A FRAMEWORK FOR CONSIDERING OPPORTUNITIES AND THREATS IN DISTRIBUTED SOFTWARE DEVELOPMENT

Pär J Ågerfalk[1], Brian Fitzgerald[1], Helena Holmström[1],
Brian Lings[2], Björn Lundell[2], Eoin Ó Conchúir[1]

Abstract

In this paper we present an overview of the field of distributed development of software systems and applications (DD). Based on an analysis of the published literature, we consider threats to communication, coordination and control in DD caused by Temporal Distance, Geographical Distance, and Socio-Cultural Distance. The analysis results in a more complete framework for reasoning in the DD domain which should be a useful resource for both academic researchers and practitioners.

1. Introduction

Distributed development of software systems and applications (DD) is an issue of increasing significance for organizations today, all the more so given the current trend towards outsourcing and globalisation. According to the World Investment Report, 2004 [51], offshoring of IT-enabled services is forecast to expand 24-fold by 2007 from a base of $1 billion in 2002. The report also notes that while US companies have been relatively active, European companies have shown less inclination to offshore services.

There are many reasons why an organisation should consider adopting a DD model, including access to a larger labour pool and a broader skills base, cost advantage, and round the clock working. This is perhaps most evident in the many cases of outsourcing of software development to low-cost countries, e.g. [12], but is also relevant in the case of, for example, utilizing local expertise to satisfy local demands.

In ideal software development teams, members have rich interactions, both formal and informal; share a common organisational culture - which promotes good coordination and facilitates effective control; represent a good mix of all required technical skills and relevant experience, made readily accessible to all team members; and are familiar with, and provided with, homogeneous tools and technologies appropriate for the project. DD adds new demands to the software development process by potentially threatening each of these ideal properties.

In this paper we present an overview of the academic body of knowledge on opportunities and threats in distributed development, as represented in peer-reviewed research articles. The paper is organized as follows. Section 2 clarifies what we mean by distributed development. Section 3

[1] Department of Computer Science and Information Systems, University of Limerick, Limerick, Ireland.
[2] University of Skövde, School of Humanities and Informatics, P.O. Box 408, SE-541 28 Skövde, Sweden.

presents the research approach adopted. Section 4 discusses the processes and dimensions to be used within a framework for reporting the literature on DD, and introduces the framework. Section 5 populates and elaborates the developed framework in order to identify important issues in DD. Finally, Section 6 briefly summarizes and concludes the paper.

2. Characterising Distributed Development

For the purpose of this research, we choose to define "development" broadly as "any software development lifecycle activity". This thus extends beyond "pure" development activities and includes, for example, deployment and maintenance. This broad definition makes sense since we do not want to restrict our analysis strictly to new software product development. We use the term "activity" in a loose sense, including any individual or collective human action at any level of granularity that serves a particular purpose. According to activity theorist Engeström [23], an activity is something that transforms an object to an outcome. Hence, a "development activity" is that individual or collective action that transforms something (abstract or concrete) into something meaningful in the context of a software systems lifecycle. Thus, we would regard an individual developer's creation of a source code document a development activity of transforming a requirements document into a piece of code. We would also regard a complete project, transforming an initial idea of a system into a working solution with documentation, associated work-processes, etc., a development activity. This means that we can regard a project as distributed without requiring all of its sub-activities to be.

Intuitively, classifying a project or development team as distributed means the team members are not co-located, but geographically spread out; we may thus say that there is a geographical distance between actors in a DD setting. However, as we shall see below, many core aspects of DD are related not to geographical distance, but rather to what can be called a "socio-cultural distance". Socio-cultural distance has to do with the fact that different people give different meanings to a situation based on their socio-cultural background and belonging. According to Orlikowski and Gash [43, p. 176], "The frames of reference held by organizational members are implicit guidelines that serve to organize and shape their interpretations of events and organizational phenomena and give these meaning." Conflicts can arise from team members coming from different cultures – both national culture and organisational culture. National or local culture encompasses an ethnic group's norms, values, spoken language and styles of communication [46, 13]. Organisational culture encompasses the working unit's norms and values, and includes the culture of systems development [13]. Culture can have a huge effect on how people interpret a certain situation, and how they react to it. Hence, having shared (or overlapping) frames of reference is a precondition for people to succeed in communication and collaboration. At the very least, each actor needs to have an understanding of and accept the others' frames of reference, and understand that these might differ from the actor's own (i.e. agree to disagree). Certainly, geographical distance may imply increased socio-cultural distance. However, the socio-cultural distance can be great even with low geographical distribution. Similarly, a huge geographical distance does not automatically mean huge socio-cultural distance. Finally, a consequence of being geographically distributed over two or more time zones is that there is also a temporal distance involved. However, neither is temporal distance confined to geographically distributed settings. Rather, a temporal distance is present as soon as team members cannot interact face-to-face. This may be due to geographical distance, but may as well be a result of, for example, shift work.

3. Research Approach Adopted

This paper presents an overview of the field of DD. Based on an analysis of the published literature, the paper provides a preliminary analysis of DD in different industrial contexts establishing basic characteristics. Inspired by Webster and Watson [52], we develop a framework to structure existing DD knowledge and studies. This has required a two-phase process of search and refinement.

