CCE'06

CHALLENGES IN COLLABORATIVE ENGINEERING

State of the Art and Future Challenges in Collaborative Design

Prague, Czech Republic

19th-20th April, 2006

in conjunction with DDECS'06

Institute of Microelectronic Systems
Technische Universität Darmstadt
Karlstr. 15, 64283 Darmstadt, Germany
http://www.mes.tu-darmstadt.de

Challenges in Collaborative Engineering Workshop Portal
http://cce.ecolleg.org

Programme

Wednesday, April 19th 2006

08:30 - 08:50 DDECS and CCE Welcome
Ondřej Novák, Bernd Straube, Leandro Soares Indrusiak

08:50 - 09:50 Keynote: Multi-Site Collaboration in System on Chip Design and Validation: The Intel Experience
Ketan Paranjape, Intel Corporation, Hillsboro, USA

09:50 - 10:20 Coffee break

10:20 - 10:30 CCE opening remarks
Leandro Soares Indrusiak, Lennart Karlsson, Kurt Sandkuhl

10:30 - 11:45 Session 1: Collaborative Engineering Environments
Session chair: Leandro Indrusiak, Darmstadt University of Technology, Germany

10:30 *Codebreaker: decentralized, cooperative and flexible support for extreme programming software development*
N. Baloian[2,3], R. Konow[1], F. Claude[2], C. Tala[1], (1) Universidad Diego Portales, Santiago, Chile, (2) Universidad de Chile, Santiago, Chile, (3) GITS, Waseda University, Saitama, Japan
10:55 *SAGE: Self-organized cooperative task management and group awareness for the coordination of distributed software development*
W. Gräther[1] , T. Koch[2] , C. Lemburg[3] , P. Manhart[4], (1) Fraunhofer FIT, Germany, (2) OrbiTeam Software GmbH, Germany, (3) Aixonix GmbH, Germany, (4) DaimlerChrysler AG, Germany
11:20 *Re-experiencing engineering meetings: knowledge reuse challenges from virtual meetings*
Peter Törlind, Andreas Larsson, Lulea Univ., Lulea, Sweden

11:45 - 12:00 Coffee break

12:00 - 13:15 Session 2: Collaboration over Distributed Resources
Session chair: Nelson Baloian, Universidad Diego Portales, Santiago, Chile

12:00 *A framework for distributed collaborative automotive testing*
Mathias Johanson, Alkit Communications AB, Sweden
12:25 *Platform for the integration of the distributed virtual component resources*
Dariusz Stachanczyk, Silesian Univ. of Technology, Gliwice, Poland
12:50 *Middleware and HCI integration for the shared access to geographically distributed devices*
Christophe Gravier, Jacques Fayolle, DIOM Laboratory, Jean Monnet University, Saint Etienne, France

13:15 - 14:30 Lunch

14:30 - 15:45 Session 3: Collaborative design
Session chair: Mathias Johanson, Alkit Communications AB, Sweden
14 :30 *Competence model for collaborative design*
Kurt Sandkuhl, Magnus Lundqvist, Vladimir Tarassov, Jönköping University, Jönköping, Sweden
14:55 *Ontology based management of designer's guidelines for motorcar manufacture*
Fredrik Elgh, Staffan Sunnersjö, Jönköping University, Jönköping, Sweden
15: 20 *A taxonomy for the collaborative design of integrated electronic systems*
Leandro Soares Indrusiak[1], Manfred Glesner[1], Ricardo Reis[2], (1)Tech. Univ. Darmstadt, Darmstadt, Germany, (2) UFRGS, Porto Alegre, Brazil

15:45 - 16:15 Coffee break

16:15 - 17:05 Session 4: Merging Engineering and Business Information Flows
Session chair: TBD
16:15 *Source network for electronic systems design*
Maciej Witczynski, Silesian Univ. of Technology, Gliwice, Poland
16:40 *An activity based simulation approach to functional product development*
Magnus Löfstrand, Tobias Larsson, Lennart Karlsson, Lulea Univ. of Technology, Lulea, Sweden

17.20 - 18.00 Panel TBD

19:00- 21:00 Welcome party together with DDECS'07

Thursday, April 20th 2006

Industrial Day Program

New Approaches to Collaborative Design - Enhancing your PLM Solution!

08:30 – 11:00 Session 5: Pilot presentations and demonstrations session

Session chair: Frank Lillehagen, Troux Technologies AS, Oslo, Norway

08:40 *The MAPPER Collaboration Platform and the AKM Collaborative Design Approach and Methodology*

Håvard Jørgensen, Troux Technologies AS, Lysaker, Norway

09:40 *The VCES presentation of cost estimation methods, and demonstration of Engineering Services, Cost Engineering (e-Mentor), and cost engineering e-Training*

Jose Rios, Cranfield University, UK.

10:40 *Hands-on Piloting, participants will be able to work with the MAPPER and V_CES pilots*

11:00 – 11:15 Coffee break

11:15 – 12:30 Session 6: MAPPER industrial design use-case pilots

Session chair: Kurt Sandkuhl, Jönköping University, Sweden

11:20 *The Fiat CRF Use-case pilots*

TBD

11:40 *The Evatronix Use-case pilots*

Wojciech Sakowski, President and CEO, Evatronix SA, Gliwice, Poland

12:00 *c-Business Challenges and Possible Solutions*

Per Høgberg, Project Manager, Kongsberg Automotive Group, Sweden

12:30 - 13:30 Lunch

13:30 – 15:15 Industry presentations and discussions on approaches and pilots

Session chair: Adam Pawlak, Silesian Univ. of Technology, Poland

13:35 *The jABC approach to Collaborative Development of Embedded Applications*

Martina Hörmann[1], Thomas Mender[1], Michael Schuster[1], Hong Trinh[1], Tiziana Margaria[2], Ralf Nagel[3], Bernhard Steffen[3], (1) IKEA IT Germany, (2) Universität Potsdam, (3) Universität Dortmund

