

# Evidence-Based Research for Improving the Work Life of Globally Distributed Teams

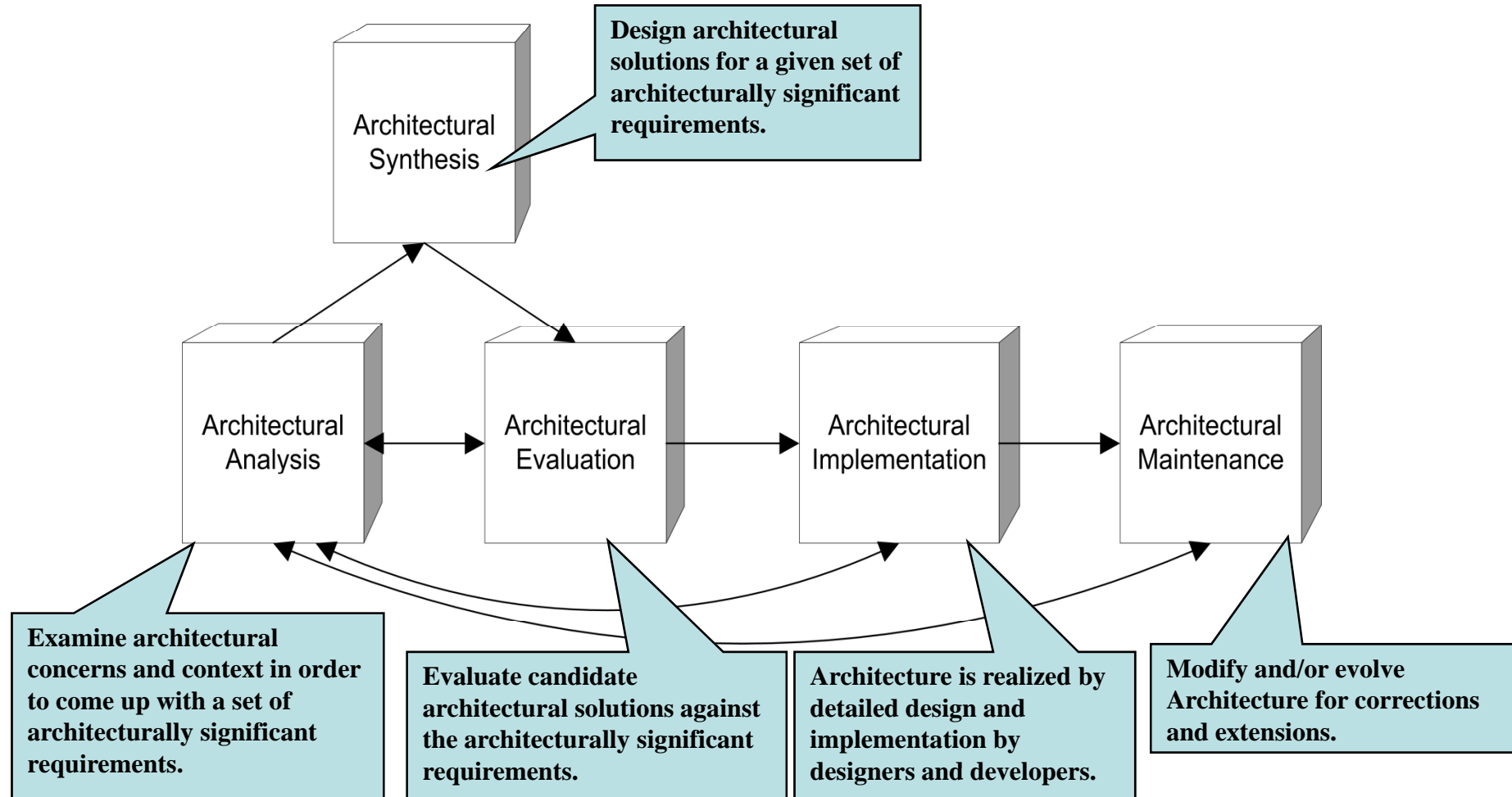
**M. Ali Babar**

*Associate Professor,  
Software Development Group  
IT University of Copenhagen, Denmark*