In the first phase, two parallel searches of the literature were conducted. These parallel searches were carried out relatively independently, thus achieving a form of triangulation in validating the resulting output. Each search was systematic, using keyword and author searches, and searches of tables of contents of Journals, and Conference and Workshop proceedings. Bibliographic databases were used to assist in forwards and backwards referencing. Papers were included if they had a core focus on DD (the primary list), or were considered highly relevant for understanding core issues raised in the DD literature (the secondary list). We also compiled an extensive note file, including quoted sections from papers which contained their major import. This allowed faster filtering in the later stages of analysis, but context was always checked against the full text.

In the second phase, which commenced when the two searches were complete, the compiled lists were combined. Another iteration of the search was undertaken based on the combined lists, with a further check using bibliographic databases. As the analysis progressed, sources considered redundant or less relevant were removed from the secondary list.

The full set of sources was then analysed with a view to developing a framework for compiling the key opportunities and threats considered to be inherent in DD. Practitioner literature was then consulted in order to check for congruence with the peer-reviewed sources.

Due to space limitations, we are not able to include all references in this paper, but are confined to a representative selection. The full list of references is available from the authors upon request.

4. A Framework for Analysing Issues in DD

For a number of years the international workshop on Global Software Development has highlighted the impact of distribution on *communication*, *coordination* and *control* within DD lifecycle activities, e.g. [20]. This view is consistent with the position taken by a number of authors who have focused on one or more of these three fundamental processes, e.g. [13, 26, 38, 39, 42, 50].

Communication is “the exchange of complete and unambiguous information - that is, the sender and receiver can reach a common understanding.” [13] The communication process concerns the transfer of knowledge and information between actors, and the tools used to facilitate such interaction. Communication is an essential process in all software development [18, 7] but becomes even more crucial in DD due to the fact that DD changes the communication context away from the “ideal” face-to-face setting [15] into a technology-mediated and thus more restricted one [1].

Coordination is “the act of integrating each task with each organisational unit, so the unit contributes to the overall objective.” [13] The coordination process concerns how this interaction makes actors interdependent on each other: “Two people have a coordination problem whenever they have common interests, or goals, and each person’s actions depend on the actions of the other.”

[15, p. 62] All software development obviously requires coordination, but DD increases this need as activities are distributed over time and space and across cultural borders, as we will discuss below.

Control is "the process of adhering to goals, policies, standards, or quality levels." [13] The control process concerns the management and reporting mechanisms put in place to make sure a development activity is progressing. Control thus relates to project management and hence the formalized structures required ensuring development of software in time, on budget and of desired quality.

The communication, coordination and control activities are affected over a number of *dimensions*, which have been well elaborated in the literature, e.g. [6, 8, 21, 24, 28, 41, 50]. These relate to temporal, geographic and socio-cultural distance.

Temporal distance is a directional measure of the dislocation in time experienced by two actors wishing to interact. Temporal distance can be caused by time zone difference or time shifting work patterns. When organising work patterns, note must be taken of both temporal overlap of parties, to facilitate communication, and temporal coverage, for example to move towards 24x7 activities. In fact, time zone difference and time shifting work patterns can work together to either increase or decrease temporal distance. For example, a one hour difference in time-zone within the EU can, because of different routines during a working day, lead to very few overlapping hours and an appearance of higher than expected temporal distance, but may offer increased temporal coverage. Conversely, an EU worker liasing with a counterpart in India working a late shift may experience low temporal distance, but such an arrangement will not offer increased temporal coverage. In general, low temporal distance improves opportunities for timely synchronous communication but may reduce management options.

Geographical distance is a directional measure of the effort required for one actor to visit another at the latter's home site. Geographical distance is best measured in ease of relocating rather than in kilometres. Two locations within the same country with a direct air link and regular flights can be considered close even if separated by great distance, but the same cannot be said of two locations which are geographically close but with little transport infrastructure and perhaps intervening borders. Further, even two actors within the same building but separated by long corridors and several floors will be impacted by geographical distance. Ease of relocating has several facets, including ease and time of travel, and necessity for visas and permits. How critical an actor is to the project in their home location may also implicitly affect perceived distance, as it will affect their ease of travelling. In general, low geographical distance offers greater scope for periods of co-located, inter-team working.

Socio-cultural distance is a directional measure of an actor's understanding of another actor's values and normative practices. As a consequence, it is possible for actor A to be socio-culturally closer to actor B than B is to A. It is a complex dimension, involving organisational culture, national culture and language, politics, and individual motivations and work ethics. It is possible to have a low socio-cultural distance between two actors from different national and cultural backgrounds who share a common organisational culture, but a high distance between two co-nationals from very different company backgrounds. At the very least, there is a need for an actor to understand and accept others' frames of reference, and accept that these might differ from the actor's own (i.e. agree to disagree). In general, low socio-cultural distance improves communication and lowers risk.

The complete framework forms a matrix in which each cell represents the impact of one dimension on one process. We present an overview of this framework in Table 1, and relate prominent DD issues, including opportunities and threats, to the relevant cells by way of illustration. This table should be considered in addition to the general characterisations given above for each process and dimension. Hence, each cell highlights only what is specific with respect to the affect of one dimension on one process. In Section 5, we elaborate on each cell in this framework.

