# Coordination and Collaboration for Globally Distributed Teams: The Case of Component-Based / Object-Oriented Software Development

Julia M. Kotlarsky[1], Kuldeep Kumar[1,2], Jos van Hillegersberg[1]

[1]Department of Decision & Information Sciences, Erasmus University
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
E-mail: {jkotlarsky, jhillegersberg}@fbk.eur.nl
Phone: +31-10-4081757, Fax: +31-10-4089010

[2]College of Business, BA 356, Florida International University,
University Park, Miami, Florida 33199, USA
E-mail: kumark@fiu.edu, Phone: +1-305-3483156, Fax: +1-305-3486331

## 1. Motivation and Theoretical Background

Global distribution of software development has become widespread over the last decade. There are a number of economical and technical trends that are likely to further accelerate the growth of distributed software development. On the technological side, ongoing innovations in information and communication technologies, made by eliminating the perception of distance, make it possible to cooperate in software development projects in a distributed mode. However, currently there is limited research and experience in this emerging area. The few existing studies ([1-3]) report numerous problems in distributed projects. The time, governance, infrastructure, and culture gaps, associated with the geographical dispersion of work, make it more difficult to manage inter-site work dependencies and to coordinate and control the distributed work. Furthermore, traditional *coordination* and *control mechanisms* are less effective in global software projects. Carmel ([1] and Van Fenema [3] suggest that these traditional mechanisms will be effective for dispersed projects only with appropriate Information and Communication Technology (ICT) support. However little is known of the success of current ICT and methodology support within globally distributed software development projects. Furthermore, current development methods do not accommodate the distribution of development activities across multiple sites. Recent trends in the software development industry in the use of Object technology and a modular or component-based architectures allow distribution of development activities over multiple sites. However knowledge is lacking on how to effectively distribute Component Based (CB) / Object-Oriented (OO) development over multiple sites and what "collaboration patterns" to use for the various activities and tasks. Therefore, *our research interest is to explore and improve the practice of development of CB/OO software systems in a globally distributed setting.*

## 2. Research Objectives and Questions

The main research question we address is ***what support (technical, methodology, and social) is needed for coordination and collaboration of distributed teams in Component-Based/Object-Oriented software development?***

Our first objective is to elicit ***Coordination and Collaboration (C&C) patterns for distributed teams developing CB/OO software.*** Following traditions of software development community (e.g. [5], [6]) by patterns we mean "best practices" explicitly represented by means of templates (models or working instructions) that could be used for coordination and collaborations of distributed teams in well-defined (easily identified/recognized) situations or for resolving well-defined issues.

- C&C patterns are to be based on issues and problems distributed teams face in practice.
- They will provide solutions/ways that are based on "best practices" or existing successful practices.
- C&C patterns will incorporate the following elements: Activities, Actors (roles), Technology, Artifacts and guidelines regarding human issues.

The second objective is to develop ***a (meta)-model of coordination and collaboration for distributed teams*.** This meta-model will provide the overall framework (architecture) for the socio-technical support for globally distributed software teams.

To study existing practices and elicit C&C patterns we have defined three sub-questions:

**Q1** What are the collaboration issues for distributed teams? What problems do they face?

In greater details - we are interested in learning how distributed teams developing CB/OO software collaborate and coordinate their work: what issues/problems do they face? What successful and non-successful practices could be identified? What procedures, methods and tools are used (when-in which situations and for which activities)? What does matter for members of distributed teams (regarding their counterparts)?

**Q2** How teams collaborate at present - "good" and "bad" practices? Why they work and Why don't? Are there alternatives?

**Q3** How can existing solutions/ways be improved and under which conditions?

## 3. Research Methodology

Research methodology we have followed is a qualitative case study methodology [7].

We are conducting case studies in four different companies (each company is a "case study"), at each company we study in depth one to three projects. For the moment we have finished data collection in two companies and collecting data in two other companies in parallel. The first case study was conducted in the company developing software between USA and Europe. Other three case studies are in companies (projects) distributed between India, Europe and USA.