# *Main Threads of Interest*



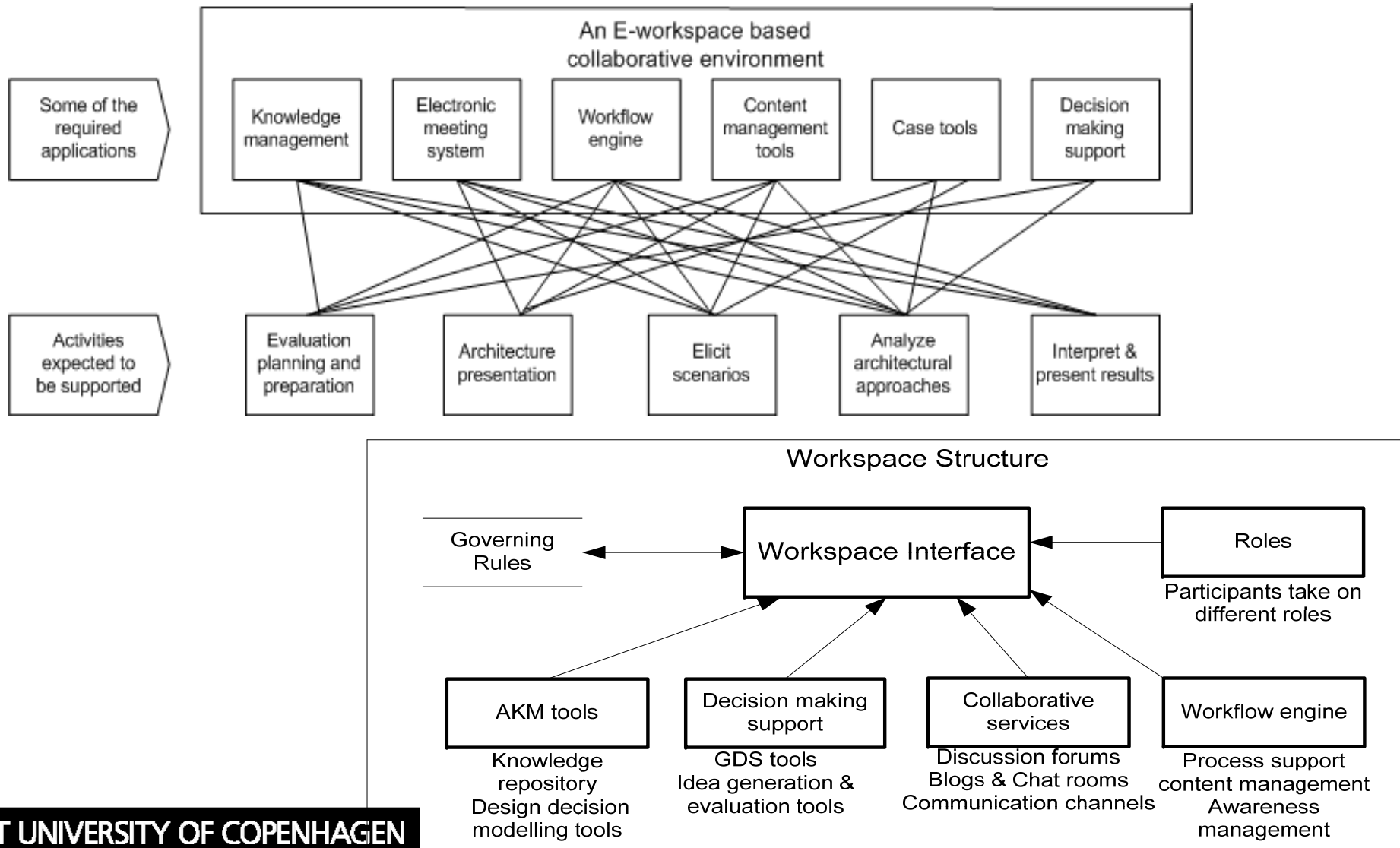
# *Focus on Architecting Process*



# *Architecture Evaluation in GSD*

- Context
  - Software architecture evaluation is an effective approach to finding erroneous and sub-optimal design decisions early
- Challenges
  - Architecture evaluation involves face-to-face meetings
  - software development teams are usually distributed
  - Collocating teams is time-consuming and expensive
  - Little support to address issues like conformity pressures, dominating personalities and cultural differences
- Proposed solution
  - Use of groupware can provide a cost effective and efficient mechanism to support architecture evaluation process

