Forming a Collaboration by Using Coordination Design

Hilda Tellioğlu
Multidisciplinary Design Group
Institute of Design and Assessment of Technology
Faculty of Informatics, Vienna University of Technology
Vienna, AUSTRIA, 1040
hilda.tellioglu@tuwien.ac.at

Abstract

This paper is about presenting the integration of Coordination Design (CoorD) as an approach to design coordination processes [11] into the formation phase of Collaboration Life Cycle (CLC) [13]. CLC defines several phases in collaborative processes and details its execution with different steps. This paper shows how CoorD enhances CLC to cover coordination issues. The new release of CLC is presented and its benefits and use are illustrated by an experiment.

1. Introduction

Collaboration is in general a structured process involving at least two people work together towards a common goal. If people do not manage to carry out an activity alone, they need others collaborating with them to achieve their goals. This causes automatically interdependencies between collaborating actors. If there is no interdependency, there is no collaboration [4].

In the field of CSCW, several work environments and communities of practice are studied to understand the interactions between cooperating actors, their routine and situated (work) practices, individual and common artifacts used to deal with dependencies [12], responsibilities, distribution of work, group and workplace awareness, representation of work [8], formal constructs [6], ordering systems [7], coordinative artifacts, etc. Besides cooperative work arrangements, cultural, organizational, technical and economic factors are studied, with more or less attention on coordination management. In the last years, very little has been done in studying coordination settings for work groups. Especially, no explicit attention has been given to mechanisms and methods for setting up coordination environments, which are not general and depend on real work settings, type of work, group structure and distributedness in time and space. If a new project is going to be started, the most time it is not clear to anyone how to arrange the cooperation and coordination from the beginning. Except deciding what tool is most suitable for the group or assessing the impacts of the use of a collaboration tool in a work group, the research does offer almost nothing about approaches and experiences with methodologies to design and implement a collaborative (work) environment [9].

This paper is based on authors' two previous research results: Collaboration Life Cycle (CLC) [13] and Coordination Design (CoorD) [11]. CLC has been developed for manufacturing [2] [1]. In the European research project MAPPER (FP6-IST-016527-MAPPER – Model-based Adaptive Product and Process Engineering), CLC is evaluated at two industrial sites as a framework to plan and manage collaborative design practices [5]. The evaluation was carried out by using paper-based artifacts like checklists, descriptions of processes, and spreadsheets for planning and coordination. Unfortunately, there was no time to integrate CLC in model-based active knowledge systems (created in MAPPER), which would deliver informed data about practicability, usefulness, efficiency, and flexibility of CLC in a distributed cooperative design project. It is necessary to find out whether CLC considers coordination issues in detail, and whether there are areas to improve regarding coordination support. On the other hand, CoorD, which is a sequence of methodologies to design coordination in a cooperation, was evaluated by using the prototype implemented based on it [10]. However, CoorD is not integrated and assessed in a bigger project, where the cooperation process spans from the initiation of a cooperation to the decomposition. It is not clear whether a coordination process of a large-scale design project would make use and furthermore benefit from CoorD.

Authors developed and studied CLC and CoorD in their previous research separately. It turned out that both concepts and approaches are useful for users. However, considering the complex structure of cooperative design work, there is a need to design a work environment, which on the one hand guides project members in organizing and maintaining their work procedure, and on the other hand, provides possibilities to make sure that the coordination established fits the organizational and task- and product-related nature of a project, is flexible enough to enable improvisations and changes during a project, and facilitates computer support for certain user-defined activities. This paper tries to fill in the gap. Its focus is the design of coordination within a CLC and its improvement by integration with CoorD to provide an evaluated approach to design and implement a collaborative work environment. It tries to answer the following questions: how a coordination-related context can be captured, how tasks and dependencies between tasks can be analyzed, how the coordination domain and rules needed to establish coordination in the work group can be created, and finally, how the created coordination rules in the coordinated work environment can be established. This paper presents the further development of the CLC. In particular, it details the process of setting up a coordinated work environment in the formation phase of the cycle. It then tries to apply the suggested design process in a real work environment to evaluate its soundness and completeness. The case is about the formation of coordinated services of deans of studies in the Faculty of Informatics at the Vienna University of Technology. The coordinated work environment used is the current web site of the faculty[1]. The results are discussed and issues for further development are addressed.

