Study of Specification Environment based on ObTS for Embedded Systems

Masao Nomura

School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 2000

Keywords: embedded system, state transition diagrams, ObTS, ObCL, ObML.

Recently, embedded systems are used in various fields with the improving performance of electric devices. On the other hand, these systems become larger and more complex. Moreover life-cycle of products in which embedded systems are used become shorter. So embedded system designers must develop these systems promptly and efficiently. Therefore they want effective development method. However no such method for embedded systems has been established.

When we design embedded systems, state transition diagrams are important because they are effective method to describe control structure of systems. Statecharts is generally taken advantage of description for state transition diagrams. Statecharts extend state transition diagrams using hierarchy, concurrency and broadcast communication. ObTS extends Statecharts using concepts of object-orientedness. In ObTS, system structure is composed by hierarchy of objects, and system behavior is expressed by behaviors of objects and inter-object communications. (ObTS has ObCL which propose specification description language based on it and ObML which is a simulation environment for it.) ObCL defines concrete syntax for ObTS computation model. It has some extensions to reuse ObCL codes for large scale codes: class, inheritance of them for reusing codes, field for multicast communication, event class for encapsulating attributes of events. ObML is the simulator which constructed on Standard ML. A system which described ObCL codes are converted to ML codes by ObCL converter, and these codes can be simulated and tested in ObML. In ObML, it is easy to make high-level test script using ML language.

In this paper, we proposed a method of describing state transition diagrams to develop systems efficiently. To achive it, high reusability of designs and program codes, and

testing/verification mechamism in the specification step are required. But the flatness of state transition diagrams deteriorate system readability when designers describe large scale systems. Therefore, it is difficult to extend and add functions into systems, and it makes development inefficient. However, it is easy to modify a part of the system which fulfills a specific function. If the specification of whole system is automatically updated to reflect modifications on parts, we develop the system efficiently. Then, we aim to work out such a method.

At first, we propose our method that slices state transition diagram of whole system to state transition diagrams of particular functions, changes them into 'basic form', and composes every functions in state transition diagrams. This technique aims to specify expansion and addition of functions into systems. Some sample using technique that state transition diagrams are changed will be shown.

Next, a case study using ObTS/ObCL to describe virtual ATM is illustrated. Applying our method to the ATM, we will show the process of slicing, changing and composing state transition diagrams, and adding a function into the ATM. Test scripts for every functions can be also applied to the composed system. And test script for the whole system can be composed directly using each scripts. This realizes higher reusability of test scripts themselves. Furthermore by making hirarchy tree of basic forms, we show an evolutionary method of describing systems, and we show that this method is high reusability system design.

Finally, a case study using ObTS/ObCL to describe IrLAP(Infra-red Link Access Protocol) which is adopted infrared communication as a large scale embedded system is examined. we show that this method can be applied to large scale embedded systems and environment using ObTS/ObCL/ObML is effective to develop these systems. By means of description using ObTS/ObCL and the result of simulation using ObML, we illustrate that there are mistakes or ambiguous designs in specification published by IrDA(Infra-red Data Assoiation).