14:00 *The Valtech Collaborative Cockpit*

Xavier Warzee et al, Valtech, Paris, France

14:25 *Co-operative and Virtual Engineering: a broad roadmap*

Jan Goossenearts et al, Univ. of Eindhoven, Eindhoven, the Netherlands

14:50 *A family of Modeling and Execution Platforms*

Håvard Jørgensen et al, Troux Technologies AS, Lysaker, Oslo, Norway

15:30 – 18:00 Hands-on Piloting, interested participants will be able to work with the MAPPER and V_CES pilots, continuation of the morning session

15:30 - 18:30 Social event in Prague

With old tram from Masarykova kolej to Prague castle. From Prague castle, walking through Charles bridge to the Bethlehem chapel.

19:30 - 24:00 Banquet - Betlehem Chapel (Old Town)

Event Objectives

CCE'06 is the 4th event in a series of workshops dedicated to industrial practices and new technologies for collaborative engineering. Collaborative Engineering aims at providing concepts, technologies and solutions for product development in dispersed engineering teams. The increased industrial demand for this innovative approach is based on the fact that networked organization struc-tures are common practice in numerous industry sectors, like automobile, aerospace, electronics or construction. Collaboration has become a key issue for agile and flexible engineering processes.

As in the previous years, the CCE workshop will have a special focus on a relevant area of collaborative engineering, but contributions from all areas are encouraged. In its 2006 edition, the focus of CCE will be on collaborative design, mainly on the application domain of electronic systems. Contributions of innovative techniques and practical experiences on collaboration at all phases of design are welcomed: specification, simulation, prototyping and implementation. In each of those phases, there are many challenges and unsolved problems from the industrial perspective, including:

- technical aspects: creating and sharing complex design flows; design tool encapsulation, selection and integration; design data modeling, management and consistency control; shared access to computational facilities, experiments, laboratories and prototyping facilities; collaboration support on design languages and tools;
- social aspects: increasing user awareness in collaborative design environments; knowledge sharing and collaborative learning; handling multi-cultural issues in collaborative design;
- organizational and economic aspects: measuring benefits of collaboration on design; organizational guidelines to foster collaboration on design; intellectual property sharing and protection.

The workshop aims at presenting concepts, technologies, and solutions for collaborative engineering in an industrial context. Researchers, software developers and end users are invited to contribute to the discussion by presenting application problems, giving experience reports, or by introducing concepts, methods, and software solutions.

Industry Day

CCE'06 hosts this year a special Industrial Day that will provide industrial perspectives on new ways of working applying innovative web-platforms for collaborative design, engineering and task management. This year's focus will be on collaborative design approaches and platforms as being prototyped and piloted in industry. Two EU projects: MAPPER, see www.mapper-ist.org, and VCES, see www.v-ces.com, will present their approaches, methodologies, platforms and industrial pilot solutions.

Event Committees

Workshop Programme Chair

Leandro Soares Indrusiak, Tech. Univ. Darmstadt, Germany

General Co-Chairs

Lennart Karlsson, Univ. Luleå, Sweden
Kurt Sandkuhl, Univ. Jönköping, Sweden

Industrial Liason

Frank Lillehagen, Troux Technologies AS, Norway

Programme Committee

Arthur Baskin, Intelligent IT, US
Matthias Bauer, Infineon Technologies, Munich, Germany
Manuel Carballeda, Thales Optronique, France
Wolfgang Deiters, Fraunhofer ISST, Germany
Luis Gomes, UNINOVA, Lisbon, Portugal
Joerg Haake, Univ. Hagen, Germany
Leandro Indrusiak, Tech. Univ. Darmstadt, Germany
Gianni Jacucci, Univ. of Trento, Italy, Programme Chair
Tom Kazmierski, Univ. Southampton, UK
Lennart Karlsson, Univ. Luleå, Sweden
Konrad Klöckner, Fraunhofer FIT, Germany
Stefan Kohut, Slovak Academy of Sciences, Slovakia
George Kovacs, Hungarian Academy of Sciences, Hungary
Frank Lillehagen, Troux Technologies AS, Norway
Wolfgang Mueller, Univ. Paderborn, Germany
Martin Ollus, VTT Industrial Systems, Finland
Ketan Paranjape, Intel Corporation, USA
Adam Pawlak, Silesian University of Technology, Poland
Johann Riedel, Nottingham Univ., UK
Kurt Sandkuhl, Univ. Jönköping, Sweden
Staffan Sunnersjö, Univ. Jönköping, Sweden
Stefan Uellner, T-Systems, Germany

Steering Committee

Mattias Bauer, Infineon Technologies, Germany
Manuel Carballeda, Thales Optronique, France
Wolfgang Mueller, Univ. Paderborn, Germany
Adam Pawlak, Silesian University of Technology, Poland
Kurt Sandkuhl, Univ. Jönköping, Sweden

Organising Committee Chairs

Hana Kubatova, Czech Technical University in Prague, Czech Republic

Piotr Penkala, CiEL, Silesian Univ. of Technology, Poland

Information for authors

The workshop proceedings will be prepared after the workshop. Please check the workshop web-site for formatting instructions of the final camera-ready full papers. Extended version of selected papers will appear in the Special Issue of the Journal: International Journal of e-Collaboration.

DDECS'06

The CCE'06 workshop accompanies the DDECS'06 event, i.e. the IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems that will be held in the same place on April 18-21, 2006.