In the next section we introduce the framework developed for this research, i.e., the new version of the CLC and the CoorD. In section 3 the experiment is presented by showing the different models and artifacts created, and the steps of the CLC carried out. Results of the experiment and design issues derived are summarized in conclusions.

2. Formation of a coordinated work environment

Collaboration Life Cycle (CLC) is an approach for systemising a collaboration process in a coordinated work environment [13]. The recursiveness of a collaboration is captured in four steps: initiation, formation, operation, and decomposition. We refer in this paper mainly to the formation phase. Besides having a life cycle for collaboration, coordination needs to be designed considering the work setting. The approach Coordination Design (CoorD) [11] helps to overcome this. It looks for coordination patterns in work practices, identifies coordinative constructs, and tries to identify structures in activities carried out. It describes how to provide elements necessary to produce computer support for coordination work.

For this paper, the task "Setup a coordinated work environment" during forming a collaboration is most relevant and therefore will be detailed in this section. So, setting up a coordination consists of the following steps. This is a revised version of the original tasks.

(1) To inquire the collaboration context, create use cases, representing tasks and artifacts used, based on interviews with stakeholders. The outcome is a set of use case diagrams.

(2) To define the collaboration environment, set the focus of the coordination, choosing between task- or artifact-based. This decision is made depending on the use cases and the type of cooperative work, whether the tasks are predefined and well-structured or ad hoc and situated [3].

(3) Assign persons to tasks and negotiate task assignments with team members.

(4) Assign realistic deadlines to tasks and communicate these with team members assigned to.

(5) Create initial and final states of coordination-related coordinative artifacts by means of state diagrams.

(6) Identify dependencies between tasks including flow dependency, shared resources and fit dependency [4] by means of activity diagrams. The result is the model of the temporal and logical order of tasks.

(7) Model the domain by means of data dictionaries and domain models.

(8) Create rules as formal organizational constructs [6] to define the constraints and hidden dependencies between entities of the domain model in a natural language.

(9) Create coordination rules in pseudocode.

(10) Associate coordination rules with tasks.

(11) To specify the IT infrastructure and to set it up, deploy the coordination rules in the coordinated work environment.

A *coordinated work environment* (CWE) is an assemblage of several organization units populated by actors, work practices, and artifacts [9]. After finding out how actors are working individually and cooperatively, work practices and common artifacts need to be studied. Especially the dependencies between tasks and the use of common artifacts need to be analyzed in order to understand the exchange between actors, roles and responsibilities actors take, the course of

[1] http://www.informatik.tuwien.ac.at/

routine work and situatedness of the actions, etc. To setup a CWE the collaboration context and environment must be defined. For that purpose both tasks and especially coordinative artifacts are important. CLC is based either on tasks or artifacts, and although the views to processes are different ones, both are modeled as workflows. In the task-based CLC, the flow of work is central and the analysis ends up in a process-oriented model, where as in the artifact-based version the focus is on the (re-)coordination around common artifacts and supports the handling with contingencies.

In the next section, we will illustrate the application of CLC in an experiment to show how its steps contribute to the design and implementation of a coordinated work environment in a real cooperative work setting.

3. The experiment

This experiment is about designing the coordinated process of students' credit transfer at our university. Being part of the team, the authors could use their tacit knowledge and conventions established in this process to define the use cases and the flow of work. In this section, we will show step by step how the coordinated work environment can be setup for collaboration between administration of deans' office, the dean and the lecturers by applying the revised CLC. With this experiment, we want to find out problems and areas for improvement by establishing a coordination environment within the CLC for a real work environment.

For this paper, we choose a subset of use cases, which we introduce here. The credit transfer process starts with a submission of a student on a certain form, called "Application Form for Credit Transfer" (AFCT). This artifact is used by several stakeholders during the process (Figure 1). To start such a credit transfer process, a student needs to fill in the AFCT on both sides and bring it to the deans' office.