Table 1: An Overview of the Framework of Issues in DD.

Process	***Dimension***		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	Reduced opportunities for synchronous communication, introducing delayed feedback. Improved record of communications.	Potential for closer proximity to market, and utilisation of remote skilled workforces. Increased cost and logistics of holding face to face meetings	Potential for stimulating innovation and sharing best practice, but also for misunderstandings.
Coordination	With appropriate division of work, coordination needs can be minimised. However, coordination costs typically increase with distance.	Increase in size and skills of labour pool can offer more flexible coordination planning. Reduced informal contact can lead to reduced trust and a lack of critical task awareness.	Potential for learning and access to richer skill set. Inconsistency in work practices can impinge on effective coordination, as can reduced cooperation through misunderstandings.
Control	Time zone effectiveness can be utilised for gaining efficient 24x7 working. Management of project artefacts may be subject to delays.	Difficult to convey vision and strategy. Communication channels often leave an audit trail, but can be threatened at key times.	Perceived threat from training low-cost 'rivals'. Different perceptions of authority/hierarchy can undermine morale. Managers must adapt to local regulations.

5. Elaborating the Framework

DD puts new demands on the software process imposed by increased complexity related to, for example, communication (formal, informal, potential lack of), coordination (time zones, social awareness, task-sharing, domain expertise, delays), cooperation (trust, teamness), control (policies, project management, power, uncertainty), culture (social, political), and technology and tools (heterogeneous technology, standardization). This means that any allocation of an issue to a single cell is necessarily arguable. The summary in table 2 therefore places some issues in several cells, as indicated in comments within the text. However, our aim is to use the framework to bring some kind of primary order out of the complex interrelationships evident from the literature.

5.1 Communication in Distributed Development

5.1.1 Temporal Distance

Time Zone Effectiveness: Although the face-to-face setting is the basic prototype for communication [15], and generally considered the best means of exchanging ideas [11], asynchronous communication (over temporal distance) can be leveraged to the distributed team's advantage. By communicating in an asynchronous manner, teams can, for example, strive for round-the-clock development [29, 31], potentially reducing the time-to-completion for the project. Also, since asynchronous communication relies on technologies such as e-mail and fax [8, 19], a written communication history is usually left [13]. This provides for increased traceability and accountability, i.e. it facilitates finding out who said what to whom, and when this was said [1]. *This is also a control issue.*

Delayed communication: Being situated across different time zones, a remotely located colleague may not be at work when their help is needed. The use of asynchronous tools over temporal distances increases the amount of time it takes to receive a response. Questions received by asynchronous communication overnight can be overwhelming for the developers beginning work in the morning [8]. The conversion of ideas into e-mail form can also increase the risk of misunderstanding [19], particularly when the content of the communication is contentious or argumentative in nature [36].

Delayed feedback: The delay in receiving a response can increase the amount of time it will take to resolve the issue at hand [8]. The problem becomes exacerbated and can drag on over days [34, 36], with increasing vulnerability costs as a result [29, 25]. It has been suggested that issues would be resolved more efficiently should the teams be collaborating co-located [8].

5.1.2 Geographical Distance

Proximity to market/customer: One advantage of DD is the possibility of being close to the target market or the customer of the product being developed [29, 31, 32]. By facilitating communication across geographical distance, distributed teams can take advantage of having software developers being placed both near the customer and in the home country, such as facilitating more effective requirements elicitation [19].

Lack of informal communication: One of the major issues highlighted in DD is the lack of informal communication that occurs within the distributed team due to geographical separation. Informal contact allows team members to develop working relationships, and allows a better flow of information about changes in the current project [33]; it is an essential part of software design and development [18, 7]. Informal contact is especially important in unstable, dynamic teams. Written documentation is inadequate when resolving misunderstandings about requirements [18, 19]. In co-located teams, informal contact, aka "coffee talk' [19], can account for about 75 minutes of the working day [33]. Both geographical and temporal distance reduces the opportunities for informal communication to take place [29, 36]. It has been found that even a small distance (30 meters) can greatly affect the level of communication between colleagues [3]. Naturally, even more attention has to be given to the effect of distance on communication in a global context.

Dependency on information and communication technologies: In DD, the dependency on information and communication technology is high. Here, technology is used for communication and, therefore, it has an impact on the most critical processes in an organisation – whether and how people communicate to coordinate their processes [49, 35]. Hence, a convenient and well working technical infrastructure for information and communication, for example, effective tools and work environments, seems to be a necessity for successful DD [22].

Increased effort to initiate contact: Having team members separated by geographical distance places a barrier on communication by increasing the effort required to initiate contact [30]. This can lead to developers taking the risk of applying minor modifications to the system without trying to make contact with the person who might have more knowledge of that part of the system [8]. As a consequence, errors may be introduced in the system, ultimately increasing the cycle time. A related factor in initiating contact is not knowing who to contact [30, 6, 9]. This can arise from the lack of informal contact with remotely located colleagues. Due to lack of informal contact, a team member cannot easily learn of the skills and precise roles of their remote colleagues.

