Ontologies in Global Software Development

Rodrigo G. C. Rocha, Ryan R. Azevedo, Sergio Mendonça
Federal Rural University of Pernambuco (UAG)
Garanhuns, Brazil
{rodrigo, ryan, sftom}@uag.ufrpe.br

Alex N. Borges, Catarina Costa, Silvio Meira
Federal University of Pernambuco (Cin - UFPE)
Recife, Brazil
{anbj, csc, srlm}@cin.ufpe.br

The distributed Software Development (DSD) has brought several competitive advantages in software industry, as well as new challenges such as communication and information sharing. In this context, the ontologies can provide benefits such as the definition, standardization and sharing of knowledge involved in the project, allowing a uniform understanding of information and facilitating the collaboration among distributed software development teams. This paper presents a systematic mapping study conducted in order to investigate which ontologies proposed for this context. This work presents evidences from each paper collected and an brief analysis of results reached. The results support the foundation for proposing and developing a feature based on ontologies to support the DSD. The Searches were performed both in manual and automatic way in a set of digital libraries engines and leading conferences in the Software Engineering field. The results support the foundation for proposing and developing a feature based on ontologies to support DSD, besides encouraging further researches that may promote advancements in this area and fostering the adoption these resources by the global software industry.

Empirical Software Engineering, Systematic Mapping Study, Distributed Software Development, Ontology.

I. Introduction

In DSD projects, the teams working are dispersed in different locations [1]. When it comes to projects with scattered teams, the limited communication and lack of sharing and knowledge cause some disadvantages: misinterpreted tasks, lack of collective consciousness relating to the work that has been developed, which issues have been brought up, difficulties following the project's plan and as to take place in a real-time discussion [2].

These challenges have encourage researchers to look for strategies that can help solve these problems, especially, the search for clear, effective information sharing mechanisms, that is essential in this environment. In this context, the use of ontologies as a standard for representing a domain's concept [3] can bring significant benefits to DSD projects, by allowing a simple share of information among dispersed teams.

This scenario motivated the conduction of a research to better understand how ontologies can give support to DSD projects and identify in what way this resource is being applied to this field. In this context, experimentation, which is a kind of empiric study, permits knowledge generation in a systematic, classifiable controlled way, which generates results with greater scientific value [4]. This paper proposes the execution of a systematic mapping study, which is another type of empiric study that is often applied in a research having a broader scope and when it is desired to find as many works as possible in the literature on a certain knowledge field [5]. Therefore, a systematic mapping is conducted to find out which ontologies have been formalized in the context of the software development in a distributed environment.

This article is organized as it follows: there is a description of methodology utilized and results found in this work on Section 2; on Section 3, there is the analysis of the results; and, finally, the concluding remarks.

II. Methodology and Results

In this research, a Systematic Mapping Study was conducted to identify ontologies supporting the DSD. And indirectly to identify tools, techniques, best practices, and models that use ontologies to support this area. An important issue in this process was to search for reviews and accurate analyses on the field, looking for current researches and open challenges related to the use of ontological resources in Distributed Software Development processes. Thus, the following research question were intended to be answered: "Which ontologies have been proposed or adopted in the context of DSD?"

The research questions were defined based on the scope of this mapping. An important issue in this process was to search for reviews and accurate analyses on the field, looking for current researches and open challenges related to the use of ontological resources in Distributed Software Development processes. Thus, the following research questions were intended to be answered:

Which ontologies have been proposed or adopted in the context of DSD?

The first step was to build a search string. Based on the research questions, we identified the main search terms and its synonyms. These definitions were based on other reviews and systematic mappings that involved the search terms: DSD and Ontology. Besides, the definitions were developed under the guidance of experts and researchers. The search string definition involved a testing phase, aiming to refine and obtaining the most appropriate string to the research objectives. The test phase was conducted using different versions of the string and performing automated searches in a few selected digital libraries, such as IEEE Digital Library and Elsevier Scopus. The first step of the test used a most comprehensive search string, composed of several terms related to ontologies and knowledge, such as: knowledge representation, knowledge management, knowledge transfer, conceptualization and concepts formalization. These searches returned several papers, but just a few were related to the ontology topic. Hence all the research terms related with Ontologies have been removed. Using a reduced version of the string, the amount of works returned decreased, however, all the relevant works to the research found in the first search have also been collected. Consequently, the tests performed with the reduced version of the string showed more efficient results. The resulting search string from this process is summarized in Table 1.

