

# WEB SERVICES-BASED COLLABORATIVE SYSTEM FOR DISTRIBUTED ENGINEERING

---

A. Pawlak, P. Fraś, P. Penkala  
*Institute of Electronics  
Silesian University of Technology  
Gliwice, POLAND  
Adam.Pawlak@polsl.pl*

*Design of complex electronic systems requires often collaboration of engineers who work in remote locations. This collaboration enabled by the network needs to be supported by seamless and secure integration of distributed design tools.*

*The paper presents a web services-based environment enabling integration of distributed design tools. It is based on TRMS (Tool Registration and Management Services) which are scalable, easily accessible, secure, and available on different platforms due to their implementation in Java.*

## 1. INTRODUCTION

Collaborative engineering for distributed product development is a new paradigm of engineering work that has become feasible due to dynamic progress in information and communication technologies. This new engineering paradigm can be enabled with innovative infrastructures, collaborative working environments, net-aware engineering tools, and new design methodologies based on re-use of globally accessible resources, as well as application of advanced security of network operations and distributed tools management.

Many research efforts were directed during recent years towards new infrastructures and collaborative working environments (CWE, 2006), like projects CoSpaces, and eCoSpace. New design methodologies, like SoC (System-on-a-Chip) design based on globally accessible IP components, are proliferating into engineering practices (Brglez, 2001), (Saucier, 2007).

The paper addresses the selected topic of collaborative engineering, namely integration of distributed engineering tools that is relevant for distributed CWES. The authors with the approach presented in the paper aim at supporting SMEs that operate in collaborative networks. Available for SMEs distributed collaborative engineering environments do not support adequately integration of tools of dispersed design groups, and do not consider distance-spanning related issues, like: firewalls, security (including user authorization, data and transfer encrypting), distributed inter-organization workflows, and remote administration of users and tools.

Although these problems have been partially solved in large global enterprises that are able to pass designs between their branches around the globe, SMEs still have major problems in entering outsourced design contracts due to a lack of affordable collaborative infrastructures.

The paper presents the new web services-based realisation of Tool Registration and Management Services (TRMS). TRMS has originally been developed within the EU E-Colleg project ([www.ecolleg.org](http://www.ecolleg.org)) (Bauer, 2001) (Schattkowsky, 2004). The solution presented in the paper is a result of R&D work conducted during the EU MAPPER project ([mapper.eu.org](http://mapper.eu.org)) (Johnsen, 2007). The new TRMS environment (also referred to as TRMS2), offers to dispersed engineers some innovative features that enhance existing distributed collaborative engineering infrastructures with new services. TRMS has been originally developed as a stand alone application (Fraś, 2004) but as a result of deployment experiments, and especially due to new requirements concerning its interoperability with other collaborative environments, its architecture has been based on web services.

The paper presents requirements, justification and an outline of the new architecture of TRMS based on web services, as well as its application in integration of engineering tools for electronic systems design. This work constitutes a part of the MAPPER collaborative framework (Johnsen, 2007) that comprises also other relevant for inter-enterprise engineering collaboration elements, like: collaborative workspace used in our applications for supporting design specification refinement (CURE, 2008), engineering use of visual knowledge models (Pawlak, 2007), and participatory design methodologies (Johnsen, 2007). Other important for TRMS issues, like the use of ontologies for tools integration have been presented elsewhere (Szlęzak, 2007).

The paper is organized as follows. Firstly, new requirements on TRMS are shortly explained; secondly, the overall architecture and TRMS services-based functionality are described. Finally, we conclude on the experiences relevant to the web services-based architecture of TRMS deployed to a distributed design of an electronic component.

## 2. CHALLENGES AND REQUIREMENTS

### 2.1 Challenges in collaborative design

Current trends in global engineering (Radeke, 1998), mass-customisation and increased competition are leading industry towards collaborative networks of design engineering and manufacturing organisations. Increasingly, products must be rapidly adapted to customer needs, leading to faster innovation cycles and more complex collaborative engineering. The core problems and challenges in the area of faster and more flexible design and manufacturing concern:

- Concurrency in all operations, increasing efficiency and decreasing time-to-market.
- Quick and inexpensive formation of networked design organisations.
- Processes and products that can be rapidly reconfigured to accommodate diverse and changing needs and opportunities. Collaborative product, process

and service engineering must thus be managed and performed across networked organisations.

