IX Triennial International SAUM Conference
on Systems, Automatic Control and Measurements
Niš, Serbia, November 22-23., 2007

Transformation of XML Model into a Block Diagram

D. H. Pršić, N. N. Nedić and A. M. Babić

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

Abstract *- Aiming at the support of an exchange and reuse of models, this paper presents transformation of a behaviour model from one format into another by means of XML. As XML is just a language for describing markup language, we have to use the language and grammar of a certain XML application. The paper takes the bond graph as a source model, and the block diagram in SIMULINK context is taken as the target one. The relation between these two fields is established through BGML format and the appropriate scheme. Simple API for XML is used for processing the XML document. The transformation process is illustrated by a small example.*

Key words*: Modelling and simulation, bond graph, block diagram, XML model format, SAX interface.*

I. INTRODUCTION

Numerous software tools for support to modelling and simulation can be found today. They include rich knowledge and experience in the form of libraries of completed models from various fields. In such a situation, the main issue is not how to develop a new model but how to integrate the one already existing into the given context. The problem is that different development environments use different languages and different formats of information. The reuse and exchange of this model between various software is not easy.

This paper proposes XML based format for communication between various modelling environments and tools. XML is the acronym of Extensible Markup Language, which was accepted by the consortium W3C [1] as a new standard for formatting documents-information. Although it was primarily intended for web pages and other narrative documents (books, textbooks, articles, etc.) read by people, today XML is increasingly used for the exchange of data between various software applications independently of the platform.

Similar problems with compatibility exist among software tools for modelling behaviour of engineering systems. Numerous software packages with libraries for support to different modelling methodologies have been developed. There arises an issue of their mutual cooperation for the purpose of reuse of engineering knowledge and experience.

Use of the XML standard is imposed as the solution of this problem. This opens the possibilities for creating connections not only toward other tools for behaviour modelling but also toward any existing (e.g. CAD system) or future tool understanding the XML format.

The document satisfying all rules of the XML grammar is said to be well-formed [2]. Thanks to the precisely written specification, numerous programs, the so-called parsers (XML processor) have been written. They can analyse an XML document and give us information about its content and structure.

By using XML, we can define our own vocabulary and domain-specific semantics. Each specific XML-based markup language is called an *XML application*. Those are, e.g. Chemical Markup Language (CML), Mathematical Markup Language (MathML) or Bond Graph Markup Language (BGML). The power of XML is hidden exactly in this flexibility and the possibility of defining a domain-specific language for describing data.

For the XML application, besides vocabulary, it is also necessary to define allowed language structures, i.e. grammar. Although we are completely free in defining our own structure of data, we have to respect certain rules (which we adopt or define by ourselves) so that we could exchange those data with others. The rules regarding the structure and content of an XML document may be defined by means of schemas written in any of several languages, such as Document Type Definition (DTD) or W3C XML Schema Language. Schemas list the elements, attributes, entities and relations that can be used, as well as their possible relationships. For example, the instructions for preparation of this paper can be regarded as a schema. At the beginning of every paper, there should be the conference title, below which there is the paper title, the authors' names, the abstract, the introduction, etc. If the document matches the constraints listed in the schema, the document is said to be valid; otherwise, the document is said to be invalid. The very process of checking the matching between the document and the given schema is called validation. If we want to write a program which can read data from a given XML document, we have to respect the schema of that document.

II. MODEL TRANSFORMATION

The structure of a model transformation process between two modelling environments is shown in Fig. 1.

The aim is to convert the internal format of a model from software modelling environment I into an acceptable form of model in software modelling environment II. The connection between these two worlds is established through an XML document. What is important is that the document does not depend on the program writing and reading it. In other words, the connection can be established between any two applications able to create and read an XML document. It means that the model remains open for some future modelling tools.