

SURVEY OF COLLABORATIVE ENGINEERING TECHNOLOGIES IN ELECTRONIC DESIGN AUTOMATION

P. Fraś, T. Kostienko, J. Magiera, A. Pawlak,
P. Penkala, D. Stachańczyk, M. Szlęzak, M. Witczyński
Silesian University of Technology
Institute of Electronics
Akademicka 16, Gliwice, Poland
pawlak@ciel.pl

***Abstract.** Some reflections on the challenges and trends in the selected domains of collaborative engineering, following the experiences of the E-Colleg project (IST-1999-11746), have been presented in the paper. The state-of-the art tools, methods, and practices in collaborative engineering environments, tool integration and description, workflow management systems, and engineering web services have been shortly summarized with the intention to identify main products and tendencies. The conclusions that were drawn, justify the strategy for further development of the TRMS – Tool Registration and Management System, which constitutes the main result of E-Colleg.*

1 Introduction

New, more efficient ways of professional collaboration are strongly required, due to steady growth in complexity of design and manufacturing of electronic systems, and an increasing pressure of time-to-market. Thanks to the improvements in ICT infrastructures, resulting in easier access to the Internet, integration of remote design automation tools, as well as closer collaboration of geographically dispersed engineering teams, are becoming feasible. However, the advanced technology, enabling dispersed engineering collaboration, is not fully used. Currently, it is mainly applied at an intra-company level. Collaboration of engineering teams from different companies still faces various obstacles.

It was the goal of the E-Colleg project to address some of these obstacles by providing a new platform for distributed collaborative engineering. In the kernel of the E-Colleg approach are the unified dynamic Tool Registration and Management Services – TRMS [9], [19]. The E-Colleg project was aimed at the definition of the infrastructure and technology for such services that suited most industrial applications in the field of embedded systems design. Since E-Colleg ended (in Dec. '2003) the work has been continued on TRMS prototype [20] deployment and further development, basing on the analysis of market demands. This paper, as a part of the state-of-the-art analysis, should advance further R&D on TRMS. As the initial studies show, such issues as EDA tool and data description and exchange, customisable collaborative infrastructures based on a workflow concept, remote

tool and data access, dynamic tools registration, and security issues are essential in this respect.

Thus, the paper is structured in the following way. First some issues on different collaborative engineering environments are discussed, next EDA tool integration and description methods are summarised, then some hints about workflow management systems follow, and finally the use of the new promising technology of web services in collaborative engineering is shortly envisioned. In the last section, general conclusions on major tendencies are gathered.

2 Collaborative infrastructures in EDA

There are some collaborative environments intended for EDA domain, which have resulted from the research carried out by various academic groups. The examples of the most relevant solutions are: WELD, REUBEN, ASTAI®, MOSCITO, and JavaCAD. They are shortly characterized below. Other examples of this type of environments are: Nelsis [16], and Odyssey [17].

WELD - Web-based ELectronic system Design is the first and almost classical solution devised for EDA, developed at the Univ. of California at Berkeley. This environment provides a reliable, scalable connection and communication mechanisms for distributed users, tools and services. WELD architecture consists of three main parts: remote servers, network services, and client applications. Remote servers give access to tools invoked from a command-line and described by internal wrappers, or to tools with a built-in support for connections based on socket or WELD protocols. Network services such as a distributed data manager, proxies, and registry services, constitute the system integration layer. Client applications enable utilization of environment network resources. WELD allows dynamic management of available tools. Each time a user needs to run a tool, its location is checked in the registry [12].

REUBEN represents another environment supporting engineering work. It has been developed in Collaborative Benchmarking Laboratory, North Carolina State University. REUBEN is a synchronous/asynchronous environment that consists of OmniDesk and OmniFlow. OmniDesk is a space for a scaleable cooperation of software agents that manage distributed tasks and workflows. A workflow description is based upon cdtML schema (XML extension), and is used for generation of graphical workflow representation in OmniFlow. The description includes all the necessary information concerning data and tools, and relations among them. Communication between system elements is transparent for a client and it is based on the TCP/IP protocol together with sockets [13].

