Advanced Infrastructure for Pan-European Collaborative Engineering

M. Bauer[1], H.J.Eikerling[2], W. Mueller[3], A.Pawlak[4,5], K.Siekierska[5], D.Soderberg[6], X.Warzee[7]
[1] *Infineon Technologies, Munich, Germany*
[2] *Siemens Business Services, Paderborn, Germany*
[3] *Paderborn University, Paderborn, Germany*
[4] *Silesian University of Technology, Gliwice, Poland*
[5] *Institute of Electron Technology, Warsaw, Poland*
[6] *FTL Systems UK Ltd, Southampton, UK*
[7] *Thales Optronique, Paris, France*

Abstract. This article presents challenges, visions, and solutions for a true Pan-European collaborative engineering infrastructure that is a target of the IST project E-COLLEG. The consortium aims at the definition of a transparent infrastructure that will enable engineers from various domains to collaborate during the design of complex heterogeneous systems. In this context, we introduce an advanced collaborative infrastructure (ACI). ACI covers dynamic Tool Registration and Management Services (TRMS) for distance-spanning, tool integration and administration, as well as open interfaces for XML-based data exchange. TRMS will constitute a backbone for E-COLLEG-related pan-European collaborative research and engineering studies by fostering a combination of most recent Plug-and-Play techniques employing agent-based communication based on XML-oriented integration technologies. This new enabling technology for collaborative engineering is validated in two application scenarios.

1. Introduction

The advancements in information and communication technologies currently reshape our work. Although new *electronic Work (eWork)* technologies penetrate different professional activities, it is in engineering, where they have a potential for the highest added-value. We claim that engineering collaboration is especially sensitive to the new technologies though it is not sufficiently recognised by current practice yet. The shifting from the traditional manufacturing paradigm to a new, virtual and agile model is globally observed. Whereas the traditional model is characterised by very limited information sharing, static organisational structure, and almost no co-operation, the virtual and agile model exhibits information sharing, collaboration, and dynamic organisation. Collaborative engineering [1] is an innovative paradigm for product development, which integrates widely distributed engineers for virtual collaboration.

Within general globalisation efforts, not only management and organisations [2] but also engineering teams become distributed over large distances. The world-wide market of Integrated Collaborative Environments (ICE) according to IDC [3] was from 50 to 60 millions licenses in 1998, and rose to 90 to 100 millions licenses at the end of 1999. The revenues for ICE markets were predicted (2000) to be between 1 billion to 1.2 billions dollars per annum for next three years. Intranet, document management, workflow solutions, as well as group calendaring and scheduling are currently the main drivers of this market.

Collaborative engineering for distributed product development integrates widely distributed engineers of virtual companies for virtual collaboration [5]. This new way of work is central to the vision of the working environment in the Information Society [11]. Internet as a backbone of the collaborative infrastructure is per se transnational in nature. From the technical point of view, existing collaborative engineering frameworks do presently not support very well the integration of very complex engineering environments when interconnecting multiple design groups and do not consider distance-spanning related issues like firewalls, security, remote administration, and distributed design flow automation. Distributed engineering development, however, needs new infrastructures, net-aware tools, and new design methodologies based on re-use in combination with advanced security and network and tool management. If supported by appropriate advanced collaborative infrastructures and tools, it radically speeds up collaboration over wide distances by connecting Intranets via the Internet. New collaborative technologies allow for fast and flexible building of virtual distributed teams and efficiently make use of distributed engineering resources.

This article investigates an Advanced Collaborative Infrastructure (ACI) with integrated dynamic Tool Registration and Management Services (TRMS), open XML-based data exchange, and collaborative tool extensions. ACI is tailored to distance-spanning engineering collaboration and security enabled data exchange [6], [12]. The infrastructure is validated through two industrial Intranet-crossing case studies between Polish and French teams, as well as between Polish and German ones applying real-world development scenarios.

The remainder of this article is structured as follows. The next section investigates existing approaches. Section 3 presents the general architecture and components for an advanced collaborative infrastructure (ACI). Section 4 describes application scenarios and the last one - a brief summary and conclusions.

2. Enabling Technologies

In complex engineering projects, engineering skills are typically dispersed among various departments located in different cities and countries. In current industrial practice modelling and synthesis tools, as well as specialised libraries of components reside often on different servers. Thus, engineers are enforced to move design data between tools using *ftp*, or traditional *telnet*. Engineers are also enforced to download new tools and to configure them in their design environments. Constraints in licences obstacle remote access to tools, especially over large distances. This results in error prone situations since installing new tools, and/or moving manually data among tools usually is not straightforward, and definitely not scalable for large, multi-site, collaborative engineering tasks.

