specification.

As we all known, GUI is more perceptible than text specification for human beings. So GUI brings more convenience. The GUI is implemented using Qt toolkit [7]. Qt is a cross-platform GUI programming applications developed by Trolltech, and it also provides a C++ class library.

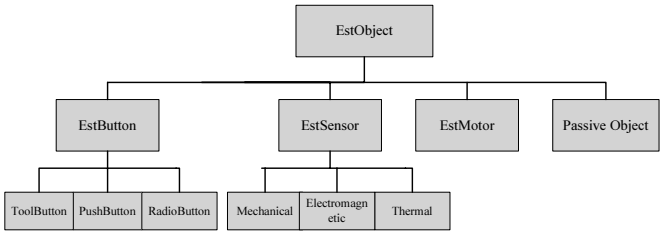
This paper is organized like as follows. In section 2, we introduce the XML system specification description in details. In section 3, we describe the method how to generate GUI simulation environment from system specification automatically. Then a simple elevator example follows in section 4. Finally, in section 5, we conclude our design works and suggest the future work.

2. Specification Description

Based on module descriptions in Esterel, we need another text specification to assist simulation of the whole system. In this specification, all signals' graphical properties and relation with others should be described. In the simulation environment, four components (e.g. button, sensor, motor

and passive objects) will be generated automatically. Mapping to C\C++ codes, four classes will be generated dynamically. As mentioned before, we only focus on control-dominated reactive systems, so it is almost enough using these four components to describe the module functions of an embedded system. The specification is also extensive to cover other more complex system in future. In an embedded system, the inputs are produced by sensors or operator commands. The outputs are commands to motors, valves, and so on. In order to simplify the final GUI generation, we propose passive objects, which can be controlled by motors and they also can activate sensors.

Figure 1 shows the hierarchy of all components.



(Figure 1: The Hierarchy of Component Objects)

From the hierarchy of component objects, it is powerful and extensive to describe a control-dominated reactive system. For the complex embedded system, our specification can be extended to cover new elements easily.

The specifications of four components are described in details.

2.1 Button Specification

The button specification, there is a button name tag (Name) which specifies the input signal name in Esterel. Moreover, it must be the same as the input signal name in Esterel. Using Qt signal/slot mechanism, when button component is activated by mouse, the signal (e.g. clicked()) will be emitted, and the slot connected with this signal will be executed at once. Usually, we define the slot to set input signals in Esterel.

2.2 Sensor Specification

The sensor specification only includes graphical. The graphical specification describes sensor components' position, shape, color and sensor name which is the same as the one in Esterel, so that it can be generated dynamically by accessing its graphical specification. The relation with passive object will be described in passive object specification. The followings are the tags described in XML specification.

Name: the sensor signal name in Esterel, which should be the same as the one in Esterel module.

Shape: shape of sensor

PosX: sensor's horizontal coordinate

PosY: sensor's vertical coordinate

Width: width of sensor

Length: length of sensor

Color: the initial color of sensor

2.3 Motor Specification

Motor component is considered as a non-graphical object but is activated by output emitted from Esterel program and controls the movement of passive object. In the motor specification, there are three elements: MotorName, Activated_By_OutputName and Control_Psi_Speed.

MotorName: the motor name

Activated_By_OutputName: the output name emitted from Esterel program

Control_Psi_Speed: the speed of passive object's movement when passive object is activated by the motor. Its value is an integer

2.4 Passive Object

The passive object specification describes the graphical properties and the relation with sensors and motors.

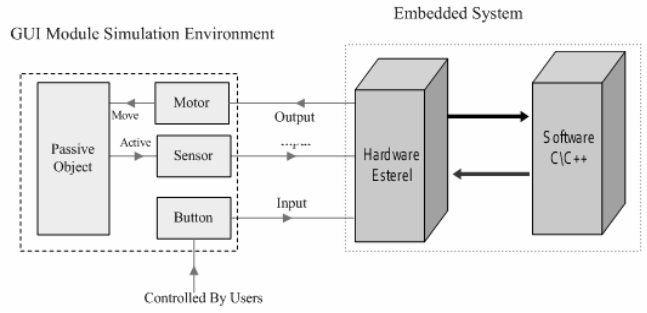
Activated_SensorName: the sensor name activated by passive object

Activated_SensorPosX: the horizontal coordinate of sensor where it will activate the sensor

Activated_SensorPosY: the vertical coordinate of sensor where to activate the sensor

Color_When_Activated: sensor color when it is activated by passive object

Controlled_By_MotorName: motor name which is output signal name in Esterel, it is to control passive object



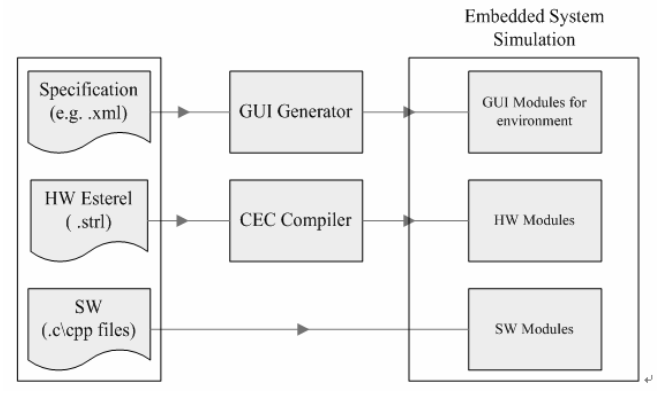
(Figure 2: The Relation between Components)

In Figure 2, we show the relations among four components. In every cycle, Button and Sensor signals are mentioned as described in hardware part in Esterel. Motors are controlled by outputs emitted from hardware part, and it will also control passive object, then the sensors may be activated by passive object. By using this mechanism, system simulation can be performed well.

