2013 IEEE 8th International Conference on Global Software Engineering

An Ontology for Task Allocation to Teams in Distributed Software Development

Anna Beatriz Marques, José Reginaldo Carvalho, Rosiane Rodrigues, Tayana Conte
Institute of Computing
Universidade Federal do Amazonas (UFAM)
Amazonas, Brazil
{anna.beatriz, reginaldo, rosiane, tayana}@icomp.ufam.edu.br

Rafael Prikladnicki, Sabrina Marczak
Computer Science School
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil
{rafaelp@pucrs.br, sabrina.marczak}@pucrs.br

***Abstract*— An adequate task allocation plan is an effective strategy to reduce collaboration issues in distributed software development. Practitioners adopt distinct processes to allocate tasks as well as diverse labels for the same activities and artifacts. This diversity is also found in literature. Task allocation proposals consider different elements and use distinct names for the same concepts. The lack of a standardized vocabulary and of an understanding of the elements involved impairs knowledge acquisition and sharing. Our paper presents a domain ontology to represent concepts related to task allocation in distributed teams. The ontology was defined based on a literature systematic mapping and on the opinion of experts. Preliminary evaluation suggests that the relationships among concepts are valid in real projects. The ontology brings awareness to managers regarding the factors related to task allocation planning and provides researchers with a framework to define processes and design tools to support such activity.**

Keywords— Distributed software development, Task allocation, Domain Ontology.

I. Introduction

In distributed software development projects the development of the products turns into the result of the collaboration between distributed teams in different locations and, in some cases, in different time zones. Despite the organizations' ability to benefit from distribution, many difficulties are often faced by distributed teams such as communication, management, and control of the activities [8]. A way to reduce these difficulties and to benefit from the collaboration within this peculiar development scenario is to adequately plan task allocation of those involved in the development life cycle.

Task allocation in software development can be defined as a process that determines the manner in which the project tasks will be executed, how and for which resources the tasks will be assigned taking into account existing constraints [13]. These tasks represent what has to be done to achieve the project goals and, ultimately, the organization's goals.

Defining tasks in distributed projects often involves defining their technical constraints related to assigning the tasks to distributed teams. Therefore, the task allocation plan should take into consideration characteristics about the teams (e.g., skills and abilities), about relationships among the distributed teams (e.g., cultural differences, common experiences, and communication overhead), and about the product to be developed itself (e.g., complexity, dependencies, and required abilities for its development) [7][4][5].

These characteristics are tightly coupled and directly related to three aspects of any distributed project [10]: (1) project duration: delays are difficult to be compensated due to the management decentralization and decision-makers be spread out among several sites; (2) product quality: to allocate tasks to inadequate or inexperienced teams will compromise the quality of the results, and imply in more delays due to rework; and (3) communication: distributed teams have higher communication needs when compared to a centralized-single-site project. They also face more barriers towards an efficient communication due to cultural differences and geographical separation. The understanding of these three aspects is fundamental for a good task allocation plan, which is, in turn, an essential factor for a successful distributed software development project.

Practitioners consider distinct criteria to allocate tasks in distributed projects, such as the product modules, product design, teams' location, teams' expertise, cultural alignment, among others. The task allocation planning process is usually defined based on the organizational background. For example, in a national distributed project it is not strongly necessary to consider cultural differences, since such differences might not be of a significant influence on the team working dynamics. On the other hand, in a distributed project cultural differences and team's location could affect or turn the communication between teams overwhelming. Therefore, specific distribution development scenarios have different constraints and require criteria to be considered in planning of task allocation. In such a diverse context, project managers need to be able to communicate and perfectly understand each other about what is involved in the process of allocating tasks to the distributed teams under their responsibilities. Thus, a shared understanding and common ground about factors or criteria to be considered to adequately plan task allocation is strongly necessary.

978-0-7695-5057-2/13 $26.00 © 2013 IEEE
DOI 10.1109/ICGSE.2013.12

However, there is still no consensus in the literature about factors, criteria and methods that should be considered when planning task allocation to distributed teams. Many proposals have been presented in literature (e.g. [4][5][7][9][11][14]). In our systematic review of literature [10] we have identified six distinct proposals, each considering different aspects of task allocation planning. For instance, we identified that four proposals consider aspects affecting the duration of the project [4][7][9][14]), only two proposals consider aspects affecting the quality of the project [5][7]), and two consider communication [7][11]). We also noticed that each proposal use different names to represent these aspects. The different criteria proposed by each solution make it difficult to project managers to analyze the situations in which each proposed solution can be applied.

Researchers need to better understand the concepts involved in task allocation to teams in order to build adequate proposals to analyze and to identify the best alternative to allocate tasks in different contexts in distributed software development (DSD) projects. In order to contribute to this important matter, we propose a domain ontology that defines the concepts related to tasks allocation to distributed teams and the relationships between each one of the concepts that comprise the proposal.

A domain ontology is the representation of a conceptualization that is captured by a vocabulary [1]. It can be used as a way of presenting the concepts of a determined domain and their relationships. The ontology that we propose aims at providing a reference (a) for a better understanding of the existing concepts related to task allocation in distributed software teams, and (b) to suggest adequate proposals to identify and analyze the various alternatives to allocate tasks to distributed teams. The proposed ontology intends to support both project managers during task allocation planning and researchers in the development of proposals to task allocation.

The remainder of this paper is organized as follows. Section II describes the concepts regarding tasks allocation to teams in distributed software development. Section III describes the method for the definition of the ontology. Section IV presents the proposed ontology for defining the task allocation process to teams in distributed projects. Section V describes the preliminary evaluation of the ontology regarding its minimum ontological commitment. Section VI discusses implications of our proposal to practitioners and researchers. Section VII concludes the paper with our final remarks and future work.

