TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2007; **2**: 402–404
Published online in Wiley InterScience (www.interscience.wiley.com). DOI:10.1002/tee.20162

Letter

Exploration of Communication Models in the Design of Distributed Embedded Systems

Kazutaka Kobayashi*, Non-Member
Takashi Shiraishi*, Non-Member
Nurul Azma Zakaria*, Non-Member
Ryosuke Yamasaki*, Non-Member
Norihiko Yoshida*[a], Member
Shuji Narazaki**, Non-Member

Distributed embedded systems involve communication in various layers, and therefore their design is more difficult than of single embedded systems. This paper presents how communication exploration can be done in a design process of distributed embedded systems using an example of event-triggered and time-triggered communication. A design process begins from abstract specification without assuming any communication category, then explores the categories in a stepwise manner, followed by physical implementation synthesis. This encourages stepwise decision making, component and framework reuse, and early stage verification.

Keywords: distributed embedded systems, network communication, model-driven architecture, stepwise refinement design

Received 27 November 2006

1. Introduction

Modern embedded systems often work in networks, which comprise *distributed embedded systems*, as found in vehicles, for example.

Embedded systems involve hardware and software, and therefore hardware/software (HW/SW) integrated design methodologies, such as System-level Design [1] and HW/SW CoDesign [2], have been gradually put into practice. Typical HW/SW integrated design proceeds as shown in Fig. 1 [3]. It is a stepwise refinement process from abstract specification to implementation.

Distributed embedded systems involve communication in various layers, from bus connections to networks, and therefore communication design is more important and difficult than in single embedded systems. An issue, from the network point of view, in the above process when applied to design of distributed embedded systems is that communication is concerned mainly with bus connections, and so communication exploration is not separated from architecture exploration, in which a suitable combination of modules is explored among several possibilities to fix an architecture model.

This paper presents how communication exploration can be done in a design process of distributed embedded systems using an example of event-triggered and time-triggered communication [4].

[a] Correspondence to: Norihiko Yoshida, yoshida@ics.saitama-u.ac.jp.
* Graduate School of Science and Engineering, Saitama University, Saitama 338-8570, Japan
** Faculty of Engineering, Nagasaki University, Nagasaki 852-8521, Japan

2. Event-triggered and Time-triggered Communication

There are two major categories of network communication in distributed embedded systems: event-triggered and time-triggered. Event-triggered communication is flexible and appropriate for soft real-time systems. Time-triggered communication, on the contrary, is deterministic, in the sense that all instants of message transmission are scheduled beforehand. This is suitable for applications in which the data traffic is of a periodic nature, and ensures dependable hard real-time message delivery which is necessary in safety-critical applications.

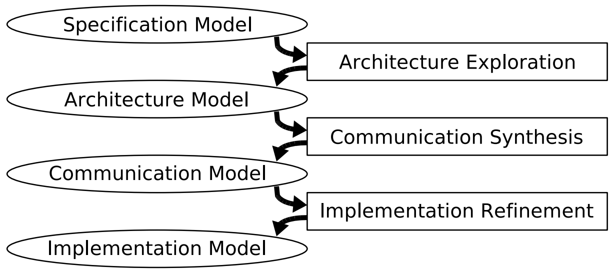

Fig. 1 Design process for embedded systems

Both the above have two sub-categories: centralized and decentralized. In centralized event-triggered/time-triggered communication, a single arbiter/scheduler manages the whole network. In decentralized event-triggered/time-triggered communication, each module is responsible for arbitration/scheduling. The former is less expensive and easier to implement, while the latter is faster and more robust owing to the absence of a single point prone to failures and load concentration.

For example, below are some protocols for in-vehicle networks:

- CAN: decentralized event-triggered;
- LIN: centralized time-triggered;
- FlexRay: decentralized time-triggered.

3. Stepwise Exploration of Communication

In the conventional design process, we must select which communication protocol to use at the beginning. Once having selected any, we cannot switch to another, even if it is found later that another is better. A complex distributed embedded system may include several categories of communication, which should be selected depending on physical constraints; thus it is sometimes difficult or impossible to select at the beginning. In addition, we cannot reuse a component or framework for other systems based on any other protocols.

Consequently, following Model-driven Architecture (MDA) discipline [5], which is now widely accepted in Software Engineering, we investigate a design process of communication that begins from abstract specification without assuming any communication category and then explores the categories in a stepwise manner followed by