After interviewing key actors, observing relevant users at work and analyzing common artifacts several use cases are created (Figure 2) (1).[2]

Depending on the descriptions of use cases and analysis of work processes, we decided to focus on tasks (2). Even if a main common artifact like AFCT is central to organize the work of different stakeholders, it is about tasks which are partly coordinated around AFCT, and partly by regulations and organizational agreements. These are part of the cooperation as much as the AFCT. In our activity diagram (Figure 4) one can see the assignment of actors to tasks (3). Students submit or resubmit their wish for credit transfer in form of AFCT. Lecturers or the dean supported by the deans' office administration staff process students' submissions. We did not add deadlines to tasks, because in the credit transfer process there are no single deadlines defined

[2]These numbers refer to the steps of the CLC presented in the previous section.

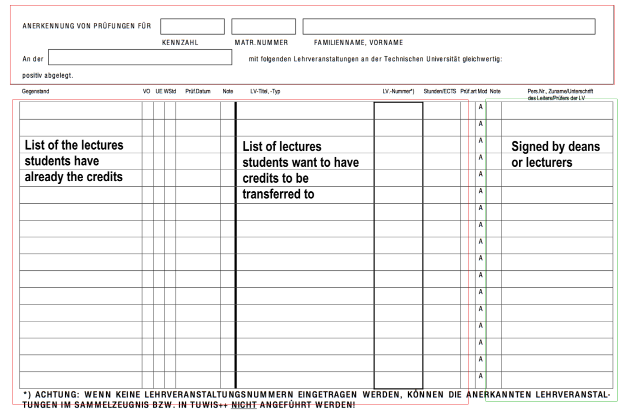

Figure 1. The second page of the application form for credit transfer (AFCT) showing the areas to be filled in by different stakeholders. Red areas filled in by students with support of deans' office administration, the green area filled in by deans or lecturers.

related to each task, rather, it depends on the time of submission (4). The submissions must be processed within one month. This is the only deadline given. All work that needs to be done to keep this only deadline must be carried out as soon as possible, mainly triggered by the common artifact AFCT.

The state diagram (Figure 3) we created to illustrate initial and final states of an AFCT has been extended to show tasks associated with state transitions (5). This is later used in the creation of rules and in consistency check related to task descriptions within the workflow (Figure 4).

To identify dependencies between tasks and subtasks the whole process is modeled by showing the temporal and logical order among them (Figure 4) (6). Additionally rules between tasks are identified (8). E.g., immediately after a submission is cancelled or signed by the dean because it is accepted, the student must be informed (showed as "(meets)" between "cancel the submission" and "inform the student", and between "sign the accepted submission" and "inform the student").

The domain model (Figure 5) is just to illustrate the entities of the credit transfer system (7). The details of the domain and the data dictionary are not shown in this paper. The main common entity is the object credit transfer consisting of submission data (lectures to be transferred, credits required, the number, type and time of the lectures, etc.).

The next step is to create rules based on the analysis of task dependencies and constraints between entities in the domain. Rules in work processes are needed if and only