II. Tasks Allocation to Teams in Distributed Software Development

The allocation of tasks to distributed teams consists in simultaneously match the right task with the right team in the correct order while defining the way to carry the tasks out [13]. Furthermore, knowing that human resources and time are limited, in order to plan carefully, project managers must consider: (a) the characteristics of the teams, (b) the characteristics of the products that will be developed, (c) the characteristics of the tasks and the relationships among them, and (d) the business goals of the organization. Some approaches have been proposed to support managers with the task allocation planning process in a distributed context.

Lamersdorf and Münch [7] suggest that an appropriate task allocation planning must consider cost of task execution and communication cost among distributed teams. Jalote and Jain [5] propose a process to allocate tasks that aims at reducing the duration of a project. Pereira, dos Santos, Ribeiro, and Elias [11] propose a recommendation framework that aims to reduce communication needs among distributed teams through identifying the most independent modules which are candidates to be decomposed into the final product and by allocating them to the most suitable teams based on technical and non technical characteristics.

The proposals above mentioned suggest that there are different factors to be considered when planning task allocation to teams in distributed software projects. Distributed teams are often composed of members with different competences, have access to different technologies, and are located in different places and time zones. These differences influence the way in which software products are modeled, developed, and integrated [12] as well as how project managers perceive what has to be considered to plan the allocation of these teams. Therefore, a shared understanding of what has to be taken into account is necessary to facilitate the work of decision-makers in allocating tasks to distributed teams.

We propose a domain ontology to contribute with literature and provide researchers and practitioners with a common understanding of the different aspects and criteria necessary to the task allocation planning of distributed software teams. This ontology was theoretically developed based on findings from a literature systematic mapping in which two other existing ontologies were used as reference, and empirically-informed based on experts' opinion. We present next how this ontology was defined.

III. Method for the Ontology Definition

In order to investigate the existing proposals in literature for supporting task allocation processes and the factors they consider we have performed a Systematic Mapping literature review. This review was performed based on Kitchenham and Charters's review protocol [6].

A. Systematic Mapping

Kitchenham and Charters [6] define Systematic Mapping (SM) as a type of systematic literature review that has research questions of exploratory nature. The purpose of a systematic mapping is to provide a general view regarding a determined research area [6].

We conducted a SM to identify the existing proposals to task allocation in distributed software development. To guide this SM, we defined the following main research question: *"Which are the existing proposals that support task allocation processes in distributed software development projects?"*. A secondary research question was also defined: *"Which tasks and teams characteristics have to be taken into account in planning task allocation of distributed software teams?"*

("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 "global software teams" OR "globally distributed development" OR "geographically distributed software development" OR "offshore software development" OR "offshoring" OR "offshore" OR "offshore outsourcing" OR "dispersed teams") AND ("task allocation" OR "task distribution")

Figure 1. The adopted search string

In order to retrieve papers to answer our research questions, we performed an automatic search in the IEEExplore, ACM Digital Library, Scopus and Compendex digital libraries using a search string containing synonym terms related to distributed software development and task allocation. Figure 1 shows the search string.

To supplement the coverage of our search, we manually searched for papers at the proceedings of the Distributed Software Development Workshop, a workshop organized in Brazil in which its proceedings are not indexed in digital libraries. Though this SM, we identified six task allocation models for distributed projects. We analyzed each model regarding the tasks and teams characteristics that have been taken into account. We present our findings next.

B. Findings from the Systematic Mapping

Results of our systematic mapping allowed us to identify the tasks and teams characteristics that have influence in the process of task allocation in distributed software teams. Table I presents the characteristics considered by each proposal. Characteristics were grouped into two main categories: (1) task-related characteristics, which describes the characteristics of the tasks that are considered by the proposal, and (2) allocation environment-related characteristics: characteristics of teams, sites, or members of teams, which includes collaboration aspects.

More specifically, the TAMRI model [7] considers tasks and remote sites characteristics to estimate the execution cost of the tasks and the communication cost between remote sites. This model is customizable, meaning that it is possible add (or remove) the desired (or undesired) characteristics to be considered in task allocation. These characteristics are modeled in Bayesian networks to represent the impact of the execution and communication costs in time, cost and quality. The Framework for Software Product Lines [11] is another proposal that also considers communication. This proposal describes that the tasks are defined based on the product architecture. Therefore, to reduce communication needs, the framework attempts to identify software modules as independent as possible. Then, the framework suggests a set of allocations considering the competences of the teams.

Mak and Kruchten [9] propose the NextMove model to prioritize tasks and allocate them to the members of teams considering their competences and availability. The 24-h software development model [5] considers skills and availability of teams to allocate tasks in order to reduce the project duration. The DIMANAGER model [4] also considers competence and availability of members to assigning tasks. Finally, the Simulation model [14] allows configuring a distributed environment considering remote sites processes and allocation strategies in order to analyze the best alternative for task allocation.