During the case studies we are collecting input and gaining the perspective of two (or more) remote groups participating in the overall project from their dispersed locations. Project is unit of analysis. Criteria for choosing projects – project distributed between at least two locations, preferably: completely new CB/OO development, specification-design and/or integration stage, current project. We collect data from the following sources:

- **Documentation & Records** – paper-based or digital documents and records that provide information on the project and sites involved (from corporate and public sources).
- **Direct observation –** attending meetings (e.g. video and phone conferences, net meetings) and other communications between sites as far as feasible.
- **Interviews** – face-to-face semi-structured interviews of project managers, team leaders and project team members, in particular people cooperating across project locations (selected to provide a heterogeneous perspective on the project). We interview about 5 people from each project. In three out of four companies we have interviewed people involved in two-three different projects, therefore in total we have about 15 interview per each of theses three companies.

While collecting empirical data we follow iterative approach – after each case study we come back to the initial theory-based categorization and modify and expand on issues and patterns based on our observations and data collected. Appendix 1 describes the overall research design.

## 4. Initial Empirical Findings

Based on the conducted case studies we have identified several factors that influence ways distributed teams collaborate. These factors could be distinguishes as *flexible factors* (those could be changed) and *fixed factors* (those cannot or hardly ever possible to change). The factors are presented at Figure 1 (*fixed factors* are in green and *flexible factors* are in blue). Furthermore, some of the factors are *intra-organizational (internal) factors* (they appear within the circle) and some are *out of organizational scope (external) factors* (they are outside the circle).

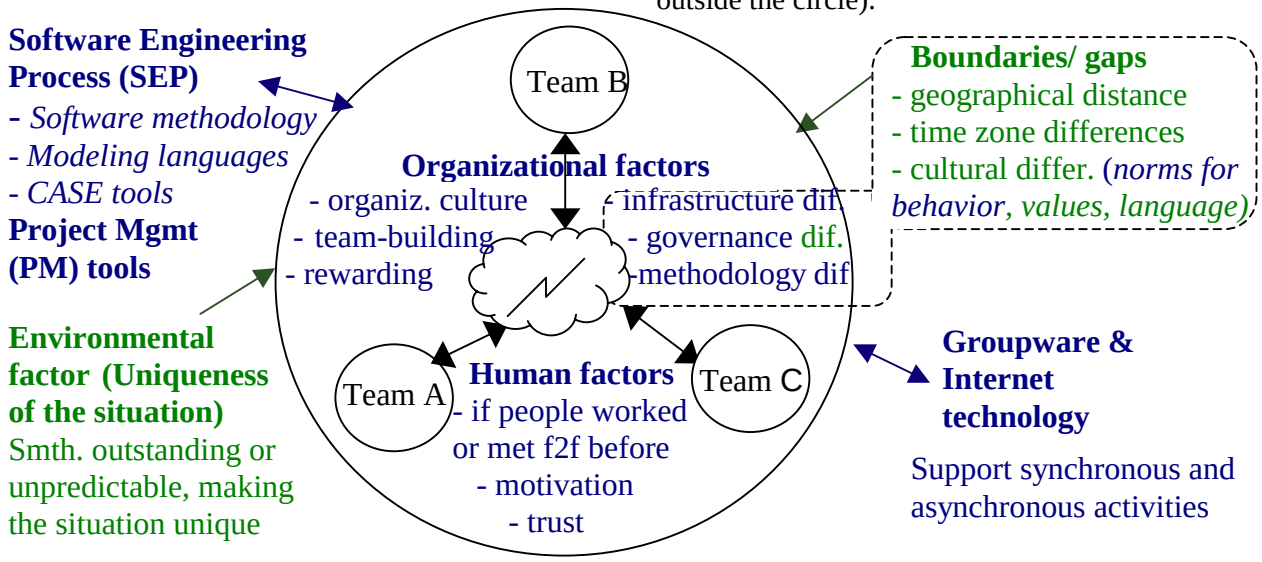



Figure 1: Factors that influence ways distributed teams collaborate.

