www.ietdl.org

Published in IET Software
Received on 15th May 2011
Revised on 30th September 2011
doi: 10.1049/iet-sen.2011.0087

Special Section: Management of Global Software Development: Opportunities, Challenges and Lessons Learned

ISSN 1751-8806

Towards an ontology for global software development

A. Vizcaíno F. García I. Caballero J.C. Villar M. Piattini

Instituto de Tecnologías y Sistemas de Información, Camino de Moledores, s/n, 13017 Ciudad Real, Spain
E-mail: Ismael.Caballero@uclm.es

Abstract: As planning an R&D project named ORIGIN addressing global software development (GSD) foundations, the authors soon became aware of the need for all the participants to share the same vision of GSD projects. The authors therefore reached the conclusion that one of the first steps should be to set up a shared and consistent GSD project-related vocabulary, since this would help to improve communication between the five companies involved in ORIGIN. After analysing existing GSD initiatives, the authors discovered that none of them really satisfied our specific needs. As a consequence, they decided to develop a new ontology, which was named O-GSD. This paper describes not only how the ontology was developed -including the reasons which led us to include each term in O-GSD- but also its usage in real contexts -what allowed us to extract some insights to refine and validate the ontology. The main contribution of this paper is the O-GSD ontology which is used and to be used in real GSD projects with the aim of helping project managers to better describe their particular GSD projects.

1 Introduction

The last few decades have witnessed an irreversible movement towards the globalisation of markets. This is especially true for those businesses related to software development processes [1–4]. Global software development (GSD) is therefore continuously gaining acceptance as a paradigm for software development, not only in terms of saving resources and reducing time-to-market, but also as an efficient means to bring together the most qualified human resources in order to ensure the highest level of quality for the software developed. Several works have already studied the benefits of this: cost saving when accessing large multi-skilled workforces, reduced time-to-market, greater productivity (programmers are distributed in places with large time differences, thus signifying that more time than the typical 8 h is available) and the proximity to market and customers [5–9]. The major challenges have also been studied: access to innovation and shared best practices, improving resource allocation, improving task modularisation, communication, or a clearer definition of the process [3, 7, 10] of this paradigm. There are also other cultural and languages [9] issues, such as the fact that global project members often come from different cultures and speak a language that, in some cases, is not their native one.

The ORIGIN (Organizaciones Intelligentes Globales Innovadoras) project came into being in order to tackle these challenges through the creation of a methodology for GSD. This project is being carried out by a consortium which is composed of five companies and two universities in Spain. Its aim is to create a framework for the management and development of software in global contexts. The project goals are divided into conceptual, methodological and technological aspects, covering the representation of organisational and global development knowledge to the development of methodologies and necessary technological support for the global development of software and the evaluation and improvement of its quality.

When starting work on the project, various problems similar to some of those described above appeared. These problems were mainly related to misunderstandings resulting from the fact that the members used similar words to indicate different meanings, or represented different things in the same way. This consequently led to inconsistencies in project representations and in the documentation. Another problem was that when a member from one site had a doubt or something was not sufficiently clear, that person often preferred not to ask the people on the other site about it since s/he did not wish to appear that s/he was not suitably qualified or did not want to bother them. This led to the sensation of working in an isolated manner. Since the goal of creating this consortium was exactly the opposite: working in collaboration and taking advantage of each site's experience, we believed that it was critical to solve these problems by facilitating communication between ORIGIN members. A shared and consistent project-related vocabulary was therefore necessary since, as is indicated in [11, 12], people cannot share knowledge if they do not speak a common language. Moreover, software engineering training and practice are quite different between cities and countries [13]. As is explained in [13], the fact of being software engineers with different training does not ensure that people working in ORIGIN have the same background in and understanding about engineering principles, since each person has his/her own background. It was therefore necessary to determine the terms and the relationships involved in global software development projects in order to ensure that all the members of the project had the same understanding of GSD (goals, factors which affect it, features, support etc.). This motivated the development of an Ontology for global software development projects (O-GSD), since an ontology would help to clarify any ambiguities in the terms used in a particular context, and would also provide the conceptualisation of a domain [14]. Furthermore, 'ontologies are becoming increasingly important in the area of software engineering as they provide a critical semantic foundation' (p. 1) [15], and they define a shared terminology for the concepts of the domain in a generic and formal manner, signifying that it can be reused and shared by different stakeholders [16].

This paper describes the O-GSD ontology and illustrates the benefits it can provide from the conceptual point of view for researchers and practitioners. For researchers O-GSD can be used as a rapid and easy means of understanding the differences between located and global software development, since O-GSD is focused on describing the typical and basic concepts that it is necessary to know to understand and represent a GSD project. The ontology is also useful for practitioners, since they can use it to describe their projects. A person who has a general view of a project, such as the project manager, can therefore instance the ontology and send this instantiation to all the stakeholders in order to attain a common and a graphic description of the project on which they will be or are already working. This instantiation will show the factors that might affect the project, their goals, the type/s of delivery model used, the teams involved, their sites, roles of the members, tools used etc. Moreover, if the stakeholders wish to add more information, this would be possible as it is an open ontology in which we have indicated the basic concepts that, according to literature and our experience, are necessary to represent a GSD project. However, researchers and practitioners might extend or adapt it according their needs, taking O-GSD as a starting point.

Besides the O-GSD we believe that it would also be interesting to provide a description of how O-GSD was created, along with its usage in a real context, from which some insights into its validity were obtained.

The remainder of the paper is structured as follows: Section 2 introduces the action research strategy followed to build the ontology. Section 3 is focused on explaining how the first version of the O-GSD was developed. In Section 4 the case study in which the ontology was applied is described. Related works are analysed in Section 5. Finally conclusions and the outline of our intended future work are provided in Section 6.

2 Strategy to build O-GSD based on the action-research method

The research method applied to build the ontology was that of action-research (A-R) [17]. Of the four variants of action-research proposed by French and Bell in [18], we agreed that the participative version was best suited to the nature of the task at hand. We therefore decided to match the research activities involved in developing O-GSD to the stages of the systematic cyclical structure of the action research method, as follows:

1. Planning: Specification and Conceptualisation of the O-GSD: Of the existing methods which can be used to build ontologies [19], we decided to develop O-GSD with the use of REFSENO [20], since this method was specifically designed for software engineering and allows several representation options of the ontology to be created that are not only restricted to representations based on first predicate logic, which might be less intuitive and familiar for the stakeholders involved in the project.
2. Action: The ontology (O-GSD v1.0) was used by GSD Engineers, as part of the participative iterations. Some reports containing feedback about the use of the ontology and its appropriateness were returned.
3. Observation: The feedback obtained from GSD Engineers was analysed in order to refine O-GSD to O-GSDv2.0.
4. Reflection: The results of the research will be shared with GSD researchers and practitioners. It will serve as a means to collect useful feedback with which to continuously refine O-GSD and improve its external validity, thus enabling it to be used in any kind of GSD project.