The **ASTAI(R)** environment, developed by the C-Lab (SBS/Univ. of Paderborn, Germany), consists of two types of applications: Tool Servers (TSs) and Client applications. Tool Server enables ASTAI(R) users to share tools. Access to these tools is through the Client application. Communication between elements of the system is based on the CORBA standard, which is one of the reasons why the system is not suitable for networks protected by firewalls or proxies. The ASTAI(R) environment contains sophisticated Workflow Management System (WfMS) which enables automation of tasks and data flows. ASTAI(R) GUI presents a design process as a number of related operations named workflow activities, whose dependencies are controlled by the WfMS. This environment allows distributed engineering teamwork to be easily defined as workflows and sub-workflows respectively. The ASTAI(R) security depends on the configuration of the operating system, as it does not contain any security mechanisms of its own. ASTAI(R)

neither encrypts data transmission, nor directly supports control of user activities and access to shared tools [14].

MOSCITO is an Internet-based multi-agent system which can be controlled and observed by the user's front-end program – the MOSCITO desktop or a web client. MOSCITO consists of three software layers: kernel layer, interface layer, and user extensions. The kernel provides functionality for basic object and data management, file handling, XML processing and communication. The Interface layer provides programming interfaces for integrating new tools, workflows and appropriate viewers. Each interface is represented by a Java class, which contains basic functionality. Communication between tools (agents) is organized as a socket-based solution. MOSCITO does not support data transport encryption and proxy servers [15].

JavaCAD is an EDA environment which employs object-oriented features of Java to describe the component based design semantics of modern digital systems. In this system Java is used as a hardware description language, allowing an abstract specification of hardware and software blocks. JavaCAD enables fuzzy matching and protection of Intellectual Property (IP) components, hierarchical design, and multiple levels of abstraction. JavaCAD utilises a client-server architecture where the IP user is a client, and the IP provider is a server. The environment is based on JavaSoft Remote Method Invocation (RMI). Additionally, it enables distributed and parallel circuit simulation [6].

It should be noticed that, despite many research efforts, widely accepted tool integration technology is still not available. Since the most known EDA CAD vendors rather concentrate attention on market position of their EDA environments, the standard integration technology may come from organisations animating industrial standardisation efforts, like Si2 (<http://www.si2.org>). In fact, there is some hope for tool integration standard in the electronic engineering community, due to the recent OpenAccess initiative.

3 Design tool description and integration

Many companies intend to use remote tools over the network to increase their design efficiency and reduce costs. The development and proliferation of the Internet-based techniques for integration of dispersed tools provide new possibilities in this domain.

A tool description should give information on the tool, its parameters and a manner in which these parameters are to be used for tool invocation. A correctly created description of the tool should enable an arbitrary connection of this tool to a workflow, i.e. a group of tasks. Communication with the tool should be based on input and output interfaces. Usually, a part of an interface is realized by command line arguments, environment variables, file data or standard input and output (information about an error code, etc.). A minimum required from the interface is delivering information about the data necessary for an action, and retrieving results and passing them to the workflow system.

Following the above characteristics, selected examples of tool description techniques are presented.

Historically, the first proposal for a standardized tool description was **TES (Tool Encapsulation Specification)** [7]. TES originated from CFI - CAD Framework Initiative, which was the consortium established to support EDA community in coordination of standardisation activities in the domain. A TES description can be characterized as a rather complicated structure which requires dedicated processing application. It is difficult to

transform it into other formats. ASTAI® is an example of an environment using the TES standard.

Open and Dynamic User Interface is an environment for invocation of educational tools in intranets and on the Internet. The environment uses its own tool description format. A tool description contains only basic information necessary to invoke the tool. A disadvantage of the description is a lack of possibility of exporting it to other environments [8].