- Integration of tools of remote groups of engineers isn't supported adequately in terms of: security, distributed inter-organization workflows, and remote administration of users and tools.

Design engineering in collaborative networks constitutes a new engineering paradigm that needs to be supported with new design methodologies, tools and practices. Although collaborative engineering has been advocated for already more than a decade, in practice, it remains typically restricted to engineering groups from large global companies, but even there support for collaboration between dispersed company's sites proves to be limited.

It has been observed (Ranta, 1999) that automobile industry was usually a source of innovation in terms of new production paradigms and the best practices in manufacturing industry. However in categories of management and control of demand-supply chain and industrial structures, electronic industry sets new paradigms through creation of global collaborative (engineering) networks (Salminen, 1997). This enables the partner companies to compete against others in terms of time to market, product flexibility and shared knowledge.

## 2.2 Requirements

The main requirements on TRMS have been formulated already in (Fraś, 2004). During R&D work conducted in MAPPER and TRMS deployments in distributed design scenarios, e.g. (Fraś, Kostienko, 2004) and (Siekierska, 2006) additional requirements have been identified and formulated. Below they are enumerated and shortly explained:

- services supporting interoperability with advanced collaborative environments
- access to particular TRMS services,
- support for secure work in the networks with NAT and firewalls,
- support for long task execution and operation tolerating Internet connection breaks,
- simple client application for common users, and
- multi-user access to invoked tasks.

As original TRMS has been developed as a stand-alone application its integration with other collaborative systems, like workspaces was a demanding task. With web services-based architecture selected services are available from outside and the system is easily integrateable using common services platforms.

Access to a particular service like user authentication is very useful in larger collaborative environments with numerous components, e.g. synchronous and asynchronous communication, and collaborative workspaces. If user authentication service can be shared among system components, the user will need to be verified once only. Further examples of use of TRMS services that should be accessed by external CE components are: tool invocation, tasks and workflows management, as well as users' management.

Secure work in collaborative environments is a must in industrial settings that use firewall systems and filters which control contents of transferred data. Network

address translation systems (NAT) constitute an additional obstacle for a direct communication with machines of an internal network.

Some remote tasks, like simulation of certain components may take long times (hours or even days). Thus, supporting of such practical cases by letting a user to initiate the remote task only without a need to maintain the connection is very useful.

Supporting users with various versions of clients depending on their role, needs and experiences simplifies the use of the system.

As tasks are often shared among dispersed users, e.g. designers working in different locations, it is very useful that each of them may monitor and control their progress

### **3. TRMS ARCHITECTURE**

A general architecture of TRMS comprises three basic components: Global Tool Lookup Server (GTLS), Tool Server (TS), and the Client application.

GTLS manages the TRMS environment whereas Tool Servers enable remote tool invocation. The Client application allows a user to control the environment and tool invocation. Each instance of the TRMS environment must comprise one GTLS server and an arbitrary number of Tool Servers and Client Applications. A quantity of Tool Servers depends on a number of installed tools that are expected to be accessible over the network. Each computer running such a tool needs an operational Tool Server.

The TRMS environment uses for communication the standard https or http protocols. Communication among components of the environment is synchronous and conducted always in a direction „to GTLS”. The GTLS server never initiates alone a connection to a Tool Server, neither to a user who uses a Client Application. The connection is initiated by the Client or TS. Tool invocation on TS is done indirectly. Firstly, a user sends a request to GTLS and from there an appropriate Tool Server takes information on its task. This mechanism allows usage of the environment in networks protected by firewall systems or using NAT. In such networks the Client application and the Tool Server operate from a security perspective, like a common browser, i.e. invoke a remote resource on a server beyond the protected network.

#### **3.1 GTLS – Global Tool Lookup Server**

The main component of TRMS is the GTLS server. It is responsible for management of elements of the environment. GTLS is also responsible for the security policy of the whole environment, registration of user activities and of access to tools, maintaining statistics, identification of an intruder attack. GTLS, as the only component of the TRMS environment should be accessible from the Internet. In such case, GTLS plays a role of a broker and a temporary repository in a communication between a Client Application and Tool Servers.