# Groupware Supported Process Model



# A framework for groupware-supported software architecture evaluation process in global software development

Muhammad Ali Babar<sup>\*,†</sup>

*Software Development Group, IT University of Copenhagen, Copenhagen, Denmark*

## SUMMARY

A Software Architecture evaluation process requires a large number of stakeholders to be collocated for evaluation sessions. Given an increasing trend to using globally distributed software development teams, organizations are likely to be discouraged from introducing disciplined software architecture evaluation practices that require collocated stakeholders. To address this issue, we propose that a software architecture evaluation can be carried out using suitable groupware systems. In this paper, we present a framework for supporting the software architecture evaluation process using a groupware system. The framework highlights the changes that are required in the existing software architecture evaluation methods. We provide an illustrated example of modeling and mapping the activities of the presented process on electronic workspaces provided by a groupware system. We also identify some of the features that a groupware system should provide to successfully support the process. Copyright © 2010 John Wiley & Sons, Ltd.



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)



The Journal of Systems and Software 79 (2006) 912–925

 **The Journal of  
Systems and  
Software**

[www.elsevier.com/locate/jss](http://www.elsevier.com/locate/jss)

## An empirical study of groupware support for distributed software architecture evaluation process

Muhammad Ali Babar \*, Barbara Kitchenham, Liming Zhu, Ian Gorton, Ross Jeffery

*Empirical Software Engineering, National ICT Australia, University of New South Wales, Sydney, Australia*

Received 6 December 2004; accepted 20 June 2005  
Available online 2 February 2006

