Third International Engineering Systems Symposium
CESUN 2012, Delft University of Technology, 18-20 June 2012

Adaptive Collaboration Support Systems: Designing Collaboration Support for Dynamic Environments

Jordan Janeiro, Stefan Werner Knoll, Stephan Lukosch, Gwendolyn Kolfschoten

Delft University of Technology
Faculty of Technology, Policy and Management
Systems Engineering Section
PO box 5015, 2600 GA Delft, The Netherlands

{j.janeiro, s.w.knoll, s.g.lukosch, g.l.kolfschoten}@tudelft.nl

Abstract. *Today, engineering systems offer a variety of local and web-based applications to support collaboration by assisting groups in structuring activities, generating and sharing data, and improving group communication. To ensure the quality of collaboration, engineering system design needs to analyze and define possible collaboration processes. Currently, engineering system design focuses on collaboration processes in a static environment. However, today's world is characterized by dynamic environments that can influence the requirements of a collaboration process and require to adapt the process during runtime. This paper introduces a new approach for engineering systems design that provides adaptive collaboration support for changing environments. This approach is based upon a conceptual architecture for engineering systems that uses data streams to analyze the dynamic environment and adapts a collaboration process on demand according to varying goals, time and data.*

Keywords. *Product Lifecycle Management, Collaboration, Collaboration Support Tool, Collaboration Process Design*

1. Introduction

A product lifecycle is a multi-stage process: beginning from a first product idea to its definition and realization, product support (including service and maintenance) and ending with the disposal of the product (Stark, 2004). Product lifecycle management (PLM) technologies support organizations in planning and controlling their product lifecycles by providing methods and tools for information and process management as well as for the integration of enterprise software (Abramovici, 2007). Besides the management of data and processes, PLM provides tools for collaboration among networked participants in product value chains (Ming et al., 2008).

Due to changes in technology, business and economy, today's organizations act in a dynamic environment that leads to new challenges for their product lifecycle (e.g.

changing market strategies or to support geographically dispersed design teams). From the literature, different needs for future PLM solutions can be identified. For example, Ming et al. (Ming et al., 2008) indicate the need for new technology solutions to support collaboration across multi-organizations and virtual teams. Here, intelligent support can be used to manage the collaboration activities that are tailored to the special needs of global and virtual teams (Hayes, Goel, Tumer, Agogino, & Regli, 2011). According to Abramovici (Abramovici, 2007), main weaknesses of given PLM solutions are the poor support of product lifecycle activities outside the production phase and missing industry standards for PLM meta-data models and processes.

According to Wood and Gray (Wood & Gray, 1991), collaboration occurs when a group of autonomous stakeholders of a problem domain engage in an interactive process, using shared rules, norms, and structures, to act or decide on issues related to that domain. As result, collaboration in engineering is difficult because of the nature of a complex product and because collaboration is a dynamic process. This makes it difficult to prescribe, a priori, an effective collaboration processes within a product life cycle. Collaboration support has been studied in various research domains such as groupware, group (decision) support systems, concurrent design tools and group facilitation (Nunamaker Jr., Briggs, Mittleman, Vogel, & Balthazard, 1996).

This paper describes on how data streams from products can be used to characterize products and to provide a basis for supporting collaboration in a dynamic environment. Within this paper, we focus on the maintenance of a product within the product lifecycle. However, we envision generalizing and applying our findings to a broader scope of the product lifecycle. In the following, we analyze the requirements for collaboration support in a dynamic engineering environment. Then, we present our architecture, an application scenario and we conclude the paper with a discussion about the use of such architecture.

2. Requirement Analysis

The following scenario resulted from a series of workshops and conference calls with twelve experts from the construction industry, in Sweden. As product engineer, maintenance engineer, service engineer and researcher, participants were deemed to be appropriate subjects for this research, because they are familiar with different process stages of the PLM and experts in their domain. During the workshops and online interviews the participants were instructed to describe their work in relation to the PLM. A structured interview was used to identify existing challenges for processes, actors and data of the PLM. These challenges were used to collect ideas on how product data streams can the improvement of the PLM.