A relational data base is used as the main data repository for GTLS. DB contains information on users and their privileges, TSs, registered tools, and workflows.

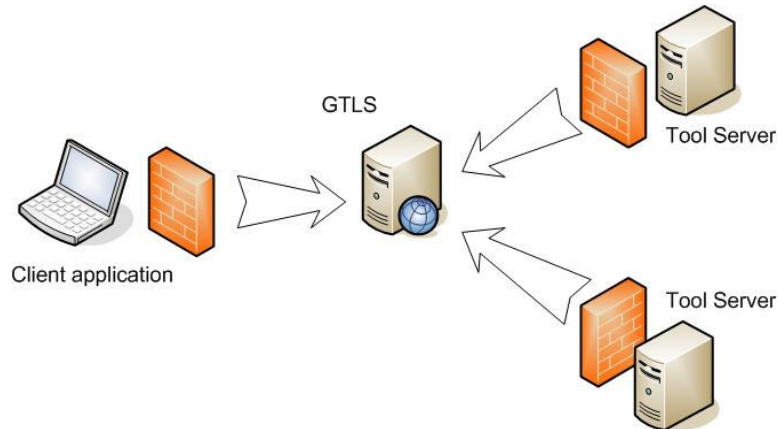


Figure 1 - General architecture of the TRMS environment

GTLS has been implemented as a set of web services (Apache AXIS). Selected web services have been shortly characterised below.

#### **Administration**

The administration services are responsible for registration and modification of data on users and their privileges, elements of the system, as well as, information on accessible tools and machines that make them available.

#### **User and Server Authentication**

Login to GTLS is the first task a user is expected to do upon invocation of the Client Application. User authentication is performed then. As soon as the user (designer) logs in, a new session is created and the user receives its session key which later on is used for authorisation. Tool Servers are authenticated automatically upon their invocation.

#### **Task Management**

Each tool that is expected to be accessible over the network needs to be registered and placed in the task queue. Registration involves determination of necessary data for tool invocation, like: tool name, complete path to tool location, and information on the Tool Server on which the tool resides (as well as additional data on the tool: its description, author, creation date). Registration can be conducted by a user with appropriate privileges. The task queue constitutes a set of refined task instances that comprise in addition to information on the tool, also data on the particular refinement (status and information on messages that result from tool invocation, like *std output*, *std error*).

#### **Workflow Management**

A workflow constitutes a set of tasks that are in the task queue. The current implementation supports sequential workflows, i.e. a new task is invoked if the

previous one has been completed. It is the GTLS server that controls a sequence of tasks invocations.

### 3.2 Tool Server

The Tool Server (TS) is responsible for controlling users' access to tools. A client invoking a tool does it through the Tool Server. The Tool Server allows for sharing of the particular tool through the Internet. Its additional task is brokerage in user authentication. The Tool Server communicates with GTLS in order to update its task queue. If a new task has been added to the queue then TS retrieves all required for tool invocation (e.g., parameters, input data) data. The task is executed in the following, and all input data are returned. As soon as, the tool completes its work, all result data are sent back to GTLS, where they wait for retrieval by the Client Application.

It is not necessary that TS remains available all the time, although this is preferred. Once a user tries to invoke a tool on TS that is off-line, information on this trial remains on GTLS until this Tool Server becomes available again. The Internet access isn't required as well during the tool invocation. In such case, messages generated by the invoked tool are buffered by the Tool Server, and with the subsequent connection to the network will be transmitted to GTLS.

### 3.3 Client application

The Client application has a simple GUI that enables login to the system, its administration and usage of available tools. It enables a full control over the environment, management of users and tasks. Each task needs to be registered by the Client application. Registration of a workflow will be possible as well. Upon an invocation by a user, information on a particular task is placed in a task queue on GTLS and awaits there for its retrieval by an appropriate Tool Server.

Figure 2 presents the client application with a workflow comprising six design tasks that are performed by an engineer during design task realisation. These design tasks are related to: service of design data repository, compilation, simulation and synthesis. They are explained in the following section.