The participants involved in the execution of the A-R cycle were the following:

- Researchers (R): Working force for ORIGIN, composed of five researchers from the Alarcos Research Group who were in charge of developing the ontology.
- Object focus of research (O): O-GSD, an ontology for GSD.
- Critical reference group (CRG**)**: This group is formed of those practitioners from the companies involved in the ORIGIN project who have real problems which need to be solved. A common understanding of GSD and the concepts related to it were also necessary. The CGR consisted of representative stakeholders from these companies.
- The benefited stakeholder (B): the GSD community.

Five researchers were involved in the development of O-GSD and the tasks were completed in 10 months.

The following section describes how the O-GSD was developed. Owing to space limitations, the paper only describes the final results obtained after the application of the Action-Research cycle.

3 Development of O-GSDv1.0

We decided to develop O-GSD by using a well-structured method to systematically build the ontology. Taking into account that O-GSD is to be used in the software engineering field, we decided to use REFSENO (representation formalism for software engineering) [20] to develop and represent this ontology. REFSENO provides constructs with which to describe concepts, their attributes and their relationships. The detailed information of the ontology is represented in REFSENO by means of a collection of tables.

The tasks carried out to develop the O-GSD according to the REFSENO formalism were:

1. Define the concept glossary from the knowledge sources.
2. Define the semantic relationships among the concepts by representing them in the unified modelling language (UML) notation and create the relationship class tables.
3. Analyse the concepts that have some kind of relationships in order to identify the commonalities among two or more concepts. It was then decided whether these commonalities are concepts (inserted for modelling reasons). If so, they were included in the glossary of concepts.
4. Identify the attributes of all the concepts and include them in the UML diagrams. Each time a new attribute type is identified, it must be included in the table of types.

5. Complete the attribute concept tables by including a non-terminal attribute.
6. Check the completeness of all the attribute tables.

For the sake of simplicity, in the development of the first version of O-GSD we decided to work only on concepts, terms and relationships (tasks T.1.–T.3), which are described in the following subsections.

3.1 Define the concept glossary from the knowledge sources

The first step was to create a glossary of terms from which to build the glossary of concepts. In order to define the most appropriate concepts for our ontology, the following steps were carried out.

3.1.1 Finding GSD terms: This first step consisted of capturing as many representative concepts for GSD as possible. Nine systematic reviews concerning GSD [2, 21–28] and the proceedings of ICGSE 2009 were analysed with the goal of finding those concepts that were narrowly related to GSD. A candidate term was considered to be relevant to GSD when it fulfilled some of the following criteria. These criteria were selected by the five authors of the paper and were later reviewed by the project managers of the companies that are collaborating with us on the ORIGIN project:

- It was an activity or process which had special relevance for GSD projects (GSD Activity).
- It was a term that characterised a GSD project of GSD (characteristic).
- It was a benefit or drawback of GSD (benefit or drawback).
- GSD goals (goal).
- It supported a GSD activity (support).
- It was a role involved in GSD (role).
- It was a tool used in GSD (tool).

Brainstorming was applied in this phase with the aim of providing a complete scope for analysis by starting from a highly representative set of terms and in order to avoid the risk of omitting key terms. This complete set of terms (111 in total, see the Appendix), was then used in the second step explained just below. The Appendix shows the number of occurrences of the selected terms in the proceedings of ICGSE 2009 and 2010. As was previously mentioned, the ICGSE 2009 was one of the initial information sources used. After the ICGSE 2010 had taken place, the table was completed with the occurrences that appeared in that edition, since these occurrences might be an indicator of the relevance of a term.

3.1.2 Filtering terms: This activity was carried out by three researchers who frequently collaborate with three of the software factories involved in the ORIGIN project. Each researcher was asked to indicate the terms shown in Appendix which should be removed. The reasons for removing these terms were the following:

- the term was very general (such as ‘global software development’ or ‘global software engineering’);
- the term did not specifically characterise GSD projects (i.e. the term ‘training’);
- the term was synonymous with another (e.g. ‘task allocation’ and ‘task distribution’ were considered to be synonyms of ‘delivery model’ which was the selected term).

A fourth researcher later compared the results and observed that almost all of the first three researchers were of the same opinion. However, there was no consensus with regard to certain terms, since some of the researchers believed that a particular term should be omitted in contrast to the other two researchers’ opinion. In order to reach a consensus, the three researchers were invited to re-evaluate the table while being fully aware of their colleagues’ opinions about each term. In this second round an agreement was reached in 92% of the decisions made. The researchers finally decided to meet to discuss the cases upon which they disagreed, and the set of terms eventually used to create the ontology was obtained as a result of this (43 terms, see Table 1). During this meeting the researchers realised that most of the ‘not sufficiently’ agreed terms were really features of other terms that had already been selected or features that could be derived from other existing ones.

Once the set of concepts for the ontology had been successfully delimited and closed, the glossary of concepts and their definition (following REFSENO) was created. Table 2 shows an excerpt of this glossary, in which not only the concepts, but also the source from which the concept has been taken appears. It is worth stating that the researchers, after studying the definitions of the various concepts as is provided by their corresponding GSD references, sought an agreement on which definitions were the most appropriate in terms of being more easily understood, or which best described the concept in a clear and unambiguous manner. The researchers occasionally had to complete or modify some definitions that did not really match perfectly with the meaning that was required within the ORIGIN project.

3.2 Define the semantic relationships among the concepts

Once the concept glossary had been created, the relationships between the concepts were established by analysing the relevant sources selected from the bibliography. These relationships were graphically represented by using UML (see Fig. 1) and were textually represented by means of REFSENO relationship tables (see Table 3).