DDECS is a forum for presenting and discussing trends, emerging results, practical applications, and hot topics in the areas of design, test and diagnosis of microelectronic circuits and systems. The workshop was held annually in Central European countries: Czech Republic (1997, 2002), Poland (1998, 2003), Slovakia (2000, 2003), and Hungary (2001, 2005). The workshop in Prague will be the ninth edition of this event.

The detailed information on DDECS'06 is available on:

http://ddecs06.felk.cvut.cz/

Abstracts of the workshop contributions

MULTI-SITE COLLABORATION IN SYSTEM ON CHIP DESIGN AND VALIDATION: THE INTEL EXPERIENCE

Ketan Paranjape
Intel Corporation
M/S JF3-218, 2111 NE 25th Avenue,
Hillsboro, OR 97123, USA
ketan.paranjape@intel.com

Extended Abstract

As companies expand their infrastructure to multiplecountries to make their presence felt in those markets, to benefit from the local knowledge and talent pool, and to take advantage of favorable economics, critical projects are partitioned and handled by these new sites. This creates problems with project partitioning, communication, decision-making and logistics. Organizations therefore need to put a solid framework in place to manage these projects. Teams must develop and follow cross-site project management and execution guidelines in order to be successful.
This talk will address these concerns in the context of Intel SoC's by walking through the integrated circuit design flows, design methodologies and standards, challenges in design and validation and finally address project planning, partitioning and execution.

A. Driving Forces – Intel Strategic Direction
Moore's Law – As the number of transistors double every 18 months, how do we take advantage of these transistors? Intel Platforms – The new mindset at Intel is to deliver platforms with more added value and performance to every end user. To accomplish this, the company has reorganized into 5 platform divisions namely Digital Enterprise, Channel Platforms, Digital Home, Mobility and Digital Health. The platform will primarily be a collection of hardware, software, technologies, standards, initiatives and services. The intent is to make sure that the platform is greater than the sum of their parts.

B. System on Chip Development
This section will focus on the Intel's SoC life cycle and discuss the following areas:
Design Flows – SoC design flow with emphasis on digital and analog design characteristics.
Design Challenges – Address design challenges in the context of verification, power, DPM, and Time to Market.
Validation Challenges – Validating complex subcomponents, creating validation environments to speed up back end validation.
Platform Validation Challenges – Address issues with early enabling, validating new drivers and software, ability to run legacy software and managing complex customer usage models.
Design Partitioning – Methodology adapted to partition designs.
Cross-Site Data Sharing – Since many Intel design projects span multiple sites and geographies, efficient global worldwide engineering collaboration is a key to future success.
However, globalization raises some significant challenges to the engineering community – what are the right tools and methodologies for such collaboration? How

to achieve fast and robust data sharing while avoiding information security compromise?

C. Multi-Site Project Management

The following sections will be addressed in the talk.

Setting Standards for Communication – When dealing with multiple site engineers have to develop standards for availability and acknowledgement and take responsibility for prioritizing communication.

Project Partitioning – Projects need to be partitioned to maximize the value of the multi sites. During project planning items like natural boundaries, proximity to vendors and customers, site expertise, time zone, communication flow, finance etc have to be considered.

Decision Making - With factors like the lack of real-time communication, the inability to have productive hallway conversations, inefficiencies in having one person make a decision without consulting all the sites, etc., cross-site decision making is extremely difficult. Group leads have to collect inputs from all the sources and be expeditious when making decisions that impact all the sites.

Cultural Impact - People at different sites often come from varying backgrounds and cultures. As a result, people may react very differently in a given situation.

Economics – This section will address travel, expatriate process, job rotations, symmetry in the labs, coordination between various project disciplines, compute and licenses etc.

D. Managing Daily Cross-Site Interaction

This section will address day-to-day concerns in running cross-site projects. In order to effectively manage and run teams, building relationships and trust is extremely important.

Setting clear roles and responsibilities and establishing clear conflict resolution mechanism is also vital. The talk will also address issues with working across different time zones and share techniques that have proved useful at Intel.

CODEBREAKER: DECENTRALIZED, COOPERATIVE AND FLEXIBLE SUPPORT FOR EXTREME PROGRAMMING SOFTWARE DEVELOPMENT

Baloian, N	Konow, R	Claude, F	Tala, C
GITS, Waseda University, Japan	Universidad Diego Portales , Chile.	Universidad de Chile, Chile.	Universidad Diego Portales , Chile.
nbaloian@dcc.uchile.cl	rkonowkr@al.udp.cl	fclaude@dcc.uchile.cl	ctalasa@al.udp.cl

In this work we will present a system called Codebreaker for supporting small and medium size software development based on an extreme programming principle. Currently the existing tools to support the collaborative development are often dedicated to large projects with many developers in a stable development environment. The system works on a peer-to-peer architecture without having a central repository. As we want to support people who may start a new development without previous preparation, a central repository may not be always available for all members at that moment. Because of this, every member of the developing group should have a copy of the project as updated as possible, even when working alone.
It allows synchronous and asynchronous collaborative working. Synchronous collaboration is supported when two or more users are on line, providing the adequate tools. For the asynchronous collaboration it provides mechanisms that allow participants which are offline synchronize their work with the rest of the team by merging the code developed off line.
It allows the inclusion of new unforeseen participants in order to support flexible and changing teams. However, the rules of the system prevent uncontrolled explosion of participants and maintains the order in the versioning of the code. This is done by introducing a rule that every part of the code should be “owned” by one member of the developing team who is responsible for keeping the “official” current version of that code.
The locking of the code is fine grained and logical oriented. This means that smaller parts of the code, which do not correspond necessarily to a physical file can be locked. It also allows to reserve names for locking of still not written code, which can be used to distribute the work among the members by just locking names of classes, methods or even variables which are still not written or used.
In order to implement some kind of role assignment among the team members CodeBreaker introduces two mechanisms which allow this with flexibility. The first one is, when a user is created it may or not receive the right of accepting new participants. The number of participants which is allowed to invite can be also be set. The second one is while receiving the ownership of a code, the right to pass this ownership to a third can be granted or not.
CodeBreaker is developed as an extension of NetBeans and it is still in the prototype implementation phase. To continue the work over this idea, we plan first to finish the development in order to test of the system in real environments. Starting with arranged experiments, with volunteers, and finishing with real teams, working on real projects. With all this information and the information of every tested team about it's past projects, the efficiency of this tool could be really measured and it could be possibly to conclude about it's effectiveness.