GUI enables invocation and termination of a task, or even a complete workflow. Once a workflow is invoked, subsequent tasks are invoked automatically by GTLS. The application enables monitoring of each invoked task status. Messages that are generated by the invoked tool (*std output*), as well as error messages (*std error*) are available as well.

Many users may concurrently control a particular workflow. They have access to all information on executed tasks and may manage all tasks. A user may anytime switch off the client application without losing any data, as all information on the invoked task and messages generated by the task are buffered on the GTLS server. Upon a new login step the client application will fetch from GTLS all data that appeared when a user wasn't present in the system.

An applet version that may be invoked through a browser has been developed as well.

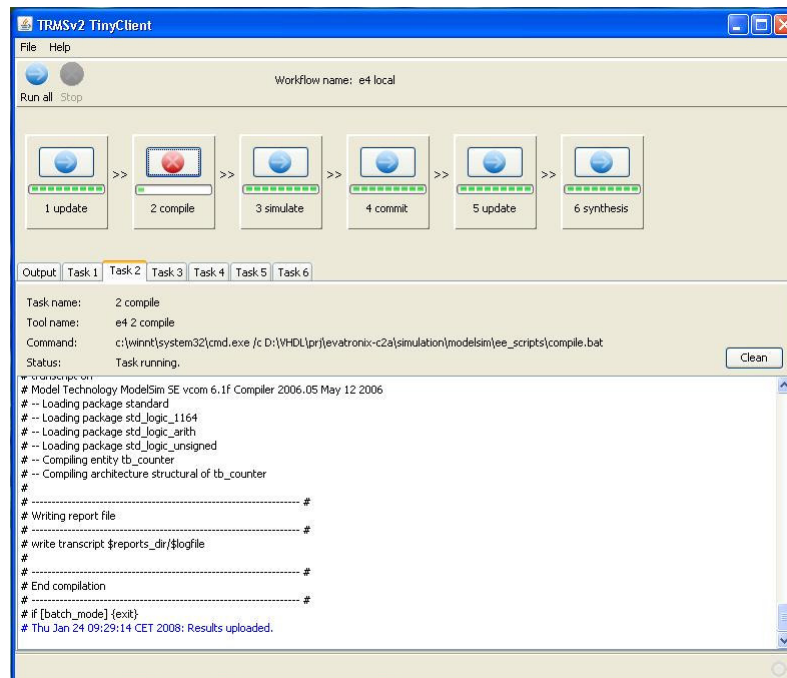


Figure 2 - Client application

### 3.4 Security measures

TRMS enables secure data transfer with authentication and authorization of users, as well as it includes security management mechanisms that allow an administrator to monitor users' activity and to execute a proper security policy. Invocation of each web service requires user authorisation which is realised using a session key. TRMS uses as default the HTTP SSL channel for communication between components what assures confidentiality of transferred data. Use of the standard Internet protocol is useful for networks protected by firewalls, e.g. corporate networks.

## 4. DEPLOYMENT OF TRMS

An experiment on distributed collaborative design and verification between two remote branches of one company – IP (virtual) electronic component design house has been performed as a part of the MAPPER project activities. It comprised the following tasks: *code development*, *synthesis*, *change notification*, *FPGA implementation*, *error handling*, and *notification of task completion*.

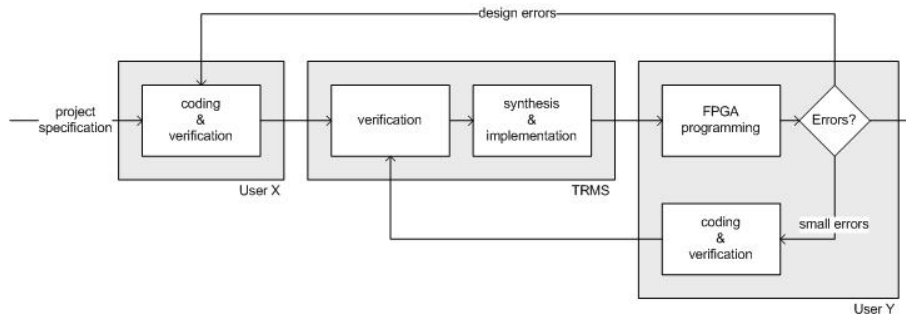


Figure 3 - Distributed design and verification flow

**Code development:** A designer X in Gliwice location prepares a RTL code and synthesis scripts using the text editor. The files are uploaded into CVS (using a local CVS client).