Providing technical infrastructure: When developing software within a global context, problems can arise with global support for third-party tools being used. Battin et al. [6] found that different versions of tools were being offered in different countries by the third-party vendors. For example, the newest version of a tool was made available in the US, with older versions still being offered in other countries. Also, export regulations may prohibit diffusion of certain technology throughout the distributed team [6].

Cost of travel: Sometimes, meeting remote colleagues face-to-face is indispensable, especially in the early phases of a project. This travel can be very expensive and time-consuming [6]. Also, there may not be direct flights between the two points of travel, increasing the journey time. Furthermore, there is much more to travel-time than flight-time [6].

5.1.3 Socio-Cultural Distance

Innovation and shared best practices: A major positive effect of globally distributed development is innovation [22]. Developers from different cultural backgrounds may work together to continuously improve a product, to innovate and to improve processes. Best practices can be shared amongst developers and between development sites.

Asynchronous communication preferred by non-native speakers: Often in globally distributed development, some or all of the developers speak English only as a second language. Having to communicate in real-time over teleconferences can be overwhelming for these people, finding it difficult to keep up with the conversation [48, 36]. Asynchronous communication allows for non-native speakers to formulate their position and to check that they are making their point clear before sending the email. Thus, non-native speakers of English tend to rely more heavily on asynchronous communication. This introduces the advantages and disadvantages of asynchronous communication, as identified earlier.

Language differences and misunderstandings: While English has become the international language for business matters, language competency is still a large stumbling block for communication within and between development teams [34]. In turn, misunderstandings can arise. Even if the whole team are native speakers of the language used in a project, problems can arise from different dialects and

local accents [14]. If a major section of the team speaks a particular language natively, unlike their remote colleagues, a feeling of alienation can arise, with non-native speakers of the major language being at a disadvantage in expressing themselves [48].

Managing frames of reference: Establishing mutual understanding is important since it increases the likelihood that communication will be successful [16]. It may be difficult for culturally- and geographically-distributed teams to achieve mutual understanding. National culture can affect how negotiations are carried out and how commitments are accepted [22]. For example, in a Norwegian-Russian project, it was found that Norwegian conversation was more "low-context" that the Russians. This caused frustrations since the Russians relied more on the context of the conversation without explicitly stating some opinions [34]. In another study of a German-Canadian project, the Germans were perceived as being blunt and stubborn, while the Canadians were viewed as being laid-back, chatty and indecisive [36]. Also, practices of agreeing to working late or not can vary between countries [10]. Altogether, this can lead to, for example, unevenly distributed information within the team and the inability to make clear which part of a message is the most important. In some projects a new common frame of reference develops, including team-specific language use, in-jokes, etc [4].

5.2 Coordination in Distributed Development

5.2.1 Temporal Distance

Time zone efficiency: Temporal distance can be seen as beneficial in terms of coordination, in that coordination costs are reduced when team members are not working at the same time [25]. The producer of a unit of work can complete the work during the off-hours of the person who requested that work. In essence, coordination costs are reduced since no direct coordination takes place when two people are not working at the same time. A side effect is that although the coordination cost as such (i.e. time spent on coordination activities, waiting for task handovers, etc) may be reduced, costs related to repairing consequences of misunderstandings, reworking, etc, may increase [25].

Reduced hours of collaboration: An obvious disadvantage of being separated by temporal difference is that the number of overlapping hours during a workday is reduced between sites [6, 36, 14]. For example, a team located in the U.S. and in Ireland can have a total of 3 overlapping hours during a workday [14]. Even a one-hour time zone difference can mean many less overlapping hours. For example, with team members in Germany working from 8am-4pm with a 12pm lunch, coordinating with UK team members working from 9am-5pm with a 1pm lunch, there are only four overlapping hours in a day [30].

Synchronous team meetings difficult: Team members might have to work flexible hours in order to coordinate with their remote colleagues through real-time teleconferences, increasing the cost and effort of coordinating regularly [6].

Availability of technical infrastructure: Available technical infrastructures, and possible incapability, greatly affect performance of DD teams. For example, most change management tools do not allow 24/7 access without disturbing engineers due to back-ups and synchronisations [22].

Coordination complexity: Software development in itself is a complex task with substantial non-routine work, and coordination itself can be costly [25]. The very nature of DD projects suggest that

it is important not to rely on one person as a coordination channel between teams, since the unavailability of this person can affect inter-team communication and coordination [6]. At the very least, it's important to manage those people closely [34]. *This is also an issue in the geographical distance dimension, and to some extent in the socio-cultural distance dimension.*

5.2.2 Geographical Distance

Access to large labour pool: By coordinating development across several countries, companies can access large labour pools of skilled workers [32, 20].