SAGE: SELF-ORGANIZED COOPERATIVE TASK MANAGEMENT AND GROUP AWARENESS FOR THE COORDINATION OF DISTRIBUTED SOFTWARE DEVELOPMENT PROCESSES

Wolfgang Gräther
Fraunhofer FIT
wolfgang.graether@fit.fraunhofer.de

Christian Lemburg
Aixonix GmbH
lemburg@aixonix.de

Thomas Koch
OrbiTeam Software GmbH
koch@orbiteam.de

Peter Manhart
DaimlerChrysler AG
peter.manhart@daimlerchrysler.com

Introduction

Coordination of distributed development processes in technical domains is often complex and subject to frequent change, which makes a formal description using workflows too complicated. Even experts cannot describe in detail how they will perform and coordinate their tasks. SAGE aims at the development of new solutions by combining approaches from software engineering with workflow technology and CSCW methods.

Task management with user-defined workflows

SAGE supports coordination of distributed development processes with a platform that establishes a shared virtual information space, which is used to create and change tasks, store and access results or agreements, to discuss and negotiate, and to stay informed about distributed development processes. Management of team membership and member roles is accomplished by the team itself (self-organized).

The concept of user-defined workflows has been developed to allow specification of individual as well as group tasks. There are atomic tasks, deadlines, tasks with subtasks and tasks for delegation. However the type of a task can be changed during the process, i.e. users have not to describe the entire process at the beginning of the collaboration.

SAGE plan to provide several services which support a high level of transparency and disburden team members from routine monitoring tasks: escalation of deadlines as well as error messages and change requests; compilation of important deadlines, documents and agreements based on relations between tasks, relations between tasks and shared workspaces, or expected results; automatic adjustment of deadlines and automatic notification about documents linked to agreements; supporting the follow-up of technical meetings. The SAGE platform will be realized as web-based system (SOA, web services) and will be based on the groupware BSCW.

Acknowledgement

We would like to thank all members of the SAGE project team. The SAGE project is partly funded by the German Federal Ministry of Education and Research.

RE-EXPERIENCING ENGINEERING MEETINGS: KNOWLEDGE REUSE CHALLENGES FROM VIRTUAL MEETINGS

Törlind, P. and Larsson, A.
Polhem Laboratory
Division of Computer Aided Design
Luleå University of Technology
SE-971 87 Luleå
SWEDEN
Peter.Torlind@ltu.se

Introduction

The aim of this paper is to explore the challenges related to knowledge and expertise sharing in globally distributed engineering design teams. Specifically, this paper deals with the problems of reusing knowledge co-created in synchronous meetings between geographically distributed individuals or teams (i.e. 'virtual meetings'). Although information systems seem to be in abundance in product developing organisations, person-to-person knowledge transfer is still critical for engineers to 'know who knows', 'know who to ask', and 'know who to trust' throughout the course of their daily work. The sharing of 'engineering experiences' in product development commonly has to do with the capturing and reusing of design histories, i.e. the information, decisions and rationale that have accumulated throughout the project. Much of this information has been captured and stored in files, documents, models and databases to provide access and reuse capabilities at a later stage in the current project, or in forthcoming projects. In this paper, we do not focus on the information that *has* actually been documented, but instead on the 'lived experience' of doing engineering work in collaboration with others, either in co-located 'face-to-face' meetings or in 'virtual meetings'. Thus, we do not discuss engineering experiences in terms of codified knowledge or documented design rationale, but rather in terms of shared experiences of real-time events in which engineers collaboratively construct meaning through discussions and interactions with both co-located and remote colleagues.

This paper highlights that while synchronous meetings often result in, for example, an improved understanding of a certain problem, and a shared understanding of the specific steps needed to reach an agreeable solution, much of the knowledge created and shared in such meetings is often not adequately documented. Even when documentation does exist, it is often fragmented and decontextualised, which of course renders it less useful for people who were not present at the particular meeting. Hence, there is an obvious gap between the knowledge shared within the meeting and the knowledge subsequently shared with those *not present* at the meeting. With regards to industry globalisation and the growing complexity of product development projects, it has become increasingly appealing to effectively share engineering experiences, such as virtual meetings, with people who for various reasons did not take part in person. This raises the need for technologies that can provide global engineering design teams with the possibility to reuse knowledge and expertise created and shared in virtual meetings, and person-to-person interactions in general, and keeping in mind that much of that knowledge and expertise is presently not formally documented.

A FRAMEWORK FOR DISTRIBUTED COLLABORATIVE AUTOMOTIVE TESTING

Mathias Johanson
Alkit Communications AB
mathias@alkit.se

Abstract: In this paper we explore the opportunities of improving the testing and verification phases of product development in the automotive industry, through a combination of broadband telematic services and tools and methods for distributed collaborative engineering. The development of a prototype system for remote vehicle testing is described, and experiences from experimental usage of the system at an automotive winter testing facility are reported. The prototype system supports real time communication of audio, video and measurement data from a vehicle at a test track, to a remote location such as an automotive company's development site. Initial results from using the system are promising, indicating a clear potential of improving the testing and verification phases of product development within the automotive sector.

