

Organization of a Microprocessor Design Process Using Internet-based Interoperable Workflows¹

Nguyen Quang Trung, Artur Kokoszka, Krystyna Siekierska, Adam Pawlak,
Dariusz Obrębski, Norbert Ługowski
*Department of Integrated Circuit and System Design
Institute of Electron Technology, Poland
{nqt,kokoszka,siek,pawlak,obrebski,lugowski}@ite.waw.pl*

Abstract

The paper presents a new approach towards large system design in distributed teams based on a workflow technology. The techniques applied introduce interoperability among design tools across computing platforms and organization borders. They result in an improved quality of wide area engineering collaboration.

After a short discussion about selected aspects of the applied workflow technology from an electronic design engineers' perspective, an Internet-based distributed design process of IP components with emphasis on specific elements of microprocessor designs is introduced. As an example, we present a workflow that enables integration and interoperability of selected IP design tasks, represented as parallel and serial sets of the workflow activities. The workflows that were employed in the design process had been developed for the purpose of design task integration in the pan-European project E-Colleg. They can be effectively adopted by distributed teams working on multiple sites, multiple platforms for remote project management.

1. Introduction

The quality of electronic design is a fuzzy notion that can be addressed from different points of view. Usually, the quality of a product is understood as the combination of quality of the design process and quality of the product itself. The quality of the design process, that we refer in this paper to, can be analyzed considering many factors; but the most important among those factors are design methodology, design process management as well as tool and data accessibility. Our work aims at organization of the

IP design process with the use of an Internet-based design environment that integrates appropriate tools, data, and services.

One of the key tasks for system designers today is finding an effective way to handle an exponential growth of design and verification complexity. A straightforward approach requires that heterogeneous designs are broken down into partitions that collaborating teams are capable to design and assure their quality on time. This methodology relies on advanced technologies enabling engineers located in remote sites to efficiently collaborate over the Internet. Nowadays, Internet and Internet-based standards enable the possibility of receiving and delivering electronic data less expensively and more securely. It is expected that the ongoing transition from static data sharing to real-time data sharing will move very fast in this decade that supports collaborative engineering among distributed teams working in both intra-company projects and company-spanning ones. The necessity for effective remote tool services, which can be used to extend the area of collaboration has become urgent.

Workflow is commonly conceived as a concept to accelerate heterogeneous working procedures, which contain a set of activities that support a specific process [2-10]. It provides procedural automation of real world processes by management of a sequence of work activities and the invocation of resources associated with those activities. In the electronic engineering domain, the workflow concept may be used by a designer for creation of the virtual design environment that integrates tools and services distributed over different sites, platforms, and enterprises. Those tools and services require a workflow management system that supports enterprise level integration dealing with many kinds of cross-domain issues.

¹ This work is supported by the European Commission within the framework of the E-COLLEG project IST-1999-11746 and the National Committee for Scientific Research (KBN) project SPUB-M-5.PR-UE/DZ.139/2000.

In order to enable collaborative design and verification of IPs in multiple-site multiple-platform environments, we have developed an executable Internet-based workflow that enables the collaborative development of a number of microcontrollers and a DSP. The workflow encapsulates required local and remote data and tools, with the use of the ASTAI® workflow management system [2] provided by our partner Siemens Business Services within the framework of the E-Colleg project [1]. It integrates design tools residing on remote servers and provides access to shared design data.

The microprocessors under concern were modeled in VHDL using simulation and synthesis tools residing on Solaris platform located in our Intranet. Next, they were implemented using implementation and configuration tools residing on a Windows NT machine with an Internet connection. Testing was done using binary test files generated with the use of a configuration tool encapsulated in the second workflow, which is located in one of our partners' site. Owing to the workflow management system, results of design steps were automatically transferred between the workflows as required.