3. Automated GUI Simulation Environment Generation

Figure 3 depicts the architecture of simulation environment generator. Using XML specification, we generate four component classes and layout in GUI simulation generator. Hardware Esterel file is compiled to HW modules for simulation using CEC. And once button components are activated by user, a timer will be triggered using QTimer module mechanism. After the timer starts, the communication between Esterel main control modules and GUI is built. At every timer internal, say a "tick", input reception from GUI and output production from Esterel are

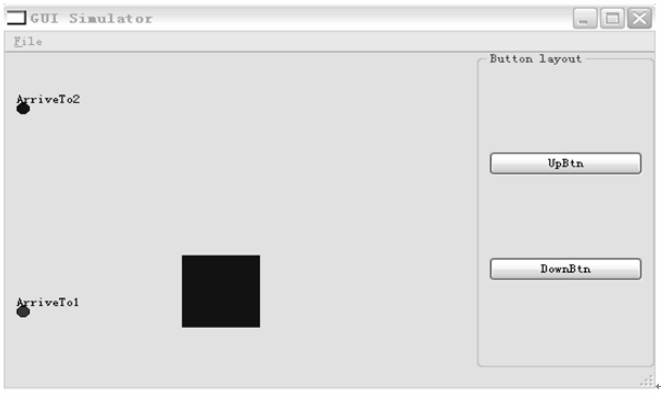
done through simulation interface. After GUI simulation environment gets the outputs, the corresponding motors will be commanded. Based on the relations among components in Figure 2, simulation of the whole system can be performed.



(Figure 3: Architecture of Simulation Environment)

4. Case Study

Our approach has been successfully tested in an elevator system. We will take elevator system with two floors as an example. In order to simplify the system, we just use two buttons here. One is to request motor car down, the other is to request motor car up. We can generate simulation APIs by compiling Esterel specification using CEC compiler. According to Esterel specification, specification of GUI components in XML is also translated into C\C++ modules. In XML specification, users should follow the patterns in every component type, and all signals should be described.



(Figure 4: A Screenshot of GUI Simulation)

In figure 4, it is a screenshot of generated GUI simulation environment. After generating the GUI modeling environment, the simulation of simple elevator system can be performed by clicking buttons and observing movement.

5. Conclusion and Future Works

This paper presents a GUI module generator for simulating an embedded system. Our contribution is, with system specification, most control-dominated reactive systems can be described using four categories, which is easy to use. Moreover, the specification can be extended in order to cover other complex systems. For the environment code generation, we can generate GUI module from system specifications automatically, and make graphical simulation of an embedded system possible.

Our future work is as following. The first one is to make system specification more powerful and extensive. As the increasing complexity of modern embedded system, it is necessary to cover many types of systems. The second one is to make the graphical user interface more convenient and friendly.

Acknowledgments.

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement)"(IITA-2007-C1090-0701-0020)

References

[1] G. Berry , "The Foundations of Esterel ", in Proof, Language and Interaction: Essays in Honor of Robin Milner, ser. Foundations of Computing Series, G. Plotkin , C. Stirling , and M. Tofte, Eds. MIT Press,Aug.2000.

[2] Sari L. Coumeri and Donald E.Thomas. “A simulation Environment for Hardware-Software co-design”. IEEE Design and Test of Computers, Pages 16-28, September 1993

[3] Stephen A. Edwards and Jia Zeng, “Code Generation in the Columbia Esterel Compiler”, Department of Computer Science, Columbia University, New York, December 2006.

[4] Tkachuk, O. Dwyer, M.B. Pasareanu, C.S, “Automated environment generation for software model checking”, Automated Software Engineering, 2003. Proceedings. 18th IEEE International Conference, Oct 2003.

[5] J.T. Buck et al., "Ptolemy: A Framework for Simulating and Prototyping Heterogeneous Systems," Int"l J. Computer Simulation, Apr. 1994, pp. 155-182

[6] Chatelain, A.Mathys,Y. Placido,“High-level architectural co-simulation using Esterel and C”, Proceedings of the Ninth International Symposium, 2001 Page(s):189 - 194

[7] Jasmin Blanchette and Mark Summerfield, C++ GUI Programming with Qt 4, Prentice Hall, June 21, 2006.

[8] XML Tutorial. http://www.w3schools.com/default.asp