Standardisation in work practices: The modularisation of work for DD requires standardisation of the software development environment, processes and practices. These standardised practices, including manuals, databases and implicit and undocumented systems, serve as points of reference to coordinate work across time and space [47]. On the other hand, allowing local variation in work practices may leverage local experience and reduce project overhead [2]. *This is also an issue in the socio-cultural distance dimension.*

Allocation of roles and team structure: In DD, there is the possibility to gain from a very large pool of expertise. In building project teams, people from different sites from all over the world can be included and project roles can be allocated to various development teams. This makes possible for a flexible team structure in that people can relocate for shorter periods, allowing for effective project management, independent of how the project is globally allocated [22]. Also, changes in allocation can adhere to the challenge of replacing isolated expertise and instead create skill-broadening tasks and effective teamwork [22]. *This is also a control issue.*

Reduced trust: Creating trust can be hindered in a DD team, since normal communication like face-to-face feedback and common experience are sources of trust which are lacking in a distributed environment [45]. Familiarity and confidence are stages of relationships that must take place before trust is formed. Achieving and maintaining trust in global teams is more difficult than in collocated teams [40]. At a distance, it is difficult to empathise with those at the other site [36]. Trust can also be corroded, for example when defects are introduced due to a developer not making the effort to contact a remote colleague before making changes to the system [8]. When there is a lack of trust, there is a lack of willingness to communicate [30]. On the other hand, studies of DD in a libre software context suggest that teams rely more on social control mechanisms than on trust [17, 27].

Lack of awareness/team spirit: The feeling of "teamness" with remote colleagues can be affected because of physical separation and lack of informal contact [6, 33, 36]. Presumably, distance affects the stages by which individuals become coherent groups or teams. Due to physical separation and lack of face-to-face contact, team members may not be aware of the details of their remote colleagues' work activities. If awareness of current work isn't spread across the whole team, misunderstandings can continue unnoticed and code conflicts can arise. It can also be difficult to determine if a remote colleague is available to be contacted at a particular time [30]. *This is also an issue in the socio-cultural distance dimension.*

Modularisation of work: According to Conway's Law, the structure of the system mirrors the structure of the organisation that designed it [30]. The nature of DD leads teams to splitting their work across feature content into well-defined independent modules [22, 47, 5]. This allows decisions to be made about each component in isolation, and reduces problems in the system integration phase

[30]. Partitioning work tasks horizontally having each site responsible for the whole lifecycle of particular functions/modules decreases interdependencies, and hence coordination costs [6]. *This is also an issue in the temporal distance dimension.*

Lack of mechanisms for creating shared understanding: Without effective mechanisms for sharing information and facilitating common understanding, managers cannot exploit the benefits of DD [32]. Inadequate dispersal of important information about a project, such as the overall architectural vision, can leave teams with a skewed perception of which tasks are on the critical path [6]. Also, with a lack of understanding of the wider system, reuse opportunities may be overlooked [32]. *This is also an issue in the temporal distance dimension, and to some extent in the socio-cultural distance dimension.*

5.2.3 Socio-Cultural Distance

Mix of skills and experiences: Globalisation, in general, achieves a constructive cross-fertilization of varying backgrounds and experiences [22], which can enrich coordination efforts between distributed teams.

Language and cultural training: An investment in language training and cultural awareness may be required if team members come from different backgrounds [34], and a compromised culture may need to be established. A "bridgehead" [13] or "liaison" [6], i.e. a person from one site working in another site and acting as a mediator between sites, may be helpful.

Lack of domain knowledge: Work on a project can require specific domain knowledge that developers coming from different backgrounds do not have [34]. Organisations can have incompatible views on a domain, based on their own particular experience and expertise [18]. For example, a Norwegian firm outsourcing work on a Norwegian tax software package to a Russian firm realised that the Russian developers did not have sufficient knowledge on the Norwegian tax system when taking on the work [34].

Doubtful of others' capabilities: Developers may be doubtful of the knowledge of team members from other sites, their capabilities and skills [6]. This impression may be overcome by promoting familiarity between teams. It has, for example, been reported that American engineers can have concerns about the competency of international engineering teams [6].

5.3 Control in Distributed Development

5.3.1 Temporal Distance

Management of project artefacts: To maintain consistency among project artefacts, a configuration management tool with centralized storage is often used. Even when working from the same central repository, it may be unclear what problems are addressed by a new version of an artefact and what status it is in (such as whether it is still being tested) [9]. Also, when a DD project involves members from different organizations (aka a virtual organization), enforcing process and artefact standards can be particularly important in maintaining consistency and interoperability between project artefacts [26]. *This is also a coordination issue.*

5.3.2 Geographical Distance

Lack of concurrent engineering principles: Synchronisation is important when teams hand off processes between sites. It requires commonly defined milestones and clear entry and exit criteria. Effectively implementing concurrent engineering principles in DD often becomes difficult because of volatile requirements, unstable specifications, the unavailability of good tools that support collaboration across time and space, and the lack of informal contact [32].

5.3.3 Socio-Cultural Distance

Perceived threat from low-cost alternatives: Employees in the higher-cost economies can feel that their jobs are under threat from their colleagues in lower-cost economies, creating a "we versus they" mentality [14]. They may see a threat to their future employment and promotion prospects – the "my job went to India and all I got was this lousy T-shirt" syndrome. As a result, they may not want to cooperate with their remote colleagues. This, in turn, affects the team's work and can compromise the benefits of globally distributed development. Apart from economic reasons, power struggles can arise between the different teams when the centre of power is not explicitly defined [36].

Adapting to local formalized norm structures: When working in a global setting, companies must learn about local formalized norm structures (applicable laws, traditions, regulations, etc). For example, applications for visas and work-permits may need to be sent some time before a trip between sites [6]. Also, different sites may prefer different development methods [26].