The remainder of our paper is organized as follows. The second section briefly presents selected aspects of the workflow technology applied to distributed design, where the characteristics of workflows significant to electronic engineers are exposed. The third section provides a general view at the Internet-based distributed IP design process, with emphasis on specific elements of microprocessor designs. The consecutive section describes a detailed fragment of the ASTAI®-based workflow that enables the interoperability of common design tasks (e.g. VHDL coding, synthesis, verification) that are represented as parallel and serial sets of the workflow activities in a flexible way. The last section contains some concluding remarks.

2. The Workflow Technology

In order to enable the collaborative working on a distributed design in EDA domain, interoperability and integration of many tools and services (e.g. word processors, simulators, logic and behavioral synthesis tools, physical design tools, test equipment, plotting tools, design component libraries and repositories, project management tools) have to be assured. A workflow organized in the context of an IT system to provide the integration and interoperability of those tools and services is customarily a suitable solution. Being based on the underlying standard Internet protocols, a workflow technology provides an open solution that offers much better chances for interoperability of tools than the past framework concept.

Workflows are defined, managed and executed by workflow management systems. Such systems may be

implemented in many techniques, employ various types of infrastructure, function in different environments ranging from small local to inter-enterprise ones, and involve interaction with many IT applications and tools across system boundaries (Fig. 1).

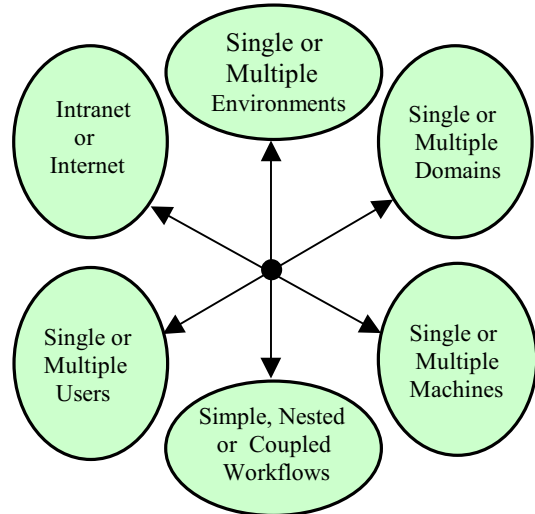


Fig. 1. A range of workflow applications

Workflow management systems provide users with the possibility of defining new *workflow types*, of managing and execution of workflows that have been defined and made available for setting up new design tasks. The term *workflow type* being used in this context refers to *workflow classes* and *workflow instances*, following object-oriented approach of presentation [2]. A workflow contains *activities* and *data objects*. Both of the notions are understood as instances of activity and data object types in a global type repository for the workflow design. A new definition of a workflow inherits from existing components of earlier workflow definitions. An example illustrating those workflow technology notions in EDA context will be presented in section 3.

Workflow engine [3] is a software service that provides run time execution environment for a workflow instance. The workflow engine is capable to invoke tools in order to activate appropriate applications for the execution of particular activities. Based on the process definition determined by object types and their attributes, and on the *workflow relevant data* (i.e. the data generated or updated by workflow applications), workflow engine interprets the process definition, controls process instances and navigation among process activities within a workflow. Workflow engine also maintains workflow control data and *workflow relevant data*, and supervisors actions for control, administration and audit purposes.