HP Service Control Manager (SCM) environment is developed by HP [10]. The SCM enables management of 'nodes' and tool invocation on different systems and platforms. The tools described in SCM are divided into three groups:

- Tools invoked in a command line,
- Scripts administering 'nodes' and execution,
- Tools working in a graphical environment (inconvenient for simultaneous work).

This description has the advantage of making data exchange and transformation of descriptions to another format possible.

TRMS-ANTS constitutes a preliminary E-Colleg environment that allows invocation of tools residing behind a firewall [9]. Information about available tools is incorporated in two files. The first file contains information about locally accessible tools, while the second one, information on tools accessible on other servers. The drawback of the solution is the necessity of control and actualization of all files on each server in case of adding or removal of any tool. These files deliver all data and additional parameters. Such a solution requires the use of additional tools (scripts analyzing files with parameters) to extract the parameter value from the file before tool invocation.

The **TRMS** environment, as already mentioned, is the main result of the E-Colleg project. The environment enables management of users and tools. TRMS uses a special XML format (E-Colleg Markup Language – ecollegML) for tool and task description. The used description enables easy tool search and easy transformation of a description to another format (tool description reuse). The transformation uses XSLT schemas. The most important elements of this description are: 'description', 'parameters', input and output. 'Description' element contains information that identifies described tool such as id, keywords, info, etc. 'Parameters' include extensions such as prefix, possible parameter selection (e.g. Verilog, VHDL) and file extensions (e.g. zip). All the descriptions have unique ID that identifies tool, task or workflow [11].

4 Workflow management systems

The workflow management technology has not gained enough popularity with the EDA community yet, though it seems that EDA designers can greatly improve their work using WfMSs (Workflow Management Systems). Since the fundamental idea of workflow is based upon process automation, workflow users can have better control of the design processes. Due to design process automation, work efficiency should be increased. This is achieved during workflow planning when some actions, which so far required user interaction, now can be omitted in the workflow. Besides, certain activities can be conducted within the WfMS architecture. Examples of these are: copying files or documents to specific network locations, report generation in case of WfMS use, and others.

Information management integrated in WfMS simplifies communication between designers. Synchronous communication serves exchange of remarks in real time, and asynchronous communication is helpful in case of contacting people working in different time zones. A central repository creates a possibility of data storage and usage in a proper moment. Coupling the repository with a version control enables access to data during different design phases. Additionally, CVS (Concurrent Versions System) allows creation of a project tree, in which the subsequently developed file versions are in the main branch (HEAD), and usage of side branches (BRANCH) allows creation of a parallel project.

Some WfMSs facilitate access to project resources via a web browser. Thus, a designer is not limited to work on his own PC in his office only. Of course, in this case, security issues are crucial. Another quality, which WfMS should possess, and it often does, is that it can facilitate data access through mobile devices. Thus, designers can retrieve information concerning the project even if they are temporarily outside their office.

In EDA processes many various tools are used. Very often these tools work under different operational systems (Windows, Linux, Unix), and so they require different hardware architectures (i386, Sparc, HP, etc.). Moreover, during the design process, a huge amount of data has to be managed and processed. With the currently existing data formats it is cumbersome to exchange data among different EDA systems, and to combine diverse tools into a complex, unified workflow.

Nevertheless, software market is persistently developing. Many software producers add to their products workflow functionality, like: SAP, Oracle Workflow, MAXPLUS+. Others provide a tool access via a web browser, e.g., Action, Ultimus, Viewstar, and W4.

Diversity of software products and solutions calls for standards in this area. Many producers try to introduce their own solutions as, de facto, standards which are often used by other parties. The most important solutions are:

- XPLD - XML Process Definition Language,
- BPEL4WS (Business Process Execution Language for Web Services) IBM
- WSFL (Web Services Flow Language) IBM,
- XLANG, XLANG (Web Services for Business Process Design). Microsoft
- WSCI Web Service Choreography Interface (WSCI), Sun, BEA, SAP, Intalio
- DMA (<http://DMAtech.info>) – The Document Management Alliance
- ODMA (<http://ODMA.info>) - The Open Document Management API

The main standardization group is the Workflow Management Coalition (<http://wfmc.org/>). It is a non-profit organization, consisting of almost 300 members that are workflow vendors, users, analysts and university/research groups. The Coalition's mission is to promote and develop the use of workflows through the establishment of standards for software terminology, interoperability and connectivity between workflow products.