Introduction

The general idea is to enable automotive test engineers to participate in vehicle testing without having to be physically present at the proving ground, through tools a methods for distributed collaborative engineering [1]. The hypothesis is that a significant amount of a test engineer's work can be performed using broadband networks and customized collaboration tools for interpersonal communication and data sharing.

Pilot study

A pilot study of the framework was conducted in collaboration between Volvo Car Corporation and Luleå University of Technology. The system was installed at Arctic Falls proving ground in Älvsbyn in northern Sweden, and at Volvo Car Corporation's development site in Gothenburg, in the southwest of Sweden.

Main results

The pilot study proved that the concept for distributed collaborative engineering in automotive testing is feasible, and revealed a great potential of improving the efficiency of the testing and verification process.

References

[1] Törlind, P., A Collaborative Framework for Distributed Winter Testing. Proceedings of eChallenges 2004, Vienna, Austria, October 2004.

PLATFORM FOR THE INTEGRATION OF THE DISTRIBUTED VIRTUAL COMPONENT RESOURCES

Dariusz Stachańczyk
Institute of Electronics, Silesian University of Technology
Gliwice, Poland
Dariusz.Stachanczyk@polsl.pl

One of the most important factors affecting a success in the IP (Intellectual Property) reuse based SoC (System-on-a-Chip) device design is accessibility of the library containing a large number of virtual components (VCs). While the availability of such a library significantly increases chances for agile collection of required system components, the effective use of the available VCs resources requires the efficient solutions that facilitate VC searching and selecting processes. The support of virtual component searching and selection processes is especially important in case of distributed design environments comprising a number of geographically dispersed design SMEs (Small or Medium Enterprises) which collaborate in the SoC device design.

Although, several solutions supporting searching and selecting of virtual components have been proposed so far, most of them still have some shortcomings, e.g. tools are expensive, integration of distributed resources is not possible or used VC description is too limited to enable efficient component selection. All of these limitations were taken into consideration during implementation of the Virtual Component Management System (VCMS) that was designed to support VC information exchange between cooperating, distributed designers groups.

The main goal of the VCMS realisation was to provide an inexpensive and easy to implement VC exchange platform that would enable integration of distributed components resources, simple and quick VC search according to specification requirements and facilitate component selection process by the direct access to description of the key VC features. Because the first stage of the component selection is usually based on the available VC description it should contain all critical for selection process information, like: VC characteristic features, VC interface and throughput data, description of the design and verification methodologies, etc. Unfortunately, different methods of the virtual component characteristic description used by their providers significantly increase the complexity of the virtual components selection process. To avoid that the unified XML based virtual component description method is used within the VCMS environment that enables SoC designer to easily find, compare with other available components and finally select the required by system specification component.

The VCMS environment comprises several, easy to implement, distributed VC catalogues. A VC catalogue stores the XML description data of the virtual components provided by a relevant SME and can be easily accessed both from an SME intranet, as well as from the Internet. The searching mechanism implemented in the VCMS enables designer to specify complex queries including requirements for several component attributes. The specified query can be sent simultaneously to all available in environment VC catalogues and the search results are presented in a form of all found in the accessible catalogues VCs list. To facilitate VCs comparison and appropriate VC selection for each found VC, regardless of its source, the unified detailed description of the component can be displayed.

MIDLLEWARE AND HCI INTEGRATION FOR THE SHARED ACCESS TO GEOGRAPHICALLY DISTRIBUTED DEVICES

Christophe GRAVIER
DIOM laboratory
christophe.gravier@univ-st-etienne.fr

Jacques FAYOLLE
DIOM Laboratory
jacques.fayolle@univ-st-etienne.fr

Abstract

This article provides details on a complete, platform independent, framework, for the remote control of high technological instruments in extended enterprises. Of course, such a challenge raises issues related to computer science paradigms. Mainly, there is no denying that the future of executive systems is made of security, scalability, authentication, "real time" accesses, multi-platform and multi-users. This implies to provide middleware and Human Computer Interactions (henceforth HCI) being able to supply a collaborative environment in order to remote control devices. At first, an exhaustive list of requirements is presented for such tools. Next, the stress is put on the software solution proposed, regarding middleware for user awareness and HCI integration. Finally a section will illustrate how the platform handles localization transparency of instruments. The user is able to see a single device by its virtual representation whereas each instrument are not geographically co-located (granting shared access to geographically distributed devices).

COMPETENCE MODEL FOR COLLABORATIVE DESIGN

Kurt Sandkuhl, Magnus Lundqvist, Vladimir Tarassov
School of Engineering at Jönköping University,
P.O. Box 1026, 55111 Jönköping, Sweden
{ Kurt.Sandkuhl | Magnus.Lundqvist | Valdimir Tarassov} @ing.hj.se

Extended Abstract

Collaborative design in dispersed groups of engineers creates various kinds of challenges to technology, organization and social environment. Work presented in this paper is located in the area of formation of teams for collaborative design. The challenge addressed is how to describe the competences needed for a planned collaborative design project in a way that those individuals best suited for the collaboration can be identified. The proposed approach is to model competences of individuals including different competence areas like cultural, professional or occupational competences.

Based on the findings from an empirical investigation carried out in Sweden during 2005, we consider competence as essential factor when selecting members for a collaborative design team. It seems that individuals when having some information demand unconsciously do a competence assessment and chose the one colleague perceived knowledgeable and informed in the relevant subject or area.

An individual usually acquires a wide range of different capabilities during his/her life experience. Most of them can be considered as competences, which can be applied in work situations. More specifically, competence can be defined as a set of all knowledge forms and personal abilities that are required for performing tasks. A competence model is a well-defined formal structure allowing for representation of these knowledge forms and abilities for an individual.