3. The Distributed Design Process

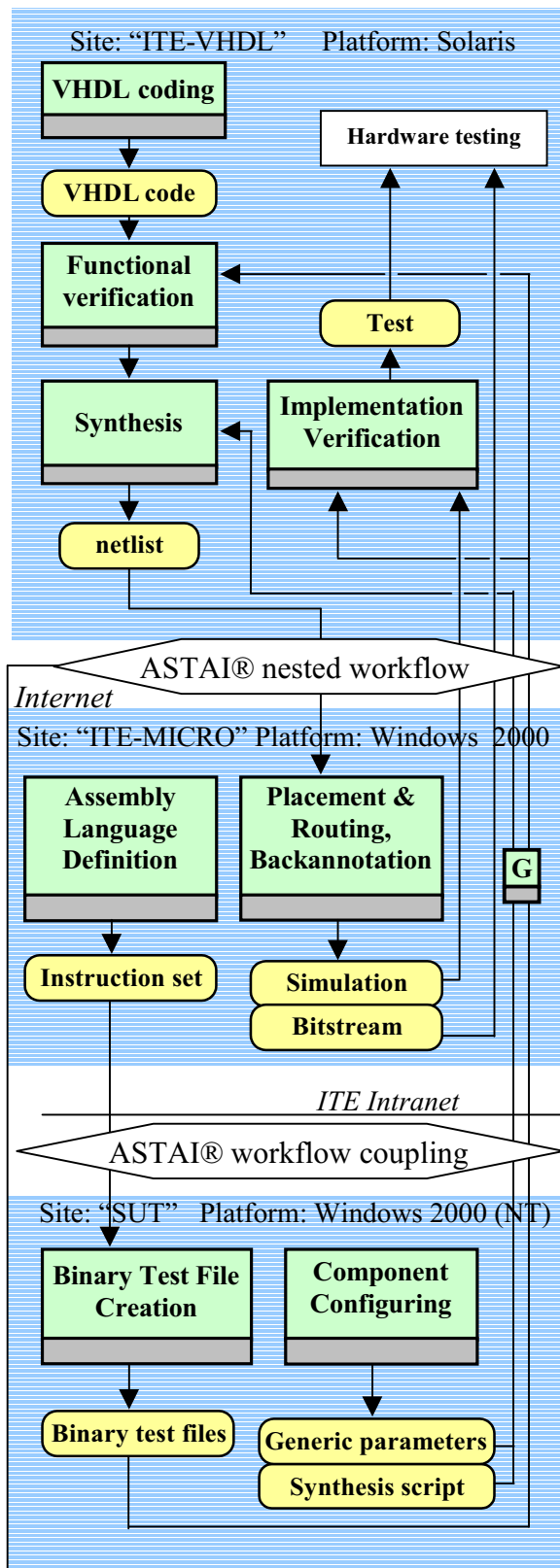


Fig. 2 Simplified workflow for microprocessor design

The design methodology currently applied in our institute (referred afterwards as ITE) is well known among IP providers. Each IP component is modeled at RT or behavioral level in VHDL. The component model is developed as a reusable IP, which is parameterized and configurable. Next, it is verified by simulation. Several VHDL-based tools can be applied at that design stage. After that, the verified model is synthesized in selected FPGA technology. As the result of logic synthesis, a netlist of the component is obtained. The next step is implementation of the netlist into selected type of FPGA device. Implementation verification of the component model is made by simulation of a VHDL model obtained from back-annotation process, and final validation of the IP – by testing the configured FPGA device on an application board using a logic verifier. In the case of design and verification of microprocessors as IP components, we have modified the above mentioned IP design process by adding some specific design steps, namely: assembly language definition, binary test file creation, component parameterization, and synthesis script generation (Fig. 2). All of these specific design steps were supported by the use of an ad hoc virtual component configuration tool named VirComp.

Returning to the point about the workflow technology notions mentioned in section 2, in EDA context, *activity* is understood as a notion related to the execution of tools and services, such as VHDL coding, functional verification, synthesis. On the other hand, *data object* is understood as a notion related to the result of the execution, such as VHDL code, netlist, test pattern. As shown in Fig. 2, activities are represented by rectangles; data objects are represented by oval boxes.

As mentioned earlier, the workflow concept has been applied to the collaborative design and verification of a number of microcontrollers and a digital signal processor. The parameterized and configurable microprocessors were developed to be used in embedded systems. The workflow specification integrates all tools used in mentioned above IP design process. Fig. 2 shows a simplified specification of the distributed design process, schematically divided into three groups, each of them encapsulates a number of design tools residing in different domains, and performs a number of design steps.