DOI 10.1007/s10664-007-9052-6

---

## Comparing distributed and face-to-face meetings for software architecture evaluation: A controlled experiment

Muhammad Ali Babar • Barbara Kitchenham •  
Ross Jeffery

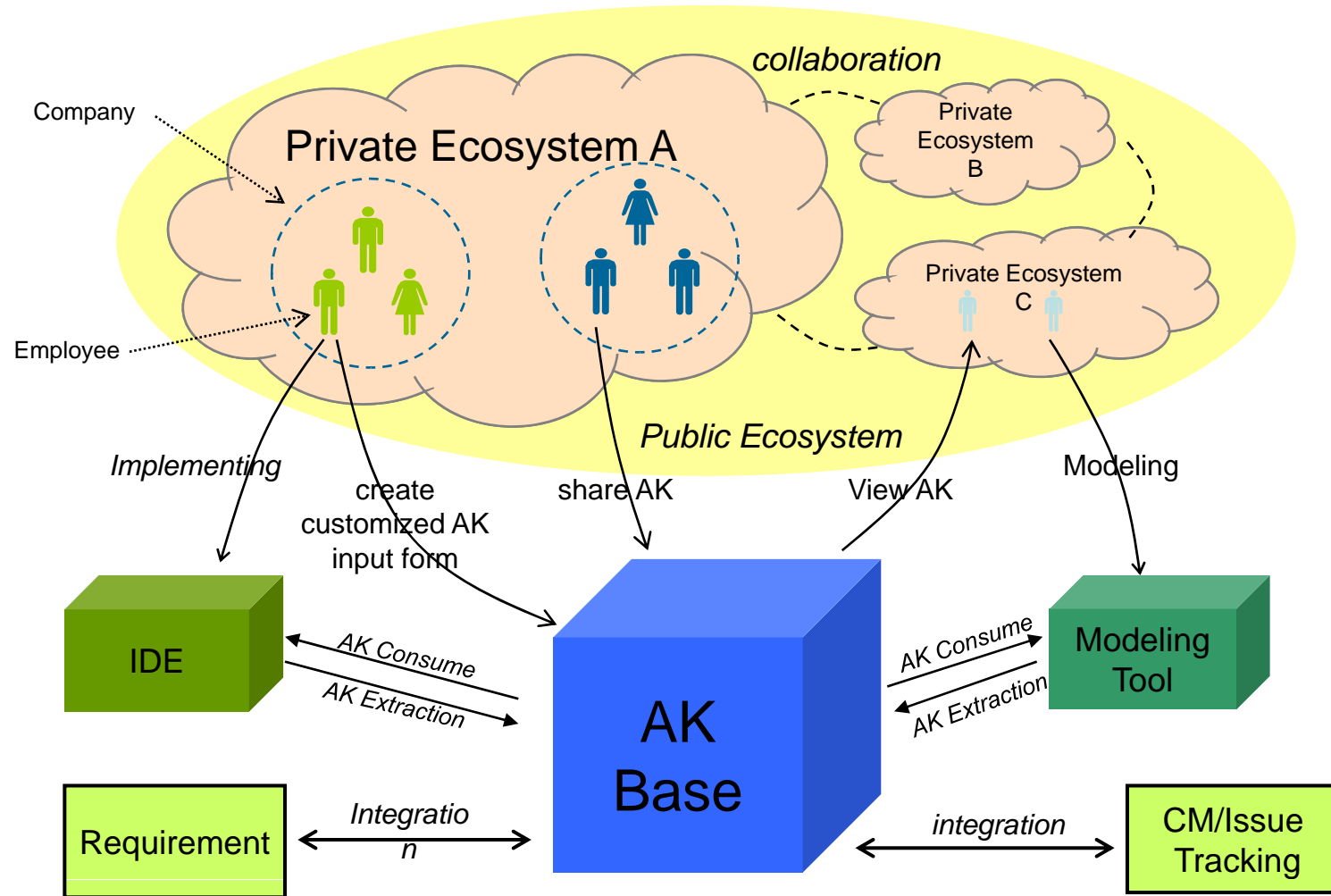
IT UNIVERSITY OF COPENHAGEN

# *Architectural Knowledge*

- Context
  - Software engineering is a knowledge intensive, architecting is more dependant on knowledge workers who go home - knowledge vaporates or grows thinner
- Challenges
  - Knowledge underpinning key design decisions hard to find
  - Capturing and organizing design decision knowledge is one of the key challenges in SA
  - Global distribution and involvement of multiple organization make knowledge flow harder
- Proposed solution
  - Building and evolving architectural knowledge ecosystems within and across organizational borders



# Architectural Knowledge Ecosystems



# Foundational Pieces of Work

The Journal of Systems and Software 83 (2010) 352–370



Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: [www.elsevier.com/locate/jss](http://www.elsevier.com/locate/jss)



## Model-Centered Customizable Architectural Design Decisions Management

Lianping Chen  
Lero—The Irish Software Engineering Research Centre  
University of Limerick  
Limerick, Ireland  
[Lianping.Chen@lero.ie](mailto:Lianping.Chen@lero.ie)

Muhammad Ali Babar  
Software Development Group  
IT University of Copenhagen  
Copenhagen, Denmark  
[malibaba@itu.dk](mailto:malibaba@itu.dk)

## A comparative study of architecture knowledge management tools <sup>☆</sup>

Antony Tang <sup>a,\*</sup>, Paris Avgeriou <sup>b</sup>, Anton Jansen <sup>b</sup>, Rafael Capilla <sup>c</sup>, Muhammad Ali Babar <sup>d</sup>

<sup>a</sup> Swinburne University of Technology, Melbourne, Australia

<sup>b</sup> University of Groningen, Groningen, The Netherlands

<sup>c</sup> Universidad Rey Juan Carlos, Madrid, Spain

<sup>d</sup> University of Limerick, Ireland

Haiqi Liang  
China Software Development Lab  
IBM  
Beijing, China  
[lianghq@cn.ibm.com](mailto:lianghq@cn.ibm.com)

### ARTICLE INFO

#### Article history:

Received 14 July 2008

Received in revised form 24 August 2009

Accepted 24 August 2009

### ABSTRACT

Recent research suggests that architectural knowledge, such as design decisions, is important and should be recorded alongside the architecture description. Different approaches have emerged to support such architectural knowledge (AK) management activities. However, there are different notions of and empha-

*Abstract*—Architectural Design Decisions (ADD) form a key element of Architectural Knowledge (AK), which plays a vital role in the software architecture process. To help manage ADDs, several tools have been proposed. However, most of them have prescribed fixed data models to be followed and do not provide sufficient customizability. Mismatches between a

required ADD model in a particular working situation can be quite large [12, 13].