Another interesting initiative is workflow patterns development that has been active since 1999 (<http://www.workflowpatterns.com>). According to its animators, workflow patterns modelling can be used as a standard for WfMSs. Many producers use this solution (COSA, FLOWer, Domino Workflow, Eastman, Visual Workflow, Forte Conductor, Meteor, Mobile, MQSeries/Workflow, Staffware, Verve Workflow, I-Flow, InConcert, Changengine, and SAP R/3 Workflow).

Adoption of the workflow technology in EDA can have many advantages, like remote tool invocation and dispersed design teams integration. The workflow technology, due to its specificity, is particularly useful for situations of repetitive processes. It must be designer friendly, as EDA design processes are often difficult to describe.

When tasks are performed once, the workflow concept is not very useful, because an amount of work done during the workflow definition can exceed benefits gained. Also

development of a new WfMS needs very profound analysis and justification, as many tools incorporate, at least partially, some sort of workflows. This is also true for EDA tools that are incorporated in Integrated Design Environments (IDE). For the majority of designers, IDE presents a sufficient work environment.

5 Web services

There are various definitions of web services. What they have in common is the use of XML (Extensible Markup Language) through SOAP (Simple Object Access Protocol). There are other standards related to web services, but it seems that they do not need to be incorporated in a base definition. Thus, a web service is a software system located on the Internet, whose public interfaces are defined and described using XML. These network available interfaces allow access to the core application code, using standard Internet technologies (XML, HTTP, SMTP, SOAP). In this way, a web service provides an abstraction layer between the application client and the application code, separating the platform from programming-language-specific details of how the application code is actually invoked. This standardized layer means that any language that supports the web service can access the application's functionality (Figure 1) [1].

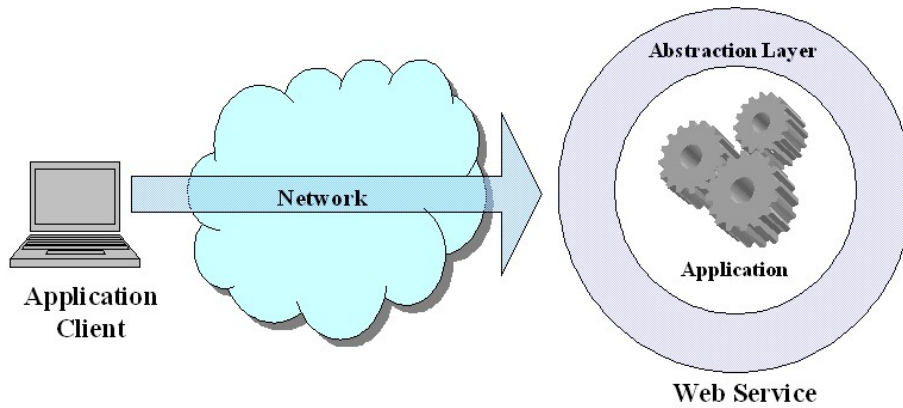


Figure 1: Access to application code through the web service abstraction layer [1].

The basic web services architecture defines an interaction between software agents as an exchange of messages between service consumer, service provider and service registry. The service provider publishes a description of the services it offers via the service registry. The service consumer searches the service registry to find a service that meets its needs and then requests the service from the service provider side. The service consumer could be a person or a program (Figure 2) [2].