The first group encapsulates design tools from Synopsys (VSS, Design Compiler, Behavioral Compiler, Design Analyzer, ...), Mentor Graphics (QuickHDL), and a text editor from Nedit.org (Nedit). All of them reside at ITE on a Solaris platform (Fig. 6). The tools perform following tasks: VHDL coding, functional verification, synthesis, and implementation verification (Fig. 2).

After the synthesis process, obtained FPGA primitive netlist in EDIF and/or XNF is automatically transferred to the second workflow group, which encapsulates design tools from Xilinx (Foundation) and ITE (VirComp); both of them locate on Windows NT platform. The transfer

could be done automatically, due to the fact that both the first group and the second one are encapsulated in the same nested workflow.

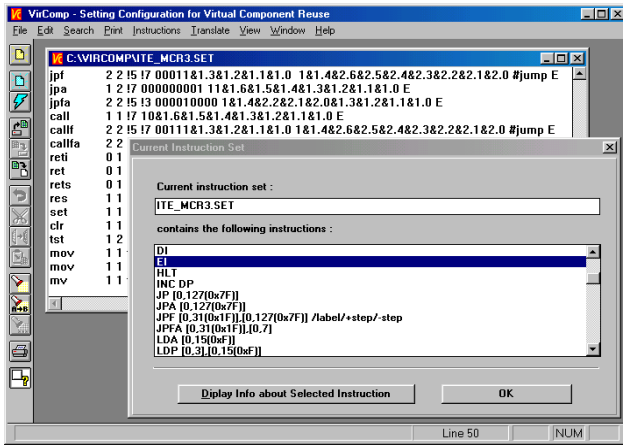


Fig.3. Definition of an assembly language

Based on the FPGA primitive netlist transferred from the Solaris platform via ASTAI® workflow management system, Xilinx Foundation implementation tool produced a simulation netlist for implementation verification, and a configuration file (bitstream) for hardware testing. Next, owing to the nested workflow mechanism, the simulation netlist was automatically sent back to the first workflow group on Solaris platform, where the implementation verification task was carried out in order to generate test pattern for the hardware testing of the microprocessors.

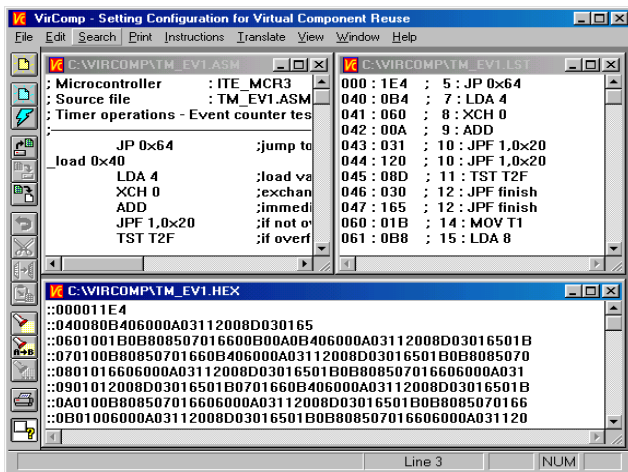
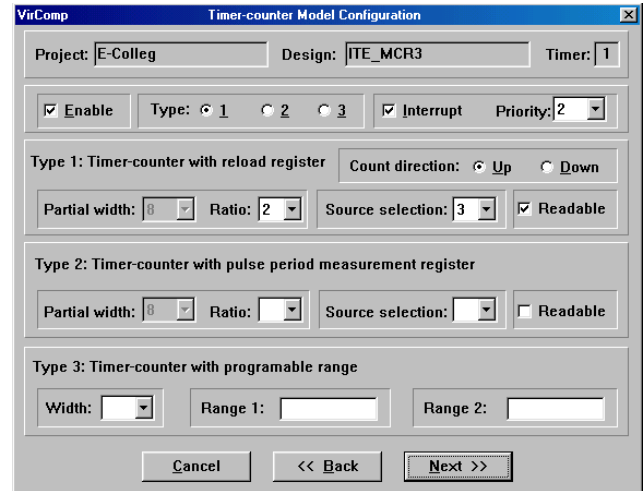