As can be observed in Fig. 1 and Table 3, ‘GSD projects’ aim to satisfy four types of ‘goals’. One of these is ‘proximity’. The concept of ‘proximity’ can be understood from any of the following points of view: ‘proximity to market’/’proximity to customer’ (this refers to the possibility of establishing subsidiaries in countries in which the company’s customers are located, or in which the company could develop software closer to their customers to increase knowledge of the local market [29]); Another kind of proximity is ‘proximity to human resources’ [34], when the subsidiaries are located near qualified or potential workers. For instance, companies can be located close to universities when the aim is to contract students that have finished their bachelor degrees, and this was actually the case of one of the companies involved in the ORIGIN project. Other goals are to ‘reduce time-to-market’ and ‘reduction in development costs’: distributing the work in countries with low labour costs helps to reduce costs and, when the follow-the-sun strategy is used, there is also a reduction in delivery to the market [5]. The last goal that is

Table 1 Set of terms finally considered to create the ontology

Id	Category	Term	Appearances	
			ICGSE 2009	ICGSE 2010
1	characteristic	geographical distance	14	5
2	characteristic	temporal distance	14	10
3	characteristic	socio-cultural distance	19	5
4	characteristic	language differences	8	12
5	characteristic	delivery model	1	0
6	characteristic	follow-the-sun	49	2
7	characteristic	GSD team	4	14
8	characteristic	module-based	2	1
9	characteristic	module	17	42
10	characteristic	phase-based	2	1
11	characteristic	phase	49	51
12	characteristic	GSD project	24	12
13	characteristic	site	420	357
14	characteristic	location	99	76
15	characteristic	time zone	83	49
16	characteristic	team member	31	22
17	characteristic	collaboration mode	2	0
18	characteristic	inter-organisational	7	7
19	characteristic	intra-organisational	4	5
20	goal	proximity	48	11
21	goal	proximity to market	2	1
22	goal	proximity to customer	2	1
23	goal	proximity to human resources	0	0
24	goal	reduction in development costs	2	1
25	goal	improve quality	2	0
26	role	GSD role	0	0
27	role	coordinator	12	6
28	role	mediator	29	1
29	role	project manager	100	70
30	role	developer	56	45
31	support	tools	648	554
32	support	control	97	108
33	support	communication	891	848
34	support	coordination	285	318
35	support	asynchronous communication	11	9
36	support	synchronous communication	18	10
37	support	blog	1	9
38	support	email	113	150
39	support	forum	18	30
40	support	newsgroups	0	1
41	support	instant messaging	48	33
42	support	video-conference	10	7
43	support	audio-conference	4	1

Table 2 Excerpt of O-GSDv2.0 concept definitions

Term	Definition	Source
socio-cultural distance	socio-cultural distance is a measure of an actor's understanding of another actor's values and normative practices	[30]
follow-the-sun	follow-the-sun implies handing on work at the end of every day from one site to the next site many time zones away (e.g. USA to India), so that the work can be advanced while one team rests for the night	[31]
GSD team	distributed members who collaborate on a common software project while working across geographic, temporal, cultural, political and organisational boundaries to accomplish an interdependent task	[32]
control	control is the process of adhering to goals, policies, standards, or quality levels	[33]
communication	communication is considered as the exchange of complete and unambiguous information – that is, the sender and receiver can reach a common understanding	[33]
coordination	coordination is the act of integrating each task with each organisational unit, so the unit contributes to the overall objective	[33]

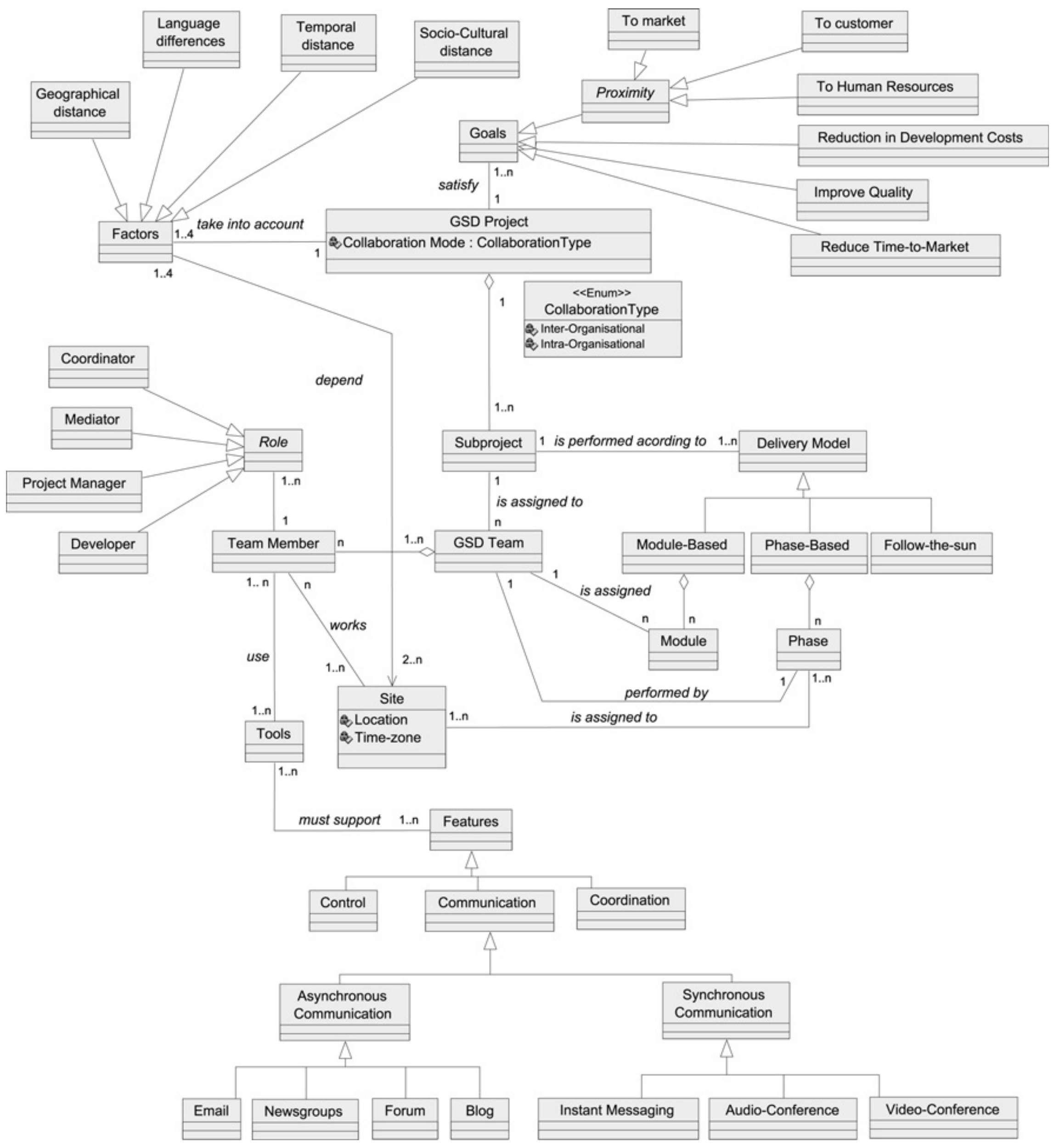

Fig. 1 *Graphical representation of O-GSD*

a common objective is 'improved quality', which can be a consequence of locating subsidiaries where the experts are [30] or to learn new strategies or habits used, for example, in other software factories belonging to the same company.