Different perceptions of authority/hierarchy: The nature of authority in a team environment can vary between cultures [37]. It has, for example, been found that Irish developers require their superiors to earn their respect, while U.S. developers give a more unquestionable respect to figures of authority [14].

5.4 Summary of Issues in Distributed Development

The main issues raised above are summarised in Table 2. Where an issue clearly relates to more than one process, or is impacted by more than one dimension, it is repeated in the table – primary effect in standard typeface, other effects in italic. This table is very much a summary, and headings may make only limited sense out of the context of the earlier text. In Table 2 we have indicated if an issue is mainly portrayed as a DD advantage or opportunity, a disadvantage or threat, or something that deserves consideration but is not easily classified as one of the two. We use the symbols ☺ for DD advantages and opportunities, ☹ for DD disadvantages and threats, and 😐 for open DD issues. Due to the complex nature of DD, this classification is obviously coarse-grained but at least serves to indicate main trends in the published peer-reviewed DD literature.

As can be seen from Table 2, the proposed framework can effectively be used to structure the many issues pertinent in DD. Although there are obvious overlaps, the framework provides a structure for discussing DD issues which can be useful for understanding the DD domain as well as being a tool for identifying problem areas where more research is needed. We can, for example, see that control issues, in general, have not been addressed to the same extent as issues related to communication and coordination. The framework also highlights that although geographical distance is perhaps the most intuitive discriminating factor in distinguishing DD from "traditional" software development,

many DD issues relate to socio-cultural and temporal distance. This is an important insight since many lessons can probably be transferred from other areas dealing with these aspects to enrich the current DD field of investigation. From Table 2 we can also conclude that most of the published DD literature seems to focus on potential threats in DD (i.e. ☹), some going on to suggest strategies for successful DD which ameliorate these threats, e.g. [6, 22, 30, 44, 45]. One future line of investigation would be to critically examine those threats and explore how and to what extent they might be leveraged into advantages.

Table 2: Framework of distributed development issues.

Process	***Dimension***		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	☺ Time zone effectiveness ☹ Delayed communication ☹ Delayed feedback	☺ Proximity to market/customer ☹ Lack of informal communication ☹ Dependency on ICT ☹ Increased effort to initiate contact ☹ Providing technical infrastructure ☹ Cost of travel	☺ Innovation and shared best practices 😐 Asynchronous communication preferred by non-native speakers ☹ Language differences and misunderstandings ☹ Managing frames of reference
Coordination	😐 Time zone efficiency ☹ Reduced hours of collaboration ☹ Synchronised team meetings difficult ☹ Availability of technical infrastructure ☹ Coordination complexity ☺ *Modularisation of work* ☹ *Lack of mechanisms for creating shared understanding* ☹ *Management of project artefacts*	☺ Access to large labour pool 😐 Standardisation in work practices ☺ Allocation of roles and team structure ☹ Reduced trust ☹ Lack of awareness/team spirit ☺ Modularisation of work ☹ Lack of mechanisms for creating shared understanding ☹ *Coordination complexity*	☺ Mix of skills and experiences 😐 Language and cultural training ☹ Lack of domain knowledge ☹ Doubtful of others' capabilities ☹ *Lack of mechanisms for creating shared understanding* 😐 *Standardisation in work practices* ☹ *Coordination complexity* ☹ *Lack of awareness/team spirit*
Control	☹ Management of project artefacts ☺ *Time zone effectiveness*	☹ Lack of concurrent engineering principles ☺ *Allocation of roles and team structure*	☹ Perceived threat from low-cost alternatives ☹ Adapting to local formalized norm structures ☹ Different perceptions of authority/hierarchy

6. Conclusion

The core challenges of DD seem to lie in the complexity of maintaining good communication, coordination and control when teams are dispersed in time (e.g. across time zones) and space, as well as socio-culturally. In this work we have elaborated on these themes, drawing on the growing body of literature in the area of DD. To structure our analysis and presentation we have developed a framework that integrates these aspects and provides a detailed overview of the DD field. Although the processes and dimensions of this framework emerge from the review, as far as we are aware the framework has never been fully articulated in the literature; in particular, the directional nature of every dimension in a DD context has not been explicitly noted before. Proven methods for successful

DD have not yet been formulated, and the presented framework may be an important tool in identifying the most pressing research issues.

References

[1] ÅGERFALK P J (2004) Investigating Actability Dimensions: A Language/Action Perspective on Criteria for Information Systems Evaluation, *Interacting with Computers*, Vol. 16, No. 5, pp. 957–988.

[2] AKMANLIGIL, M. and PALVIA, P.C. (2004) Strategies for global information systems development, *Information & Management*, Vol. 42, No. 1, pp. 45-59.

[3] ALLEN, T.J. (1977) *Managing the Flow of Technology*, MIT Press.

[4] ARMOUR, P.G. (2002) The organism and the mechanism of projects, *Communications of the ACM*, Vol. 45, No. 5, pp. 17-20.