In order to develop a competence model for collaborative design, which represents all essential and desirable competences needed, we have to understand the nature of collaborative design. Collaborative design can be defined as design task performed in a dispersed group of workers with a joint collaboration objective. This leads to at least three areas, which should be taken into account

- The nature of engineering design work itself,
- The work of design teams as compared to individuals,
- The effects of distributing design work as compared to co-located design.

Although all three areas have been thoroughly researched, a systematic analysis of competences required is not available. Our approach is to derive a competence model based on the analysis of literature from the above areas.

Furthermore, the paper addresses the question, which formal representation to use for competence models in order to make competence models available for computer supported retrieval or matching. We have chosen to use an ontology language to represent competences. We consider this method to be well suited for formalization of competence models because it allows for capture of the rich semantics of competence. Each competence item can be represented with a concept, competence measurement or description with a property, and competence sub-items with relations to other concepts. To accommodate the identified perspectives, the model is designed to include three major parts: General Competence, Cultural Competence, and Occupational Competence.

ONTOLOGY BASED MANAGEMENT OF DESIGNER'S GUIDELINES FOR MOTORCAR MANUFACTURE

Fredrik Elgh, Staffan Sunnersjö
School of Engineering,
Department of Mechanical Engineering
Jönköping University

Background

Ten years ago the product life cycles in the motorcar industry was such that the development of a new car model usually meant setting up a new assembly line, or even a new manufacturing plant, for this particular model. This production facility could then be adapted to the requirements of the new model. With today's high product variety and shorter life cycles, this is no longer possible. Instead the new car designs must be adapted to existing production facilities so that these can be used for several car models, often run simultaneously and in an arbitrary, order driven sequence on the same line operated by the same personnel.

This entirely new production paradigm relies on production constraints being well defined, understood and applied by the car designers. Project planning and working practices with frequent interchange of information between production and design departments is a necessity. There is also a strong need of a more formalised definition of manufacturing constraints. A natural way is to represent these constraints as *manufacturing requirements* in analogy with the *functional requirements* defined by the department for product planning.

The car company studied in this work has over the years compiled design rules related to the manufacturing process in order to guide the design work towards solutions with good manufacturability in existing facilities. The rule base is being gradually expanded and today comprises more than 1500 rules. It is to be expected that the knowledge base will become substantially larger over time, which may incur problems with maintenance and easy access for designers and production engineers.

The purpose of this work is to explore ontology based solutions to handle the growing production related information source, so that relevant information can always be retrieved in a flexible manner for a variety of needs among designers and production engineers. It is important to choose dynamic solutions, which allow the guide lines to change frequently as this will occur naturally as product, processes and experiences evolve over time.

Several information tree structures relating to the product, the manufacturing system, the organization and the rule base exist, but they are not formally linked together. In our work we make use of the existing structures and link them using appropriately named links. We also propose the introduction of a new structure describing the generic functions of the manufacturing system, MSF. This tree structure is a suitable tool to link product related objects to their associated production equipment at varying levels of detail. We also use the rule inference facility to reduce the number of explicitly defined. The approach is exemplified by a small example relating to the hood system for a car.

A TAXONOMY FOR THE COLLABORATIVE DESIGN OF INTEGRATED ELECTRONIC SYSTEMS

Leandro Soares Indrusiak, Manfred Glesner
Institute of Microelectronic Systems
Technische Universität Darmstadt
Karlstr. 15, 64283 Darmstadt, Germany
[lsi, glesner] @mes.tu-darmstadt.de

Ricardo Reis
Instituto de Informática
Universidade Federal do Rio Grande do Sul
C. P. 15064 - 91501-970 Porto Alegre, Brasil
reis@inf.ufrgs.br

Extended Abstract

The research topic called collaborative design can be considered as a blend of computer-aided design and computer supported collaborative work - usually associated to the terms CAD and CSCW, respectively. Both fields incorporate already a significant amount of knowledge. The interdisciplinary research involving both areas is also well developed, specially in the areas of CAD for mechanical and civil engineering. However, the techniques supporting collaboration among mechanical and civil engineers differ significantly than those needed by integrated systems designers, which are the main focus of this paper. The major difference lies on the fact that both civil and mechanical engineers design with very little abstraction, and thus collaborate over design models that are almost exclusively geometrical. In the case of integrated systems design, most of the interaction and collaboration among designers happens at the abstract level, and the geometrical models (the circuit layout description) is either automatically generated or handcrafted by a designer individually. Perhaps the only exception to this scenario may be the floorplanning of really large general-purpose integrated circuits, where the leaders of the several design groups involved in the project actively interact on the partition of the chip real estate. In this paper, we focus instead on the most common scenario, which is often found in small and medium-size companies and research institutions, where a heterogeneous group of designers must bring together different knowledge and expertise in order to design an electronic system – usually application-specific – based on an abstract functional description and a set of constraints regarding price, usability and performance.

To better analyze the different technologies supporting collaborative design and to assess their applicability to the integrated systems domain, we propose a taxonomy that regards the objects that are being shared among the designers: design data, design knowledge, methodologies, tools and resources. Such decision was based on the intention to avoid a pure technical analysis (as it would be, for instance, a taxonomy based on the consistency control mechanisms or the already famous time-space taxonomy), focusing instead on the relevance of the techniques to the end user. By doing that, we expect to contribute to the reusability of the surveyed techniques, as they are presented under the light of the actual problem they are designed to solve.

The taxonomy was loosely devised from a survey of several techniques focusing specifically on the collaborative design of integrated electronic systems and circuits, but a few seminal contributions borrowed from the CSCW literature are also considered due to its potential applicability within the integrated systems domain.