On the other hand, a 'GSD project' largely depends on four 'factors' [29]: 'geographical distance', as this often causes problems in communication and coordination [35] 'language differences' which creates misunderstandings that may cause delays [30], 'temporal distance' which is also another important factor because co-workers work in different 'time zones' (attribute of site) according to the site on which they are located and their working times might not overlap, which leads to the need to empower Asynchronous communication to the detriment of the more effective synchronous communication [36]. The last aspect is socio-cultural distance, since this may cause a lack of trust among members, along with a loss of team spirit [37]. The factors depend on how many 'locations' (attribute) are involved in the project and where they are.

A 'GSD project' could consist of several 'subprojects', and the different activities are distributed throughout the different locations by following a type of 'delivery model' [38]. This model can be 'module-based', 'phase-based' or 'follow-the-sun' [37]. The first (module-based) consists of dividing the project into modules and allocating them among the different locations. The second model (based phase), assigns a 'phase' of development to each location; for example, the analysis and design phases could be carried out in one country and the implementation in another. Follow-the-sun [31] is based on the idea of prolonging the working day, so that when the people in one country stop work it is continued by people in another country where the day is starting. The countries should consequently have different 'time zones' [39].

A 'GSD team' contains one or more 'team members', who can play different roles: coordinator, mediator, (sub)project manager, developer. We attempted to avoid complicating the ontology by assuming that all the roles that exist in a

Table 3 O-GSD relationships table

Name of the relationships	Interrelated concepts	Cardinality	Definition
depend	factors – location	2..*	there is a dependency between the factors of GSD and two or more locations
is assigned to	GSD team – module	1..*	the development modules are assigned to a team
is assigned to	site – phase	1..*	one or more phases are assigned to one or more locations
is assigned to	subproject – GSD team	1..1	each sub-project is assigned to a development team
is performed according to	subproject – delivery model	1..*	each subproject will develop a delivery model type: module-based, phase-based or follow-the-sun, according to the criteria adopted
must support	tools – features	1..*	the tools used by members must support one or more features of control, communication and collaboration
performed by	phase – team	1..1	each phase is performed by a team
satisfy	GSD project- goals	1..4	this relationship indicates that each GSD project must satisfy between one and four goals of the GSD
take into account	GSD project – factors	1..4	this relationship indicates that each GSD project must take into account between one and four factors of the GSD
use	team members – tools	1..*	members of a development team use one or more tools to work
works	site – team member	1..*	this relationship indicates that one or more team members could work on a site

co-located development and that are already well known by software engineers, such as analyst, designer, tester etc. are included in the developer role. All of them could be therefore represented in the ontology under such concept. Members are assisted in their work by means of GSD supporting tools. A GSD supporting tool includes one or more ‘features’, which are often classified according to the typical 3Cs GSD categories (although other features could be added. For more information about tools see [40, 41]): ‘control’, ‘communication’ and ‘collaboration’ [37]. Communication can be classified as: ‘synchronous’ and ‘synchronous’ [36]. Asynchronous communication is supported by features such as ‘Email’ [42], ‘newsgroups’ [43], ‘forums’ [36] and ‘blogs’. Finally, synchronous communication can be supported by means of ‘instant messaging’, ‘audio-conference’ and/or ‘video-conference’.

4 Application of O-GSD to represent a GSD project

The first version of the O-GSD was applied to represent a GSD project being carried out by INDRA software labs, which is one of the companies of which the Critical Reference Group is composed. Indra Software Labs (ISL) is a subsidiary company of the INDRA group, which specialises in software development. ISL is characterised by the near and off-shoring work in its different software factories and the development and application of its own methodologies, architectures and tools to promote high productivity and quality. ISL operates as a unique virtual centre, providing software development services in more than 25 countries, 24 h a day, 365 days per year, making its factories one of the company’s main assets. ISL offices are distributed throughout 30 countries in Europe, Africa, The Americas and Asia.

The goals of this instantiation were:

- Validating the ontology, checking whether it was possible to represent all the information that a project manager needs to know about a GSD project.
- Validating the relationships among concepts.
- Obtaining feedback from a real representation, in order to detect limitations of O-GSD and make new refinements.

Owing to its nature, ISL constituted a very promising scenario in which to carry out the preliminary validation of the proposed ontology. This was done by selecting one of the company’s most representative GSD projects. Some details of the project are omitted or described with generic names owing to confidentiality issues. The project, named ‘SCEPYLT G6’, consisted of the development of a piece of software for the management of security in the commercialisation of security devices between countries in the European Union. This application aimed to reduce the administrative work load that has to be assumed by suppliers, and was composed of the following components/modules:

- security device transfer management,
- order managements between European Union countries,
- permission and license management,
- security device transit management,
- communication management.

Five ISL factories located in Spain, France, Germany, United Kingdom and Italy participated in this project. The O-GSD ontology was used to represent this GSD project and some refinements were incorporated into O-GSD v2 as a result of its application. Fig. 2 represents the project when using O-GSD 2.