TABLE I. COMPARISON TABLE AMONG CHARACTERISTICS CONSIDERED BY THE PROPOSALS IDENTIFIED IN THE SYSTEMATIC MAPPING

Characteristics		**TAMRI [7][8]**	**24-h software development model [5]**	**Framework for Software Product Lines [11]**	**NextMove [9]**	**Simulation model [14]**	**DIMANAGER [4]**
Characteristics related to tasks	Needed proximity to customer	X					
	Needed knowledge and skills	X	X	X			X
	Execution time/effort estimative	X	X		X		
	Tasks dependences	X	X	X	X		
	Required resources		X				
	Features			X			
	Deadlines				X		
	Related artifacts				X		
Characteristics related to allocation environment	Cost rate	X					
	Proximity to customer	X					
	Team members knowledge and skills	X	X	X	X		X
	Process maturity	X					
	Time zone, language and cultural differences	X					
	Common experiences among teams	X					
	Communication infrastructure	X				X	
	Working time		X	X			
	Technical performance			X			
	Familiarity with related artifacts				X		
	Effort				X		
	Availability						X

During the detailed analysis of these proposals, we identified that there is no theoretical basis with significant depth for the allocation of tasks to distributed teams. The theoretical basis used by the identified proposals is often summarized and focuses in a determined aspect. For instance, the authors that created the TAMRI model [7] formulated its theoretical basis grounded on a qualitative study. As a result, none of the identified proposals cover all aspects that have been identified in our study. Suck lack of completeness gives us the opportunity to propose an ontology that relates all aspects suggested in literature offering project managers with a guide of what factors to account for when planning task allocation to distributed software teams.

We also identified that there is no consensus regarding the factors that should be considered in the allocation of tasks to distributed teams in distributed projects. The existing proposals consider different factors based on the utilization of different nomenclature. Furthermore, they do not clearly state in which situation these factors are feasible, confirming that there is a need for the establishment of a common ground regarding the involved concepts of task allocation to distributed teams.

C. *Defining the proposed ontology*

Domain ontologies are used as a way of presenting the concepts of a determined domain and their relationships. According to Chandrasekaran and Josephson [1], ontology is the representation of a conceptualization that is captured by a vocabulary. Such vocabulary is specialized for an application domain. The definition of a common vocabulary for task allocation among distributed teams that represents the relevant factors of such domain can contribute to: (a) knowledge sharing, and (b) increasing the semantic interoperability with other domains.

Among the studied proposals in our SM, we identified the NextMove model [9], which is based on an ontology that represents the policy for coordination and allocation of tasks. The concepts within this ontology represent the abstract elements of the process: (a) role, (b) activity, and (c) work product; and physic elements of the project: (a) team member, (b) task, and (c) artifact. The goal of using such concepts is to provide flexibility during the execution of the project. Table 1 showed the attributes from each concept. However, the ontology did not offer a complete view of the task allocation process in distributed development since it does not have, for instance, a representation of the distribution or different costs of teams. This lack of representation can be explained by the actual goal of the solution, which is the allocation of tasks to teams with more suitable abilities and more familiarity with the involved artifacts.

Considering the identified limitations of the proposals studied, we defined an ontology that incorporated the concepts captured during the SM to represent the process of task allocation to teams in a distributed software project. To define our ontology we used the Systematic Approach for Building Ontologies (SABiO) proposed by Falbo [2] as a reference.

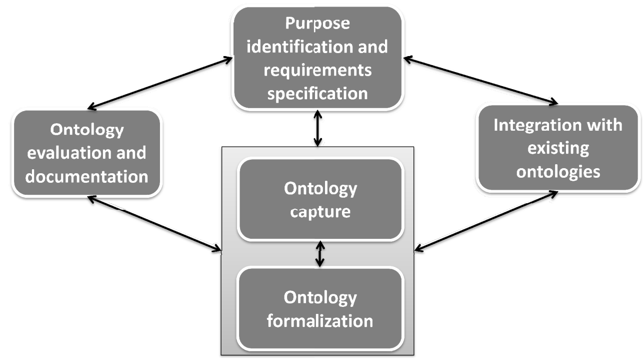

Figure 2. Steps from the SABiO - Systematic Approach for Building Ontologies [2]

The SABiO approach guides the creation of domain ontologies by proposing well-defined and clear steps to be followed. Figure 2 illustrates the steps to be taken. The steps are detailed below.

- Purpose identification and requirement specification: it concerned to the precise identification of the ontology purpose and its intended uses, i.e. the competence of the ontology. Competency questions are used to achieve such goal.

- Ontology capture: the goal is to capture the domain of conceptualization based on the ontology competence. Relevant concepts and relations should be identified and organized. A model using a graphical language and a dictionary of terms should be used to aid communication with domain experts.

- Ontology formalization: it aims to explicitly represent the conceptualization captured in a formal language.

- Integration of existing ontologies: during the capturing or the formalization steps, it can be necessary to integrate the current ontology with existing ones in order to use previously established conceptualizations.

- Ontology evaluation: the ontology must be evaluated to check whether it satisfies the requirement specifications. It should be evaluated with respect to the ontology competence and some design quality criteria, such those proposed by Gruber [7].

- Documentation: all the ontology development must be documented.

As the ontology proposed in this work was based on the results of our SM (See section III.B), we needed to include some concepts regarding software processes (activity, artifact) and software organizations (human resources, teams, organizations). Therefore, we reutilized these concepts from the Software Process Ontology (SPO) [3] and the Software Enterprise Ontology (SEO) [15]. The SPO provides concepts related to activities, artifacts and procedures, while the SEO defines concepts related to resources, competences, organizations and project. These ontologies were respectively created following the SABiO approach.

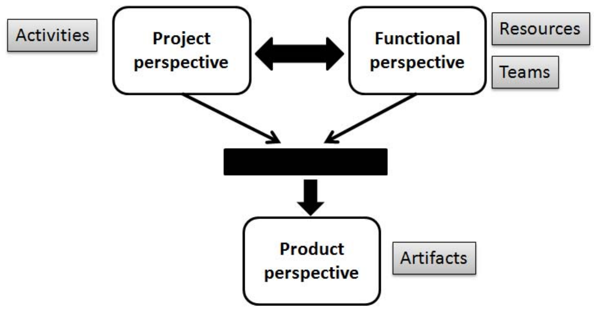

Figure 3. Perspectives of the proposed ontology

SPO and SEO are based on traditional projects. In order to incorporate the concepts captured from specific aspects of distributed projects, which are not considered by the adopted ontologies, we proposed three perspectives based on an organizational structure, namely: (a) product perspective ("What will be done?"); (b) functional perspective ("Who will do it?"); and (c) project perspective ("How will it be done?"). Figure 3 shows the relationship between these perspectives.