Fig.4. Translation into binary reference test file

On the other hand, the VirComp configuration tool was used to define a configurable assembly language for every developed microprocessor based on its machine codes (Fig. 3). The obtained instruction set was transferred to the

workflow located at our project partner's site at Silesian University of Technology (SUT).

Data transfers between ITE and SUT employed workflow coupling, the mechanism provided by ASTAI® workflow management system that allows cross-domain data flow integration between two or more workflows working in different environments.



```

tpw11_lab: if T1_ENB=1 and T1_INT=1
and T1_TYP = Type_1 const generate
u_tpw1 : tpw generic map(DW, T1_RAT,
T1_DIR, T1_RDB, T1_SLW)
portmap(clk,rst,t1_ie,t1_oe,cl_sel,
fl_sel,fl_clr,cl_src,data_z,t1_ovf,
t1_irf,fl_tst,irq(T1_PRI));
end generate tpw11_lab;

```

Fig. 5 Setting configuration for reuse of a timer

The third group, located at SUT, encapsulates the virtual component configuration tool running on Windows NT platform. Based on the instruction set defined at ITE, the verification team invoked built in VirComp assembler to translate test files written in assembly language into binary reference test files (Fig. 4). As the assembler had been designed with special emphasis on configurability, the designer could modify grammars of assembly languages, for example to add new instructions or to remove some existing ones any time as required. Next tasks performed in SUT's domain were parameterization of previously developed at ITE microprocessor models and generation of appropriate synthesis scripts with the use of VirComp configuration tool . An example in Fig. 5 shows how a user can choose parameters for a timer in order to set up the configuration of a timer-counter unit. Based on the configurability range of actually developed microprocessor, the designer can choose appropriate

parameters of the virtual components that come into the microprocessor model. The configuration tool then checks the chosen parameters for possible errors and reports them to the user. If there is no error found, it translates the information on chosen parameters into an appropriate synthesis script and a VHDL design package.

Results of activities performed in the SUT workflow environment were transferred back to the ITE, also using workflow coupling mechanism. The binary test files were used in the simulation of developed microprocessor models for the purpose of functional verification and implementation verification. On the other hand, the VHDL design package containing generics parameters, as well as the synthesis script, were used by a Synopsys synthesis tool (e.g. Design Compiler in case of microcontroller design, or Behavioral Compiler in case of DSP design).

The small rectangle denoted by character “G” in the second workflow group in Fig. 2 represents a set of activities residing on the ITE machine with Internet connection, which performs the role of a buffer between the first workflow group on Solaris platform residing in ITE Intranet and the third workflow group residing in SUT.

Hardware testing performed at ITE was the last step of the microprocessor design process. That step, however, could not be encapsulated into the workflow, due to the fact that the workflow management system does not support the operating system functioning in the logic verifier.

4. Workflow Encapsulation of Design Tasks

The fragment of the workflow performed at ITE on a Solaris platform, which is responsible for encapsulation of VHDL coding, functional verification, and synthesis processes, is introduced in Fig. 6. The mentioned above fragment is also a workflow itself, then it will be referred afterwards in this section simply as a workflow.

As seen in the figure, the workflow contains successive activities executed during the IP design process at RT level. It integrates three simulation tools: Synopsys VSS, Mentor Graphics QuickHDL, and Model Technology ModelSim, which can be alternately chosen, and two synthesis tools from Synopsys: Design Compiler and FC2.