The experts indicated that in modern product development, different knowledge domains are integrated in order to develop new services and sustainable product solutions. In line with (Abramovici, 2007), the construction industry recognizes a need for new PLM systems that support product lifecycle activities outside the production phase. Currently, when the product is introduced to the market the link between the manufacturers and customers is usually broken. Experts suggest to keep this link by using product sensors, which generate telemetric data about product usage that can be used as feedback to the manufacturer. In this way, manufacturers can provide proactive maintenance services, in which, maintenance teams use telemetric data to diagnose malfunctioning products and execute preventive actions, avoiding equipment downtime. The key concept for tracking equipment data allows companies to migrate their traditional fail and fix (FAF) methodology into a predict and prevent (PAP) methodology. PAP addresses the fundamental needs of predictive intelligence tools to monitor the degradation of an equipment usage in order to allow interventions to be taken before a unscheduled downtime or unexpected breakdown (Levrat, Iung, & Crespo, 2008).

2.1. Scenario

In the following scenario, a monitoring infrastructure is connected to a construction machine to keep it continuously functioning. For example, a machine can contain sensors to identify the position of the machine, measure fuel levels, vibration of the engine, temperature, and speed. If machine degradation is detected, e.g. the engine of a machine operating over certain thresholds, the monitoring infrastructure immediately reports a problem to the maintenance expert. The expert mobilizes a maintenance team to perform preventive actions. These actions aim to avoid machine breakdown. For this purpose, experts have to quickly analyze and understand machine problems, and to identify solutions. Besides intrinsic maintenance activities, this process requires collaboration. The team has to identify machine failures, malfunctioning components, its causes and consequences, and define action plans in a dynamic situation.

Planning collaboration in such a dynamic situation introduces challenges that are not traditionally considered in collaboration process design. These challenges originate in the dynamics and short problem solving cycles in the process, causing uncertainty about the time available, the goals, requirements and participants for the collaboration process. This situation creates a challenge for designers. Traditionally, collaboration process designers create process on beforehand (Kolfschoten & Vreede, 2009). However, a situation in which time, group composition and goals vary, forces designers and group members to constantly adapt the process during runtime. It is difficult to include a collaboration designer or facilitator in a maintenance team, whenever machines are about to breakdown. Generally, manufacturing companies do not employ these experts. Therefore, a core challenge in this scenario is to capture and model the knowledge of collaboration process designs and reuse such knowledge to adapt the processes during runtime.

2.2. Challenges of Collaboration

The above scenario indicates that in dynamic situations like maintenance: (i) the original goal of the session can dynamically change, (ii) the time planned for a collaboration session can suddenly vary, (iii) the participants that are required to solve the problem can change, and (iv) the collaboration process is constantly redesigned during the session. This matches the four key design concerns in collaboration processes: the goal, the resources, the participants, and the tools/techniques to support the problem solving process (Kolfschoten & de Vreede, 2009). The next list describes the challenges for collaboration design in more details:

1. *Varying Goal*: One of the pre-requirements for designing collaboration processes is to understand the desired goal of the collaboration. In traditional approaches, the designer of the process has to understand in detail the outcomes that should be generated to design the process accordingly. However, in dynamic situations, like in industrial machine maintenance processes, the goal of the collaboration might change. For example, a machine might break down, and its failure (e.g. temperature rise) could accelerate to create a potentially unsafe situation. Because of a change in degradation speed, the collaboration support system used by maintenance team has to support detection of changed requirements to the intervention, and dynamically change the collaboration process to mitigate the risk, and guide the group to decide on an intervention.
2. *Varying Resources*: A normal practice when designing a collaboration process is to get information about the collaboration task, the group, the available resources, and in particular the total amount of time that is available to collaborate (Kolfschoten & de Vreede, 2009). Based on this information, the designer carefully chooses different facilitation techniques of to support during the collaboration. In dynamic situations, the time available might change, which requires process adaptation. For example, a maintenance team can face a hard deadline due to an underperforming cooling sub-system of a machine but the deadline might change due to faster deterioration. However, in dynamic situations, the collaboration process should be flexible. Therefore, the collaboration support system should be able to adapt the process, and the techniques and interfaces used.
3. *Changing Participants*: A collaboration process is designed for a specific group of people. These people need to have the expertise and decision power to resolve the problem, and they need to be committed and motivated to solve the problem (Kolfschoten & de Vreede, 2009). If the goal and resources available for a collaborative problem solving process are changing, this might also require changes in the people required to solve the problem. When the problem becomes more severe, or time pressure increases, it might be required to invite experts or decision makers into the process, and also, when remote participants loose connectivity, they might need to be replaced.
4. *Changing Needs for Collaboration*: A collaboration process has to be designed on demand in real-time, whenever collaboration occurs in dynamic situations. In dynamic situations, the necessary context information is previously unavailable, burdening designers to prepare a collaboration process for a specific session. In such circumstances, designers have to consider all possible alternatives for the collaboration session, creating a complete but generic process, which makes the collaboration process design unsuitable. In this context, a collaboration support system should react to context information and, based on it, adapt the collaboration process and in real-time the collaboration process.