SOURCE NETWORK FOR ELECTRONIC SYSTEMS DESIGN

Maciej Witczyński
Silesian University of Technology, Gliwice, Poland
witczynski@ciel.pl

The requirement of time-to-market shortening forces electronic companies to seek for new design methods that improve design efficiency. A new IP (Intellectual Property) re-use design paradigm can be a solution for development of new SoCs (System-on-a-Chip). This type of design is especially justified in a supply chain of several electronic companies that are IP component providers and form a complex collaborative network. This collaborative network has a form of a source network (SN) or a dynamic design virtual enterprise (VE). The source network is a stable, but not static, group of companies that have developed a preparedness to co-operate in case of a specific customer demand [1]. The dynamic design virtual enterprise is a temporary consortium of different partners established to fulfil a value adding design tasks. Fast and efficient creation of a dynamic design VE within SN in response to market demands can influence time-to-market decrease.

The paper concentrates on efficient acting of source network for electronic systems design. SN management model that describes the structure of the network, main roles, and its functionalities is depicted. Four main characters of member companies are distinguished: Network Manager – preferably one company for the whole network, Network Broker – one or few companies in the network, Project Manager / VE Coordinator – potentially all members, in a particular moment the company that won the tender for the project coordination, Network Member – all companies in the network. The paper also proposes a sketch of a management-oriented process model that includes data flow and distribution of activities in regards to core business processes of the network. Basing on this model a VE initiation module, which is a part of planned IT system for SN support, has been under development.

Following, this "work-in-progress" paper addresses various technological issues that must be decided to realize a complete IT system for SN support. Especially interesting are Web Services (WS) as a base technology for source network implementation and IT system modeling languages which should compose the web services used. Composition of web services can be realized with usage of dedicated modeling languages, like WSCDL (Web Services Choreography Description Language), WSCI (Web Service Choreography Interface), BPEL4WS (Business Process Execution Language for Web Services) [2]. At the end a general scenario example of SoC design initiation is presented in which WS-based semi-automatic VE initiation process including functionalities, like: partners search and project planning are underlined.

The presented IT solution is based on the moderately simple model without special complicated functionalities, like negotiation or contracting phases that should be rather accomplished with human assistance. A simplicity of the solutions assures the possibility for easy realization of main functionalities required in the source network. Applying WS technology gives additional benefits and among them the most important is a possibility to act in heterogeneous systems of various platforms.

References

[1] Witczyński, M., Pawlak, A., "Virtual Organisation in the Electronic Sector" in Virtual Organizations – Systems and Practices – edited by Camarinha-Matos, L. M.; Afsarmanesh, H.; Ollus, M., Springer 2005, X, ISBN: 0-387-23755-0.

[2] State of the Art in Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability - Work package A1.1 – Public deliverable of ATHENA project (www.athena-ip.org).

AN ACTIVITY BASED SIMULATION APPROACH TO FUNCTIONAL PRODUCT DEVELOPMENT

Magnus Löfstrand
Division of Computer Aided Design,
Luleå University of Technology,
Sweden,
Magnus.Lofstrand@ltu.se

Tobias Larsson, Lennart Karlsson
Division of Computer Aided Design,
Luleå University of Technology,
Sweden,
Tobias.c.Larsson@ltu.se

1 Functional Product Development Challenge Today's Work Practice

This paper discusses the new demands that are placed on tools and methods used in industrial product development due to the transformation of industrial companies from hardware producers to function providers, including some effects on collaborative engineering. Traditionally, manufacturing industry focuses on providing excellent goods, i.e. hardware. Services occur on an aftermarket, as an add-on to the developed hardware. By supplying functions, companies can gain control of the aftermarket. The responsibility and availability of the functions provided by hardware remains with the function provider as well as the responsibility for maintenance and spare parts. This approach, a new business mode, is a response to a necessity for business-to-business collaboration to gain economy–of-scale partnerships in the extended enterprise and ultimately to be able to develop competitive offers [1] [2]. Hence, the shift in view is a move towards providing functions, taking a lifecycle commitment for the hardware as well as optimizing the availability of its function in the customers' system. The redirection from hardware development to a process where development of functions is in focus is hereafter referred to as Functional Product Development.

The challenge when merging the business and technology domains into total offers that are to be developed in the FPD process is that new tools and methods have to emerge that concurrently take care of all the aspects of the lifecycle of the provided function, i.e. adding to the existing complexity of managing all parameters of a provided function. The paper discusses an activity based modeling and simulation approach to the functional product development process where hardware development activities are combined with service activities into a FPD simulation system. The activity based simulation approach is realized in the industry standard simulation environment Matlab. For example, effects of a design change, captured as an activity in a simulation tool, could be displayed in different ways to people with different functions in the design team or its management. In this setting, the challenge in getting a culturally and functionally non uniform distributed team to collaborate effectively over distance lies partly in being able to describe the design processes and identifying what activities that truly are important for, for example, the through put time of the functional product development process. This, in turn, requires partly new collaborative tools and methods for collaborative work.

References

[1] Löfstrand.M., Larsson, T., Karlsson, L.: Demands on Engineering Design Culture for Implementing Functional Products, Proceedings of the 15th International Conferance on Engineering Design, Melbourne, August 15-18, 2005

[2] Alonso-Rasgado, T., Thompson, G., Elfström, B.: The design of functional (total care) products, Journal of Engineering Design, Vol.15 no.6, December 2004, pp. 515-540.