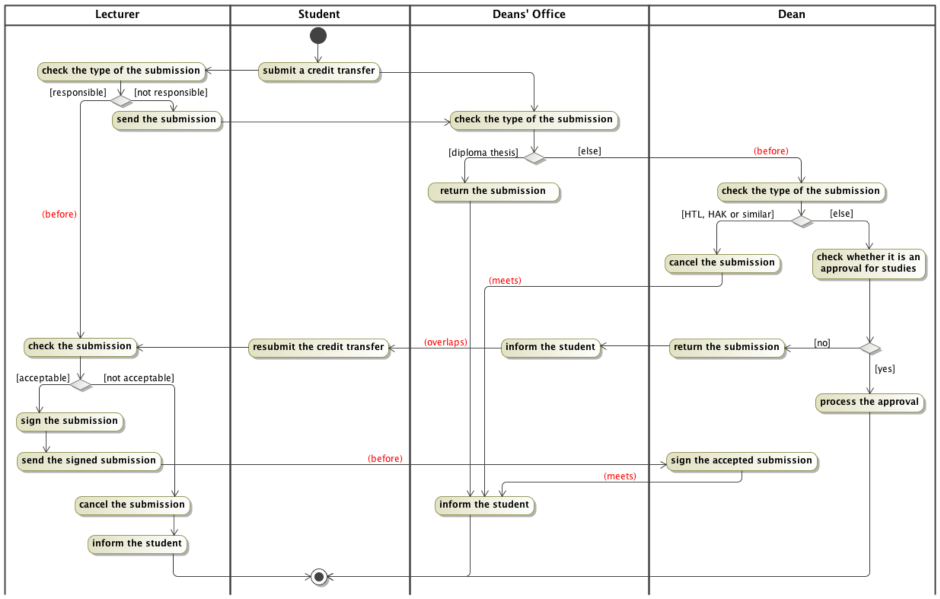

Figure 4. Tasks and their dependencies. Rules are shown as relations between certain tasks, e.g., (meets) or (overlaps) marked red in this diagram.

if there is a need for coordination between actors because there is a work-related dependency between them. Otherwise the workflow covers the logical and temporal order of tasks. Rules are valid not only for the system used (in this case the web-based CWE), they are also guidelines for actors, which define the process of credit transfer for all involved. Guidelines are crucial for common understanding of collaborative work process and articulation work. Analyzing the relations, associations, and hierarchies between entities and tasks, restrictions and norms could be identified. CoorD uses rules to illustrate these hidden dependencies between artifacts and tasks. There were several iterations with users to ensure that rules are complete and correct (8). In the following, the six rules formulated first in natural language, as users can describe and agree upon, are listed.

R_1: `If an AFCT is submitted, the type of the submission must be checked at deans' office, before it is checked by the dean (before-relation).`

R_2: `If an AFCT is submitted, the type of the submission must be checked before its content can be checked (before-relation).`

R_3: `Because if it is a certain type of submission (of HTL, HAK or similar studies) or it is an approval of studies, it cannot be accepted (a guideline for decision making).`

R_4: `Students can resubmit their submission to the lecturer of the according course, before even they are notified about the acceptance or rejection of their submission at the deans' office (overlaps-relation).`

R_5: `Immediately after a submission has been cancelled by the dean, the student must be notified (meets-relation).`

R_6: `If a submission must be approved by the lecturer, the dean waits for his/her signature before s/he can sign it too (before-relation).`

R_7: `Immediately after a submission has been approved, the student must be notified (meets-relation).`

By analyzing activity and state diagrams and by considering all rules, we defined coordination rules to show the main dependencies between tasks (9). Some rules are important to cover the execution order of tasks carried out by different actors, like CR_1. Here is necessary first to check the submission on an administrative level, before process it regarding its content. The rule CR_1 takes care of this constraint. Another example is the flow order of tasks like in CR_2, CR_3, CR_4, CR_5, and CR_6. The R_3 is not implemented as a CR, because it is a guideline for actors, which is defined as a rule for decision making. This is implemented by the workflow engine directly in the flow of work.

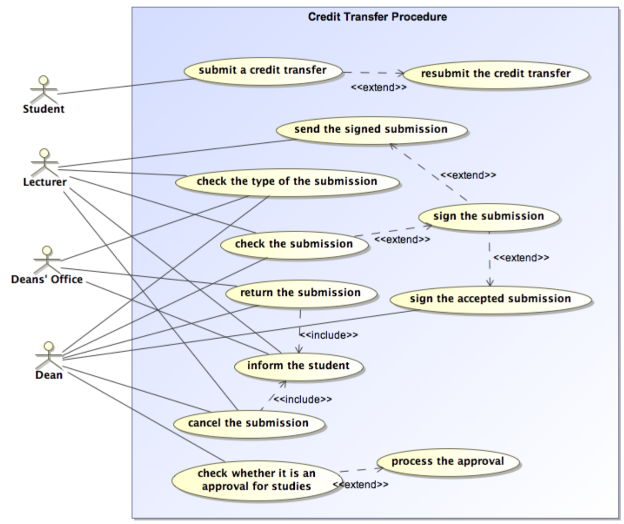

Figure 2. Use cases of the credit transfer procedure at our university.