We will explain how these factors influence way teams collaborate:

Some of **boundaries/gaps** – *geographical distance* and *time zones differences* – are external and fixed factors. Studying distributed teams means that distance is a pre-condition – there is always distance between sites. Regarding time differences – time window when working hours overlap defines when synchronous communications are possible. In the case of Australia and New York or India and San Francisco – there is no overlap is working hours. Therefore teams working with half-day overlap (e.g. India and The Netherlands) will collaborate differently from those where there is twelve hours time difference.
*Cultural differences* are rather fixed then flexible factor. On both - individual and national levels – norms of behavior are difficult to change, however often people who work closely with other countries somehow adjust to the "culture" of their counterparts, specifically if they did spend a long time (at least a couple of months) in the country they collaborate with. However values and language are very difficult to change (if possible at all). Therefore cultural differences is very important factor that influences ways people from different cultures collaborate.
Some of gaps are internal – these are gaps between remote sites involved in the same project. These are differences in *methodologies* (or processes followed), *governance* (differences in management structure, rules, implicit practices) and *infrastructure gaps* that include access to basic facilities like electricity and telecommunications (capacity), also different (versions of) hardware, platforms, and applications. Gaps in technical infrastructures and methodologies could be reduced, even eliminated, governance differences also possible to reduce however it is quite difficult.

Next factor is **Groupware[1] and Internet technology**. This is flexible external factor – because on the one hand there are a lot of tools and technologies available on the market. On the other hand – availability of tools within project sites limit choice of people involved in remote collaboration.
The goal of Groupware is to allow remote communications and to support collaboration between people at remote locations. Table 1 summarizes Groupware capabilities (grouped based on synchronous and/or asynchronous mode of support they provide for distributed parties - teams or individuals).

| **Synchronous** | **Asynchronous** |
|---|---|
| ➢ video-conference<br>➢ phone-conference<br>➢ e-chat<br>➢ e-whiteboard | ➢ e-mail<br>➢ voice-mail<br>➢ video-mail<br>➢ calendar/scheduling<br>➢ discussion list<br>➢ e-workspace |
| **Synchronous & Asynchronous** | |
| ➢ application/document sharing<br>➢ *(remote access, file transfer, bulletin board)*<br>➢ workflow management | |

Table 1: Groupware capabilities

**Software Engineering Process (SEP)** that cover *software methodology*, *modeling languages* and *CASE tools*, and **Project Management (PM) tools** are flexible external factors. Similarly to the previous factor (Groupware and Internet technology), tools and methodologies available within project organization limit choice of what can be used for collaboration between distributed teams. At the same time this "limited" choice could be changed by acquiring tools from the variety available at the market, by developing internal tools and/or adopting or developing new methodologies/techniques.
Two of the four companies we are studying developing CB software using OO approach, other two companies develop CB but not OO software. Therefore we will be able to analyze if teams developing OO software coordinate and collaborate differently from those not using OO approach (and if yes – what are the differences).

Within project organization we have distinguished between two (internal) factors they apply on different levels - organizational and individual levels - **Organizational** and **Human** factors respectively, both factors are flexible.

On the individual level we have seen that people who *met their remote counterpart face-to-face* at least for a short period of time and have some history of working jointly from different locations – they collaborate more closely and more successfully then team members who never met face-to-face. Personal contact and joint working experience help to build *trust* between remote parties. Furthermore, *motivation* on the individual

[1] By Groupware we mean concept of Groupware technology and not any specific tool or vendor. There are many different tools by different vendors that provide same capabilities/features.

level is very important for successful collaboration between distributed teams.

On the organizational level *organizational culture*, *team-building* activities (e.g. if they are taking place, what kind of activities, if they are part of organizational culture), *rewarding* (not about money but rather personal recognition etc.) are very important for successful collaboration between remote team members.
*Methodology*, *infrastructure* and *governance gaps* (appear within the circle at the Figure 1) could be considered as organizational factors as well because these are gaps on the organizational level between the sites involved in the same project.