As can be observed in Fig. 2, the SCEPYLT G6 project, which was intra-organisational, that is, developed between ISL factories located in Spain and the rest of Europe, confronted the following GSD factors: language differences (five different languages) and temporal distance (GTM + 1 as the maximum time zone difference). The main goals to be satisfied when developing this project as a global project were the reduction of costs and the improvement of quality. The global project was divided into five subprojects in which the phase-based delivery model was followed. The Spanish factory was in charge of requirements, functional analysis, technical design and implementation. The other

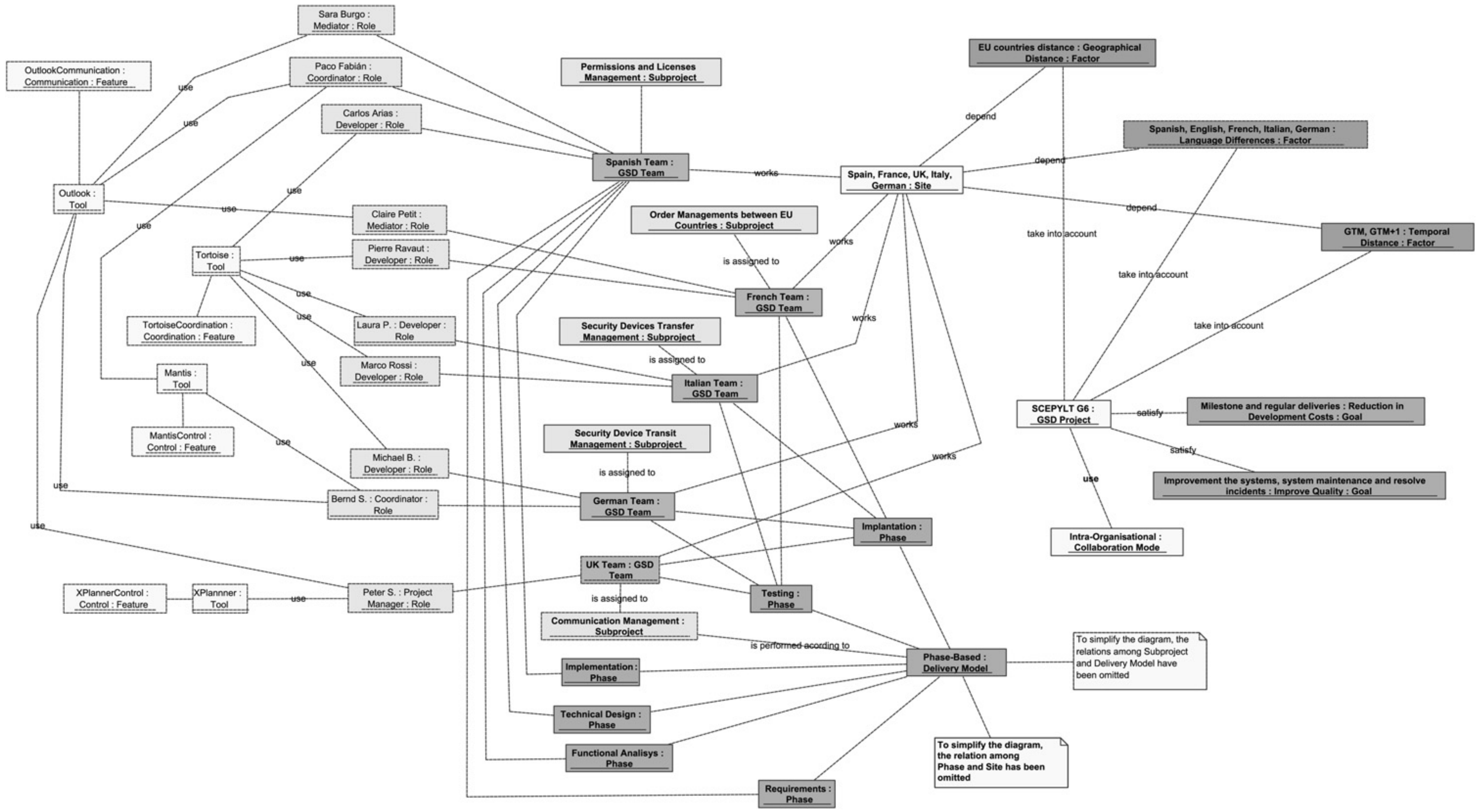
Sara Burgo : Mediator : Role
OutlookCommunication : Communication : Feature
Paco Fabián : Coordinator : Role
Permissions and Licenses Management : Subproject
EU countries distance : Geographical Distance : Factor
Carlos Arias : Developer : Role
Spanish, English, French, Italian, German : Language Differences : Factor
Spanish Team : GSD Team
works
Spain, France, UK, Italy, German : Site
depend
Outlook : Tool
use
Claire Petit : Mediator : Role
Order Managements between EU Countries : Subproject
GTM, GTM+1 : Temporal Distance : Factor
Tortoise : Tool
Pierre Ravaut : Developer : Role
is assigned to
take into account
French Team : GSD Team
TortoiseCoordination : Coordination : Feature
Laura P. : Developer : Role
Security Devices Transfer Management : Subproject
Marco Rossi : Developer : Role
Italian Team : GSD Team
Mantis : Tool
SCEPYLT G6 : GSD Project
satisfy
Milestone and regular deliveries : Reduction in Development Costs : Goal
MantisControl : Control : Feature
Security Device Transit Management : Subproject
Michael B. : Developer : Role
Improvement the systems, system maintenance and resolve incidents : Improve Quality : Goal
Bernd S. : Coordinator : Role
German Team : GSD Team
Implantation : Phase
Intra-Organisational : Collaboration Mode
UK Team : GSD Team
XPlannerControl : Control : Feature
XPlannner : Tool
Peter S. : Project Manager : Role
Testing : Phase
Communication Management : Subproject
is performed acording to
Phase-Based : Delivery Model
To simplify the diagram, the relations among Subproject and Delivery Model have been omitted
Implementation : Phase
Technical Design : Phase
Functional Analisys : Phase
To simplify the diagram, the relation among Phase and Site has been omitted
Requirements : Phase

Fig. 2 *Instantiation of SCEPYLT project in O-GSD*

factories involved carried out the testing and implantation. Mantis and XPlanner tools were used to support control of GSD and the coordination was managed by using TortoiseSVN. The synchronous communication between the members of different sites was supported by Microsoft Communicator as the Instant Messaging tool, and Audio Conferences and Video Conferences tools were also used. Asynchronous communication was supported by e-mail (outlook). The information about members is also shown in Fig. 2.

One of the main conclusions obtained as a result of the application of O-GSD was that the ontology covered 100% of the concepts required by ISL to represent the GSD project. It was not necessary to incorporate new terms, so we can state that O-GSD passed the first validation. However, some important properties with which to characterise GSD projects were obtained, such as the collaboration mode: inter-organisational and intra-organisational.