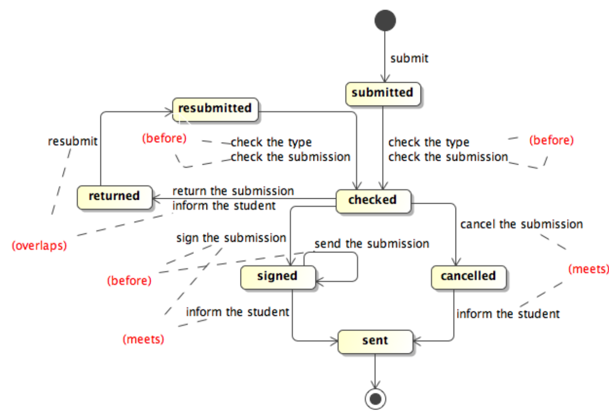

Figure 3. Initial and final states of an application form for credit transfer (AFCT). Rules are shown as relations between certain tasks, e.g., (meets) or (overlaps) marked red in this diagram.

CR$_1$: `checkTheTypeOfSubmissionByDeansOffice is before checkTheTypeOfSubmissionByDean`

CR$_2$: `checkTheTypeOfSubmission is before checkTheSubmission`

CR$_3$: `informTheStudent overlaps resubmitTheAFCT`

CR$_4$: `cancelTheSubmission meets informTheStudent`

CR$_5$: `sendTheSignedSubmissionByLecturer is before signTheAcceptedSubmission`

CR$_6$: `signTheAcceptedSubmission meets informTheStudent`

Tasks associate domain entities with coordination rules. Certain tasks are assigned to certain entities. Figure 6 shows how these associations are implemented in the prototype. In our case we used a relational database to deploy the coordination rules associated with tasks (10) (11).

4. Conclusions

This paper shows how CoorD is integrated in CLC to improve the life cycle with regard to coordination issues. What are the benefits of this? The enhanced CLC provide both predefined and situated coordination [3]. Integration of CoorD into CLC improved the process of designing collaborative work environments: Defining the collaboration environment, e.g., whether it is task- or artifact-focused, assigning persons and deadlines to tasks were missing in the

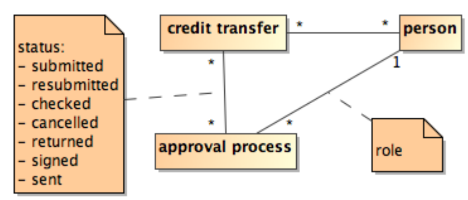

Figure 5. An overview of the domain model.

previous version. These are necessary especially from the management point of view. A new step of analyzing coordination issues in detail is added, which is an important extension to collaboration support provided by CoorD. This paper shows exactly this integration. It shows on the one hand conceptually the new version of CLC, and on the other, practically its application in a real collaborative work environment like the credit transfer process at our university.

CLC can be task- or artifact-based where as both types can be implemented in a workflow. In modeling terms this means that workflows used in CoorD can be created by focusing on tasks carried out by actors and visualizing them using activity diagrams, or by focusing on artifacts used in the workflow and visualizing them using state diagrams. No matter in which form, CoorD extents the CLC in the formation phase, especially in setting up the coordinated work environment. The design of the CWE used in a collaboration setting plays a crucial role: Easiness to access common artifacts, enabling awareness of others and others' work, managing dependencies in collaboration, having

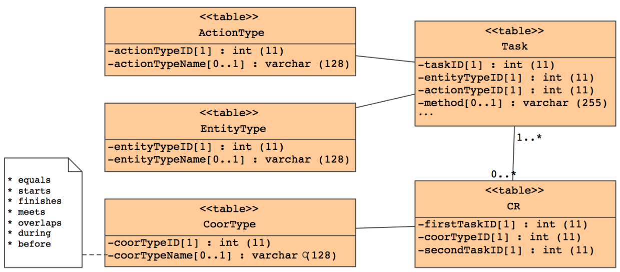

Figure 6. Associations between entities, tasks, and coordination rules [11].

overview of tasks and progress of work, etc. Although there are standard approaches how to build up a collaboration environment, each cooperating group depending on their work habits, product type or work processes established is different, works differently, cooperates differently and needs a customizable environment for its needs. CoorD tries to provide an approach to this problem and shows how to implement an IT environment to support a specific collaboration setting.