[5] BASS, M. and PAULISH, D. (2004) Global Software Development Process Research at Siemens, In *The 3rd International Workshop on Global Software Development*, (co-located with ICSE 2004), pp. 11-14, <gsd2004.cs.uvic.ca/docs/proceedings.pdf >

[6] BATTIN, R.D., CROCKER, R., KREIDLER, J. and Subramanian, K. (2001) Leveraging resources in global software development, *IEEE Software*, Vol. 18, No. 2, pp. 70-77.

[7] BECK K (2000) Extreme Programming Explained: Embrace Change, Addison-Wesley, Reading.

[8] BOLAND, D. and FITZGERALD, B. (2004) Transitioning from a Co-Located to a Globally-Distributed Software Development Team: A Case Study and Analog Devices Inc., In *The 3rd International Workshop on Global Software Development*, (co-located with ICSE 2004), pp. 4-7, <gsd2004.cs.uvic.ca/docs/proceedings.pdf>

[9] BRAUN, A., DUTOIT, A.H. and BRUGGE, B. (2003) A Software Architecture for Knowledge Acquisition and Retrieval for Global Distributed Teams, In *International Workshop on Global Software Development*, (co-located with ICSE 2003), pp. 24-29.

[10] BRANNEN, M.Y. and SALK, J.E. (2000) Partnering across borders: Negotiating organizational culture in a German-Japanese joint venture, *Human Relations*, Vol. 53, No. 4, pp. 451-487.

[11] CARMEL, E. (1999) Global Software Teams: Collaborating Across Borders and Time Zones, Prentice Hall, Upper Saddle River.

[12] CARMEL, E. (2003) Introduction to the Special Issue of EJISD: The Emergence of Software Exporting Industries in Dozens of Developing and Emerging Economies, The Electronic Journal on Information Systems in Developing Countries, <www.ejisdc.org>

[13] CARMEL, E. and AGARWAL, R. (2001) Tactical approaches for alleviating distance in global software development, IEEE Software, Vol. 18, No. 2, pp. 22-29.

[14] CASEY, V. and RICHARDSON, I. (2004) Practical Experience of Virtual Team Software Development, In *European Software Process Improvement (EUROSPI) 2004*, Trondheim, Norway.

[15] CLARK, H.H. (1996) *Using Language*, Cambridge University Press, Cambridge.

[16] CRAMTON, C.D. (2001) The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration, *Organization Science*, Vol. 12, No. 3, pp. 346-371.

[17] CROWSTON, K., ANNABI, H., HOWISON, J. and MASANGO, C. (2005) Effective work practices for FLOSS development: A model and propositions, In *Proceedings of the 38th Hawaii International Conference on System Sciences – 2005*, IEEE Computer Society, pp. 1-9.

[18] CURTIS, B., KRASNER, H. and ISCOE, N. (1988) A Field Study of the Software Design Process for Large Systems, *Communications of the ACM*, Vol. 31, No. 11, pp. 1268-1287.

[19] DAMIAN, D.E. and ZOWGHI, D. (2002) The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization, In *Proceedings IEEE Joint International Conference on Requirements Engineering*, IEEE Computer Society, Los Alamitos, pp. 319-328.

[20] DAMIAN, D., LANUBILE, F. and OPPENHEIMER, H.L. (2003) Addressing the Challenges of Software Industry Globalization: The Workshop on Global Software Development, In *Proceedings 25th International Conference on Software Engineering*, IEEE Computer Society, Los Alamitos, pp. 793-794.

[21] DELONE, W., ESPINOSA, J. A., LEE, G. and CARMEL, E. (2005) Bridging Global Boundaries for IS Project Success, In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1*, IEEE Computer Society, pp. 1-10.

[22] EBERT, C. and DE NEVE, P. (2001) Surviving Global Software Development, *IEEE Software*, Vol. 18, No. 2, pp. 62-69.

[23] ENGESTRÖM, Y. (2000) Activity Theory as a Framework for Analyzing and Redesigning Work, Ergonomics, 43(7), pp. 960–974.

[24] ESPINOSA, A. and CARMEL, E. (2003) The Impact of Time Separation on Coordination in Global Software Teams: a Conceptual Foundation, *Software Process Improvement and Practice*, Vol. 8, pp. 249-266.

[25] ESPINOSA, J. A. and CARMEL, E. (2004) The Effect of Time Separation on Coordination Costs in Global Software Teams: A Dyad Model, In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1*, IEEE Computer Society, pp. 1-10.

[26] EVARISTO, J.R., SCUDDER, R., DESOUZA, K.C. and SATO, O. (2004) A dimensional analysis of geographically distributed project teams: a case study, *Journal of Engineering and Technology Management*, Vol. 21, No. 3, pp. 175-189.

[27] GALLIVAN, M. J. (2001) Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies, *Information Systems Journal*, Vol. 11, No. 4, pp. 277-304.

[28] GHOSH, T., YATES, J.A. and ORLIKOWSKI, W.J. (2004) Using Communication Norms for Coordination: Evidence from a Distributed Team, In *2004 — Twenty-Fifth International Conference on Information Systems*, Association for Information Systems, pp. 115-127.

[29] GRINTER, R.E., HERBSLEB, J.D. and PERRY, D.E. (1999) The Geography of Coordination: Dealing with Distance in R&D Work, In *Proceedings on the ACM SIGGROUP Conference on International Conference on Supporting Group Work*, ACM Press, New York, pp. 306-315.