One more factor that should be taken into account while studying how distributed teams work is **Environmental factor** (i.e. **Uniqueness of the situation**) – something outstanding, unpredictable, making the situation (project) we are studying unique or exceptional. It can be something related to product, e.g. very special (important for organization) product or something completely unpredictable, out of control (e.g. September 11$^{th}$ events). These factors affect a way teams work, in a multinational team it might have either positive or negative influence. Positive – for instance unite remote team members if they feel ownership of the product and developed it as it is their "child". Or negative - like September 11th events created a lot of tensions in multi-national teams.

For each project we study as part of empirical data collection, we cover all the above mentioned factors in order to understand better when (under which scenarios) the collected practices could apply. Furthermore, we are tracking variables such as size of company and project, experience level of team members and other factors that can influence success or failure of a project and necessary for conducting cross-case analysis.

## 5. Background for Data Analysis and Further Steps

To study and benchmark globally distributed projects (as a theoretical background), we have developed a Framework of Global Software Development Project (Figure 2) that identifies *functional areas* on which distributed teams collaborate during various stages of a project [4]. We use these *functional areas* as a base to group issues, coordination and collaboration patterns and tools we identify during empirical data collection.

The Framework identifies and ties together different elements of global software projects - product, methodology, project organization (i.e. people) and plans, and relevant coordination and control activities.