Task allocation in traditional projects focuses on the project perspective, which is related to aspects regarding the activities that will be performed. On the order hand, the allocation in distributed projects needs to detail the functional perspective, which is the perspective that gathers many aspects of the team. For instance, geographical location, expertise, and how much would cost to assign task to a given team are functional aspects. The integration of the project and functional perspectives will lead to the construction of the different artifacts that will result in the final product.

The requirements of an ontology are represented by competence questions. The competence questions from the integrated ontologies were the basis for the formulation of the other questions. Since the integrated ontologies are applicable to the context of traditional software development, it was necessary to formulate further competence questions to represent the peculiarities of the allocation of tasks in distributed projects. Such questions should consider the project, functional and product perspectives.

Table II describes the competence questions for each perspective. Note that the referenced questions were posed by the two adopted ontologies, while the remaining questions were proposed by the authors. The latter are indicated as "Proposed" in the table.

In order to identify the concepts and relationships encompassed by the competence questions, we considered the characteristics that are addressed in the task allocation planning process mapped by our SM in Section III.B along with actual scenarios, based on case studies on a following-the-sun global project environment. These sources were usefull to drive the creation of new concepts whenever necessary. The StarUML tool [16] was used to design the graphical model that illustrates our proposed ontology.

TABLE II. COMPENTENCE QUESTIONS OF THE PROPOSED ONTOLOGY

Perspective	Competence question	Reference
Product	Q1. What is the nature of the artifact?	Proposed
	Q2. What is the origin of the artifact?	Proposed
	Q3. How can an artifact be decomposed?	Proposed
	Q4. What is the complexity of an artifact?	Proposed
	Q5. Which artifacts are input for a determined activity?	[3]
	Q6. Which artifacts are produced by a determined activity?	[3]
	Q7. What is the importance of the artifact?	Proposed
	Q8. What is the priority of the artifact?	Proposed
Project	Q9. What is the nature of an activity?	[3]
	Q10. How can an activity be decomposed?	[3]
	Q11. Which activities must precede a determined activity?	[3]
	Q12. What procedures can be adopted in the execution of an activity?	[3]
	Q13. In which activities of an organization is a determined competence required?	[3]
	Q14. Which are the constraints of a determined activity?	Proposed
	Q15. Which is the nature of a constraint?	Proposed
Functional	Q16. Which Organizations participate in a determined project?	Proposed
	Q17. What is the role of an organization in a determined project?	Proposed
	Q18. Which are the teams of a determined project?	[17]
	Q19. Which people, within the organization, posses a determined competence?	[15]
	Q20. Which are the levels of influence of a human resource in a language?	Proposed
	Q21. Which people are allocated to the execution of a determined activity?	[17]
	Q22. What is the performance of a determined team in the execution of a given activity?	Proposed
	Q23. What is the cost of an activity?	Proposed
	Q24. Which teams attend to a determined constraint in a given activity?	Proposed
	Q25. Which teams interact with each other during the execution of a determined activity?	Proposed
	Q26. Which are the cultural characteristics of a team?	Proposed
	Q27. What is the time overlap between teams that interact with each other?	Proposed
	Q28. Which teams have common experiences?	Proposed
	Q29. Which are the goals in a project?	[17]
	Q30. What is the priority from a determined goal?	[17]

IV. The Ontology for Defining the Process of Task Allocation to Teams in Distributed Software Projects

The proposed ontology represents concepts related to task allocation in distributed projects. We aimed at capturing concepts that describe planning tasks in distributed software development projects and aspects that influence decision-making during task allocation planning in such projects. These concepts are related to the three pespectives that guided the definition of competence questions mentioned in Section III.

As we deal with the allocation of tasks or activities, the main concept of the ontology is *activity*. Also important are the following concepts, all closely related to *activity*:

- *Artifact:* defined as the input or output of an *activity*;
- *Competence*: what expertises are necessary for its realization;
- *Constraint*: what are the limitations that restricts the activity realization;
- *Human resource*: who are the people responsible for the activity realization. It is also the concept that will detail the characteristics of teams and their geographic dispersion.

To facilitate the comprehension of the ontology, we decided to present it in fragments organized by the main concepts that compose the ontology itself, which are: artifacts, activities, competences, teams, organizations, and project teams. Also, the concepts adopted from the Software Process Ontology (SPO) [3] and from the Software Enterprise Ontology (SEO) [15] will be identified in the representation by the SPO and SEO labels, respectively. For instance, Fig. 4 shows a fragment of the ontology that deals with the concepts related to *artifact*, adopted from the Software Process Ontology [3].

ARTIFACT FRAGMENT: An *artifact* represents the resulting products from the development process [3]. Furthermore, in the highest composition level, an artifact is the final product, the ultimate distributed software development output. On the other hand, on the lowest level, an artifact represents each partial product that might be an outcome of the development. As partial products one may have, for instance, a requirements document, a software module, or a test procedure. Consequently, an artifact can be the either the input to or the outcome of an *activity*.

