Version-Based Description of Cooperation Aspects: Motivations and Objectives

Lahouaria Bendoukha
Hamburg University, Computer Science Department
Working Group for Software Development
Vogt-Koelln-Str. 30
22527 Hamburg, Germany
Bendoukh@informatik.uni-hamburg.de

Abstract

The purpose of our work is to improve the understanding of cooperative work in organizations. Three main services of a cooperative work system are identified: Collaboration: management of shared objects, Coordination: ordering of tasks within the process, and Communication: exchange of information between users. We aim at first to bring out the implicit cooperation support in complex object versioning. For this purpose we have made explicit some common properties to both research areas: cooperation and version modeling. Based on those properties, we will show that the description of entity evolution by means of versioning, is a natural means to describe the application domain knowledge, particularly the cooperative work aspects. The basic idea consists in capturing the cooperation aspects by keeping track of entity dependencies, and by controlling object change propagation through these dependencies. We will present the research question and the objectives of our doctoral work as well as the approach we are following. An overview of some possible benefits of our approach deduced from the actual state of our work is also given.

Keywords: cooperative work, groupware, entity dependency, entity versioning

1. Collaboration aspect as a key for understanding the cooperation

A cooperative work allows several users to cooperate around a common task through a shared space. The most important fields considered in the literature are: the multidisciplinary field computer supported cooperative work (CSCW), business processes and information technology. Other fields are also investigated in the literature: information systems, databases, tool integration and configuration management.

Technology mediated workgroups systems are sometimes called CSCW systems or groupware. According to the three complementary models of Ellis [1] from the point of view of the user (or users), we identify three services of a groupware covering those models: *Collaboration*: management of shared objects, *Coordination*: ordering of activities in the process, and *Communication*: exchange of information between users.

Unfortunately, these services have received each one different interest and degree of importance according to the system area. For instance, Workflow systems emphasizes on coordination aspect, multimedia on communication aspect. This leads to the fact, that there is no unifying model allowing the support of the three dimensions for a given application domain. Moreover, collaboration (object sharing) has received less attention.

The support of object evolution and the eventual resulting change propagation is to our point of view an important issue in the support of the collaboration aspect. On the other hand, we believe that starting from the collaboration aspect, if we master the support of object evolution, we will then succeed to describe easily also the other cooperation aspects crossing the coordination and the communication. As an application area, we consider during our doctoral work the university teaching-planning problem.

2. Bringing out implicit cooperation support in version modeling

The reflection on the objectives mentioned above led us naturally to think about version concept. The advantages of the concept of version have already been illustrated in a huge amount of research papers in different research areas. We focus on the particular interest of versioning for cooperation support. We mean by implicit cooperation the fact that human activity is not explicitly understood. In order to make explicit the correspondences between the both areas: cooperation and version modeling, we try to classify some common properties to them.

- *Both version and cooperation modeling are application- or work-oriented*: it is impossible to formalize a version entity evolution. The task of determining what constitutes a version of an entity and the semantics for creating and retrieving versions is always left to the user.
- *The time and the space* are key factors for both areas: in version modeling, this is materialized by the well-known different type of version evolution: revisions, alternatives, equivalencies, variants and merging versions. On the other hand, the usual topology dimensions of groupware systems are also based on the temporality (synchronous or asynchronous) and the localization of the participants (at the same or different place).
- *Version types deal with the evolution of all key concepts in cooperative work*: versions are used to model evolution of different type of entities from objects to processes, and from simple to complex entities. Version modeling community suffer unfortunately of several difficulties caused by this diversity. We believe that this diversity is precisely needed for the support of cooperative work. Those unrelated purposes are

all found to be the necessary elements in cooperative work. Indeed, If we consider the key concepts in a conceptual model for the development of CSCW system[2], such as objets, tools, subjects, community, rules, work processes, constructions, we realize that a such description of cooperative work is a good area for concretion of the diverse version modeling.