TABLE I. SEARCH TERMS

Population	(Ontology or Ontologies)
Intervention	AND (Distributed Software Development OR Global Software Development OR Collaborative Software Development OR Global Software Engineering OR Globally Distributed Work OR Collaborative Software Engineering OR Distributed Development OR Distributed Team OR Distributed Teams OR Global Software Team OR Global Software Teams OR Globally Distributed Development OR Geographically Distributed Software Development OR Offshore Software Development OR Offshoring OR Offshore Outsourcing OR Dispersed Team OR Dispersed Teams OR Distributed Software Project OR Multi-site Software Development OR Distributed Environment of Software OR Outsourced Software Project OR Virtual Team OR Virtual Teams)
Outcome	AND (Technique OR Techniques OR Method OR Methods OR Tool OR Tools OR Software OR Program OR Programs OR System OR Systems OR Model OR Models OR Framework OR Frameworks OR Methodology OR Methodologies OR Good Practice OR Good Practices OR Best Practice OR Best Practices OR Lesson OR Lessons OR Learned OR Success Factor OR Success Factors)

The search strategy used to map the primary studies involves automated searches through well-known digital library search engines. They were chosen based on the relevance for the computer science community, and the availability of papers on this field on the Internet or with libraries, which have partnership with the Federal University of Pernambuco. The search has been performed in the following digital libraries:

- ACM Digital Library (http://dl.acm.org);
- EI Compendex (http://www.engineeringvillage2.org);
- IEEE Digital Library (http://ieeexplore.ieee.org);
- Science Direct (http://www.sciencedirect.com);
- Scopus (http://www.scopus.com/home.url).

The search process also involved manual searches in conference proceedings in the research area. In this stage, the research considered some of the main conferences related to the subject, considered to be more relevant. The conferences defined to perform the manual searches were: International Conference on Global Software Engineering (ICGSE) and Workshop of Distributed Software Development (WDSD). Besides the research in conference proceedings, the manual search involved conversations with experts in DSD, resulting in the inclusion of some articles they found important.

Finally, in the search process, no limitation was imposed regarding to the initial period of publications. The final deadline for the publication of articles was December, 2011, date of the last stage of the search process. Regarding the manual searches, the conferences selected were surveyed from its first year of achievement until the 2011 edition. Studies published later in 2012 have not been considered in the research, in order to produce a more homogeneous result and also to allow a possible future update of this mapping study, which may consider publications from this date.

The searches for the primary studies were conducted according to the research plans defined in the protocol. The search process retrieved 1588 studies from the chosen scientific databases. The Table 2 summarizes the selection process results of the primary studies. The first column presents the digital libraries used in this study and the conferences where manual searches were performed. The second column represents the number of papers retrieved in the search process.

The third column shows the number of selected papers after the first step of selection process, which consisted of evaluating title, abstract and keywords to exclude studies clearly irrelevant to this search. The fourth, fifth, and sixth columns present the number of papers excluded after the second stage (selection process). Finally, the last two columns show, respectively, the number of papers included in the mapping and the percentage of inclusion by research source.

TABLE II. PRIMARY STUDY SELECTION PROCESS RESULTS

Source	**Search Results**	**Relevant Studies**	**Irrelevant**	**Repeated or Duplicated**	**Incomplete**	**Primary Studies**	**% Included**
ACM	236	37	26	7	2	2	5%
EI Compendex	53	21	5	16	0	0	0%
IEEE	550	122	88	9	6	19	50%
Science Direct	225	38	27	7	1	3	8%
Scopus	365	66	23	30	2	11	29%
Manual	157	15	11	3	0	1	3%

Experts	2	2	0	0	0	2	5%
Total	**1588**	**301**	**180**	**72**	**11**	**38**	**100 %**

Analyzing the Table 2, the EI Compendex looks the less efficient digital library, with the lowest number of papers returned. However, the EI Compendex provided a more accurate list of papers in comparison with other research sources. For being the last search performed, the 16 papers found in the EI Compendex base that were considered relevant to the study were excluded because they were duplicates, i.e., they were returned and accounted in other search engines in this research previously.