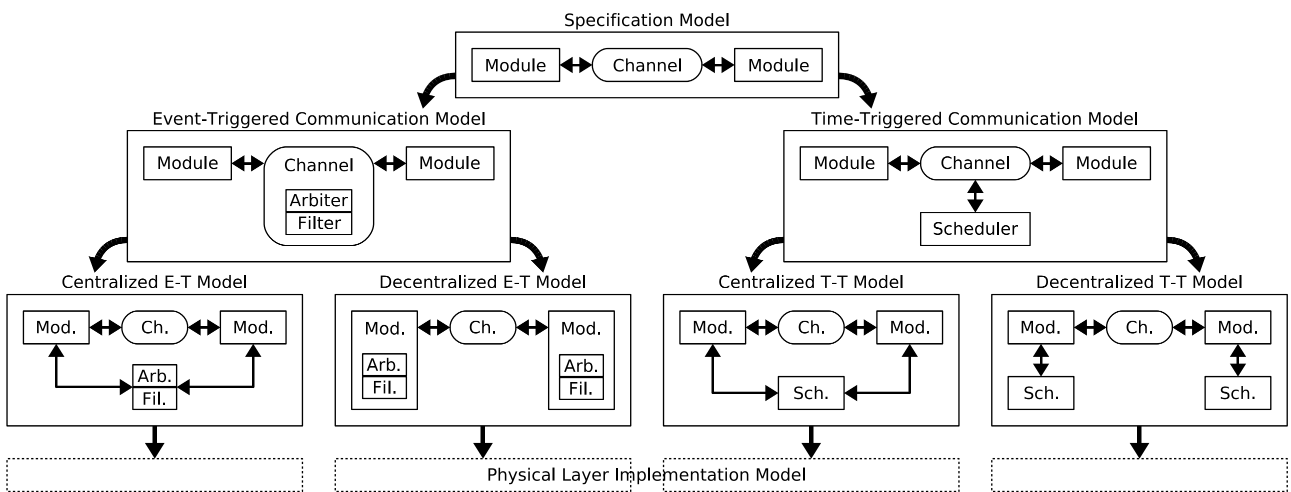

Fig. 2 Stepwise exploration of communication models

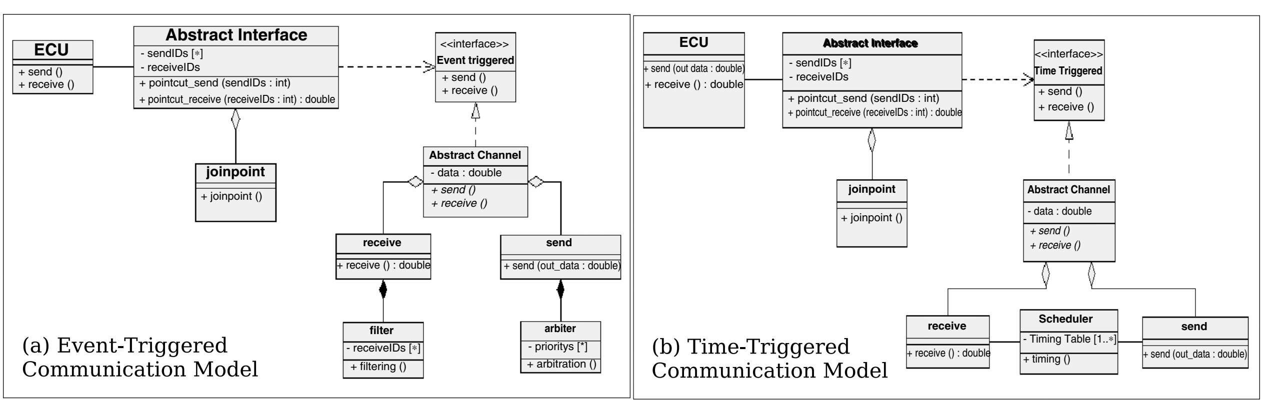

Fig. 3 UML class diagrams of communication models

physical implementation synthesis. Figure 2 shows the result in outline. Models are actually described in UML, and two examples, class diagrams of event-triggered and time-triggered communication models, are shown in Fig. 3, where 'EUC' (Electronic Control Unit) is an embedded module, and 'joinpoint' is a tool class for framework reuse.

At each model, a designer verifies its correctness, and then selects which way to go, considering advantages of each path such as presented in Section 2, on the basis of some estimation or profiling that reflects requirements and constraints. The designer makes a decision, not at once at the beginning, but in a stepwise manner gradually fixing details. Also, the designer verifies the system, not at once after implementing it, but in a stepwise manner.

4. Related Works and Concluding Remarks

There is a research trend to apply UML and MDA to HW/SW integrated design, which started a few years ago (see Proc. 2006 Workshop on UML for SoC Design in conjunction with ACM/IEEE 43th Design Automation Conf.). However, there has been no study on communication exploration yet.

Stepwise exploration encourages stepwise decision making, component and framework reuse, and early stage verification, all of which accelerate design processes. This letter applies it to the design of distributed embedded systems. This letter also contributes toward integrated design of event-triggered and time-triggered communication, which are used separately at present.

We have verified our approach by interpreting the models and the design process in the SpecC modeling language [3]. Our ongoing studies are: (i) interpreting them in executable UML [5], (ii) formalizing semantics-preserving transformation between models to build an automatic CAD tool, and (iii) investigating some real-world applications.

References

(1) Keutzer K, Malik S, Newton AR, Rabaey J, Sangiovanni-Vincentelli A. System level design: orthogonalization of concerns and platform-based design. *IEEE Transactions* 2000; **CAD19(12)**:1523–1543.

(2) Balarin F, Chiodo M, Giusto P, Hsieh H, Jurecska A, Lavagno L, Passerone C, Sangiovanni-Vincentelli A, Sentovich E, Suzuki K, Tabbara B. *Hardware-Software Co-Design of Embedded Systems*. Kluwer, Norwell, MA, USA, 1997.

(3) Gajski DD, Zhu J, Dömer R, Gerstlauer A, Zhao S. *SpecC: Specification Language and Methodology*. Kluwer, Norwell, MA, USA, 2000.

(4) Kobayashi K, Yamasaki R, Yoshida N, Narazaki S. Exploration of communication specifications in system level design. *IEICE Technical Report* 2006; **SLDM-127**:175–179 (in Japanese).

(5) Mellor SJ, Scott K, Uhl A, Weise D. MDA Distilled: Principles of Model-Driven Architecture. Addison-Wesley, Boston, MA, USA, 2004.