- *Implicit cooperation support in complex object versioning:* in database community, the problem of maintaining the consistency of evolving complex entities is well overcame. These applications are supposed implicitly to be in interaction with several types of users. Some techniques are proposed to control change propagation between related entities. They are concerned, by the way, for instance, in which entity versions which are linked to a given entity are propagated. We think that these techniques, when adapted to collaboration requirements, could be of a great interest for cooperative applications.

Thus, we think that in version modeling, the notion of cooperating persons by sharing object is evident. It is confused with the notion even of object versions.

3. Limitations of the actual implicit support of the cooperation in version modeling

The conventional techniques in databases bring out an implicit support of the cooperation but they are concentrated on systems instead of work . The problem of maintaining the consistency of evolving complex entities is well treated, nevertheless, the main concern in the proposed models is the preservation of database consistency but from a database point of view. Thus, users and tasks are not considered.

This is why we think that additional concepts dealing with persons and tasks would be helpful. The integration of such results in cooperative work support is an open issue.

4. Towards the Cooperative-view concept

If we consider the entities to be versioned, which are not only objects but also tasks and persons, we can describe different types of entity-dependencies (object-dependencies, task-dependencies and person-dependencies). According to our objectives, Object, person and task combinations identify our proposed generic concept which we call cooperative-view. It allows the identification of fine-grained shared contexts in a cooperative work. Each entity in the cooperative-view could be simple or complex. For instance, a complex object may participate in several tasks and a task could act on several objects. In our work, all the entities could be "versionable". *A* person (or a team) may lose or earn new roles. These different status are described by its versions. Concerning task *versions*, we are interested particularly in the most pertinent status of the degree to fill up a task, which are: "activated", "in work" and "finished". Finally, different types of object versions are also considered. Cooperative-views are able to evolve, each time at least one entity evolves. We speak then about cooperative-view versions.

5. Possible benefits

In order to have an idea about the possible benefits of our work, we propose to consider the cooperation support in the context of software development processes. Since STEPS (Software Technology for Evolutionary Participative Systems development) and Tools & Materials approaches require the understanding of the organizational constraints and working practices within projects [3], we want to investigate the possible integration of our results into the proposed techniques based on both approaches such as cooperation pictures techniques [4]. If we take into account our objectives dealing with entity dependencies**,** several forms of cooperation become invisible by means of such techniques. Besides of "what" and "how", entity dependencies improve the understanding about the "why".

6. Conclusion

The need of entity versioning has been amplified with the emergence of areas such as cooperation support. From the user point of view, the capacities of versions to support diverse forms of evolution, make them rich in semantic in the sense, that also when they describe the object evolution, the information about its creation such as, “why”, “when”, “whom”, etc. may give more understanding about the cooperation aspects. Moreover, versions are good if we do not want to automate, but to answer to questions pertinent for cooperation support such as what-if ? And why ? For all these reasons, we think that version modeling is a significant part of the more general cooperation modeling topic which must be taken into account. We think that introducing cooperation requirements into version modeling will bridge the gap existing between both research areas: cooperation support and version modeling, which have so far brought several common problems to researchers in these two areas.

Acknowledgement

The author would like to thank Prof. Dr Christiane Floyd and Dr. Ingrid Wetzel for their constructive comments, their suggestions and their support through many discussions.

7. References

[1] C. (Skip) Ellis, J. Wainer, “A conceptual Model of Groupware”, ACM CSCW 94 Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina, USA, 1994, pp. 79-88.

[2] Kari Kuutti, “Debates in IS and CSCW Research: Anticipating System Design for Post-Fordist Work”, in Information Technology and Changes in Organisational Work, W. Orlikowski, et al., Editors. 1996, Chapman & Hall: London, p. 287-308.

[3] C. Floyd, C. Züllighoven, H. Budde, R., & Keil-Slawik, R.(Eds.), Software Development and Reality Construction, Springer Verlag, Berlin, 1992.

[4] I. Wetzel, “Information Systems Development with Anticipation of Change: Focussing on Professional Bureaucracies”, Proc. Of Hawaii International Conference on System Sciences, HICCS-34, IEEE Computer Society, Maui, Januray 2001.