Other sources that have also been underperforming were the ACM and Science Direct, requiring even more work in the selection of studies. Despite the large amount of studies returned in these two sources, it can be observed the low number of selected articles from them. It is also interesting to notice that the IEEE and Scopus are the libraries that hold the largest number of published papers related to the research topic of this study, representing almost 80% (30 articles) of primary studies included combined. Besides the results of the automatic searches (92% - 35 articles), it can be observed that a few studies (8% - 3 articles) included in the research were found from manual searches or indications of specialists in DSD.

By analyzing Table 2 it is also possible to observe a small number of primary studies returned by search engines of the digital libraries and by manual searches when compared with other mappings and systematic reviews in the field of Software Engineering. This happens especially due the fact that the theme of this research is relatively recent, with many ongoing studies. Furthermore, of the 1588 papers returned in searches, only 38 were included in the research. Therewith is noticeable that the queries presented a considerable level of noise, since only 2,4% of returned studies were really relevant for the search. Many issues may contribute to this result, such as the use of an inappropriate search string or the inefficiency of automated search engines, as discussed by Kitchenham [10].

An analysis of the results of the quality evaluation process, shows that 13 out of 38 primary studies included in this research (34%) have been classified as Excellent, 12 (32%) as Very Good, 10 (26%) as Good e 3 (8%) as Regular, but none being considered Bad. The complete results of the quality evaluation are available at (http://www.rgcrocha.com/ease2013).

This systematic mapping did not restrict the period of publications, although all selected studies were published between 2001 and 2011. This evidences that studies involving the use of ontologies to support DSD are still recent. Hence, most studies (80% - 31 articles) were published between 2006 and 2011, which therefore portraits the relevance this particular topic has been acquiring recently.

The complete protocol is available at (http://www.rgcrocha.com/ms). This question aims to find out which are the ontologies normalized on the DSD context, i.e. To answer this research question, 4 ontologies have been found. The Table 1 presents the proposed ontologies in the distributed context. The first column presents the name and identifier from each ontology. The second column shows a description of each one.

TABLE III. ONTOLOGIES FOR DSD

Models	Description
OFFLOSC [PS10]	This ontology is formalized in the context of open-source software development communities. Its goal is help coordinate activities, management of resources and knowledge sharing. It is composed by 46 classes and describes the concepts related to open-source communities such as actors, artifacts, activities, operations, relationships and resources.
Knowledge Management Ontologies [PS18]	A set of ontologies that formalize structural concepts of DSD environments, directed to knowledge management. It describes concepts of software artifacts, environment problems, interaction among the distributed development teammates, infrastructure, business rules and general information of the project.
Open Source Communities [PS32]	This ontology is also formalized in the context of open-source software development and its main purpose is to compose a project knowledge basis having semantically related, categorized data, which allows the execution of semantic searches and data inferences by smart agents. It is composed of 6 classes that describe concepts of actor's relations, rules, activities, processes, artifacts and tools from open-source communities' projects.
OntoDISEN [PS35]	This ontology is formalized in the DSD Project scenario and is used to aid the establishment of communication between distributed teams. It is integrated to a textual information-spreading model, enabling sharing information in distributed environments to be comprehended by all the software engineers in a clear, homogeneous way. It describes concepts of elements that are represented and shared in a DSD environment, such as users, tools, other environments, activities and processes.

III. DATA ANALYSIS

This section presents an brief analysis of the results found in this study. Besides, it is of interest to notice that numerous another resources were found in this research. Like models, tools and best practices that use ontologies for better their activities. Each resource found is focused in a specific area from Software Engineering aiming to improve this area.

Based on results, it is evident that the development phases that are benefiting from the use of ontologies are: process, management, requirements and design. On the other hand, some important branches have not been fully approached, for example, quality and tests, which involves lots of information management activities, and may have a considerable evolution with the utilization of ontologies as means to standardize, manage and share knowledge.

By answering the research question from this mapping, there have been found four works that propose some ontologies especially developed for distributed software development, according to what is presented previously. Since these ontologies have been designed specifically for distributed teams, they bear the concepts and features required to work in this environment. Noteworthy to mention that two of the four ontologies were developed for open-source software development communities. The free dynamic nature of this environment poses challenges to the coordination of activities and knowledge sharing.

Therefore, the use of ontologies as a support to open-source software development simplifies the management of knowledge resources in the communities. Noticeable that several other works use ontologies to solve or mitigate challenges and in DSD environments, however, these ontologies are not specific for this environment. They are ontologies for Software Engineering, but if applied on DSD projects, they might help. The Table 4 depicts the several ontologies found in this study.