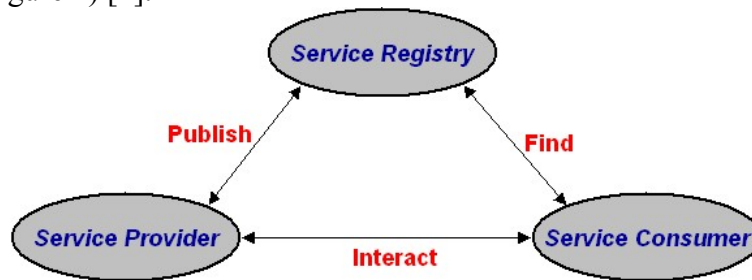


Figure 2: Web services basic architecture.

TRMS is based upon the same architectural concept. In this way, TRMS is the Java realization of web services, dedicated to electronic engineering. However, it differs from the web services described by W3C organization [5]. The difference between them is that ordinary web services are more standardized and employ such common technologies, like: WSDL (Web Services Description Language), and UDDI (Universal Description, Discovery and Integration). WSDL is an XML technology that describes the interface of a web service. It standardizes the way a web service represents the input and output parameters of an invocation externally, the function's structure, the nature of the invocation (in only, in/out, etc.), and the service's protocol binding. WSDL allows distant clients to automatically understand how to interact with a web service. UDDI provides a worldwide registry of web services for advertisement, discovery, and integration purposes. Business analysts and technologists use UDDI to discover available web services by searching for names, identifiers, categories, or the specifications implemented by the web service [2]. The similarity between web services and TRMS is that XML and SOAP are used as a packaging format for transporting XML documents over the network by means of standard Internet protocols – HTTP in case of TRMS and SMTP, HTTP, and FTP in case of regular web services. By having a standard transport mechanism, heterogeneous clients and servers can become interoperable.

Web services include standards, which are their advantage, but they have also certain limitations, especially in perspective of their utilization in EDA. While TRMS is rather a web service system designed for engineering applications, the standard web services are designed for business processes and applications. Though they might be applied in the area of engineering tools, not much has been done in this field, so far. Lack of centralised management of resources, necessary for platform integration inside enterprise networks, is another drawback of web services, as compared to TRMS. Although web services can be built of other web services, thus making creation of more complicated workflows possible, it seems that, as yet, it would be quite difficult to realize complex engineering design tasks or flows.

The web services limitations we have pointed out should be considered as a challenge rather than obstacles not to overcome. Web services seem to be promising in respect of successful interoperation of any number of applications or application components between organizations in a platform-neutral, language-neutral fashion. Such interoperation will bring heterogeneity to the world of distributed computing once and for all [2]. For that reason this technology is also very promising for deployment in Collaborative Engineering for EDA. Engineering tools for SoC (System-on-Chip) modelling, simulation, verification, etc. can be described with WSDL, and encapsulated in a web service, which will make them remotely accessible via web service interface and UDDI registry. The first trials in this field have been already accomplished [3].

To support collaborative distributed SoC design, the idea of web services must be completed with clear, widely accepted way for secure, reliable transactions and orchestration, i.e. platform integration. Some additional functionality for data, and, especially IP (Intellectual Property) components sharing, exchange and trading are also needed. An advantage this technology is supposed to have is its capability of integrating engineering tools as web services into complex design flows, which is essential for the collaborative engineering concept [4]. If all these challenges are successfully met, the web services technology could enhance collaboration among electronic industry actors, supporting the IP-based SoC design and distributed EDA performance.

5 Conclusions and future work

The prime motivation of the E-Colleg project was to develop a prototype of the collaborative infrastructure to support SMEs in integration into a value-added chain of complex design processes, like the one for SoC (System-on-Chip). With the TRMS prototype being now available, along with the first deployments of this prototype, the TRMS development team analyses strategies for further R&D of the infrastructure. This paper monitors some of the results of this analysis in respect of tool description and integration, workflow management and web services.

The analysis has already helped to improve the ecollegML-based tool description used in TRMS. This results in an easier search for design tools, due to improved identification of tool descriptions, and simplified extending of tool descriptions, as well as their transformation to another formats.