**Synthesis:** The designer X in Gliwice launches the following tasks in TRMS:

- Upload of the source code from the CVS repository to the remote machine where the synthesis tool is installed.
- Execution of compilation and simulation of a design and subsequently the synthesis script on the remote machine. The synthesis tool is installed in location Gliwice on the machine other than the designer's one with the TRMS Tool Server running. The designer can control the progress of the synthesis process (it can be stopped any time, e.g., when a designer notices that something goes wrong).
- Commitment of the synthesis result files into the CVS repository.

**Change notification:** The designer X in location Gliwice and a designer Y in location Bielsko get an e-mail notification about changes in the CVS repository. In addition, any designer (from Gliwice or Bielsko) can log into the TRMS system and check the status of the synthesis process.

**FPGA implementation:** The designer Y in location Bielsko downloads from the CVS repository (using a local CVS client) the implementation results needed for programming an FPGA chip.

**Error handling:** In case of design or implementation errors one can identify the following cases:

a) Errors that can be handled by the engineer in Bielsko (e.g., FPGA pin assignment):

In this case the engineer in Bielsko fixes these errors, uploads modified files to the CVS repository and uses the TRMS remotely to re-run synthesis on the machine located in Gliwice (the tasks described in synthesis).

b) Errors that have to be handled by engineers in Gliwice (e.g., more serious design bugs):

The engineer in Bielsko notifies the engineer in Gliwice that a serious error has been found, and from now we return to code development.



**Notification of task completion:** If there are no new bugs the design verification is completed.

The experiment has demonstrated that TRMS can accelerate designers' activities in case of sequential operations. A workflow solution enables automation of selected design tasks. Another advantage of using TRMS is the ability provided to engineers located in Bielsko to invoke tools which are installed in the Gliwice branch of the company. Upon installation of TRMS they can work with remote tools smoothly.

Main restrictions of the current TRMS version that have been pointed to by the application engineers are:

- TRMS technology for workflow definition and design tasks automation in general, requires training before a novice engineer can apply it efficiently,
- TRMS can invoke command line scripts only,
- The workflow engine of TRMS supports currently sequential workflows without decision blocks and loops only.

## 5. CONCLUSIONS

The presented TRMS architecture and its functionality fulfil requirements for a secure collaborative infrastructure. The TRMS infrastructure supports tool integration and management. In our opinion, TRMS is a solution for distributed cooperating engineers and gives them a possibility for a simple invocation of distant tools. Distributed engineering work will become much easier if engineers are able to cooperate in larger networks with many registered tools. Their work will be accelerated and thus work costs will be reduced. A registered tool will provide a service to distributed design engineers. It is expected that the overload due to tool installation and configuration processes may be reduced substantially. Additionally, encrypted and digitally signed messages assure a high level of security in sensitive design information exchange. TRMS constitutes the infrastructure well tailored to distance-spanning engineering collaboration in complex distributed product design.

The TRMS architecture based on web services has the following advantages:

- Enables easier integration with other collaborative environments,
- The use of the standard HTTPS protocol enables control of the network traffic,
- GTLS as a communication broker enables use of tools that are installed in local networks on machines that are not visible from outside,
- The new architecture supports also tools that require long computation times,
- The environment is robust enough for transient problems in accessing the network,
- It reduces demand for a broad bandwidth in accessing the network, and speeds up the overall the environment.

Basing the TRMS architecture on web services opens it for further improvements that will be due to forecasted new standardisation initiatives related to collaborative services (MAPPER D13, 2008).

Further R&D related to TRMS includes enhanced workflow management system and improved support for engineering teamwork with both synchronous and asynchronous collaboration.

### 5.1 Acknowledgments

Authors would like to acknowledge collaboration with Szymon Grzybek and Wojciech Sakowski from Evatronix SA on TRMS deployment in the distributed design and verification of the USB transceiver virtual component.