There are tools available to handle existing problems, like: Intranet solutions (Netscape, Microsoft, ...), document management solutions (Open Text, Interleaf, Documentum, ...), workflow solutions (StaffWare, Keyflow Filenet, Banyan, ...), and group calendaring and scheduling (Lotus/IBM Notes, Microsoft Exchange, ...). However, those tools mainly offer partial solutions and do not combine very well due to limited support for necessary standards. The standards to be considered are: Corba, XML [4], SNMP, etc.

Our conclusion of the market analysis is quite simple: today, there is no commercial offer which is open to third party tools and support collaborative engineering tasks over the Internet covering all needs companies are concerned with.

3. Advanced Collaborative Infrastructure

In order to face challenges of wide area engineering collaboration, a European project with strong industrial representation has been set up in the frame of the 5FP IST programme. The E-COLLEG project develops of an Advanced Collaborative Infrastructure (ACI) as its primary objective. ACI is tailored to distance-spanning engineering collaboration and security enabled data exchange.

ACI comprises four major components:

1. Basic Services
2. Advanced Tool Registration and Management Services
3. XML-based Integration Technologies
4. Collaborative Extensions to Design Tools

3.1 Basic Services

Basic services within a collaborative engineering infrastructure cover a flexible communication backbone, as well as a set of basic services for

- engineering data management
- engineering work flow management and automation
- basic tool encapsulation

The basic communication backbone has to support standard communication services. Today, TCP/IP-based protocols are of major importance, i.e., ftp, telnet, ssh, http, etc. Additional protocol support of PPP, LDAP is appreciated. For engineering data management we can distinguish services for course grain (e.g., files) and fine grain data objects (e.g., Corba). Detailed requirements are mainly due to individual engineering tools and applications. One main aspect for basic services of an engineering infrastructure is the support of basic work flow automation. This mainly covers means for description, implementation, and monitoring of distributed workflows. This comprises a basic underlying virtual machine which interprets distributed workflows enabled to perform basic data exchange via the communication backbone and to launch remote tools for exchanged data. In this context, graphical tools for remote configuration and monitoring, as well as, web enabled interfaces to seamlessly integrate into a standard environment are appreciated. Finally, we need some basic support for tool encapsulation. Here, a straightforward solution is compliant to the Tool Encapsulation Specification language which was defined as a standard for integration of EDA tools.

Extensive experiments with basic services are conducted within E-COLLEG that use the ASTAI® integration platform [10]. These experiments confirm that security issues, and especially firewalls, constitute a real challenge for the integration of distributed tools.

3.2 Advanced Tool Registration and Management

Currently available basic infrastructures mainly support passive tool integration and management with no or little support in tool registration, installation, and discovery. In particular for engineering tools this can be a time consuming activity when to be maintained in larger network due to various versions of frequently migrated tools and supported (sub)standards. Therefore, we are aiming at advanced unified dynamic tool-registration and management services (TRMS). TRMS stands for the combined concepts of universal plug and play and Jini™ technologies applied to the needs of distance-spanning tool integration. Once registered, tools will provide their service for distributed design teams. Advanced service discovery technologies will connect to the most appropriate service with respect to optimal availability within the current configuration of the virtual engineering network.

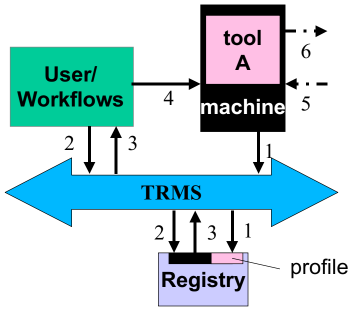

Figure 1. Dynamic Tool-Registration and Management Services (TRMS).

Thus, TRMS basically denotes concepts from Jini™. In a first step, a tool installed on a specific machine registers with its combined hardware/software profile specifying the tool class (text editor, digital simulator, ...), version, possible import and export formats including different versions, operating systems, available main memory, machine type etc. Once being registered (Step 1 in Figure 1) TRMS manages tool discovery and data exchange in five steps (Steps 2 - 6 in Figure 1) specified below:

(2) User or underlying workflow engine contacts the registry for a specific service identifying tool specific properties;
(3) Registry evaluates required properties and selects a tool; if more than one tool matches the requirements one is being selected w.r.t. static or dynamic criteria (e.g. current workload of a machine);
(4) User or underlying workflow engine launches the tool;
(5) Tool fetches design data for input;
(6) Tool generates output, workflow engine copies output to destination.

For distance-spanning tool access TRMS has to include some application specific pre-processors for design data format transformation. Furthermore, it has to include standard plugins like encryption and decryption compliant to existing security standards like PGP [12]. Plugins have to be configured individually to the needs and requirements of individual applications and industrial context.