THE JABC APPROACH TO COLLABORATIVE DEVELOPMENT OF EMBEDDED APPLICATIONS

Martina Hörmann, Thomas Mender, Michael Schuster, Hong Trinh
IKEA IT Germany
martina.hoermann@memo.ikea.com
hong.trinh@memo.ikea.com

Tiziana Margaria
Universität Potsdam
margaria@cs.uni-potsdam.de

Ralf Nagel, Bernhard Steffen
Universität Dortmund
ralf.nagel@cs.uni-dortmund.de
steffen@cs.uni-dortmund.de

We address the design of an integrated document management system inside a supply chain management application provided by IKEA IT Germany. Our approach to the model-driven collaborative design of this system uses the jABC [1] for model driven mediation and choreography, to complement a RUP-based [3] development process. Although a widely recognized development best practice, RUP with UML does not adequately cover our needs: while the whole project addresses processes, most of the models support primarily a static view instead of a behavioral description. As a consequence, the impact of changes to an individual process model remains undetected.

Although this is not an embedded system in the more traditional sense, it involves a number of characteristics that make it a kind of *system of embedded systems*:

- the document management system sits in the background, essentially as a controller/ executor serving the overall delivery process.
- it has strict real time constraints. Admittedly, some of them are on a different time scale as usual for embedded systems, but they are as stringent and as business critical as in traditional embedded systems.
- its reliability is business critical. In particular this requires the integration of flexible mechanisms for fault tolerance.
- it is realized as a network of platforms, ranging from pure data management to systems steering the loading and unloading of vehicles or monitoring the progress of a shipment.

jABC [1], a flexible framework for service development based on Lightweight Process Coordination [2], allows users to easily develop services and applications by composing reusable building-blocks into (flow-)graph structures. An extensible set of plugins provides additional functionalities that adequately support all the activities needed along the development lifecycle, like animation, rapid prototyping, formal verification, debugging, code generation, and evolution. jABC does not substitute but rather enhance other modelling practices like the UML-based RUP, which is used in our process to design the single components. In particular, global modelling and verification plugins make it possible to check consistency vertically, horizontally, and over changes.

References

[1] jABC Website: www.jabc.de

[2] Margaria, T., Steffen, B.: Lightweight coarse-grained coordination: a scalable system-level approach. STTT 5 2-3 (2004) pp. 107-123.

[3] Rational Unified Process. http://www-306.ibm.com/software/awdtools/rup/

THE VALTECH COLLABORATIVE ENVIRONMENT

Xavier Warzee
Valtech Technologies Consulting
Grande Arche de la Défense
92044 Paris la Defense
+33 (0)6 62 60 87 87
xavier.warzee@valtech.fr

Introduction

This paper presents the current effort from Valtech to build a collaborative cockpit to enable global agile software project delivery, bridging the gap between project stakeholders and the development team. Software projects are more and more developed in a distributed way. This phenomenon leads to shorter development cycles to check the status of projects and detect as early as possible failures to meet customers' expectations. Two levels of responses exist to manage these distributed projects: agile project management and software factory. In our paper, we will focus on software factory solutions.

The Software factory

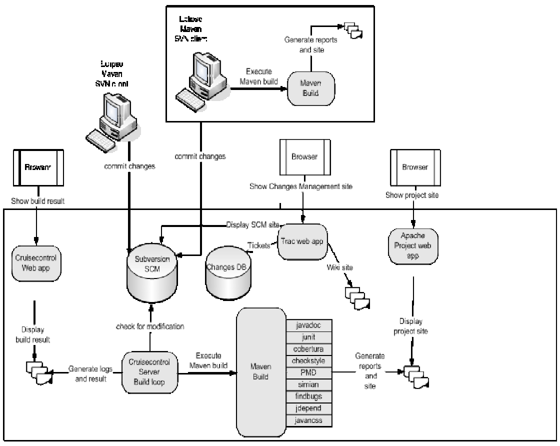

Figure 1 Software Factory architecture

Software factories mean here test driven development, automation of tests, continuous integration and ease of continuous improvement of code, design and architecture. We will present how "opensource" solutions, such as CruiseControl, Maven, Trac, JUnit, Subversion, etc., combined together with the **SCRUM** method may help to offer a powerful collaborative environment to manage distributed teams for software developments.

References

[1] Martin Fowler, Matthew Foemmel: Continuous Integration (http://www.martinfowler.com/articles/continuousIntegration.html)

[2] Mike Roberts: Enterprise Continuous Integration Using Binary Dependencies, Conference XP 2004

[3] Ken Schwaber, Mike Beedle: Agile Software Development with SCRUM, Prentice Hall.

CO-OPERATIVE & VIRTUAL ENGINEERING: A BROAD ROADMAP

Jan Goossenaerts[1], Christian Brecher[2], Frank Possel-Dölken[2], Keith Popplewell[3]
1 Eindhoven University of Technology, Faculty of Technology Management,
P.O. Box 513, Paviljoen D12, 5600 MB Eindhoven, Netherlands
j.b.m.goossenaerts@tm.tue.nl

2 Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Steinbachstr. 53B, 52074 Aachen, Germany
f.possel-doelken@wzl.rwth-aachen.de

3 Coventry University, Manufacturing and Systems Design Research Group,
Priory Street, Coventry, CV1 5FB, United Kingdom
k.popplewell@coventry.ac.uk

Abstract

The virtualisation and the distribution of product development activities have lead to new challenges for organisation and technology in European companies. The global competitiveness can only be assured by developing effective and lasting strategies for creating and managing innovation. The IMS NoE Special Interest Group 6 has analysed problems of co-operative engineering as well as methods and tools of virtual engineering of extended products.

Based on these analyses a multi-stakeholder roadmap is proposed that articulates public and private sector roles in coping with future engineering challenges. The public sector role targets the creation of a knowledge intensive global business eco-system conducive for balanced private sector innovation and sustainable growth. The private sector roles evolve under strategies that implement proven co-operative and virtual engineering practices with a focus on value creation.