[30] HERBSLEB, J.D. and GRINTER, R.E. (1999) Splitting the Organization and Integrating the Code: Conway's Law Revisited, In *Proceedings of the 21st International Conference on Software Engineering (ICSE'99)*, ACM Press, New York, pp. 85-95.

[31] HERBSLEB, J. D., MOCKUS, A., FINHOLT, T. A. and GRINTER, R. E. (2000) Distance, Dependencies, and Delay in a Global Collaboration, In *CSCW 2000 – ACM 2000 Conference on Computer Supported Cooperative Work*, ACM Press, New York, pp. 319-328.

[32] HERBSLEB, J.D. and MOITRA, D. (2001) Guest Editors' Introduction: Global Software Development, *IEEE Software*, Vol. 18, No. 2, pp. 16-20.

[33] HERBSLEB, J.D. and MOCKUS, A. (2003) An Empirical Study of Speed and Communication in Globally Distributed Software Development, *IEEE Transactions on Software Engineering*, Vol. 29, No. 6, pp. 481-494.

[34] IMSLAND, V., SAHAY, S. and WARTIAINEN, Y. (2003) Key issues in Managing a Global Software Outsourcing relationship between a Norwegian and Russian firm: Some Practical Implications, In *26th Information Systems Research Seminar in Scandinavia*, Finland.

[35] KAROLAK, D. (1998) Global software development: managing virtual teams and environments, Wiley/IEEE Computer Society, Los Alamitos.

[36] KIEL, L. (2003) Experiences in Distributed Development: A Case Study, In *International Workshop on Global Software Development: GSD 2003*, (co-located with ICSE 2003), pp. 44-47.

[37] KRISHNA, S., SAHAY, S. and WALSHAM, G. (2004) Managing Cross-Cultural Issues in Global Software Outsourcing, *Communications of the ACM*, Vol. 47, No. 4, pp. 62-66.

[38] MALONE, T.W. and CROWSTON, K. (1994) The interdisciplinary study of coordination, *ACM Computing Surveys*, Vol. 26, No. 1, pp. 87-119.

[39] MCCHESNEY, I.R. and GALLAGHER, S. (2004) Communication and co-ordination practices in software engineering projects, *Information and Software Technology*, Vol. 46, No. 7, pp. 473-489.

[40] MCDONOUGH, E.F., KAHN, K.B. and BARCZAK, G. (2001) An investigation of the use of global, virtual, and colocated new product development teams, *Journal of Product Innovation Management*, Vol. 18, No. 2, pp. 110-120.

[41] NICHOLSON, B. and SAHAY, S. (2001) Some political and cultural issues in the globalisation of software development: case experience from Britain and India, *Information and Organization*, Vol. 11, No. 1, pp. 25-43.

[42] NURMI, A., HALLIKAINEN, P. and ROSSI, M. (2005) Coordination of Outsourced Information System Development in Multiple Customer Environment – A Case Study of a Joint Information System Development Project, In *Proceedings of the 38th Hawaii International Conference on System Sciences – 2005*, IEEE Computer Society, Los Alamitos, pp. 1-10.

[43] ORLIKOWSKI, W.J. and GASH, D.C. (1994) Technological Frames: Making Sense of Information Technology in Organizations, ACM Transactions on Information Systems, Vol. 12, No. 2, pp. 174–207.

[44] PAASIVAARA, M. and LASSENIUS, C. (2003) Collaboration Practices in Global Inter-organizational Software Development Projects, *Software Process Improvement and Practice*, Vol. 8, pp. 183-199.

[45] PYYSIÄINEN, J. (2003) Building Trust in Global Inter-Organizational Software Development Projects: Problems and Practices, In *International Workshop on Global Software Development*, (co-located with ICSE 2003), pp. 69-74, <gsd2003.cs.uvic.ca/gsd2003proceedings.pdf>

[46] ROBEY, D., KHOO, H.M. and POWERS, C. (2000) Situated Learning in Cross-Functional Virtual Teams, *IEEE Transactions on Professional Communications*, Vol. 43, No. 1, pp. 51-66.

[47] SAHAY, S. (2003) Global software alliances: the challenge of ”standardization’, *Scandinavian Journal of Information Systems*, Vol. 15, pp. 3-21.

[48] SARKER, S. and SAHAY, S. (2002) Information Systems Development by US-Norwegian Virtual Teams: Implications of Time and Space, In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, Los Alamitos, 10p.

[49] SPROULL, L. and KIESLER, S. (1991) Connection: New Ways of Working in the Networked Organization, MIT Press, Cambridge.

[50] SUTANTO, J., KANKANHALLI, A. and TAN, B.C.Y. (2004) Task Coordination in Global Virtual Teams, In *2004 — Twenty-Fifth International Conference on Information Systems*, Association for Information Systems, pp. 807-819.

[51] UNITED NATIONS (2004) World Investment Report 2004 – The Shift Towards Services, United Nations Conference on Trade and Development, New York and Geneva, 468p.

[52] WEBSTER J and WATSON R T (2002) Analyzing the Past to Prepare for the Future: Writing a Literature Review, *MIS Quarterly*, Vol. 26, No. 2, pp. xiii–xxiii.