

E. Holz, M. Wasowski, D. Witaszek, S. Lau<sup>††</sup>

## 1 INSYDE – Methodology Overview

The main objective of the INSYDE project is to define, implement, and demonstrate a comprehensive methodology for the design of hybrid hardware/software systems. Stages in the INSYDE Methodology are analysis, system design, detailed design, and validation, each with its own milestone (Fig. 1). The INSYDE Methodology is not devoted to one particular development strategy, such as the waterfall model, prototyping e.g.: the stages can possibly overlap and iterations between them can occur.

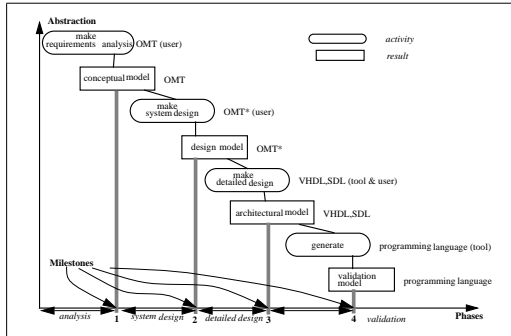


Figure 1:

### 1.1 Analysis

The INSYDE Methodology starts with the analysis of the system to be developed. The major goal of the analysis phase is to get a thorough understanding of the system under development. Using the INSYDE Methodology, one should be focused during the analysis phase on capturing the functional requirements of a system. The functional requirements of a system are defined as those requirements, that describe the functionality of a system, as opposed to the requirements, that specify the performance, technology, and so on. This should not be confused with the functional model as defined by J. Rumbaugh in [1]. The functional requirements capture leads to a first OMT description of the system. If this OMT description is sufficient to serve as the conceptual model, then the design phase can be entered. Otherwise transformations, modifications, or refinements of the OMT description are necessary to get a conceptual model. Since the OMT methodology does not provide a formal base, the OMT description of the conceptual model only has informal semantics. Its interpretation, however, should be clear and precise enough to serve as a starting point for the system design.

<sup>†</sup>. This work has been supported by the European Community, ESPRIT-III-Project INSYDE

<sup>††</sup>. E. Holz, M. Wasowski, D. Witaszek and S. Lau are with Humboldt University Berlin

### 1.2 Design

Design in the INSYDE Methodology can be divided in system design and detailed design. System design is concerned with the specification of subsystems and the interactions between them. Subsystems can be hardware or software. Detailed design deals with hardware and software subsystems separately. The conceptual model, which is a result of the analysis phase, is the basis for the system design phase. The result of the system design phase is the system design model, which serves as a starting point for the detailed design, for which SDL and VHDL are used.

As such, the system design model will be transformed (supported by software tools) to SDL and VHDL. The language used for the system design model should have well defined syntax and semantics. Therefore OMT\* will be introduced as a language for the system design model. OMT\* is a dialect of OMT which contains less but well described syntactical constructs. Each of these constructs has a small number of well defined meanings. Therefore, the OMT\* constructs can be translated by software tools for detailed design. OMT\* is tailored for the two specification languages SDL [2] (for software) and VHDL [4] (for hardware). That means, OMT\* includes SDL-related OMT constructs as well as VHDL-related OMT constructs.

#### System Design

In the INSYDE Methodology the first step of the design phase is the differentiation between OMT classes or structures, being necessary for the further system development, and classes or structures, which were introduced only to support problem understanding and no longer have to be taken into consideration. Then the OMT structure classification goes on with the differentiation between classes and structures representing

- the environment (not to be specified more detailed),
- the software parts of the system (later to be described in SDL),
- the hardware parts of the system (later to be described in VHDL), and
- the interfaces between software and hardware parts (later to be described in SDL as well as in VHDL).

