Chapter 15
The Language Specification
PEARL for Co–Designing
Embedded Systems

Roman Gumzej
University of Maribor, Slovenia

Wolfgang A. Halang
Fernuniversität, Germany

ABSTRACT
The Specification PEARL language and methodology for hardware/software co-design of embedded control systems is presented. The Specification PEARL language has its origins in standard Multiprocessor PEARL, and was enhanced as foreseen by the standard to model layered and asymmetrical multiprocessor system architectures. Its constructs were extended by additional parameters for model verification and validation by schedulability analysis. Graphical symbols were introduced for Specification PEARL constructs to enable graphical modeling while maintaining their semantic background. Originally, Specification PEARL was intended as a superlayer for normal PEARL programs, implementing the Multiprocessor PEARL specification. Later on, however, it evolved to a methodology to specify and co-design applications for other target programming languages, such as C and C++, based on the superior programming model of PEARL. To supplement structural modeling by a behavioural model, Timed State Transition Diagrams were defined to model program tasks. A system model is verified for coherence by model checking as well as for temporal feasibility with co-simulation. The resulting information is used to correct and fine-tune a current model. To enhance availability and safety as well as to support maintainability, dynamic re-configuration planning is included in the software-to-hardware mapping specifications. Since UML has become the de facto standard also for designing embedded control systems, a UML profile for the methodology was defined using UML’s extension mechanisms.

DOI: 10.4018/978-1-60566-800-0.ch015
INTRODUCTION

Since embedded real-time systems depend on both – hardware and software – in order to fulfill their duties safely and efficiently, they should be designed in a holistic way taking both aspects into consideration. This chapter presents the results of a project whose purpose was to devise a language, methodology, and development environment to achieve this – Specification PEARL (Gumzej, 2004).

The Specification PEARL (S-PEARL) language is based on the notation of standard Multiprocessor PEARL (DIN 66253-3, 1989), which was extended in the manner foreseen by the standard by the following features:

- enhanced constructs to describe hardware configurations,
- enhanced constructs to describe software configurations,
- construct properties endowed with timing information (where applicable),
- graphical representation of constructs to enable graphical modeling,
- constructs to specify communication and its characteristics (peripheral and process connections, physical and logical connections, transmission protocols), as well as
- constructs to specify both conditions and methods of carrying out dynamic re-configurations.

The Specification PEARL co-design methodology enables to construct a conceptual system model, whereby its hardware and software architectures may be described in parallel. It supports a project’s lifecycle as shown in Fig. 1. After a system model is built from its hardware and software specifications within the co-design environment, it is checked for consistency, and its temporal feasibility is analysed with the associated co-simulation environment (Gumzej, 2004). These steps may be repeated multiple times while introducing changes to the software/hardware designs in order to correct/improve the system model. Once the designer is satisfied with the model, a target platform model can be produced from the design environment for cross-compilation to the target platform.

The project evolution process from specification of system architectures, via software design, verification and validation, to deployment of the system model is straightforward and based on the waterfall model, leaving the designer with no semantic gap between specification and design formalisms as well as implementation and verification of models produced. The resulting feasible models retain their validity even after enhancement to fully functional applications in case their timing limitations are not violated.

The main benefits of using the Specification PEARL methodology to co-design embedded control applications are the abilities to:

- reason on system integration at an early stage,
- introduce timing constraints wherever applicable into a design, and
- check the temporal feasibility of a design before implementation.

BACKGROUND

Designing real-time systems has always been a challenge. Only recently, however, adequate methodical tools have formally been introduced. Naturally, they were present from the beginning. But due to the large diversity of the areas in which embedded systems are deployed, special formalisms have been developed and used in, e.g., telecommunications, industrial control, aerospace, or mechatronics. Embedded systems constitute a large subset of real-time systems and, hence, these two terms are often used interchangeably. Owing to the increase in complexity of information systems and applications in general, as well
Related Content

Applying the Certification’s Standards to the Simulation Study Steps
www.igi-global.com/chapter/applying-certification-standards-simulation-study/55982?camid=4v1a

Disruptive Space Technologies
www.igi-global.com/article/disruptive-space-technologies/75305?camid=4v1a

Designing for Human Factors in the Technology-Intensive Domain of Fighter Aircraft
www.igi-global.com/article/designing-for-human-factors-in-the-technology-intensive-domain-of-fighter-aircraft/104510?camid=4v1a

Simulators as an Essential Tool for Shaping the Competence of the Aviation Personnel
www.igi-global.com/chapter/simulators-as-an-essential-tool-for-shaping-the-competence-of-the-aviation-personnel/223725?camid=4v1a