In the ORIGIN project, O-GSD has contributed towards obtaining a common understanding about the project according to the answers of five stakeholders who were asked about its utility. Four of them stated that the instantiation of the ontology helped them to reflect on the global setting challenges and to always bear in mind the goals of the project. Furthermore, all of them said that it was like ‘a project map’ where they could easily consult information related to the project, for example the role of a particular person or which team member she/he belonged to. All of them also agree that it facilitates communication between the different team members as all of them share the same ‘project map’, that is, the same view of the project.

5 Related work

There is currently an important interest in developing ontologies in the software engineering field. A proof of this fact is the increasing number of workshops focused on this, such as the workshop on ontology, conceptualisation and epistemology for software and system engineering (ONTOSE), the workshop on ontologies and metamodelling software and date engineering (WOMSDE) or the workshop on semantic-web enabled software engineering (SWESE). However, to the best of our knowledge, none of the existing works satisfy the specific needs that appear in the context of the ORIGIN project with regard to the conceptualisation of GSD projects.

Nevertheless, there are some existing works of interest in the field. In [15], the authors describe a software engineering ontology model. The knowledge related to software engineering was obtained from the software engineering textbook [44] and from the SWEBOK. The authors of this ontology agreed with our purpose when developing the ontology: to enable communication between software engineers in order to understand common software engineering knowledge. However, their ontology is considerably more general than O-GSD as it deals with the entire software engineering domain, whilst we focus on GSD projects.

In [13], the authors define five ontologies to be used in a multi-site software engineering environment: a business domain ontology to characterise the fundamental knowledge about a particular domain since all software is designed to solve a business need such as accounting or a customer service etc.; a software engineering ontology in which software engineering principles and aspects are described; a project management ontology, created to enable all the stakeholders to have consistent knowledge when discussing project-related matters; an issues ontology, divided into ontological, technical and managerial issues; and the solution ontology, which is related to the knowledge of issues and solutions that drive the project and product success. These ontologies are developed to serve intelligent software agents which support multi-site software development. Of these ontologies, that which is most directly related to O-GSD is the project management ontology. However, it does not fulfil our purpose as it does not contain any particular terms that are specific to GSD. In [45] the authors propose a software engineering sub-ontology to enable remote team members to browse, search for and share data in a distributed software engineering project environment. One of their main goals is therefore to enable communication between computer systems. The authors thus provide two subontologies, a generic ontology and an application-specific ontology. The generic ontology is a set of software engineering terms, including the vocabulary, the semantic interconnections and some simple rules of inference and logic for software development. It provides the vocabulary for the terms in software engineering, along with an application-specific ontology which is an explicit specification of object-oriented development in software engineering. The approaches of both ontologies are helping to transform explicit knowledge to conceptual knowledge representation with the aim of using software agents to access data from this project-ontology repository. The approach of this ontology is thus different to ours. In summary, the main contribution of this work is the provided conceptualisation focused on the area of GSD.

6 Conclusions and future work

In this paper an ontology for GSD has been described, which was developed to fulfil a real need in the ORIGIN R&D project: to facilitate communication between members and to avoid misunderstanding.

O-GSD has been obtained as a result of continuous feedback between the researchers and practitioners involved in the ORIGIN project, and it is therefore the fruit of a consensus between them. An in-depth analysis of related GSD literature was used to establish a suitable conceptualisation of the domain. O-GSD has been tested in live GSD projects, which reported useful feedback with which to refine it until its current version was obtained (2.0). This ontology has provided a common understanding of GSD projects and promotes the usage of the same terminology, thus facilitating the communication between the practitioners and researchers involved in ORIGIN. The other contribution is the ontology itself, which will be used by the companies to represent the knowledge related to their GSD projects.

Ontologies are frequently grouped into two main categories, depending on whether they are used to describe the knowledge of a domain (domain ontologies) or whether they are used as software artifacts in software development processes [16]. This paper has focused on the role of O-GSD as domain ontology, which as a summary of the obtained conclusions from this work, might be useful from the following perspectives:

- Academic, since the ontology provides significant knowledge which must be considered in GSD projects.

Students or researchers who wish to start learning about this topic can therefore use the ontology to see at a glance what the concepts most widely used in GSD are (features, goals, delivery models etc.).

• Industrial, as practitioners can use the ontology to represent global projects and share information about them. The fact of instancing the ontology will help companies to discover cultural differences, languages and other features regarding their project and it might be a means to detect and avoid possible problems that could appear during the development of a global project.

This conceptualisation can be used as a starting point to enhance and extend specific parts of global software development characteristics in order to fulfil other related goals.

O-GSD could also serve as 'software artefact' to support software engineering processes. A possible scenario of this is to use O-GSD to build metamodels and associated DSLs (domain specific languages). In this context, the specific concepts covering GSD can be used to complement and or customise relevant metamodels about software engineering processes such as ISO 24744 [46] and SPEM 2 [47] to support GSD projects.

As future work more instantiations of the ontology in other projects will be carried out and more experts in GSD will be involved to provide further validation and improvements.

7 Acknowledgments

Thank you very much to the reviewers. Their suggestions have helped us to improve this paper. This work has been funded by the ORIGIN project (IDI-2010043 (1–5)), funded by CDTI and FEDER; the PEGASO/MAGO project (Ministerio de Ciencia e Innovación MICINN and Fondos FEDER, TIN2009-13718-C02-01). We also are grateful to practitioners from ISL who actively collaborated in the validation of the ontology.

8 References

1 Boehm, B.: 'A view of 20th and 21st century software engineering'. ICGSE, 2006

2 Jiménez, M., Piattini, M., Vizcaíno, A.: 'Challenges and improvements in distributed software development: a systematic review', *Adv. Softw. Eng.*, 2009, **2009**, Article ID 710971, 14 pages, doi:10.1155/2009/710971

3 Nguyen, T., Wolf, T., Damian, D.: 'Global software development and delay: does distance still matter?'. IEEE Int. Conf. on Global Software Engineering, 2008

4 Conchúir, E., Ågerfalk, P.J., Olsson, H.H., Fitzgerald, B.: 'Global software development: where are the benefits?', *Commun. ACM*, 2009, **52**, (8), pp. 127–131

5 Ågerfalk, P.J., Fitzgerald, B., Holmström, H., Conchúir, E.: 'Benefits of global software development: the known and unknown'. Int. Conf. on Software Process, 2008

6 Humphrey, W.S.: 'Introduction to the team software processs', Part of the SEI Series in Software Engineering, Addison-Wesley Professional, 1999, Reading (MA, USA)