TABLE IV. LIST OF ONTOLOGIES

Field	Primary Study
Software Component	PS01
Business Domain	PS02, PS09, PS16, PS27, PS31
Software Engineering	PS02, PS06, PS07, PS08, PS21, PS28, PS30, PS38
Project Management	PS02
Problems and Solutions	PS02
Collaborative Structure	PS03, PS04, PS14, PS23
Team Division and Role Assignment	PS05, PS13
General Project Data	PS06, PS08, PS19, PS20, PS22, PS25, PS27, PS30, PS31, PS33, PS34
Software Requisits	PS09, PS16, PS24, PS29, PS31
Code and Bugs	PS14
Software Artifacts	PS17
Software Tests	PS26
Software Design and Architecture	PS27, PS37
Linguistics Services	PS34
Constitution of Electronic Contract	PS36

As seen on Table 3, there are numerous tools that utilize nonspecific-to-DSD ontologies only to mitigate challenges and limitations. These tools are distributed and used as support in the various project parts, from actual Software Engineering branches to specific project activities.

With these results, it is clear that there are a lot of advantages in using ontologies to support DSD, specially to generate solutions aiming at mitigating the communication, collaboration, knowledge flow management, coordination of project activities and knowledge, and process management issues.

IV. CONCLUDING REMARKS

The DSD work environments are very complex and there are no mature practices for this context since it is relatively new. In this sense, ontologies can bring benefits such as a shared understanding of information, ease of communication among distributed teams and effectiveness in information management.

This work presents evidences from each paper collected and an analysis of the results reached. The results support the foundation for proposing and developing a feature based on ontologies to support the DSD. This research aimed to identify ontologies formalized in DSD context and resources (models, tools, techniques and best practices) that use ontologies to support the DSD. Most studies have been being published from 2006 to the present. Through results, it is possible to affirm that ontologies were essential for some researches and some teams and projects already use tools based on ontologies aiming to establish information sharing and to improve the software development process as a whole. Is possible to view all the Sistematic Mapping Results in Borge's work [7].

From this work, some research can be developed: through of OOPS! Tool [8] (OntOlogy Pitfall Scanner!), is possible to detect some of the most common pitfalls of four ontologies found. Futhermore, development of an ontology to map all the DSD domain; presentation of solutions to assist project management in such an environment, proposing solutions to test process and software quality in DSD; and to indicate and use tools to support collaboration among distributed teams with the attachment of knowledge through ontologies.

REFERENCES

[1] Carmel, E. 1999. Global Software Teams – Collaborating Across Borders and Time-Zones. Prentice Hall, NJ, EUA.

[2] Wongthongtham, P., Chang, E., Dillon, T. S. 2004. Ontology-based Multi-agent System to Multi-site Software Development. In Proceedings of the Workshop on Quantitative Techniques for Software Agile Process (QUTE-SWAP). (Newport Beach, USA, Oct. 31 - Nov. 05, 2004). ACM, New York, NY, 66-75. DOI= http://doi.acm.org/10.1145/1151433.1151443.

[3] Gruber, T., A. 1993. Translaction Approach to Portable Ontology Specifications. Technical Report, Knowledge Systems Laboratory, Computer Science Department at Stanford University.

[4] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A. 2000. Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell, MA, USA

[5] Kitchenham, B. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report. Keele University at Staffordshire and University of Durham at Durham.

[6] Kitchenham, B. 2004. Procedures for performing systematic reviews. Technical Report. Keele University at Staffordshire.

[7] Borges, Alex., Rocha, Rodrigo., Costa, Catarina., Tomaz, Hilário., Soares, Sérgio., Meira, Silvio. Ontologies Supporting the Distributed Software Development: a Systematic Mapping Study. In Proceedings of the International Conference on Evaluation & Assessment in Software Engineering (EASE). Porto de Galinhas, PE, Brasil. 2013.

[8] Poveda Villalon, Maria and Suárez-Figueroa, Mari Carmen and Gómez-Pérez, A. (2010) A Double Classification of Common Pitfalls in Ontologies. In: Workshop on Ontology Quality (OntoQual 2010), Co-located with EKAW 2010 , October 15, 2010, Lisbon, Portugal.