2.3. Requirements for a Collaboration Support System

Based on the above challenges, different requirements for a collaboration support within a PLM system can be identified. The requirements for collaboration support system are as follows:

(R1) - to detect the need for collaboration support: a PLM system needs to provide tools and methods to analyze data streams from product lifecycle to detect the need for collaboration;
(R2) - to find the right experts: a PLM system needs to provide tools and methods to support group formation and collaboration in virtual team;
(R3) - to find the right information: a PLM system needs to provide tools and methods to filter relevant information from the product lifecycle data streams, information that is needed to support collaboration;
(R4) - to support collaboration in a dynamic environment: a PLM system need to provide tools and methods that support the design and execution of collaboration under the constraints of varying goals, resources and changing participants time and goals, which results out of a constantly changing data stream.

3. An Architecture for Adaptive Collaboration Support Systems

This section presents an architecture for adaptive collaboration support systems that use data streams from products to constantly adapt a collaboration process model during runtime. A collaboration process model represents the sequence of activities that a group performs towards a goal. Such models describe the way people interact with each other to accomplish a certain goal. Similar to a workflow management system a collaboration support system can use the underlying process logic of a described collaboration workflow to support the configuration of a collaboration system.

Our approach of a collaboration process model uses a pattern design approach to divide the collaboration process into generic collaboration procedures; work tactics for an intended behavior and outcomes of a group that are needed to achieve the intended goal of a collaboration process. These work tactics form a pattern language for collaboration, which allows a process designer to describe a collaboration process by a sequence of collaboration procedures (Briggs et al., 2009). A collaboration support system can make use of this pattern language to adapt a collaboration process during runtime in different ways. One approach is to analyze the collaboration context by using information provided from data streams from products and the used technology during the execution of a collaboration procedure. The resulting contextual information can be used to adapt the technology used by activating functionalities or providing data that supports a group during the collaboration process. Another approach is to use decision points between the collaboration procedures to adapt the sequence of collaboration procedures in relation to given collaboration context. Here, the collaboration support system analyzes the outcome of a previous collaboration procedure and the information provided from data streams to select an appropriate collaboration procedure that helps the group to achieve the goal of the collaboration process.

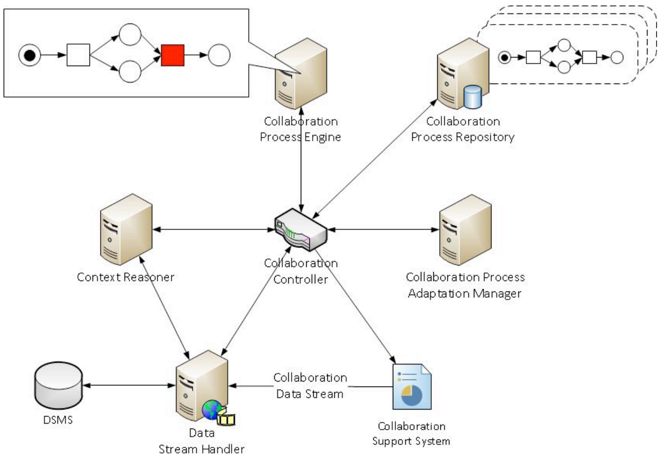

Fig. 1. Architecture for Execution of Collaboration Processes.

The proposed architecture uses data streams to adapt a collaboration process model (presented in the Figure 1). It contains entities to store, execute and adapt collaboration process models, to manage and analyze context information from data streams and to configure suitable tools that are used during the collaboration process. The following subsections describe each entity in detail to clarify its role in the system.