7 Ebert, C., Neve, P.D.: 'Surviving global software development', *IEEE Softw.*, 2001, **18**, (2), pp. 62–69

8 Cataldo, M., Herbsleb, J.D.: 'Communication networks in geographically distributed software development'. CSCW '08: Proc. 2008 ACM Conf. on Computer Supported Cooperative Work, 2008

9 Damian, D., Zowghi, D.: 'The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization'. Requirements Engineering (RE'02), 2002

10 Solingen, R.V., Valkema, M.: 'The impact of number of sites in a follow the sun setting on the actual and perceived working speed and accuracy: a controlled experiment'. 2010 Fifth IEEE Int. Conf. on Global Software Engineering, 2010

11 Davenport, T.H., Prusak, L.: 'Working knowledge: how organizations manage what they know' (H.B.P., 1998), p. 224

12 Niederman, F., Tan, F.B.: 'Emerging markets managing global IT teams: considering cultural dynamics', *Commun. ACM*, 2011, **54**, (4), pp. 24–27

13 Wongthongtham, P., Chang, E., Dillon, T., Sommerville, I.: 'Ontology-based multi-site software development methodology and tools', *J. Syst. Archit.*, 2006, **52**, pp. 640–653

14 Gruber, T.R.: 'Toward principles for the design of ontologies used for knowledge sharing', *Int. J. Hum.-Comput. Stud.*, 1995, **43**, (5–6), pp. 907–928

15 Wongthongtham, P., Chang, E., Dillon, T., Sommerville, I.: 'Development of a software engineering ontology for multisite software development', *IEEE Trans. Knowl. Data Eng.*, 2009, **21**, (8), pp. 1205–1217

16 García, F., Ruiz, F., Calero, C., *et al.*: 'Effective use of ontologies in software measurement', *Knowl. Eng. Rev.*, 2009, **24**, (1), pp. 23–40

17 Barney, S., Wohlin, C., Hu, G., Aurum, A.: 'Creating software product value in China', *IEEE Softw.*, 2009, **26**, (4), pp. 84–90

18 Sangwan, R., Bass, M., Mullick, N., Paulish, D.J., Kazmeier, J.: 'Global Software Development Handbook, 2006: Boca Raton, FL: Auerbach publications

19 Kock, K., Lau, F.: 'Information systems action research: serving two demanding masters', *Inf. Technol. People*, 2001, **14**, (1), pp. 6–11

20 Tautz, C., Von Wangenheim, C.G.: 'REFSENO: a representation formalism for software engineering ontologies, Fraunhofer IESE-Report NÂº 015.98/E, version 1.1, 20 October 1998

21 Šmite, D., Wholin, C., Feldt, R., Gorschek, T.: 'Reporting empirical research in global software engineering: a classification scheme'. Int. Conf. on IEEE Global Software Engineering (ICGSE), 2008

22 Prikladnicki, R., Audy, J.L.N.: 'Process models in the practice of distributed software development: a systematic review of the literature', *Inf. Softw. Technol.*, 2010, **52**, (8), pp. 779–791

23 Prikladnicki, R., Audy, J.L.N., Shull, F.: 'Patterns in effective distributed software development', *IEEE Softw.*, 2010, **27**, (2), pp. 12–15

24 Khan, S.U., Niazi, M., Ahmad, R.: 'Critical success factors for offshore software development outsourcing vendors: a systematic literature review'. Int. Conf. on Global Software Engineering (ICGSE), 2009

25 Khan, S.U., Niazi, M., Ahmad, R.: 'Critical barriers for offshore software development outsourcing vendors: a systematic literature review'. Software Engineering Conf., 2009

26 Persson, J.S., Mathiassen, L., Boeg, J., Madsen, T.S., Steinson, F.: 'Managing risks in distributed software projects: an integrative framework', *IEEE Trans. Eng. Manage.*, 2009, **56**, (3), pp. 508–532

27 Hossain, E., Babar, M.A., Paik, H.: 'Using scrum in global software development: a systematic literature review'. Int. Conf. on Global Software Engineering (ICGSE), 2009

28 Šmite, D., Wohlin, C., Gorschek, T., Feldt, R.: 'Empirical evidence in global software engineering: a systematic review', *Empir. Softw. Eng.*, 2010, **15**, (1), pp. 91–118

29 Herbsleb, J.D., Moitra, D.: 'Guest editors' introduction: global software development', *IEEE* Softw., 2001, **18**, (2), pp. 16–20

30 Ågerfalk, P., Fitzgerald, B., Holmström, H., Lings, B., Lundell, B., Conchúir, E.: 'A framework for considering opportunities and threats in distributed software development'. Int. Workshop on Distributed Software Development, 2005

31 Carmel, E., Espinosa, J.A., Dubinsky, Y.: 'Follow the sun: workflow in global software development: conceptual foundations', *J. Manage. Inf. Syst.*, 2010, **27**, (1), pp. 17–38

32 Barney, H.T., Moe, N.B., Low, G.C., Aurum, A.: 'Indian intimacy ends as the Chinese connection commences: changing offshore relationships'. Third Global Sourcing Workshop, 2009

33 Carmel, E., Agarwal, R.: 'Tactical approaches for alleviating distance in global software development', *IEEE Softw.*, 2001, **18**, (2), pp. 22–29

34 Richardson, I., Casey, V., Zage, D., Zage, W.: 'Global software development – the challenges', S.T.R. 278, University of Limerick, Ball State University, 2005

35 Lamersdorf, A., Münch, J.: 'Studying the impact of global software development characteristics on project goals: a causal model', *Open Softw. Eng. J.* (*special issue on 'Global Softw. Dev. Chall.*), 2010, **4**, pp. 2–13

36 Portillo-Rodríguez, J., Vizcaíno, A., Ebert, C., Piattini, M.: 'Tools to support global software development processes: a survey'. Fifth Int. Conf. on Global Software Development (ICGSE'2010), 2010

37 Conchúir, E.: 'Global software development: a multiple-case study of the realisation of the benefits' (University of Limerick, Limerick, Ireland, 2010), p. 262

38 Deshpande, S., Richardson, I.: 'Management at the outsourcing destination – global software development in India'. Fourth IEEE Int. Conf. on Global Software Engineering, 2009, ICGSE 2009, 2009