3.3 *XML-BASED INTEGRATION TECHNOLOGIES*

XML has been demonstrated as an open flexible standard for data exchange. In the framework of an advanced collaborative infrastructure we see two valuable application areas. Though tool encapsulation specification has already been covered by the CFI-standard TES [15], we see a necessity for an XML based solution which can be easily derived from TES. Like the LISP-based TES language, XML-based solutions are very flexible w.r.t. configurability and genericity. On the other hand, there is a real need for a fine grain XML-based data exchange. This domain is presently underdeveloped. There are single solutions for, e.g., XML-based data exchange for VHDL models, but there is currently no standardisation effort for DTDs in the field of EDA data exchange.

Additional efforts have to be invested to exchange not only passive design data but to wrap passive data with some behaviour in order to reduce communication efforts between different sites. This would basically result in active mobile agents, e.g., covering information at destination. However, agent-based data exchange also covers a high security risk. Active objects basically have to be limited to only execute "harmless" behaviour. We thus need to investigate data attacking scenarios before coming up with an industry compliant solution.

3.4 Collaborative Extensions to Design Tools

An advanced distributed infrastructure also requires the support of advanced features for design tools. Design verification, for instance, is the single most time consuming set of services within most embedded system design flows. These services bring together all of the architects, designers, the verification experts and the realisation experts around a single “virtual system” to ensure that the evolving system performs as specified. Each of these experts may be in a distinct location, yet they must work together to probe and modify versions of a common design. Design tools basically have to be customised to enable concurrent, distributed, multi-user operation, e.g., concurrent version control of online data, concurrent view of source code and waveforms represented in XML. In general this requires the extension of individual tools by:

- enhanced distributed access to a live design database, and
- enhanced graphical views to deliver simultaneous, coherent, writable and remote views into live design database.

4. Application Scenarios

The previously introduced services are defined and evaluated in two considerably complex industrial scenarios that constitute a validation strategy of E-COLLEG. The first scenario comprises a collaborative development of a digital camera. The second one is a collaborative development of VHDL-based system level testbenches.

Both applications are designed and verified by a distributed team of engineers in a framework of a multi-site infrastructure coupling the design flows of a design team from major industries with SMEs involved. The first scenario is mainly focussed on tool aspects, i.e., integration of heterogeneous tools. A distributed design methodology developed by the THO/ITE team that is enabled by the applied workflow technology of the ASTAI🎭 platform [10] is described below briefly.

4.1 Scenario for Distributed Design of a Digital Camera

A camera is a complex optronic system, which requires a design team including engineers skilled in digital electronics and optronic domains. A number of involved tasks, tools and complexity of the design (reuse-oriented architecture comprising functional components that are parameterised and made configurable) justifies use of this particular design as the validation scenario for the E-COLLEG collaborative technology. The scenario involves two partners: Thales Optronique (THO) as product developer and manufacturer, and the Institute of Electron Technology (ITE) as a IP provider.

Figure 2 presents a schematic of the digital camera design flow, divided into four main parts, that are executable in the collaborative environment. Each part contains tools dedicated to a different application domain and requires different engineers' expertise. The whole workflow integrates tools at different partner sites residing on remote servers and accessible through Internet/Intranet and common data processed during this design.

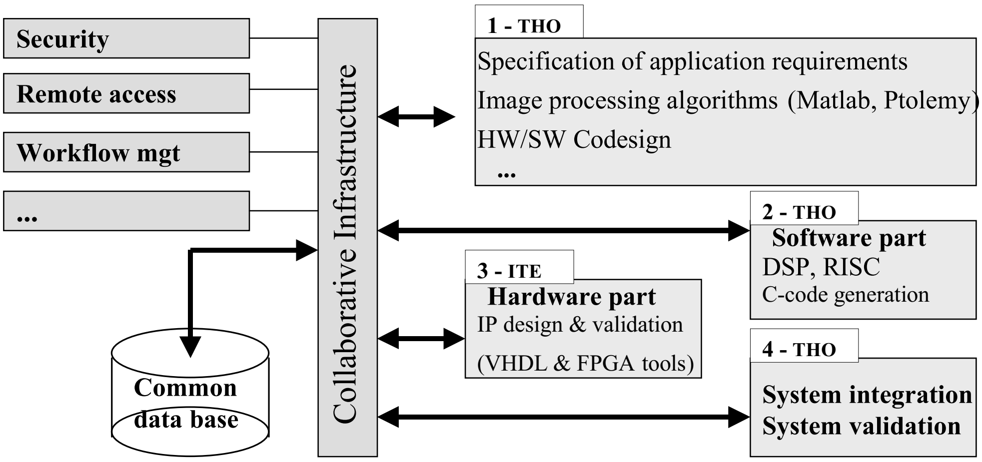

Figure 2. Multi-site workflow for collaborative design of a digital camera.