## 6. REFERENCES

1. Bauer M., *et al.*, "Advanced Infrastructure for Pan-European Collaborative Engineering". In: Stanford-Smith B; Chiozza E: E-work and E-commerce, Novel solutions and practices for the global networked economy. IOS Press / Ohmsha, Berlin, 2001.
2. Brglez F., E-Design Concepts and Practice: An Overview, IP SoC Grenoble, 2001.
3. CoSpaces IP project 1<sup>st</sup> Annual Conference, 26-28.09.2007, Leiria, Portugal.
4. CURE - Collaborative Universal Remote Education, <http://cure.sourceforge.net/>
5. CWE, 1st Conf. on Collaborative Working Environments for Business and Industry (CWE'06), Brussels, 10-11.05.2006, Conference Report.
6. eCoSpace project Newsletter Nb. 3, June 2007.
7. Fraś P., Kostienko T., Magiera J., Pawlak J., Penkala P., Stachańczyk D., Szlęzak M., Witczynski M. "TRMS Deployment in Distributed Engineering Applications", Proc. CCE'04, Tatranska Lomnica, Slovakia, April 18-21 2004 (ISBN 91-975604-0-5).
8. Fraś P., *et al.*, "Collaborative infrastructure for distance - spanning concurrent engineering". PRO-VE'04, in Luis M. Camarinha-Matos (Ed.) "Virtual Enterprises and Collaborative Networks". 5th IFIP Working Conf. on Virtual Enterprises, Toulouse, Kluwer Academic Publishers, 2004.
9. Haake J., Schümmer T., "Supporting Effective Collaborative Engineering". Proc. Workshop on Challenges in Collaborative Engineering, Krakow, 11-13.04.2007, Lecture Notes in Informatics (LNI), Vol. 120, GI - Gesellschaft für Informatik, Bonn, 2007.
10. Johnsen S., *et al.*, "Model-based Adaptive Product and Process Engineering". New Technologies for the Intelligent Design and Operation of Manufacturing Networks, Rabe, M.; Mihók, P. (eds), Stuttgart, Fraunhofer IRB Verlag 2007.
11. MAPPER D13 "Standardisation Activities Report", Feb. 2008, <http://mapper.eu.org/miug/>.
12. Pawlak A., Jørgensen H., Penkala P., Fraś P., "Business Process and Workflow Management for Design of Electronic Systems- Balancing Flexibility and Control". as [8].
13. Radeke E., "GEN - Global Engineering Networking". Proc. Conf. on Integration in Manufacturing, 1998, Goteborg.
14. Ranta, J., "Globalisation of manufacturing – A new paradigm emerging from electronics industry". IST99, European Conference on Information Society Technologies, Helsinki (Finland).
15. Salminen V., Buckley E., Malinen, P., Ritvas, J., Silakoski S., Sauer, A., "Global engineering networking – Turning engineering knowledge into an accessible corporate asset". Proceedings of the 11<sup>th</sup> Conf. on Engineering Design (ICED'97), Tampere (Finland), vol. 25, p. 165-172.
16. Saucier G., "IP website as a catalyst of IP industry". Proc. Conf. on IP Based Electronic System, Grenoble, 5-6<sup>th</sup> Dec. 2007.
17. Schattkowsky T., Mueller W., Pawlak A. "Workflow Management Middleware for Secure Distance-Spanning Collaborative Engineering". In L. Fischer (ed.) The Workflow Handbook 2004, WfMC, Lighthouse Point, USA, 2004.
18. Siekierska K. *et al.* "Distributed collaborative design of IP components in the TRMS environment". Microelectronics Reliability, Elsevier Journal, vol. 46, 2006, 5-6.
19. Szlęzak M., Pawlak A., Wojciechowski K., "Markup Language Based Design Tool Integration Method Supporting Collaborative Engineering". Pre-Proceedings of CollABD'07, 1<sup>st</sup> Workshop on Integrated Practices for the 21<sup>st</sup> Century: Collaborative Working Environments, (G. Carrara and Y. Kalay (eds), Sapienza Univ., Rome, 13-15.12.2007.