39 Damian, D.: 'Global software development: growing opportunities, ongoing challenges', *Softw. Process Improv. Pract.*, 2003, **8**, (4), pp. 179–182

40 Portillo-Rodríguez, J., Ebert, C., Vizcaíno, A.: 'Tools and technologies for distributed teams', *IEEE* Softw., 2010, **27**, (4), pp. 10–14

41 Lanubile, F., Ebert, C., Prikladnicki, C., Vizcaíno, A.: 'Collaboration tools for global software engineering', *IEEE Softw.*, 2010, **27**, (2), pp. 52–55

42 Niinimaki, T.: 'Reflecting the choice and usage of communication tools in GSD projects with media synchronicity theory'. 2010 Fifth IEEE Int. Conf. on Global Software Engineering (ICGSE), 2010

43 Fortuna, B., Mendes, E., Milic-Frayling, N.: 'Improving the classification of newsgroup messages through social network analysis'. Proc. 16 ACM Conf. on Information and Knowledge Management, 2007

44 Sommerville, I.: 'Software engineering' Seventh Edition, Addison-Wesley 2004, Harlow (Essex, England)

45 Wongthongtham, P., Chang, E., Cheah, C., Dillon, T.S.: 'Software engineering sub-ontology for specific software development'. 29th Annual IEEE/NASA Software Engineering Workshop, 2005

46 ISO/IEC: 'ISO/IEC 24744:2007. Software engineering – metamodel for development methodologies', 2007

47 OMG: 'Software & systems process engineering metamodel specification (SPEM) Version 2.0', 2008

9 Appendix

The complete set of terms are given in Table 4.

Table 4 Complete set of terms

Id	Category	Term	Appearances	
			ICGSE 2009	ICGSE 2010
1	benefit or drawback	communication problems	12	36
2	benefit or drawback	cultural differences	59	113
3	benefit or drawback	cultural diversity	17	60
4	benefit or drawback	cultural issues	14	13
5	benefit or drawback	dispersion	55	8
6	benefit or drawback	distance	331	159
7	benefit or drawback	geographical distance	14	5
8	benefit or drawback	lack of trust	31	19
9	benefit or drawback	language differences	8	12
10	benefit or drawback	socio-cultural distance	19	5
11	benefit or drawback	temporal distance	14	10
12	benefit or drawback	time zone differences	13	10
13	characteristic	awareness	86	120
14	characteristic	collaborative software engineering	5	10
15	characteristic	collaborative work	11	11
16	characteristic	delivery model	1	0
17	characteristic	distributed development (DD)	121	80
18	characteristic	distributed sites	20	8
10	characteristic	distributed software development (DSD)	141	166
20	characteristic	distributed software engineering	6	18
21	characteristic	distributed teams	109	108
22	characteristic	distributed work	12	38
23	characteristic	follow-the-sun	49	2
24	characteristic	geographical location	8	0
25	characteristic	geographically distributed software development	6	9
26	characteristic	geographically distributed teams	4	6
27	characteristic	global collaboration	10	10
28	characteristic	global software development (GSD)	384	441
29	characteristic	global software engineering	341	313
30	characteristic	global software teams	46	24
31	characteristic	global teams	15	21
32	characteristic	global virtual teams	27	19
33	characteristic	globally distributed development	10	6
34	characteristic	GSD teams	4	14
35	characteristic	knowledge transfer	11	67
36	characteristic	location	99	76
37	characteristic	module-based	2	1
38	characteristic	module	17	42
39	characteristic	national culture	20	15
40	characteristic	native language	19	9
41	characteristic	number of sites	56	48

Continued

Table 4 *Continued*

Id	Category	Term	Appearances	
			ICGSE 2009	ICGSE 2010
42	characteristic	offshore	245	193
43	characteristic	offshore outsourcing	29	40
44	characteristic	offshore software development	27	17
45	characteristic	offshoring	82	30
46	characteristic	organisations	172	159
47	characteristic	outsourcing	332	154
48	characteristic	participating organisations	1	0
49	characteristic	phase-based	2	1
50	characteristic	phase	49	51
51	characteristic	GSD project	24	12
52	characteristic	round-the-clock	9	3
53	characteristic	sites	420	357
54	characteristic	socio-cultural	39	13
55	characteristic	software development projects	123	76
56	characteristic	software development teams	45	24
57	characteristic	team member	31	22
58	characteristic	teams	972	687
59	characteristic	teamwork	11	10
60	characteristic	time shift	8	3
61	characteristic	time zone	83	49
62	characteristic	virtual teams	86	55
63	goal	productivity	78	112
64	goal	proximity	48	11
65	goal	proximity to market	2	1
66	goal	proximity to customer	2	1
67	goal	proximity to human resources	0	0
68	goal	reduction in development costs	2	1
69	goal	improve quality	2	0
70	goal	quality	381	291
71	GSD activity	asynchronous communication	11	9
72	GSD activity	control	97	108
73	GSD activity	communication	891	848
74	GSD activity	collaboration	402	311
75	GSD activity	cooperation	39	29
76	GSD activity	coordination	285	318
77	GSD activity	distribution	193	85
78	GSD activity	F2f communication	0	0
79	GSD activity	F2f interaction	0	0
80	GSD activity	F2f meetings	4	0
81	GSD activity	informal communication	26	24
82	GSD activity	interaction	109	86
83	GSD activity	meetings	296	95
84	GSD activity	motivation	35	42
85	GSD activity	synchronous communication	18	10
86	GSD activity	task allocation	104	36
87	GSD activity	task distribution	19	6
88	GSD activity	training	99	197
89	role	coordinator	12	6
90	role	mediator	29	1
91	role	project manager	100	70
92	support	audio	49	22
93	support	blog	1	9
94	support	chat	72	68
95	support	collaborative technologies	26	2
96	support	communication media	39	30
97	support	email	113	150
98	support	forum	18	30
99	support	instant messaging	48	33
100	support	netmeeting	5	5

Continued

Table 4 *Continued*

Id	Category	Term	Appearances	
			ICGSE 2009	ICGSE 2010
101	support	newsgroups	0	1
102	support	phone	34	28
103	support	skype	3	5
104	support	social networks	35	30
105	support	teleconferencing	15	3
106	support	telephone	51	50
107	support	tools	648	554
108	support	video	37	49
109	support	video-conference	11	7
110	tools	collaborative tools	16	8
111	tools	repository	88	76

Classification of terms and their occurrences in ICGSE'09 and ICGSE'10 proceedings