The steps (part 1, 2 & 3 in Figure 2) that represent different design domains will be performed at several departments of THO, while the part strictly related to hardware component IP design and verification is under responsibility of ITE .

4.2 SCENARIO FOR COLLABORATIVE TESTBENCH DEVELOPMENT

This application scenario focuses on re-use of test environments in collaborative dispersed environments. A component, its documentation and its test environment is a bundle that may be regarded as IP (Intellectual Property) if all rights on it are reserved by a company. The development and trade of IPs is a big challenge for SMEs to co-operate with large companies focusing on a secure and fast data exchange and tool access. Activities cover design, test generation and validation. But test generation and simulation became a real bottleneck in increase of product efficiency. The new collaborative design methodology that has been undertaken by E-COLLEG aims at integration of distributed installations of the testbench environment called Reflective [13]. Reflective addresses the issue of testbench re-use. It will be applied in the distributed workflow spanning Infineon Technologies in Germany and the lab of the Silesian University of Technology in Poland.

The teams involved in both experiments will have to be dynamically modified during the course of the project. Since team members will stay in remote locations their intersections, as well as constant access to automatically updated, distributed project documentation are essential. All these distributed, collaborative design issues will have to be supported by the advanced collaborative infrastructure ACI.

5. Conclusions and Future Work

We have identified challenges, visions, and preliminary solutions for an advanced collaborative distance-spanning infrastructure for non-trivial engineering workflows. Advanced services are build based on basic services for basic communication, workflow management, and tool encapsulation. They cover dynamic tool registration and management, exchange of active, fine grain XML-based design data, as well as collaborative extensions to existing design tools.

These concepts are validated in four phases in the two industry-driven application scenarios. During the first phase the basic infrastructure has already been installed. It allows to check for additional requirements, to select high priority requirements and to focus on the implementation of a prototype on the most relevant ones. In this phase the ASTAI® [10] environment from SBS constitutes the core of the basic collaborative infrastructure (BCI). The currently running second phase comprises validation of the BCI in application scenarios that has been shortly addressed in the paper. The following phase will develop and deploy the advanced collaborative infrastructure (ACI) with the functionality that has been shortly presented. In the final phase, ACI will be validated again in both application scenarios. The results will be measured using the earlier defined metrics.

ACKNOWLEDGEMENTS

The work described herein is funded under the IST programme by the European Commission under #IST-1999-11746 - E-COLLEG.

References

[1] M. Cutkosky, J. Tenenbaum, J. Glicksman, Madefast: an exercise in collaborative engineering over the Internet, Communications of the ACM, Sept. 1996, vol. 39, no. 9.
[2] J. Laitinen, M. Ollus, M. Hannus, Global Engineering and Manufacturing in Enterprise Networks - GLOBEMEN, eBusiness and eWork 2000, 18-20 Oct. 2000, Madrid.
[3] Collaboration : Technology and Market Dynamics, IDC report for DARPA, June 1997
[4] Charles F. Goldfarb, Paul Prescod, The XML Handbook, Third Edition Prentice Hall, 2000.
[5] J. Lipnack, J. Stamps. Virtual Teams: Wiley John & Sons, Inc. 2000
[6] W. Mueller, et al.: Security Concepts for Agent-Based Systems, Proc. SCASE 2001, Enschede.
[7] W. Mueller: A Visual Framework for Agent-Based Specification, IEEE Visual Languages, USA, 2000.
[8] R. Orfali, Dan Harkley, Client Server Programming with Java and Corba Wiley, John & Sons, 1998.
[9] J. Willis J., Z. Li, T-P. Lin, Use of Embedded Scheduling to Compile VHDL for Effective Parallel Simulation, EuroDAC / EuroVHDL, Sept. 1995.
[10] ASTAI® home page: http://www.c-lab.de/astair.
[11] A. Pawlak, W. Cellary, A. Smirnov, X. Warzee, J. Willis, Collaborative Engineering based on Web, Advances in Information Technologies: The Business Challenge, Roger *et al.* (Eds.) IOS Press, 1997.
[12] W. Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall, 1998.
[13] M. Bauer; W. Ecker, REFLECIVE, a reusable extendable flexible testbench, Proc. of the 32th Design Automation Conference, Annahein/USA, June 1997.
[14] CFI, Inter-Tool Communication Programming Interface, CFI document no. dis-92-S-3, CAD Framework Initiative Inc., 1992, Austin, USA.
[15] CFI, Tool Encapsulation Specification, Version 1.0.0, CAD Framework Initiative Inc., 1992, Austin, USA.