It has been observed that remote tool invocation is becoming a widely accepted technology with new methods of tool invocation and its standard XML-based reusable description. It may improve efficiency and quality of a design process due to access to the best available over the network tools, but also due to a simplified search for required and available tools. With the systematic increase of the broadband and secure access to tools the remote tools invocation will be more and more accepted in industry.

Further development of collaborative engineering environments (including TRMS) should be based on common standards like, XML and SOAP because of flexibility of solutions, implementation easiness, and possibility of data exchange through firewalls. Adoption of the workflow technology and other web services related standards can give many additional advantages, like seamless remote tool invocation and dispersed design teams integration. The workflow technology is devised for situations in which repetitive processes are used, which is often a case in electronic system design.

References

- [1] Tidwell, D., Snell, J., Kulchenko, P.: Programming Web Services with SOAP, O'Reilly, ISBN: 0-596-00095-2, December 2001,.
- [2] Chappell, D., Jewell, T.: Java Web Services, O'Reilly, ISBN: 0-596-00269-6, March 2002.
- [3] Eikerling, H-J., Thronicke, W., Bublitz S.: Provision and Integration of EDA Web-Services Using WSDL-based Markup, Forum on Design Languages, Lyon 2001.
- [4] Rettberg, A., Thronicke, W.: Embedded System Design based on Webservices, Proceedings of the Design Automation and Test in Europe Conference (DATE), Paris, March 2002.
- [5] W3C recommendations, notes, and drafts – <http://www.w3.org/>
- [6] Dalpasso, Bogliolo, Benini, Specification and validation of distributed IP-based designs with JavaCAD, DATE, 1999
- [7] Manual: Tool Integration With TES, <http://www.pb.izm.fhg.de/lip/manual/tes.htm>
- [8] Priidu Paomets, An Open and Dynamic User Interface to the CAD-system, Tallinn May 1998
- [9] Schattkowsky, T., Müller, W., Distributed Engineering Environment For The Design Of Electronic Systems, CCE'03, Poznań, April 2003
- [10] HP Servicecontrol Manager Overview & Features, <http://www.software.hp.com/scmgr>
- [11] Szlęzak, M.: XML-based tool description and integration, IWCIT, Gliwice, September 2003
- [12] Chan F., Spiller M., Newton R.: WELD- An Environment for Web-based Electronic Design, Proc. DAC 1998, San Francisco.
- [13] Lavana, H.: A: Universally Configurable Architecture for Taskflow-Oriented Design of a Distributed Collaborative Computing Environment, Thesis PhD, 2000

- [14] ASTAI(R) home page, <http://www.c-lab.de/astair>
- [15] Schneider, A., Ivask E.: Internet-based collaborative system design using Moscito, Workshop on Challenges in Collaborative Engineering (CCE'03), 15th-16th, April 2003, Poznań, Poland.
- [16] The Nelsis CAD Framework home age, <http://www.ddtc.dimes.tudelft.nl/nelsis>
- [17] The Odyssey home page, <http://www.eecs.umich.edu/odyssey/>
- [18] Witczynski, M., Pawlak, A., Kostienko T., Fraś P., Szlezak, M., Magiera, J.: Tool management as a key integration feature for an engineering design virtual organisation, Proc. 9th Int. Conf. of Concurrent Enterprising, Espoo, Finland, 16-18 June 2003
- [19] Kostienko, T., Pawlak, A., Szlezak, M., Fraś, P., Magiera, J., Witczyński, M.: Development of TRMS/GTLS - Global Tool Lookup Services. Challenges in Collaborative Engineering, E-Colleg Workshop accompanying DDECS'03 IEEE Workshop, Poznań, June 2003.
- [20] P. Frasz, T. Kostienko, J. Magiera, A. Pawlak, P. Penkala, D. Stachanczyk, M. Szlezak, M. Witczynski, TRMS Deployment in Distributed Engineering Applications, Proc. CCE'04, Tatranska Lomnica, Slovakia, April 18-21 2004.