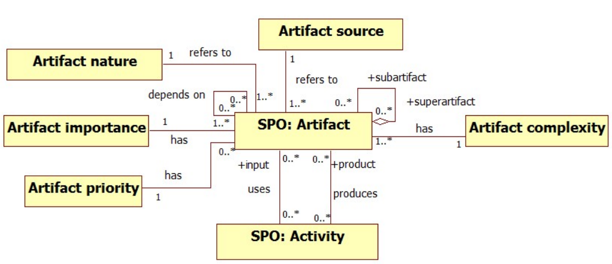

Figure. 4. Ontology – Fragment I: Artifacts and related concepts

Artifact Importance and Priority: An artifact has *importance* and *priority*, which are inter-related concepts. Usually the more important an artifact is, the higher is its priority. However, they are different attributes. The importance of an artifact is not supposed to change, while its priority depends on the project current status. *Artifact* also has an associated *complexity*; which is defined as the technical difficulty to develop it. Complexity is a relevant concept for task allocation to teams since it might determine the time and effort necessary for its development.

Subartifacts: Artifacts can also be decomposed into inter-dependent sub artifacts without considering defining the order of precedence of their development. For instance, a possible situation is to assign the coding and testing of a set of features to distinct teams working in different time zones simulating a following-the-sun scenario. Both coding and testing are inter-dependent; the artifact finishes when both finish. An *artifact* that was decomposed into *subartifacts* is a *superartifact*.

Artifact source: The *artifact source* indicates if an artifact was developed or acquired which is a particular characteristic in distributed projects defined by a "make-or-buy" decision [13].

Artifact nature: The *artifact nature* describes the artifact relationship with the organization. It can be part of the core business of the organization, and thus should not be outsourced, or it can be an artifact built on top of the application layer (not touching sensitive technology), and as a consequence a candidate to be outsourced. It might be the outcome of a supporting activity (e.g. system test), or it can be part of the organization's strategic marketing, enabling many inovative features and thus, demanding to be closely controlled (e.g. an acquired JVM for a java-based device).

ACTIVITY FRAGMENT: Figure 5 shows the fragment that represents the concepts related to *activity*. In this fragment we defined concepts to represent the constraints that an activity can have and, therefore, must be taken into consideration when allocating tasks to distributed teams.

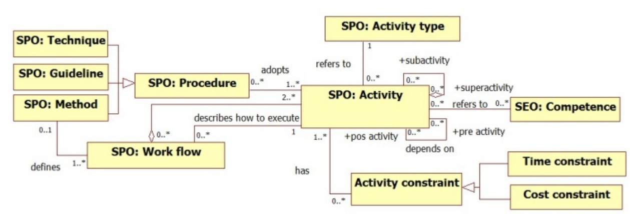

Figure 5. Ontology – Fragment II: Activities and related concepts

Activity type: As defined in the SPO [3], an *activity* is related to an *activity type*. A type defines if the activity is related to testing or development, for example.

Activity competence and procedure: To be developed an *activity* requires *competences*, adopts *procedures*, and uses the *artifacts* that were produced by other activities.

Activity constraint: An activity has *constraints* that must be respected during its development. Time and costs necessary for the execution of an activity are classic examples of constraints. *Time constraints* are related to the time in which an activity must be carried out, such as start and finish time. *Cost constraints* are associated to the monetary values that must be considered for the development of an activity.

Subactivity and preactivity:An activity can be decomposed into *subactivities* and can depend on the realization of other activities, named *preactivities*.

COMPETENCE FRAGMENT: Competences must be considered in order to idenfity which team has the competences needed for the activity development. Figure 6 shows the fragment of the proposed ontology regarding competence and its related concepts, which captures the aspects that helps in the analysis of the allocation possibilities among different teams.

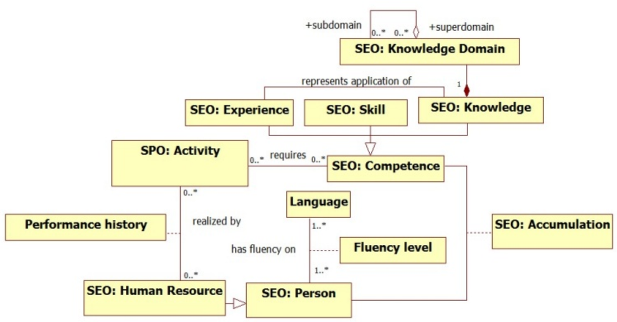

Figure 6. Ontology – Fragment III: Competences and related concepts

Competence: People build up competences throughout their professional carrier, making human resources capable of carrying out the defined activities for a project.

Knowledge, ability and experience: A competence can be classified as *knowledge*, *abilities (or skill)*, and *experiences*. *Knowledge* is the appropriation of objects through the definition, clear perception, analysis, complete apprehension, or other forms of appropriation by ones mind. Furthermore, *knowledge* is composed by various *knowledge domains* that can be wider domains, named *super domains*, composed by *sub domains* [15]. *Experience* represents the application of knowledge and is, therefore, acquired through practice. Finally, *abilities* are acquired skills not associated to an activity or specific knowledge domain [15]. For intance, good writting skills is considered an ability.

Human resource and performance history: The concept of *performance history* represents the performance of a *human resource* during the development of an *activity* and can contribute in the decision-making process.

Language and fluency level: As distributed projects can be dispersed in a global context, the concept of *language* was created and related to the concept of *human resource*. As the fluency in a *language* has a strong impact in communication, this concept is relevant and needed in our ontology.

TEAM FRAGMENT: To represent the characteristics of the distributed teams and the collaboration among them in a distributed project, we included concepts related to geographical and cultural differences among teams (refer to Fig. 7). A *team* is composed by *human resources*. For a given *human resource* of a specific *team* performing an *activity*, the *team* must meet all *activity´s constraint*s. A *project team* is a *team* that is involved in a specific project.

Historical work: Considering that the final product will be the result of the collaboration between teams, we included a relationship that describes that a team works with another team. We included the concept of *historical work*, which describes situations the teams have succesfully worked together.

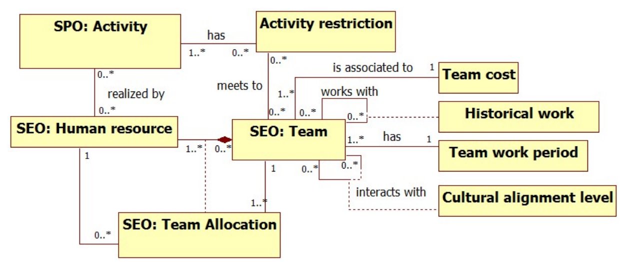

Figure 7. Ontology – Fragment IV: Teams and related concepts

Team work period, cultural alignment and cost: To represent the time dispersion between teams, we included the concept of *team working period*. Moreover, in order to represent cultural differences among teams, we included a relationship that describes that a team interacts with another team and the concept of *cultural alignment level*. These concept will be used to describe the level of similarity of the existing cultures based on factors like language, time zone, and geographic dispersion. Finally, the concept of *team cost* was included to represent the associated cost of maintaining a team.

ORGANIZATION FRAGMENT: Figure 8 shows the fragment of the ontology regarding human resources and the organization to which they belong to.

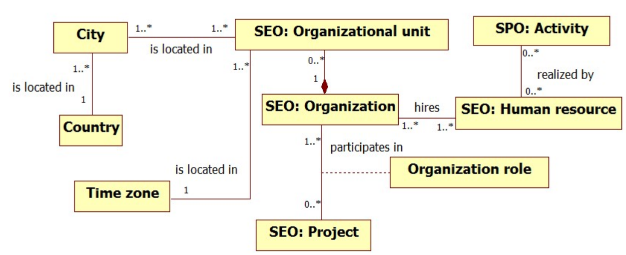

Figure 8. Ontology - Fragment V: Organizations and related concepts

In the SEO [15], the authors represent the concepts regarding the hiring of human resources by the organizations. As in a distributed context many organizations participate in a given project, we added new concepts to this fragment in order to represent the distribution of these organizations.

Organization role: One or more *organizations* can participate in a project, playing different *organization roles*, such as headquarter office, subsidiary office, or subcontracted office. The headquarter office is the one that controls the project as a whole and is the owner of the final product. A subsidiary office is an association to the headquarter office, but it does not hold overall control over the project. Furthermore, a subcontracted office offers services or develops part of the final product.

Organizational unit and time zone: An *organization* can also be composed of *organizational units* that can be dispersedly located. Consequently, an *organizational unit* can be located in one or more *countries* within a determined *time zone*.

PROJECT TEAM FRAGMENT: The participation of an organization in a *project* is given through the hiring of *human resources* to its *organizational units*. These human resources will perform activities in the project once they get allocated to the *project teams*. Figure 9 shows the fragment of the ontology regarding the *project teams* and the related concepts. A project can have one or more project teams.

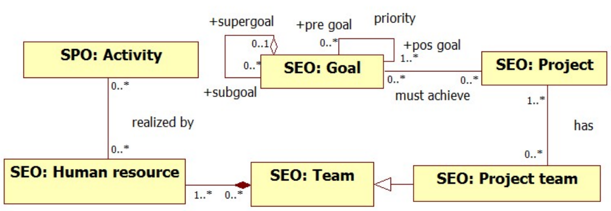

Figure 9. Ontology - Fragment VI: Project teams and the related concepts (Source: Falbo and Bertollo [3] and Santos et al [15])

Goal: As defined by Santos et al.[15], a project must achieve one or more *goals*. A *goal* can be decomposed into *subgoals*. Some goals can have higher priority than others. Goals with higher priority are denominated *postgoals*. The tasks allocation process must meet the projects goals.

V. PRELIMINARY EVALUATION OF OUR ONTOLOGY

Our proposed ontology was defined based on the projects' characteristics identified in our systematic mapping and on the opinion of experts in distributed software projects. To represent some of the related concepts we adopted pre-defined concepts from two other existing ontologies. In order to preliminarily evaluate our ontology regarding the validity of the included concepts and their relationships, we interviewed five project managers of distributed projects (identified as PM1-PM4 here). Due to physical distance, three of them filled out an online interview form. We followed-up when necessary.

The interview guide was elaborated to obtain information about a distributed project that the interviewee had participated in. The interviewees were instructed to think about a recent project in order to answer to the questions based on a concrete scenario. We asked about characteristics of: (1) the project, (2) the teams, (3) the tasks and artifacts, and (4) the task allocation process adopted. The interview guide is presentd in Table III.

We mainly aimed at validating the relationships related to the new concepts included to build the proposed ontology. With the interviewee feedback, we obtained information that enabled us to confirm the relationships related to the dispersion of the *teams* and the distribution of the *tasks* (or *activities*). Relationships related to the costs were not confirmed at this time. We present excerpts of the interviews that suggest confirmation of the proposed new concepts.

ARTIFACTS: Three of the respondents reported that an artifact has a related importance as the relationship **"Artifact has Artifact importance"** expresses. They also reported that most of the important artifacts are often developed by the headoffice. One of the project managers said:

"The product modules have different importance. The core technologies that belong to software enterprise are normally developed by the headoffice. Over time, as the partners acquire more expertise, practice and performance, the headoffice will assigning the most important product modules to them." [PM4]

TABLE III. INTERVIEW GUIDE

	Questions
Charact. of the project	What was the final product developed by the project?
	Were there more than one organization involved in the project? Or were the teams from a unique organization?
	What was your role as project manager?
Characterization of the teams	What was the dispersion level among the teams? Were the teams located in different countries or time zones?
	Were there differences on the work period of the teams?
	(If yes) How were these differences reduced?
	Were there cultural differences among the teams?
	(If yes) What were the existing cultural differences? How were they reduced?
Characterization of artifacts/activities	How was the final product decomposed into artifacts to allow the distribution?
	Were there artifacts with differents importance levels?
	(If yes) Has the importance of an artifact influenced in its distribution?
	What were the types of activities developed?
	Were some types of activities outsourced?
Characterization of task allocation process	How was the task allocation to teams planned?
	What were the criteria considered?
	(If not mentioned) Were the commom experiences of teams that have previously worked together considered?
	(If not mentioned) Was the cultural alignment between the teams considered?
	(If yes) In which way have the cultural alignment been verified?
	Were there different levels of importance for the task allocation criteria adopted?

However, this relationship is not always considered in a project as reported by another respondent.

"The most important artifacts are requirements list, technical especification of requirements, architecture design, and test plan. The importance of an artifact does not influence in its distribution because the artifacts are equaly distributed to obtain feedback from all the teams." [PM1]

The first quotation suggests the validation of the relationship "**Artifact has Artifact Source**" that is related to the "make-or-buy" decision and the relationship "**Artifact has Artifact Nature**" that is related to the strategy of the organization or its interest on develop itself the artifact. According to one of the respondents an artifact has a complexity associated to it:

"There were artifacts that were more complex than others. It required more experienced resources to work on it." [PM2]

TEAMS AND ORGANIZATIONS: All the respondents confirmed that an organization has an organization role while participating in a project. The following quotation ilustrates it:

"I have examples of projects of testing and projects of development. [...] Projects that were developed not only by a single organization but jointly with vendors who provided part of the hardware components." [PM4]

The dispersion of the organization units were also confirmed by the respondents. We could confirm the relationships "**The organization unit is located in time zone**" and "**The organization unit is located in city**. The following quotation ilustrates these relationships:

"The dispersion among the teams was among different countries and time zones. Usually, our projects involve two different countries with different time zones." [PM1]

We could also identify the relevance of the relationship "**The team has team work period**" as indicated by one of the managers:

"We worked jointly with Europe, China, and Taiwan. In Europe the time zone difference was of 5 or 6 hours. In China and Taiwan was of 10 or 11 hours. When we needed to talk to someone from another country we had to stay in the office after the end of the workday." [PM4]

The cultural diversity and common experience among the teams are cited by the subjects suggesting the confirmation of the concept "**Cultural alignments level**" that is related to the relationship "**Team interacts with team**" and the concept "**Historical work**" related to the relationship "**Team works with team**". However we noted that these concepts are not always considered due to lack of choice as showed in the following quotations:

"The fact that some teams have already worked together was not considered in the allocation of tasks because the resources were scarce and the priority was the optimization. [...] the cultural aligment among teams was not considered by the same reason." [PM2]

HUMAN RESOURCES AND ACTIVITIES: We could identify the importance of the following relationship related to the human resources concept "**The human resource has fluency level on language**" as ilustrated below:

"[...] some of the human resources speak English in a comprehensible way which facilitates the communication. However there were some others with more difficulties speaking English even within the same organization." [PM4]

Furthermore, we identified that competences are considered in task allocation as the relationships **"The activity requires competence"** and "**The person accumulate competence**" describe as illustrated by the following quotations:

"[...] in most cases you have no choice, you have to work with that partner because he has the resources that have the competence needed or offer the service that we need." [PM4]

"The task allocation is done according the technical competences required."[PM3]

The concept "**Performance history**" related to the relationship "**The activity is developed by human resource**" could also be identified as the following quotation ilustrates:

"[...] as the partners will acquire more expertise, practice and performance, the headoffice will assign them the most important product modules." [PM4]

This preliminary evaluation suggests that the relationships and concepts defined in the proposed ontology to represent the dispersion of the teams are valid to real distributed projects. Some concepts related to cost were not cited by the respondents. We could not confirm the relationship "**The activity has activity constraint**" related to restricitons of an activity and "**The team is associated to team cost**" related to the cost of a team. These results could be an indication that the task allocation is actually concerned more in quality of the product instead of the cost reduction. However other assumptions must be verified since the evaluation with the liminted number of experts were based on specific cases and thus our results cannot be generalized yet.

VI. IMPLICATIONS

Our proposed ontology is useful for providing a commom understanding about what concepts are related to task allocation to teams in distributed projects. Although there are some proposals in literature to support this process, there is no consensus on what has to be considered in task allocation planning. We aimed at representing aspects that can be considered in task allocation to distributed teams as geographical dispersion, cultural differences, cost difference, characteristics related to the collaboration among teams and among organizations involved in a distributed project. Practitioners can benefit from this ontology by analyzing the relationships between the concepts and what they describe as presented in this paper. Researchers can direct efforts to analyzing for which distribution development scenarios the existing proposals are viable and to develop new proposals to scenarios that have not been addressed yet in literature.

Although the ontology aims at showing all the aspects that should be considered in the task allocation of teams our preliminary evaluation suggests that some concepts might not be considered in some scenarios. However, all the proposed concepts were considered relavant to some extent.

VII. Concluding Remarks

Task allocation to teams in distributed projects defines the sequence in which the tasks will be executed, how and for which teams the tasks will be assigned. The teams can be inserted into a global context, by adding different time zones or cultures, among other characteristics that define distribution.