Prescribing users to follow a fixed ADD model that does not fit to their needs can cause significant problems. People are forced to adapt their way of thinking and describing that thinking according to a particular prescriptive ADD model.

## Tool Support for Automating Architectural Knowledge Extraction

Aman-ul-haq, Muhammad Ali Babar  
Lero, University of Limerick Ireland  
[Aman.ul.haq@lero.ie](mailto:Aman.ul.haq@lero.ie), [malibaba@lero.ie](mailto:malibaba@lero.ie)

### Abstract

Development of large projects is a knowledge intensive task. Applying knowledge management techniques to project activities can enhance productivity and reduce risks of failures. However, it has been observed that knowledge management activities suffer from problems such as unavailability of structured information and lack of incentives to put

importance of AK, many organizations have started paying more attention towards codification of architectural knowledge underpinning their architectural processes and artefacts [2]. Architectural information is usually documented in lengthy documents which pose problems such as: 1) locating relevant information inside a long document is time consuming and difficult task; and 2) traceability among different architectural elements is lost. To overcome these problems, research

## The Application of Knowledge-Sharing Workspace Paradigm for Software Architecture Processes

Muhammad Ali Babar

Lero, the Irish Software Engineering Research Centre, University of Limerick, Ireland  
[malibaba@lero.ie](mailto:malibaba@lero.ie)

### ABSTRACT

Management of architectural knowledge is vital for improving an organisation's capabilities in software architecture. Recently, there have been many efforts to develop various kinds of tools for managing architectural knowledge. However, most of these efforts overlook the fact that most of the working teams in today's organisations are distributed. This paper proposes the application of electronic workspace paradigm for capturing and sharing knowledge to support the software architecture processes.

Though, researchers emphasized the importance of Design Rationale (DR) in software design and the challenges involved in documenting it [6, 7] more than 20 years ago, it was Bosch's paper [8] that has drawn significant attention to architectural knowledge management research. Apart from researchers, practitioners also reported that design decisions and their rationale are not rigorously documented [9]. Lack of suitable methodological and tool support has been described as one of the main reasons for this situation [10]. In response to the increasing

# ***Awareness among & between Teams***

- Context
  - Globally distributes teams and team members remain unaware of each other – its hard to create the team feelings
- Challenges
  - All sorts of tools (video, chats, screen sharing tools, repository and emails) are heavily used – but these can be inefficient or ineffective
  - People unaware of others' whereabouts and activities
  - Status signs can be misleading or annoying
- Proposed solution
  - Infrastructure for supporting pervasive awareness to create and maintain the feelings of teamness across borders

# Pervasive Awareness Panel

The screenshot displays three overlapping windows from the Pervasive Awareness Panel:

- PervasiveWindow:** Shows contact information for Antonio Lagrotteria (PM), including his email (lagrotteri@gmail.com), phone number (+45 21255128), and time zone (GMT +2.00h). It also displays status for Skype, MSN, and Bluetooth, along with a meeting check interface for June 12, 2010, from 10:00-12:00. A "Check Available" button and an "Add to GCalendar" checkbox are visible.
- Choose the day:** A calendar window for June 2010, with the 3rd of the month highlighted in red.
- ColleagueFrame:** A window showing a list of project participants for "Web Services": Lagrotteria (highlighted in green), Calahorra, Tell, and Pascelupo.