After classifying the structures of the conceptual model (described in OMT), the software, hardware and interface structures should be described in OMT\*. Therefore those structures important for the further system development should be modified to get

- a description of the structures representing the software part by SDL-related OMT\* constructs,
- a description of the structures representing the hardware part by VHDL-related OMT\* constructs, and
- a description of the structures representing the interfaces between software and hardware by OMT\* constructs which are SDL-related as well as VHDL-related.

if such a modification is possible. (Otherwise, the OMT constructs not describable in SDL-/VHDL-related OMT\* have to be carried forward to the system design model.) The methodology for system design in INSYDE is based on the OMT system design methodology ([1], chapter 9). OMT\* should therefore contain as much as possible of the OMT language constructs, excluding only those that are not suited for design purposes (according to [1]) or that are not translatable to SDL nor VHDL. A more thorough discussion on OMT\* and the transition from OMT to OMT\* can be found [3] and [5]. The system design phase is completed with the system design model, described in OMT\* (as far as possible).

#### Detailed Design

The transformation of the SDL-/VHDL-related OMT\* descriptions into SDL/VHDL is done by software tools as automatically as possible, with human intervention where appropriate. The result of the transformation is a system description, which consists of parts described in SDL, and parts described in VHDL. OMT constructs not describable in OMT\* have to be transformed by hand and linked to the SDL or VHDL part, respectively. The hybrid system specification can be refined or modified following the general guidelines for SDL and VHDL. This way a formal architectural model of the hybrid hardware/software system is created, which contains all information needed to describe the required architecture and functionality of the system and makes code generation and simulation possible.

### 1.3 Validation

Two aspects have to be investigated by the validation:

- validation of the self-consistency of the specification (e.g. absence of deadlocks, stability in error cases) and
- validation of consistency with other specifications (on other stages of the development process or w.r.t. requirements or environment specifications).

In the INSYDE Methodology the validation phase is realized by the co-simulation of the hybrid system specification (Heterogeneous Approach): An SDL simulator and a VHDL simulator are used to simulate the SDL and VHDL part of the system specification; the interaction and synchronization between both parts is realized by a Coupling Module, which connects the simulators. An experimental variant, which aims at studying the functionality of this Coupling Module and the general principles of co-simulation, is the integrated simulation (Single Base Approach): The architectural model

(based on SDL and VHDL) is translated into C++, and special runtime-libraries are used to execute the simulation. Problems related to a physical distribution of co-simulation are not considered in the Single Base Approach. Therefore, the Coupling Module for the Heterogeneous Approach must provide the same functionality as for the Single Base Approach with additional features related to the distributed simulation.

## 2 Summary

The whole development process can be divided into several phases, where specific activities are carried out and results are produced. The functional requirements analysis results, after some iterations, in a conceptual model in OMT. The system design, which iterates over the OMT structure classification and the transition from OMT to OMT\*, produces an OMT\* system design model. This system design model is transformed into SDL/VHDL automatically as far as possible, and partially by the user. Additional information for the further specification development can be taken from the OMT\* system design model.

## 3 Acknowledgements

This work is as a part of the ESPRIT-III-Project INSYDE supported by the European Community. We want to thank our project partners, esp. J. Fischer, P. Roques, J.-P. Delpioux, L. Cuyper, J. Heirbaut, K. Verschaeve, V. Mariatos and N. Kyrloglou for the fruitful discussions and their contributions

## 4 References

- [1] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen: *Object Oriented Modelling and Design*, Prentice-Hall, International Editions, 1991.
- [2] CCITT: *CCITT Specification and Description Language SDL*; Recommendation Z.100 (SDL'92), Geneva, 1993.
- [3] E. Holz, D. Witaszek, M. Wasowski, S. Lau, J. Fischer, P. Roques, L. Cuyper, V. Mariatos, and N. Kyrloglou: *INSYDE Deliverable D1.1: Technology Assessment*; ESPRIT P8641 Report, 1994.
- [4] Navabi Z.: *VHDL Analysis and Modelling of Digital Systems*, McGraw-Hill, Inc., 1993.
- [5] E. Holz et al.: *The INSYDE Methodology -Ad-Hoc Document* ESPRIT P8641 Report, 1994