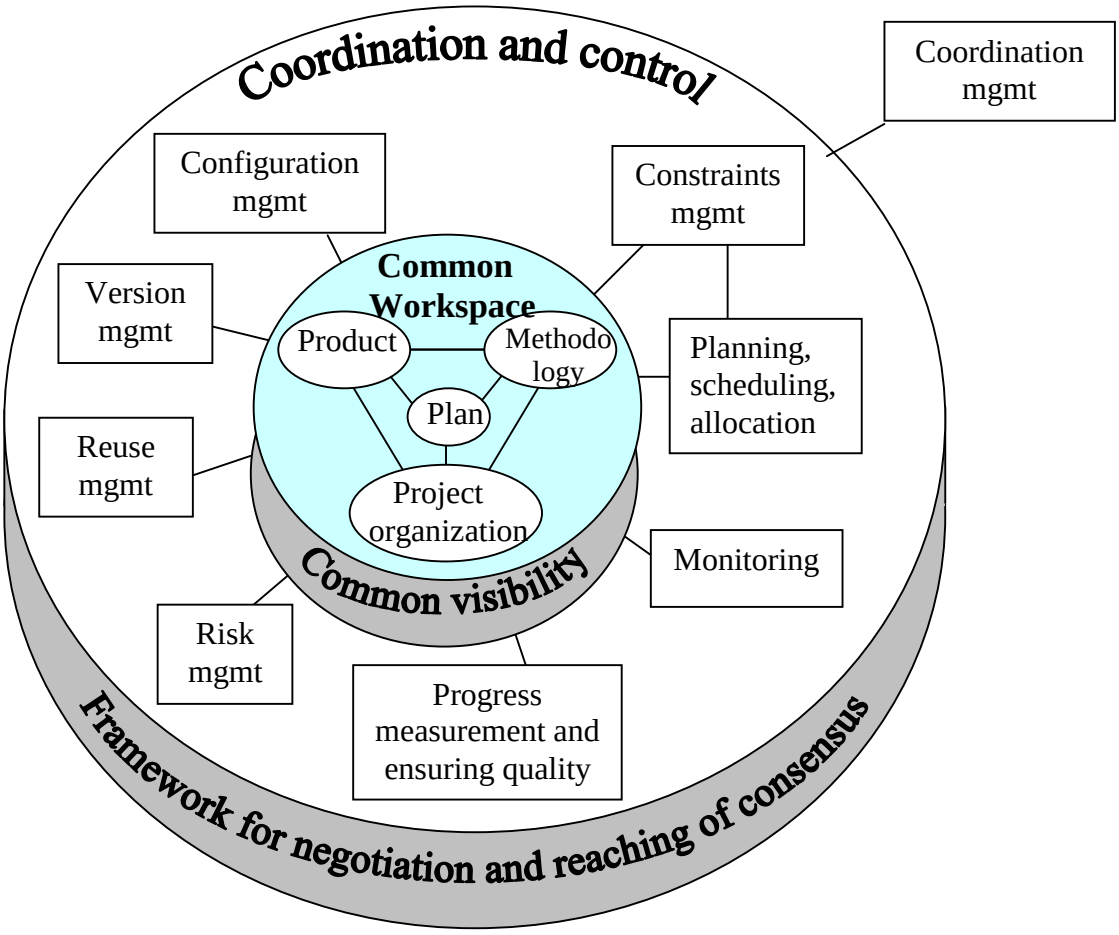



Figure 2: The Framework of Global Software Development Project

As for now, we have finished data collection in two companies and "half way" in two other companies. We have visited one of two remote sites at each of two other companies. The two sites left will be visited in May-June 2002.
At the moment we are transcribing the collected interviews. After all cases are completed and interviews transcribed, we will analyze the collected practices - each case separately and cross-case analysis [8]) and available theories in order to elicit Coordination and Collaboration patterns for distributed teams.

While analyzing existing practices we are collecting, we group them based on two following categories:
1) Which activity (functional areas) and/or elements of global software projects does it apply (based on the Framework, Figure 2).
2) State (i.e. scenario) of each of the factors presented at Figure 1. This is important in order to analyze which practice/pattern applicable at which situation.

During the workshop we will present some of the practices that we have observed during data collection and describe how and under which circumstances (scenarios) they are working.

## 6. Implications of the Research

Our research is relevant for both research and management practice.
In terms of practical contribution, we will provide guidelines for distributed software teams based on "best" or "successful" practices.
Tool vendors - commercial or academic (e.g. OPEN [9], [10]) – we will provide with requirements of distributed software teams - what capabilities teams need, what is missing in the existing tools they use now.
For scientific community our research will be an exploratory study of quite a new research area - global teams developing CB/OO software. Meta-model and coordination and collaboration patterns will be our scientific contribution.