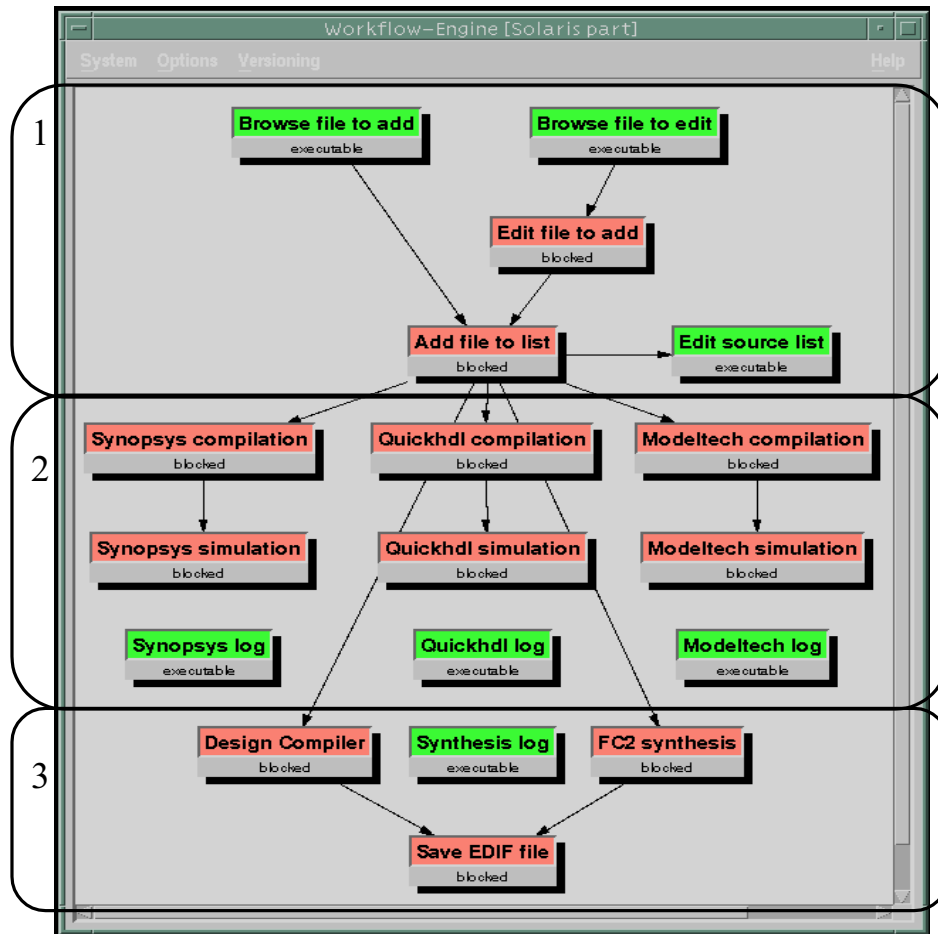


Figure 6. A part of the workflow on a Solaris platform

The first part of the workflow showed in Fig. 6 is responsible for selecting and editing source, library and test files. A designer is given the possibility of creating a new file, choosing an existing one with or without editing it. Moreover, he can add selected files into a file list associated with the workflow. After that, selected files are copied into workflow subdirectory; so each workflow has its own copy of them. Therefore, it is impossible to change source files used in one workflow by another one. The user can change the file position in appropriate file list for the purpose of modifying the order of files to be compiled.

The second part of the workflow is responsible for compilation and simulation of files coming from the file list. As the result of compilation, the whole hierarchy of compiler specific directories including configuration files and the results of the compilation process is created. If the compilation process completes successfully, the designer can start the simulation and view its result. If the simulation is too computing power consuming, he can choose a machine on which the simulation should be done. All messages generated during compilation and simulation processes are redirected to associated log files.

The third part of the workflow is responsible for the synthesis process, which uses the synthesis script coming from the SUT domain. If the synthesis process completes correctly, the user obtains a netlist that can be transferred to XILINX Foundation implementation tool working on Windows NT platform.