IT UNIVERSITY OF COPENHAGEN

- Knowledge worker's activities painted by different colours
- Awareness Information
  - Name, role, country, phone number, time zone
- Buddy Status
  - Skype, MSN, Arduino, Bluetooth status, next meeting
- Checking availability:
  - Check availability on Google Calendar events, add or delete events, choose day and time available

# ***NexGen CDE***

- Context
  - Software development teams need supportive technologies for collaboration, communication and coordination
- Challenges
  - Integrated and end-to-end support missing without using dozens of different tools
  - Some commercial tools (IBM SameTime and MS Communicator) available but across vendor integration problem and expensive
  - Not much support for Just In Time (JIT) Composition and Use of tooling services for collaboration
  - Misalignment between tools, processes, and culture
- Proposed solution
  - Service-Oriented Infrastructure for compositing and using Just-in-Time collaborative environments

# ***Acknowledgements***

- Ideas and implementation contributions from my colleagues, PhD and Masters students
  - Lianping Chen, Paolo Tell, Antonio Lagrotteria
- Global Interaction Research Initiative (GIRI)



## 6th IEEE International Conference on Global Software Engineering

Helsinki, Finland | August 15-18.8. 2011

<http://www.icgse.org>



### *Organizing Committee*

#### **General Chair**

Casper Lassenius, Aalto Uni, Finland

#### **Program Chairs**

M. Ali Babar, ITU Copenhagen, Denmark

Christian Lescher, Siemens Corporate Technology, Germany

#### **Steering Committee Coordinator**

Christof Ebert, Vector Consulting, Germany

#### **Finance Chair**

Alberto Avritzer, Siemens Corporate Research, USA

#### **Publicity Chair**

Rafael Pradladnicki

#### **Tutorials Chair**

Morten Jensen, Schlumberger

#### **Organizing Chair**

Maria Paasivaara, Aalto Uni, Finland

#### **Organizing Committee**

Tuomas Niinimäki, Arttu Piri, Mary-Ann

Wikström, Aalto Uni, Finland

#### **Webmaster**

Jyrki Airola, Aalto Uni, Finland

### *Call for Papers*

#### *Global Software Engineering Challenges for the Next Decade*

Software and IT industries are today truly global - and so is software engineering. The diversity of culture and the dispersion over time and space require novel techniques, tools, and practices from many disciplines to overcome challenges and to take advantage of opportunities global software engineering offers.

The 6th International Conference on Global Software Engineering (ICGSE) will bring together researchers and practitioners interested in exploring how globally distributed teams work and how the challenges can be met.

The ICGSE 2011 organizing committee invites the academic and industrial communities to contribute to the expertise of the international global software engineering community. Conference attendees will get an opportunity to explore the current state of practice in this area as well as new thoughts and trends that will shape the future. In 2011, to increase the focus on industry, we are including an industry track, for which papers will be separately reviewed.

Topics of interest include, but are not restricted to:

- Strategic issues in distributed development: business models (offshoring, nearshoring, outsourcing etc.), business case development, work distribution models, supplier selection and management, cost-benefit-risk analysis
- Managing distributed software development: planning, team building, project and SLA management, managing diversity, measurements and evaluation
- Methods and tools for distributed software development requirements engineering, design, coding, verification, testing and maintenance
- Communication, coordination and collaboration
- Knowledge management in distributed development
- Getting started with global software engineering
- Empirical studies of distributed teams and lessons learnt
- Software Process paradigms in the global environment (planned, agile etc)
- Teaching global software development

### *Paper submission*

Submissions are invited for technical papers describing unpublished, original work, in any of the following categories:

- **Research papers** that describe theoretical frameworks, technical solutions, or empirical studies. Maximum length 10 pages.
- **Industry papers** that describe problems, lessons learned, solutions implemented and challenges encountered in industrial global software engineering environments. Maximum length 5 pages.
- **Industry extended abstracts** that summarize global software engineering approaches and experiences in industry. Maximum length 2 pages.
- **Educational papers** that report experiences in teaching global software engineering. This includes not only education at the university level, but also corporate training. Maximum length 5 pages.

All submissions will be handled electronically through [www.icgse.org](http://www.icgse.org). Papers should be submitted as PDF files which conform to the two-column IEEE CS Press format (<http://www.computer.org/portal/site/cspress/index.jsp>). Accepted papers will be published in the conference proceedings by IEEE CS Press and be available in the IEEE CS Digital Library. Selected papers will be invited for submission to a special issue of XXX journal. Conference details are available at <http://www.icgse.org>.