VIII Triennial International SAUM Conference
on Systems, Automatic Control and Measurements
Belgrade, Serbia, November 5-6., 2004

Bond Graph Concepts in Object Oriented Modeling

D.H. Pršić and N.N. Nedić

Faculty of Mechanical Engineering Kraljevo, Dositejeva 19, Kraljevo, University of Kragujevac
E-mail: prsic.d@maskv.edu.yu, nedic.n@maskv.edu.yu

Abstract *- At the technical components abstraction level, bond graph provides a systematic and unified technique to represent the dynamic behavior of engineering systems. On the other side, in programming, as the last phase in the simulation model definition, the object-oriented methodology is present more often recently. In such way, the program looks like as a model of real world. Therefore, it is necessary to transform methodology from one to another abstraction level. In this paper we attend to present how Unified Modeling Language (UML) can be extended on bond graph modeling concepts.*

Key words: *physical system modeling, bond graph modeling, object oriented modeling, unified modeling language*

I. INTRODUCTION

Modeling is iterative system describing process aimed to better system knowledge and easier communication. A model always represents only some a real system abstraction from different point of view. From control point of view, description of system dynamic behavior is very important. Therefore, energetic aspect of system description, based on bond graph method is useful [1,2].

Creation of the simulation model has a few phases, that is, at a few abstraction levels ([3,4,5]). First, at physical components level, we have to identify system's components and their interconnections. For the graphical representation, usually application dependent icons or non-standardized symbols are used. In the next phase, relevant physical processes and their interactions taking place in the components are specified. The real world system is conceptualized in terms of idealized physical mechanisms and their interactions. Network model representing the system is constructed. For quantitative analyses of the dynamic behavior the mathematical model is needed. The mathematical relations that characterize the physical mechanisms and their interactions are specified. Usually, mathematical model is expressed graphically in computational causal form by means of block diagrams.

In that way, the modeling process is structured by the few abstraction levels which create hierarchy in the sense that the degree of details increases from physical component level to the mathematical level.

Nowadays, we can distinguish two modeling methodologies [6]:

- process oriented approach: system is structured by procedures (operations) which describe behavior from the highest to the lowest level.
- object oriented approach: system is decomposed into structural elements which can interact among themselves.

The essential thing which defines one modeling methodology is language for model description. What can be expressed in model is determined by the language in which they are stated. The language specified what terms can be incorporated in model (the vocabulary) and how these can be meaningfully combined (the grammar). The expressiveness of language is large if it has large vocabulary and unrestrictive grammar. Along with expressiveness, the language also determines how a model is to be encoded (the notation). Unfortunately, it is impossible to find one single language that is both powerful and appropriate to formulate model in all modeling phases.

An approach that can circumvent this problem is to use set of languages which are powerful in specific area and together provide expressiveness and representations that are needed. But, simultaneous application of multiple model formulation introduces the need for establishing relation between the different formulation of model. One of the major trouble is the presence of discontinuities that occur within the development process. This discontinuities are caused by the lack of formal relationships between different notations. This discontinuities also make it difficult to trace the linkages between system requirements and the implementations that are supposed to satisfy them. Maintaining this linkage is important to ensure not only that all the requirements are met, but also that the effects of any change can be determined precisely in terms of its effect on the original requirements.

The second approach is based on extending of the used language, on some abstraction level, in order to be used on some another level [7]. Our work is based on this approach.

If we look on modeling process from the simulation point of view, the end aim of the modeling process is to find the model which can be performed on a computer. That is, the model end form is a computer program. Therefore, the usually used modeling process structure is something modified, (Fig.1). As can be seen from this figure, the end phase of modeling process is not explicit mathematical description of a system, but the software model. In fact, the software model is not some a new model in a sense with some new information about the system then only the implementation of the modeling process. Our task is, by set of transformations, to come from physical components level to simulation programming layer.

From above mentioned, it is very obvious the idea that for physical systems modeling we should use the software system modeling methodology. Advantages of this approach are: