Collaboration ontology : applying collaboration knowledge to a generic group support system

Knoll, Stefan; Plumbaum, T; Hoffman, J.F.; de Luca, EW

Publication date
2010

Document Version
Final published version

Published in
Proceedings of the GDN 2010

Citation (APA)
Knoll, S., Plumbaum, T., Hoffman, J. F., & de Luca, EW. (2010). Collaboration ontology : applying collaboration knowledge to a generic group support system. In G. J. De Vreede (Ed.), *Proceedings of the GDN 2010* (pp. 1-37). Omaha : Center for Collaboration Science.

Collaboration Ontology: Applying Collaboration Knowledge to a Generic Group Support System

Stefan Werner Knoll[1], Till Plumbaum[2], Jan Leif Hoffmann[3], and Ernesto William De Luca[2]

[1]Department of Simulation and Graphics, Faculty of Computer Science, University of Magdeburg, Germany
[2]DAI-Labor, Technical University of Berlin, Germany
[3]Department of Distributed Systems, Faculty of Computer Science, University of Magdeburg, Germany

Abstract. Collaboration is a social and interactive process, where participants join efforts toward a group goal. A Group Support Systems (GSS) can improve the productivity of collaboration work by structuring activities and improving communication. However, knowledge is needed for the faithful appropriation of GSS technology.
In this paper we present ongoing research about a supporting framework for a generic GSS that adapts the GSS technology automatically to a logical model of a collaboration process. Drawing on Collaboration Engineering and the logical model, the paper explores the use of an ontology to capture and share knowledge about collaboration.
As a result, we present a collaboration ontology that builds a common vocabulary for the key concepts of collaboration and the relations and dependencies between them. A generic GSS can use the ontology to enable knowledge exchange between experts and practitioners. This will increase the potential benefits of a GSS for collaboration.

Keywords: Collaboration, Ontology, GSS, ThinkLet, ThinXel

1 Introduction

More and more organizations are faced with the need to innovate. This is due to a number of factors, including globalization and liberalization of markets, geographical development and an ever growing number of new technologies. To be innovative, organizations use collaboration to combine the potential and expertise of their employees.

Terveen (1995) defines collaboration as the process of a group where participants work together to achieve a shared goal. Research has shown that collaboration is affected by the characteristics of the group, the task, the context, and the technology used (Dennis et al., 1988; Nunamaker Jr. et al., 1991). The resulting group behaviors can lead to different effects (like social loafing, production blocking or synergy effects), which influence the efficiency of the collaboration and with it the innovation process. To recognize and manage these effects, organizations need experience in design and execution of collaboration processes.

The advent of the World Wide Web has led to the development of different web-based applications that changed collaboration by creating a common ground for interaction among several cultures for global teams. An example of this technical support for collaboration is a Group Support System (GSS), a meeting environment based on information technology. A GSS offers a variety of local and web-based tools that link a group via computers and assists them in structuring activities and improving communication (Nunamaker Jr. et al., 1991; de Vreede et al., 2003). The provided applications can be adapted in different ways to implement a collaboration process. DeSanctis and Poole (1994) argue that the faithful appropriation of the intended process is fundamental. Briggs et al. (2003) point out that the adaption of a GSS to a designed collaboration process can lead to a high conceptual load, i.e. practitioners find it difficult to understand how GSS technology can be used to support a collaboration process. As a result knowledge about the collaboration process and the used GSS technology is needed to adapt GSS tools appropriately and advantageously.

In this paper we explore the use of an ontology in collaboration process design, to capture and share knowledge about collaboration. We believe that this knowledge can be used to obtain a deeper and better understanding of collaboration. It can be used to develop new GSS functionalities that could reduce the experience needed for design and execution of collaboration processes (Knoll et al., 2009a).

2 State of the Art

Today, more than one-hundred web-based applications for collaboration exist that provide different functionalities and mechanism to execute collaboration activities (Mittlemann et al., 2008). Some of these applications ensure the faithful appropriation of the tool by providing limited functionalities for a specific collaboration activity. In contrast, by using generic applications like a GSS which provides a set of common functionalities that can be configured in a number of ways to implement complete collaboration work practice, experience is needed. To ensure faithful appropriation of GSS technology, organizations use professional facilitators who have expertise in design and execution of collaboration involving GSS. However, economic and political factors can prevent hiring external skilled facilitators and as a result organizations cannot benefit from collaboration knowledge of such a facilitator. In this case the efficiency of collaboration is not guaranteed. This situation leads to challenges in collaboration:

- How can a faithful appropriation of GSS technology be supported?
- How can collaboration knowledge be transferred to reduce needed experience for collaboration involving a GSS?

Briggs et al. (2003) assume that the experience needed for design and execution of collaboration can be reduced by packing and transferring knowledge about collaboration. They introduce Collaboration Engineering (CE) as an approach to design collaboration work practices for recurring high-value tasks that can be executed by practitioners without ongoing support from professional facilitators. To reach this goal, CE classifies collaboration into six key patterns of collaboration and introduces design patterns for best facilitation practice, called thinkLets (Briggs et al., 2003; 2006). ThinkLets form a pattern language for collaboration and are defined as named, scripted and reusable collaborative activity for creating a known pattern of collaboration among people working together toward a goal (Briggs et al., 2006). The original specification of a thinkLet based on the concept of design pattern introduced by Alexander (1979) and comprises the components (de Vreede et al., 2006):

- *Identification* – contains a name attribute, which is intended to emphasize the specific group dynamics the thinkLet invokes;
- *Script* – contains rules for a participant in a defined role for creating the required pattern of collaboration. These rules describe the actions a participant has to execute using the capabilities under some set of constraints;
- *Selection Guide* – contains different attributes such as patterns of collaboration to support the practitioner in the selection of a thinkLet.

A formal specification of a thinkLet as a technology-independent logical design element is given by the thinkLet class diagram (Kolfschoten et al., 2006). This specification uses the unified modelling language (UML) notation to illustrate the key concepts and relations of a thinkLet. An essential component is the concept Rule, whose instances define the script of a thinkLet. A Rule defines the Actions a Participant must do individually in a given Role, the Constraints under which he or she must act, and the Capabilities they will require to execute the Actions.

According to the given design approach for collaboration processes (Kolfschoten and de Vreede, 2009), a collaborative process will be designed as a sequence of the design pattern thinkLet. Each thinkLet will transfer knowledge about the used technology (e.g. GSS) and its configuration for a given task as well as facilitation skills that are needed to engender the pattern of collaboration. A designed collaboration process will be documented as a paper-based handbook. Research indicates that a practitioner who is trained in using thinkLets can repeatedly engender the patterns of collaboration for a designed collaboration process by following the description of the handbook (de Vreede and Briggs, 2005).

Our research analyzes the possibility to design a generic GSS for collaboration based on the existing theories and methods of CE. The goal is faithful appropriation of a generic GSS by adjusting the technology automatically to a designed collaboration process. Furthermore, the generic GSS should provide functionalities that help collaboration experts capturing and sharing their knowledge about collaboration. Also, practitioners with limited expertise should be able to use this knowledge for designing and selecting collaboration processes that fit to a given task or context. As a GSS aims to improve collaboration by a technology, we derived the following requirements for generic GSS technology:

- A generic GSS needs to support design, configuration and execution of a collaboration process.
- A generic GSS needs to cope with the effects that result from the characteristic of a collaboration process and influence the efficiency of collaboration (like social loafing, production blocking or synergy effect).

- A generic GSS needs to increase the usability of the technology by reducing the mental effort required to understand and work with the technology.

According to Briggs et al. (2003), we propose to use a pattern language for collaboration to package and exchange methods and best practice for collaboration.

2.1 A Prototype of a Generic GSS

In earlier work, we analyzed the applicability of the Collaboration Engineering approach to logical process descriptions similar to the concept workflow of Business Process Engineering (Hollingsworth, 1995). The resulting logical model for collaboration, called Group Process Modeling Language (GPML) (Knoll et al., 2008), illustrates process information and describes their influence on the collaboration process. The GPML adopts the design pattern thinkLet as a process template that creates one known collaboration pattern. ThinkLets can be combined to collaboration processes. These can be adapted to a group goal by the configuration of their parameters and activities. Like Kolfschoten et al. (2006) we divided a collaboration process into atomic activities of a participant, like *add*, *select* and *move*. In contrast to the fundamental concept Rule, we think that a rule needs to capture the used facilitator instruction and should lead to a defined action of a participant. As a result, we introduced a design approach for a reusable instruction element called thinXel, which represents an instance of a Rule. The concept thinXel is originally defined as an atomic facilitator instruction, leading to a response of the participants, that has a well-defined function in the context of the group goal (Knoll et al., 2007). By using the concepts thinkLet and thinXel, GPML can define personalized processes for the participants of a group. GPML distinguishes between an individual participant and a group of participants which allows us to illustrate concurrent processes of participants with different roles. We have used this property of GPML to design a method called Participant Flow Algorithm (Knoll et al., 2009b), which uses the logical design to compute the active activity and the next step of the participants.

Our first application of GPML and the Participant Flow Algorithm is a web-based GSS prototype that links a group via the Internet and implements the activities of a collaboration process via a website (Knoll et al., 2009b).

Currently, the prototype provides no support to secure the efficient combination of thinkLets as well as no functionalities for the design of guidelines for the configuration of the logical model by other practitioner. Equally important, the prototype provides no functionality to capture and share knowledge that results in context with the execution of a collaborative process. We think that this knowledge is necessary to obtain a deeper and better understanding of collaboration. Further the knowledge can be used to develop new design guidelines for the efficient design and execution of collaboration (Knoll et al., 2009a). For this reason, we analyzed the applicability of an ontology to capture knowledge about collaboration. We think that the use of an ontology in collaboration process design would strongly extend the possibility to capture and share knowledge about collaboration. Thus, in the next sections we give a brief introduction to the concept of ontologies and how we intend to uses them in our GSS.

3 Why an Ontology?

By definition, an ontology is an explicit formal specification of the terms and relations between them in a domain of interest (Gruber, 1993). They define a common vocabulary and a common understanding of information in a domain. This allows sharing information between people and software agents.

To reach the long-term goal of this work, enabling GSSs to serve as libraries of reusable knowledge that can be triggered by other applications, an ontology is the perfect basis. By building a common vocabulary for collaboration tasks and defining relations and dependencies between them, we enable information exchange between agents. Those agents, sharing the ontology, do not need to have the same knowledge base; each agent knows facts the others do not know. This is in our opinion a great strength of an ontology approach for collaboration systems; each agent can query other agents for collaboration information and therefore enhance his collaboration process.

The difficulties with the ontology approach lies in the design phase of such an ontology. The creation of an ontology has to take into account different perspectives to create an ontology that represents the domain. Therefore, different methodologies are developed and evaluated to allow an objective ontology creation (Cistani, 2005).

The next section gives an overview of existing work in the area of collaboration ontologies. Afterwards, we introduce the used methodology and our collaboration ontology.

4 Existing Collaboration Ontologies

In this section we discuss the use of given ontologies to capture and share knowledge about collaboration. We focus on the possibility to model the given methods and concepts of CE and the GPML with the concepts of an ontology.

To capture knowledge about collaboration, different ontologies have been developed. Oliveira et al. (2007) present a domain ontology for collaboration in the context of collaborative web browsing. They divide the collaboration ontology into the sub-ontologies cooperation, communication and coordination.

Regarding CE, the ontology could model a design pattern by the concepts Participant, Participation, CollaborativeAction and Protocol. Yet, it lacks a concept to identify a collaboration pattern. In contrast to the GPML, the concept Participation only distinguished between atomic events with or without the exchange of a message. The ontology provides no concept to define conditions for the participant flow, which are needed to define sequences of intended activities in relation to the resulting behaviors of the participants. Also, the concept Protocol does not enforce the definition of a relation between a rule and an activity of the participant. However, we think that we need to define the relation between a single rule and activity to gather new knowledge about collaboration.

Another approach for an ontology-based process definition is given by Rajsiri et al. (2008). They define a collaboration network ontology that is composed of a collaboration ontology and a collaboration process ontology. The collaboration ontology regards the characterization of collaborative network, details and abstract services of participants. The collaborative process ontology defines the task of the participants at a functional level, which has input and output resources. The ontology could model a design pattern by the concepts: Participant, Role, Abstract Service, Business Service, Resource, Common Goal and Dependency B/W Services of Participants. The latter only defines conditions for a message flow between the activities of participants. As a result the ontology does not model a participant flow that can be adapted in relation to the resulting behaviors or generated resources. However, we think that the approach to look at collaboration from different points of view is an interesting approach to gather new knowledge about collaboration.

In summary, most of the given collaboration ontologies represent a special domain of collaboration. As a result, no common collaboration ontology exists that can be used to model process descriptions for different kinds of collaboration. Therefore, we developed a new collaboration ontology that bases on the design pattern thinkLet and GPML.

5 Research Method

The objective in developing an ontology is to share a common understanding of the structure of information among people or software agents (Gruber, 1993). Today, several methods and methodologies for developing ontologies exist (Corcho et al., 2003). Uschold and Gruninger (1996) present a skeletal methodology for ontology engineering. We adopted it with different methods and technologies for ontology building (Gruninger and Fox, 1995; Pinto and Martins, 2004). The research approach considers the following stages:

1. *Ontology purpose and scope* – We conducted a literature research on collaboration and a case scenario of a collaboration workshop. To determine the scope of the ontology, we defined a set of questions that our ontology should be able to answer, called competency questions.
2. *Ontology capture and formalization* – We explored and structured all potentially relevant terms and phrases in a brainstorming session and used the resulting groups to capture key concepts and relationships. A graphical representation was used to build a conceptual model. We analyzed the integration of existing ontologies to use previously established conceptualizations. The conceptual model was transformed into a formal model and coded.
3. *Ontology evaluation* – We evaluated the ontology in respect to the purpose and its intended use. In doing so, we used the competency questions to verify the ontology regarding its consistency and completeness.
4. *Ontology documentation* – We documented the concepts and relationships in a data dictionary, where each concept is describes by its name, description, cardinality, etc.

We used this approach to develop a collaboration ontology to capture knowledge about collaboration. In the next section we present our research results more in detail.

6 Collaboration Ontology

In this section we present our collaboration ontology. We do this accordingly to the skeletal methodology shown in the previous section. At the end of this section, one knows how we defined the purpose and scope of the ontology, captured and formalized it and how we did the evaluation. A detailed documentation can be found on the internet (collaborationontology, 2010).

6.1 Ontology Purpose and Scope

The purpose of an ontology depends on the kind of user. Hence, we determined potential user groups of a collaboration ontology. According to Collaboration Engineering (Kolfschoten and de Vreede, 2009), we identified three user groups that have different knowledge bases.

- The collaboration engineer: Specialist in designing collaboration processes. Knows about factors that affect the outcome of a collaboration process.
- The practitioner: A domain expert with knowledge about the needed resources, the client goal and the stakes of a collaboration process.
- The participant: Participates in a collaboration process and knows about necessary steps to reach the client goal.

Any of these user groups can use the ontology to share its individual knowledge and therefore provide knowledge that enhances the design, adaption and execution of collaboration.

Besides Collaboration Engineering, we identified further user groups.

- The professional facilitator: Specialist in designing and executing collaboration processes. Potential user for sharing knowledge via the ontology by modeling common facilitation techniques.
- The software engineer: Utilizes the ontology to share knowledge between different applications he or she develops.

To identify the scope of the ontology, we searched for competency questions, i.e. questions that are utilized to identify areas of knowledge the ontology should contain. We conducted a literature research on collaboration and group facilitation (VanGundy, 1988; Schumann, 2005; Briggs and de Vreede, 2009). Furthermore, we analyzed case scenarios of collaboration workshops of a consulting company (Zephram, 2010). These face-to-face workshops are booked from different organizations to generate new product ideas, improve customer value or find new business areas. Each process represents a combination of different collaboration techniques involving participants with different roles like facilitators, experts, freelancers or an organizational staff in the background that prepares the workshop and executes follow-up tasks.

We inspected the facilitator script (agenda, cue cards) and the activities of the organizational staff. Further, we participated as freelancers in some of these workshops.

We came up with a set of competency questions that our collaboration ontology should be able to answer. Our design approach for a collaboration ontology used the existing methods and concepts of CE and the GPML. Therefore we defined competency questions with particular regard to the design pattern thinkLet and thinXel. To model the workflow of a participant, we asked what the next collaboration pattern or atomic activity of a participant in the collaboration process is.

In contrast to the collaboration ontology by Oliveira et al. (2007), we looked at the participants of a collaboration process in more detail. We adopted the design pattern thinkLet that defines the actions of a participant in relation to its skills, a defined role and the capabilities (de Vreede et al., 2006). Each activity of a participant will be executed in a defined location and generate different kind of artifacts. To further define the actions of a participant we asked what kind of equipment or technology is needed. The case scenarios showed us that the activities of participants are also related to the elements money and time. Further, we focused on organizational elements like food and logistics.

Similar to Rajsiri et al. (2008), we looked at collaboration from different points of views. Therefore we asked for the objectives of a client, a collaboration process, a collaboration pattern (thinkLets) and an atomic activity (thinXels). To measure the achievement of these objectives, we asked how results are represented and how they would compare to previously defined objectives.

6.2 Ontology Capture, Formalization and Verification

Capturing of the collaboration ontology demands to identify abstract entities (key concepts), naming important properties and defining relationships between the entities. These entities are based on the given concepts of CE and GPML.

After the identification of key concepts we depicted the entities to create a graphical conceptual model. Like Rajsiri et al. (2008) we divided the key concepts into different ontologies: One named collaboration ontology (co) that describes the external point of view on collaboration of a client, and another one named collaboration process ontology (cpo) that contains the concepts belonging to the internal description of collaboration processes. The interface between these two is the concept “cpo:CollaborationProcess”, which is present in both ontologies.

Iteratively, we added unique relationship descriptors between entities and properties like names or descriptions. While creating, we verified it. We took the competency questions that we defined in the beginning and tried to answer them abstractly by use of the ontologies. We observed that in some cases we were not able to do so. In those cases we refined the ontologies. After this step, our created ontologies held all the knowledge we wanted them to hold. Each key concept, property and relationship is unambiguously described in a dictionary.

In the next section we will present the key concepts of the ontologies. For all details, please consult our website (collaborationontology, 2010). Please note that in the following subsections names of concepts are capitalized.

6.2.1 The Collaboration Ontology

The collaboration ontology (shown in Fig. 1) is abbreviated co. Its purpose is to describe collaboration from an external point of view. We define the following key concepts outlined in the conceptual dictionary in Table 1.

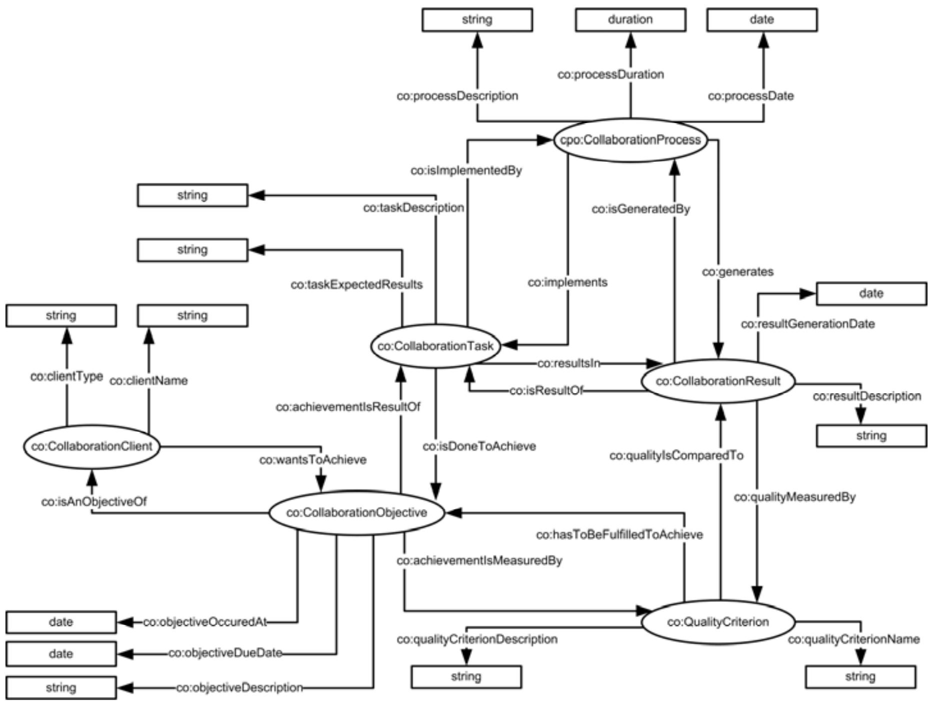

Fig. 1. Graphical Representation of the Collaboration Ontology (co)

Table 1. Conceptual Dictionary of the Collaboration Ontology (co)

Concept	Description
CollaborationClient	Denotes a person or a group of people that has the need for collaboration
CollaborationObjective	Denotes a goal set by the CollaborationClient to describe a desired result; motivation for the CP
CollaborationTask	Denotes a step to achieve the CollaborationObjective in a collaborative manner
CollaborationProcess	Denotes the part in which participants interact for the purpose of collaboration; the implementation of a given CollaborationTask
CollaborationResult	Denotes the actual outcome of a CP
QualityCriterion	Denotes a criterion that results from the CollaborationObjective and against which the CollaborationsResult is evaluated

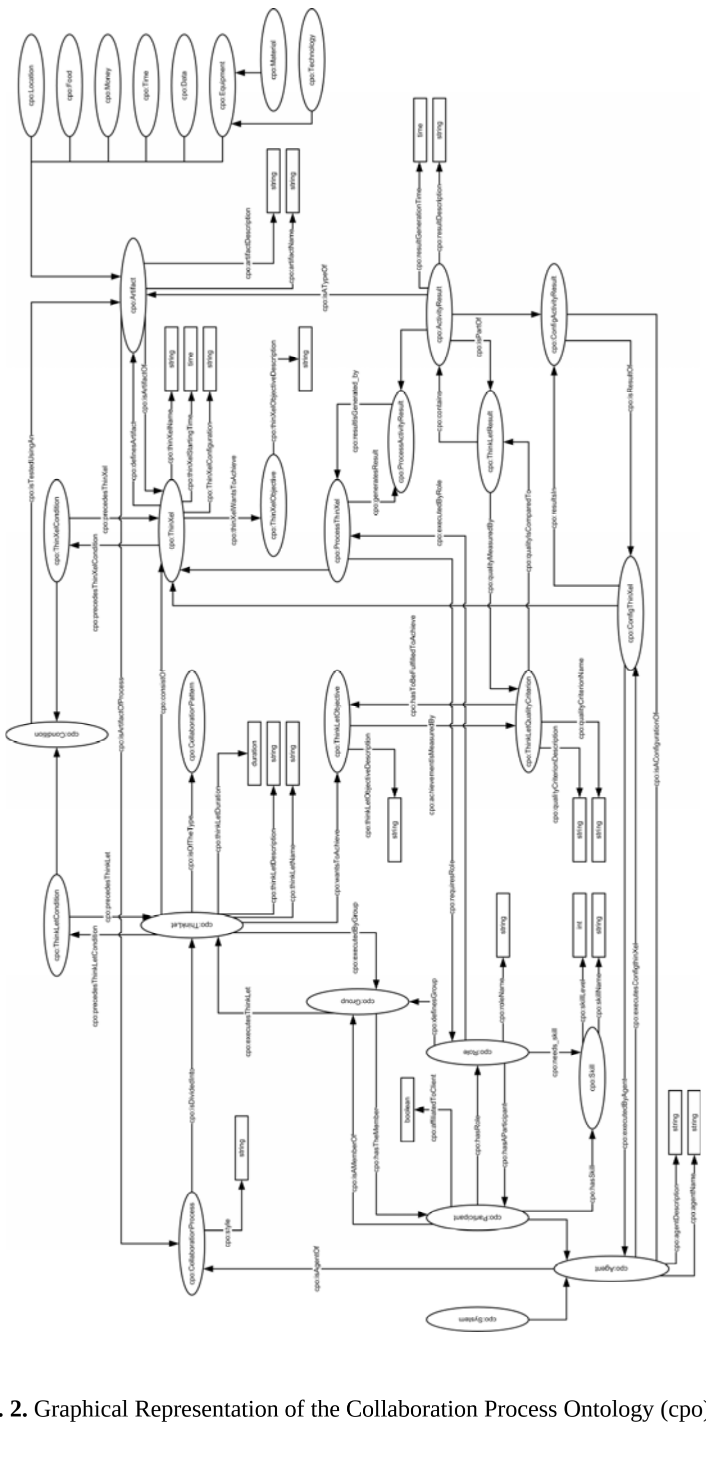

Fig. 2. Graphical Representation of the Collaboration Process Ontology (cpo)

The concept CollaborationClient denotes a possible client of a collaboration process, e.g. a company, an organization or a single person. A client has one or more objectives (CollaborationObjective) which should be achieved by usage of a collaboration process. According to the objectives, one or more tasks (CollaborationTasks) can be instantiated by the client or a collaboration engineer. The concept CollaborationProcess is a designed process description with the goal to implement such a task and achieve a defined objective. CollaborationProcess represents the collaboration process ontology (abbreviated cpo) which will be defined in the next subsection. The outcome of the collaboration process is stored in the concept CollaborationResult. To make a statement regarding the quality or fitness of the results we have defined the concept QualityCriteria, which can be measured against the CollaborationObjective.

6.2.2 The Collaboration Process Ontology

The second ontology is called collaboration process ontology (cpo) and describes the inner setup of a collaboration process (shown in Fig. 2). It comprises agents (persons or process-supporting machines like computers), structural information about the collaboration process in a pattern paradigm and artifacts that are produced or consumed by the process.

First, we describe the concepts concerning the concept Agent (Table 2). An Agent is either a System or a Participant. It is the entity that executes an activity that is related to the collaboration process. As a result of our case scenarios of collaboration workshop, a System is a placeholder for some entity doing background work, for example sorting lists of ideas. This can be done by a machine or a human being; the difference is not important to the collaboration process. A Participant, on the other hand, is a human being taking part in a collaboration process. This entity has certain Skills that can be a prerequisite of a Role in a process. Roles are defined by the process description and abstractly denote a set of behaviors, rights and obligations. Furthermore, Participants can be assigned to a Group for collaboration work.

Table 2. Conceptual Dictionary of the Collaboration Process Ontology (cpo), excerpt concerning Agents

Concept	Description
Agent	Denotes a person, system, software, etc.
System	Denotes an agent that is not a participant; a machine, e.g. a computer, software
Participant	Denotes a person that participates at a collaboration process
Skill	Denotes an ability of a certain level that is a requirement or property of a role or participant to fulfill a task
Role	Denotes a set of behaviors, rights and obligations
Group	Denotes some participants that in a collaboration process work together as a group

Second, we take a look at the collaboration process (Table 3 and Table 4). We use a pattern approach that is based on CE and the GPML and therefore define the key concepts ThinkLet and ThinXel. A ThinkLet denotes the concept of a scripted and reusable collaborative activity for creating a known pattern of collaboration (Briggs et al., 2006) by a Group. We implement the pattern of collaboration by the concept CollaborationPatterns. By using CollaborationPatterns, the six key patterns of collaboration can be described. Further it is possible to add new collaboration patterns, if the resulting knowledge indicates it. Based on our experience with collaboration workshops, we suggest adding a social collaboration pattern that can be used to introduce the participants to each other or to level social behaviors in a group. To measure the achievement of a collaboration pattern, we defined the concept ThinkLetObjective and ThinkLetQualityCriteria. A collaboration process can be divided into a sequence of different ThinkLets. To model a defined order of ThinkLets we implement a precondition called ThinkLetCondition which must be fulfilled in order that it is executed.

According to GPML, a thinkLet consists of different thinXels, which are defined as an atomic facilitator instruction leading to an activity of a participant (Knoll et al. 2009b). Furthermore, in the context of the ontology, the concept ThinXel is similar to the concept Rule (Kolfschoten et al., 2006) and defines the relation between an agent, an intended activity, and an intended result under use of certain artifacts. We use the concept ThinXelObjectives to represent the intended activity. The result of a ThinXel is denoted by the concept ActivityResult. The concept ActivityResult changes Artifacts. However, a ThinXel has no quality criteria because one does not check the quality of the result of an activity as part of a thinXel but as dedicated part of a thinkLet. Similar to the concepts ThinkLet, a ThinXel can have a precondition called ThinXelCondition.

Table 3. Conceptual Dictionary of the Collaboration Process Ontology (cpo), excerpt concerning the CollaborationProcess

Concept	Description
CollaborationProcess	Denotes the part in which participants interact for the purpose of collaboration; the implementation of a given CollaborationTask
ThinkLet	Denotes a design pattern for collaborative work, i.e. all relevant information to create a pattern of collaboration
CollaborationPattern	Denotes an abstract design pattern for collaboration
ThinkLetObjective	Denotes a goal set by the collaboration engineer to describe a desired result; motivation for the ThinkLet
ThinkLetQualityCriterion	Denotes a criterion that results from the ThinkLetObjective and against which the ThinkLetResult is evaluated
ThinkLetCondition	Denotes a condition of a ThinkLet that can be testet and results a logical value
ThinkLetResult	Denotes the actual outcome of a ThinkLet
Condition	Denotes a condition that can be tested and results a logical value

We distinguish the concept ThinXel into the subclasses ConfigThinXels and ProcessThinXel. A ConfigThinXel inherits from ThinXel and represents an activity with the intention to change the existing context. This change can be necessary to provide a required context for follow-up activities of a participant. For example, the activity to provide pen and paper is required to be done before the activity to write an idea on a sheet of paper can be executed. The result of a ConfigThinXels is defined by ConfigActivityResults which inherit from ActivityResult. A ConfigThinXel is executed by an agent, what means it can be executed by any participant or external person, but also, in some cases, by a computer or machine.

A ProcessThinXel inherits from ThinXel and represents an activity for a defined role with the intention to change the actual outcome of a thinkLet. It results in a ProcessActivityResult inheriting from ActivityResult. A ProcessThinXel requires a role. Because of this it can only be executed by a participant, ergo a human being.

Third, we describe the Artifacts and concepts inheriting from it (Table 5). Artifacts denote products consumed or produced by the process. They are specialized by the concepts Data, Time, Money, Location, Food and Equipment. Equipment for itself can be divided in Technology (like a video projector) and Material (like pens or paper). Artifacts describe most of the relevant context of ThinXels and therefore of ThinkLets and the whole collaboration process.

Table 4. Conceptual Dictionary of the Collaboration Process Ontology (cpo), excerpt concerning ThinXels

Concept	Description
ThinXel	Denotes a single step in a ThinkLet. It defines the relation between an agent, an instruction, an intended activity, and an intended result under use of certain artifacts
ThinXelObjective	Denotes a goal set by the collaboration engineer to describe a desired result; motivation for the ThinXel
ActivityResult	Denotes a result of a thinXel activity
ThinXelCondition	Denotes a condition of a ThinXel that can be tested and results a logical value
ConfigThinXel	Denotes a ThinXel that is specified for an agent that leads to a context change. Its intended use is to instruct the system or facilitator.
ConfigActivityResult	Denotes a ActivityResult of a ConfigThinXel; changes the setting of a context (agents, artifacts)
ProcessThinXel	Denotes a ThinXel that is specified for a participant with a predefined role and creates or changes artifacts in a specific context. It leads to a thinking step
ProcessActivityResult	Denotes the ActivityResult gained by a ProcessThinXel

Table 5. Conceptual Dictionary of the Collaboration Process Ontology (cpo), excerpt concerning Artifact

Concept	Description
Artifact	Denotes any kind of resource used in the collaboration process
Data	Denotes an artifact that describes a single piece of information, e.g. an idea
Time	Denotes an artifact that describes a time duration
Location	Denotes an artifact that describes where a collaboration process or parts of it take place
Equipment	Denotes an artifact that describes a single piece of equipment, i.e. material or technology
Material	Denotes a piece of equipment that describes the material used in the collaboration process, e.g. pens, paper, post-it etc.
Technology	Denotes a piece of equipment based on technology, e.g. a watch, video projector, etc.
Money	Denotes an artifact that describes the monetary demands of certain steps in the collaboration process
Food	Denotes an artifact that describes food, e.g. coffee or snacks

Currently we use OWL (the Web Ontology Language) to represent the collaboration ontology in a formal language. In our case, it is written in XML syntax and built using RDF Schemas, which base on a larger vocabulary and stronger syntax than RDF (Herman et al., 2004). It allows an exact description of information and relationships between them. We will use the formal language to explore the possibility to define and store given thinkLets as templates into a digital library of collaboration processes. The generic GSS could use this library to increase the benefit of the GSS for Collaboration.

7 Future Work

In future work, we will use the collaboration ontology to develop new functionalities for the generic GSS that will reduce the experience needed for design and execution of collaboration processes. In this context, we plan to use the presented ontology for information retrieval and machine learning approaches. Data and information collected with the generic GSS and the ontology constitute the basis for

- Learning recommendation for collaboration tasks,
- Retrieval of matching collaboration tasks, and
- Fuzzy Collaboration Task Matching.

Learning recommendations: Based on data collected from previous collaboration processes, we want to learn relations between thinkLet and thinXel and the needed resources and participants. Based on the learned relations we then want to recommend collaboration tasks fulfilling certain conditions.

Retrieval: Similar to the recommendations described above, the collected data are used as the basis for learning relations. But, in contrast to the first scenario, the goal is to support collaboration engineers.

Fuzzy Matching: The fuzziness of information has been also taken into account. Based on the ontology, we could introduce a fuzzy matching method that can help users in finding interdisciplinary collaboration processes.

We appeal to other researchers to take part in evaluating, using, and enhancing our proposed ontology.

8 References

Alexander, C. (1979), *The timeless way of building*. Oxford University Press, New York.

Briggs, R.O., de Vreede, G.-J. (2009), *ThinkLets: Building blocks for concerted collaboration*.

Briggs, R.O., Kolfschoten, G.L., de Vreede, G.-J., Dean, D.L. (2006), Defining key concepts for collaboration engineering. In: *12th Americas Conference on Information Systems*.

Briggs, R.O., de Vreede, G.-J., Nunamaker Jr., J.F. (2003), Collaboration engineering with thinkLets to pursue sustained success with group support systems. *Journal of Management Information Systems*, 19(4), 31-64.

Collaborationontology, http://collaborationontology.org/. (*last accessed on 2010-04-29*).

Corcho, O., Fernández-López, M., Gómez-Pérez, A. (2003), Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, 46(1), 41-64.

Cristani, M., Cuel, R. (2005), A Survey on Ontology Creation Methodologies, *Int. J. Semantic Web Inf. Syst.*, 1(2), 49-69.

Dennis, A.R., George, J.F., Jessup, L.M., Nunamaker Jr., J.F., Vogel, D.R. (1988), Information technology to support electronic meetings. *Journal of Management Information Systems*, 12(4), 591-624.

DeSanctis, G., Poole, M.S. (1994), Capturing the complexity in advanced technology use: Adaptive structuration theory. *Journal Organization Science*, 5(2), 121-147.

Gruber, T.R. (1993), A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199-220.

Gruninger, M., Fox, M.S. (1995), Methodology for the Design and Evaluation of Ontologies. In: *Workshop on Basic Ontological Issues in Knowledge Sharing.*

Herman, I., Swick, R., Brickley, D. (2004), Resource Description Framework (RDF). *Technical report*, W3C.

Hollingsworth, D. (1995), The Workflow Management Coalition Reference Model. *Technical Report WFMC-TC-1003*, Workflow Management Coalition.

Knoll S. W., De Luca E. W., Horton G., and Nürnberger A. (2009a) Integrating Semantic Web and Web 2.0 Technologies for supporting Collaboration Work, In: Proceedings of the 17. Workshop on Adaptivity and User Modeling in Interactive Systems.In conjunction with LWA 2009, GI joint workshop event "Learning, Knowledge and Adaptivity". Darmstadt, Germany, 2009.

Knoll, S.W., Hoerning, M., Horton, G. (2009b), Applying a thinkLet- and thinXel-based group process modeling language: A prototype of a universal group support system. In: *42nd Hawaii International Conference on System Sciences*. IEEE Computer Society Press, Los Alamitos.

Knoll, S.W., Hoerning, M., Horton, G. (2008), A design approach for a universal group support system using thinkLets and thinXels. In: *Group Decision and Negotiation*.

Knoll, S.W., Chelvier, R., Horton, G. (2007), Formalized online creativity using thinXels. In: *10th European Conference on Creativity and Innovation*.

Kolfschoten, G.L., de Vreede, G.-J. (2009), A design approach for collaboration processes: A multimethod design science study in collaboration engineering. *Journal of Management Information Systems*, 26(1), 225-256.

Kolfschoten, G.L., Briggs, R.O., de Vreede, G.-J., Jacobs, P.H.M., Appelman, J.H. (2006), A conceptual foundation of the thinkLet concept for Collaboration Engineering. *International Journal of Human-Computer Studies*, 64, 611-621.

Mittlemann, D.D., Briggs, R.O., Murphy, J., Davis, A. (2008), Toward a taxonomy of groupware technologies. In: *Groupware: Design, Implementation and Use*, pp. 305-317, Springer-Verlag Berlin / Heidelberg.

Nunamaker Jr., J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R., George, J.F. (1991), Electronic meeting systems to support group work. *Communications of the ACM*, 34(7), pp. 40-61.

Oliveira, F.F., Antunes, J.C.P., & Guizzardi, R.S.S. (2007), Towards a collaboration Ontology. In: *2nd Workshop on Ontologies and Metamodels in Software and Data Engineering.*

Pinto, H.S., Martins, J.P. (2004), Ontologies: How can they be built? *Knowledge and Information Systems*, 6(4), pp. 441-464.

Rajsiri, V., Vatcharaphun, J.-P., Bénaben, F., Pingaud, H. (2008), Collaborative process definition using an ontology-based approach. In: *Pervasive Collaborative Networks*, pp. 205-212, Springer, Boston.

Schumann, S. (Ed.). (2005), *The IAF handbook of group facilitation: Best practices from the leading organization in facilitation*. Jossey-Bass, San Francisco.

Terveen, L.G. (1995), Overview of human-computer collaboration. *Knowledge-Based Systems*, 8(2-3), pp. 67-81.

Uschold, M., Gruninger, M. (1996), Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2).

VanGundy, A.B. (1988), *Techniques of Structured Problem Solving*. Van Nostrand Reinhold Company Inc., New York.

de Vreede, G.-J., Kolfschoten, G.L., Briggs, R.O. (2006), ThinkLets: A collaboration engineering pattern language. International Journal of Computer Applications in Technology 25(2-3), pp. 140-154.

de Vreede, G.-J., Briggs, R.O. (2005), Collaboration engineering: Designing repeatable processes for high-value collaborative tasks. In: *38th Hawaii International Conference on System Sciences*. IEEE Computer Society Press.

de Vreede, G.-J., Vogel, D.R., Kolfschoten, G.L., Wien, J.S. (2003), Fifteen years of GSS in the field: A comparison across time and national boundaries. In: *36th Hawaii International Conference on System Sciences*. IEEE Computer Society Press.

Zephram, http://www.zephram.de/?lang=en. (*last accessed on 2010-02-18*).