In this sense, the task allocation to teams that participate in a distributed project must be planned based on the factors that can influence the decision-making process. Many proposals have been developed within this context. However, there is no consensus regarding the factors that must be considered, and the problem of an efficient decision-making tool to promote the benefit of distributed software development remains opened.

In this paper we propose an ontology that defines the concepts related to tasks allocation in distributed teams and the relationships between such concepts. The proposed ontology provides a reference (a) for a better understanding of the existing concepts related to task allocation to teams in distributed projects, and (b) to suggest adequate proposals to identify and analyze the various alternatives to allocate tasks to distributed teams. Many tools have been developed based on ontologies like the ODEs (Ontology-based Software Development Environment).

The results of a preliminary evaluation based on the opinion of experienced experts in distributed projects indicate that the relationships represented by the proposed ontology are valid in real distributed projects. These results suggest that project managers can use the ontology to guide their decision making about what consider in task allocation planning. This is possible because the ontology provides flexibility to represent diverse distribution development scenarios. Likewise, researchers can benefit from this ontology by analyzing which distribution development scenarios are addressed by existing proposals and guiding efforts to develop new proposals for not yet addressed distribution development scenarios.

As future work, we aim to conducting more interviews in order to verify whether the cost concepts are relevant and should remain as part of the ontology. We also aim to elaborate a guideline to support managers in using the proposed ontology. This guideline could help the managers to identify what aspects must be considered during the process of task allocation to distributed teams. Also, we aim to establishing a mapping to an ontology of optimization to facilitate the identification of optimization techniques that can be used to solve problems of distribution development scenarios.

Acknowledgments

This work has been partially supported by the FTS-Brasil Project CNPq (483125/2010-5) and by CAPES. We thank the PDTI Program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91) and FAPEAM for the financial support. We also kindly thank the experts for their collaboration and time.

References

[1] Chandrasekaran, B., Josephson, J.R., R., B.V. (1999), "What Are Ontologies, and Why Do We Need Them?" IEEE Intelligent Systems & their applications, v. 14, n. 1 pp. 20-26.

[2] Falbo, R. A. (2004) "Experiences in Using a Method for Building Domain Ontologies", 4th International Conference on Quality Software, IEEE Computer Society, Braunschweig, Germany, pp. 162-169.

[3] Falbo, R.A., Bertollo, G., (2005) "Establishing a Common Vocabulary for Software Organizations Understand Software Processes", International Workshop on Vocabularies, Ontologies and Rules for The Enterprise, Enschede, The Netherlands.

[4] Huzita, E., Silva, S., Wiese, I., Tait, T., Quináia, M., e Schiavoni, F. (2008) "A solution set to support Distributed Software Development", II WDDS, 101-110. (In Portuguese)

[5] Jalote P. e Jain, G. (2006) "Assigning tasks in a 24-h software development model," J. of Syst. and Softw., vol. 79, 2006, pp. 904-911.

[6] Kitchenham, B. Charters, S. (2007) "Guidelines for performing systematic literature reviews in software engineering", v. 2.3, EBSE Technical Report EBSE-2007-01.

[7] Lamersdorf, A., Münch, J. (2010) "A multi-criteria distribution model for global software development projects," Journal of the Brazilian Computer Society, pp.1-19.

[8] Lamersdorf, A., Münch J., Rombach, D. (2009) "A Survey on the State of the Practice in Distributed Software Development: Criteria for Task Allocation,", IEEE Int'l Conf. on Global Software Eng., pp.41-50.

[9] Mak, D.K.M., Kruchten, P.B. (2007) "Task coordination in an agile distributed software development environment," Canadian Conf. on Electrical and Computer Engineering, Ottawa, ON, Canada, pp.606-611.

[10] Marques, A. B., Rodrigues, R. F., Prikladnicki, R., & Conte, T. (2011). "Task alloction in distributed software projects: Analyze of the existing proposals". II Brazilian Congress of Software: Theory and Practice, V Distributed Software Development Workshop, pp.1-8 (In Portuguese).

[11] Pereira, T.A.B., Dos santos, V.S., Ribeiro, B.L., Elias, G. (2010) "A recommendation framework for allocating global software teams in software product line projects," Int'l. Workshop on Recommendation Systems for Software Eng., New York, USA: ACM, 2010, pp. 36-40.

[12] Prikladnicki, R., Marczak, S., Conte, T., De Souza, C., Audy, J., Kroll, J., Marques, A.B. e Orsoletta, R. (2011) "The Evolution and Impact on research in Distributed Software Development in Brazil", XXV Brazilian Symposium on Software Eng., special track: SBES 25 years.

[13] Project Management Institute (2004). A Guide To The Project Management Body Of Knowledge (4th ed.).

[14] Setamanit, S.-On, Wakeland, W., Raffo, D. (2007) "Using simulation to evaluate global software development task allocation strategies: Research Sections," Software Process, vol. 12, 2007, pp.491-503.

[15] Santos, G.; Montoni, M. A.; Rocha, A. R. C.; Figueiredo, S.; Mafra, S.; Albuquerque, A; Paret, D.B.; Amaral, D.B. "Knowledge Management in a Software Development Environment to Support Software Process Deployment". Lect. Notes in Artificial Intelligence, v. 3782, pp.24-36.

[16] StarUML. "The Open Source UML/MDA Platform" available in staruml.sourceforge.net.

[17] Villela, K. ; Santos, G. ; Schnaider, L. ; Rocha, A. R. C. da ; Travassos, G. H. . The Use of an Enterprise Ontology to Support Knowledge Management in Software Development Environments. Journal of the Brazilian Computer Society, v. 11, n.2, pp.45-59.