Different approaches in CLC complement each other. Activity diagrams show the dependencies between tasks and possible parallel or sequential paths in task execution, where as state diagrams present possible loops and inefficiencies in accessing and processing (common) artifacts. Modeling both views enables capturing discontinuity and gaps in cooperation. In our experiment this happened to us several times. We then went back and reconsidered the tasks or artifacts, and revised our model of the collaboration mechanism. The other benefit of using both the activity and the state diagram is to identify the rules based on dependencies and constraints. If only one is used, it will be not complete and there may be something left out of consideration.

In sum, this research work enabled to evaluate, revise, and further develop both frameworks (CLC and CoorD). It showed bottlenecks and proved their applicability and usefulness in collaboration settings after their revision. The experiment helped to discover leaks in the approach. However, the implementation of the rules and coordination rules is not completed yet. Currently, it is a working prototype and needs to be developed further. Additional work is needed to design and implement CoorD web services to provide an interface for implementation of coordination rules on web-based platforms, which is already the preferred IT environment of distributed collaborative work groups.

References

[1] S. G. Johnsen, R. K. Rolfsen, and H. Tellioğlu. An approach for model based requirement engineering of participative engineering methodology. In A. Pawlak et al., editors, *Proceedings of the 5th International Workshop on Challenges in Collaborative Engineering (CCE'07): Coordination of Collaborative Engineering - State of the Art and Future Challenges*, pages 21–32, GI-Edition, Lecture Notes in Informatics, Cracow, Poland, April 11-13 2007.

[2] S. G. Johnsen, T. Schmmer, J. Haake, A. Pawlak, H. Jorgensen, K. Sandkuhl, J. Stirna, H. Tellioğlu, and G. Jacucci. *New Technologies for the Intelligent Design and Operation of Manufacturing Networks*, chapter Model-based Adaptive Product and Process Engineering. Frauenhofer IRB Verlag, Stuttgart, 2007.

[3] N. Lundberg and H. Tellioğlu. Understanding complex coordination processes in health care. *Scandinavian Journal of Information Systems*, 1999.

[4] T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.

[5] K. Sandkuhl, H. Tellioğlu, and S. G. Johnsen. Orchestrating economic, socio-technical and technical validation using visual modelling. In *Proceedings of the 16th European Conference on Information Systems*, Galway, Ireland, June 9-11 2008.

[6] K. Schmidt. Of maps and scripts. the status of formal constructs in cooperative work. *Paper presented at the GROUP97. ACM Conference on Supporting Group Work, Phoenix, Arizona, 16-19 November 1997*, pages 1–10, 1997.

[7] K. Schmidt and I. Wagner. Ordering systems: Coordinative practices in architectural design and planning. In *Proceedings of GROUP'03*, pages 274–283, 2003.

[8] L. Suchman. Making work visible. *Communications of the ACM*, 38(9):56–64, 1995.

[9] H. Tellioğlu. Modeling coordinated work: Definition and application of the model 'coordinated work environment'. *The Journal of Supercomputing. An International Journal of High-Performance Computer Design, Analysis, and Use*, 24:161–171, 2003.

[10] H. Tellioğlu. CoMex – a mechanism for coordination of task execution in group work. *Proceedings of the 1st International Workshop on Computer Supported Activity Coordination, CSAC 2004, in conjunction with ICEIS 2004 (The 6th International Conference on Enterprise Information Systems)*, pages 13–20, April 2004.

[11] H. Tellioğlu. Coordination design. *Proceedings of the 20th International Conference on Advanced Information Networking and Applications, AINA2006*, 1:425–432, April 18-20 2006.

[12] H. Tellioğlu. *Cooperative Work*. Habilitation thesis. Vienna University of Technology, Faculty of Informatics, Vienna, 2007.

[13] H. Tellioğlu. Collaboration life cycle. *Proceedings of the 2008 International Symposium on Collaborative Technologies and Systems, CTS 2008*, pages 121–126, May 19-23 2008.