The workflow can invoke all design tools from the NT platform, so the user can do all his work using a Windows NT machine. But if he wants to observe the simulation results from a Solaris machine, he should have an appropriate service installed on his Windows NT machine.

5. Conclusions

Collaborative working in multiple-site, multiple-platform environments using traditional Internet-based tools (e.g. ftp, telnet) is time consuming and error prone. The approach based on the workflow concept gives new opportunities for efficient distributed engineering. Improvements can be foreseen in reduction of design cost and design time.

In the paper, we have presented an Internet-based workflow organized in the context of a semiconductor IP design to provide the integration and interoperability of design tools and data. We also described the deployment of the developed workflow in a multiple-site, multiple-platform environment for collaborative design and verification of parameterized and configurable microprocessors to be used in embedded systems.

The work proceeds on refinement and extension of the workflow implementation in terms of encapsulated design tools, data, and services, as well as the work on the definition of quality measures for design processes employing workflow technology are being done.

References

- [1] E-Colleg project home page: <http://www.ecolleg.org>
- [2] C-LAB: ASTAI® Manual v. 2.2.1, Paderborn, 2001, www.c-lab.de
- [3] Workflow Management Coalition, www.wfmc.org.
- [4] Cutkosky, M., Tenenbaum J., Gliksman J.: *Madefast: An Exercise in Collaborative Engineering over the Internet*, ACM Communications, Sept. 1996, vol. 39, no. 9, http://madefast.stanford.edu/ACM_paper.html.
- [5] Kokoszka A., Nguyen Q. T., Siekierska K., Pawlak A., Obrębski D., Ługowski N.: *Distributed Design of Semiconductor IP Based on The Workflow Concept*, Proc. of the 4th IEEE DDECS Workshop, Győr, Hungary, Apr. 18-20, 2001, pp. 299-306.
- [6] Kokoszka A., *The application of a workflow concept in a distributed design process of a digital camera based on the ASTAI® environment*, MS diploma in Polish, Warsaw University of Technology, Oct. 2001.
- [7] Lavana H., Khetawat A., Brglez F., Koźmiński K.: *Executable Workflows: A Paradigm for Collaborative Design on the Internet*, Proc. 34th DAC, 9-13 June 1997, Anaheim.
- [8] Pawlak A., Cellary W., Smirnov A., Warzee X., Willis J.: *Collaborative Engineering Based on Web – How far to go?*, Advances in Information Technologies: The Business Challenges, ROGER, J.-Y. et al. (Eds), IOS Press, 1997.
- [9] University of California at Berkeley, *WELD: Web-based Electronic Design*, <http://www-cad.eecs.berkeley.edu/Respep/Research/weld/index.html>
- [10] Beca L. et al., *TANGO - a Collaborative Environment for the World-Wide Web*, Northeast Parallel Architectures Center, Syracuse, USA, <http://www.npac.syr.edu/tango>
- [11] Józwiak L.: *Modern Concepts of Quality and Their Relationship to Design Reuse and Model Libraries*, Hardware Component Modeling, Kluwer Academic Publisher, 1996.
- [12] Fujimura, A.: *Quality on Time*, Plenary Speech at the 2nd IEEE ISQED, San José, USA, Mar. 28, 2001.
- [13] Bricaud, P. - Keating, M.: *Reuse Methodology Manual for System-on-a-chip Designs*, Kluwer Academic Publishers, 1999.
- [14] Nguyen Q. T., Siekierska K.: *Soft Core Based Model of a Microcomputer Family*, Proc. of the 2nd IEEE ISQED, San José, USA, Mar. 26-28, 2001, IEEE CS Press, CA 2001.
- [15] Synopsys: *Online Documentation for ver. 2001.08*, 2001.
- [16] Xilinx Inc.: *Documentation Foundation Series 3.1i*, 2000.