## 7. Expectations from the Workshop on Global Software Development

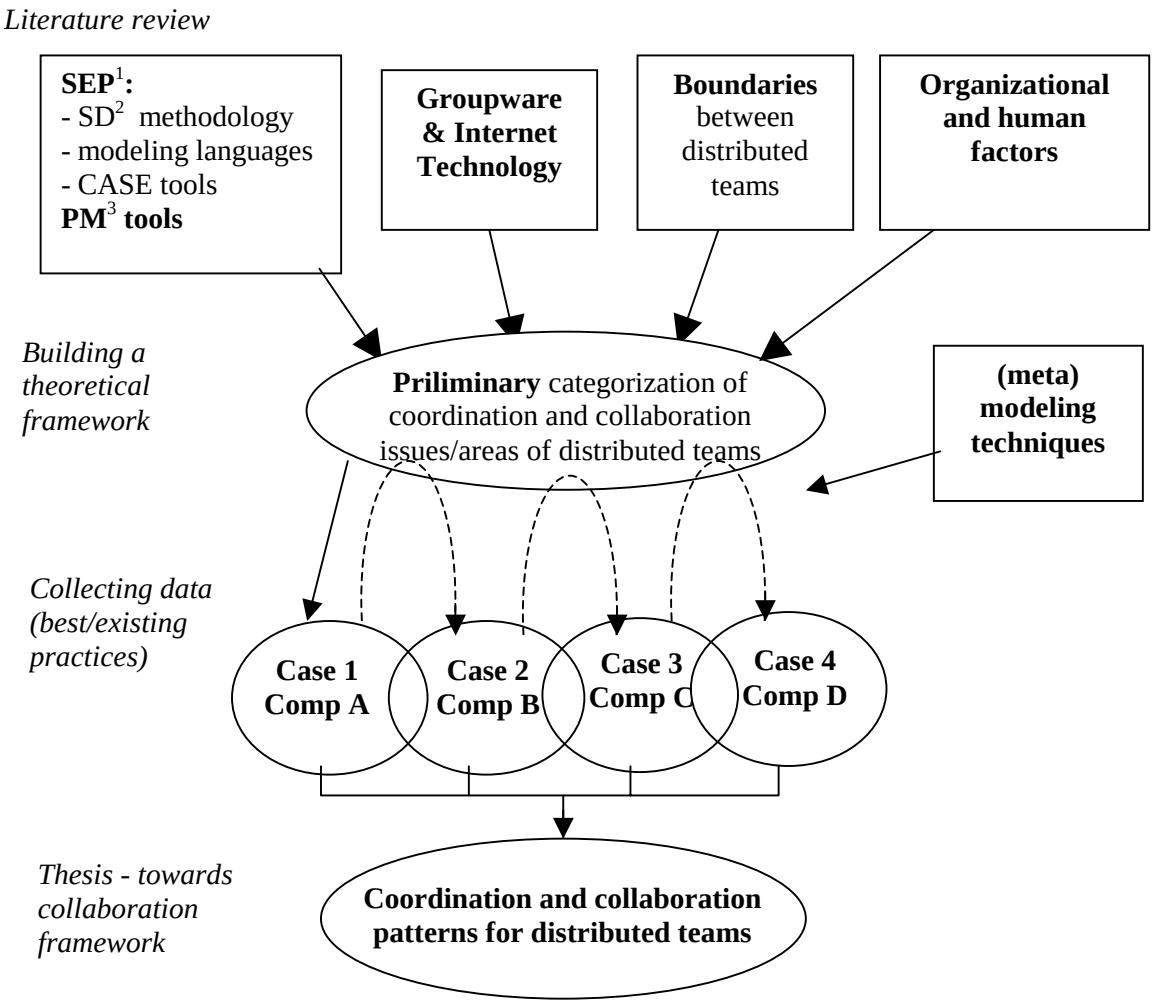
We are very interested in meeting people involved in global software development and researched working in these fascinating and quite new area.
From practitioners we are interested to hear about their experience of working in distributed teams - what problems do they face, success and failure stories (and why it is succeeded or failed) etc.
From other researches – what interesting observations did they come across while studying distributed software teams and findings. How are they researching a topic – theories and research methodologies used.

## 8. References


1. Carmel, E., *Global Software Teams: Collaborating Across Borders and Time Zones*. 1st ed. 1999, Upper Saddle River, NJ: Prentice-Hall P T R.
2. Karolak, D.W., *Global Software Development: Managing Virtual Teams and Environments*. 1st ed. 1999.
3. Van Fenema, P.C., *Coordination and Control of Polycontextual, Geographically Dispersed Temporary Systems: The Case of Global Software Projects.*, in *Department of Decision and Information Sciences*. 2002 (forthcoming), Erasmus University, Rotterdam School of Management: Rotterdam, The Netherlands.
4. Kotlarsky, J., J. Van Hillegersberg, and K. Kumar. *Towards a Global Software Development Architecture*. in *2001 IRMA International Conference*. May 2001. Toronto, Canada.
5. Fowler, M., *Analysis Patterns*. 1996: Addison Wesley.
6. Gamma, E., *et al.*, *Design Patterns*. 1994: Addison Wesley.
7. Yin, R.K., *Case Study Research: Design and Methods*. Vol. 6. 1994, Newbury Park, CA: Sage.
8. Miles, M.B. and A.M. Huberman, *Qualitative Data Analysis*. 2nd ed. 1994, Thousand Oaks, CA: Sage.
9. Firesmith, D. and B. Henderson-Sellers, *The OPEN Process Framework*. 2001: Addison Wesley.
10. Graham, I., B. Henderson-Sellers, and H. Younessi, *The Open Process Specification (Open Series)*. 1997: Addison Weslev.

## Appendix 1

## Research design




*Comments:*

[1] SEP – Software Engineering Process
[2] SD – Software Development
[3] PM – Project Management