3.1. Data Stream Handler

The data stream handler represents an interface to communicate with the data stream management system (DSMS) (SmartVortex 2011). The DSMS manages and processes incoming data streams from the product into information that can be used during the collaboration process. The data stream handler is responsible for preparing queries for the DSMS using a query language and for retrieving data required to support collaboration and for analyzing the collaboration context. This entity also gathers data stream from the collaboration support system and sends it to the DSMS as part of the collaboration data stream.

3.2. Context Reasoner

The context reasoner (CoRe) uses the retrieved data from the data stream handler and transforms it into processed information that we name context. Thereby, the entity reasons over different data streams to describe different kinds of context such as the operational context of a machine, power consumption, and equipment location (SmartVortex, 2011). It can also create a collaboration context, which describes a collaboration environment and collaboration scenarios.

In this paper context, refers to three properties: (i) expert availability, (ii) time availability and (iii) collaboration process progression. The first property refers to any data that can be inferred to discover the experts that are available or will be available to solve a certain kind of problem. The second property refers to the available time the group has to solve a problem. The third property refers to the activity, within the process, in which the group is executing a certain action. Such properties are used as a reference to design the adaptation of collaboration processes

3.3. Collaboration Process Repository

The collaboration process repository (CPRepo) contains all the collaboration process models that are either designed by a process designer or are adapted during the course of collaboration. Each model describes a sequence of collaboration procedures that use different pieces of process information to define the workflow of a collaboration process. The model describes e.g. the people involved in the process, the type and order of collaboration procedures including decision points to adapt the process, the type and value of the data elements that are used or developed during the process, the contextual information in which the process is used, the tools of the collaboration process and their variations in relation to a specific context.

3.4. Collaboration Process Engine

The collaboration process engine (CPEngine) is responsible for reading, interpreting and executing the collaboration process model. This entity continuously assesses the course of collaboration and recognizes whenever it has to perform the transition between the work tactics of a collaboration process model. This entity also cooperates with the context provider entity, providing the history of a collaboration workflow, as performed along the collaboration process.

3.5. Collaboration Process Adaptation Manager

The collaboration process adaptation manager (CPAM) uses predefined rules for the selection of a predefined collaboration process models from the CPRepo and its adaptation during the collaboration process. These rules are related to different factors like the amount of time that a group has to find a solution to prevent a machine breakdown (response time). Whenever the CPAM is aware of the time in which a group has to accomplish a task, it selects an appropriate collaboration process model. Such selection is based on a matchmaking process that considers different factors like the available response time and the average expected time required for a collaboration process model to be accomplished. Collaboration process models are selected when their expected duration match the required response time. In this case, the CPAM selects a model from the CPRepo and sends it to the CPEngine for its execution.

3.6. Collaboration Controller

The collaboration controller acts as a controller for the collaboration system. It is basically responsible to mediate the communication with the other entities. This entity constantly processes the data generated by every team's participant, analyzing patterns that can trigger collaboration, addressing the requirement R1. The Collaboration Controller is also responsible for identifying the team that should be gathered to be part of the machine diagnosis process, addressing the requirement R2.

4. Application Scenario: Collaboration Support in Diagnosis Processes

The system architecture, introduced in the previous section, represents abstract entities that describe and execute a collaboration process in a dynamic environment. The diagnosis process is a dynamic collaboration process for problem solving in machine maintenance (rf. Section 2). In this process experts need to constantly assess the status of a machine, be available for short notice collaboration and use appropriate tools to analyse the machine data.

According to the ideas proposed by the previous architecture, a company could use a remote monitoring infrastructure connected to a machine to assess its performance. The previously introduced architecture of adaptive collaboration support systems can manage and process incoming data streams from the product with the DSMS. The DSMS and the CoRe use predefined rules to analyze data streams from product lifecycle to detect the need for collaboration (refer to requirement R1) like machine degradation e.g. the engine of a machine operating over a certain threshold. Here, the CoRe immediately reports a machine degradation problem back to the system.

The system analyses the collaboration context, such as: available experts, skills of the experts, available time and amount of data; and based on this information forms a group of expert for the diagnosis process (refer to requirement R2). The CPAM uses the context information provided from the CoRe to select an appropriate collaboration process model from the CPRepo and reuses or adapts it to the current requirements of the collaboration situation. Based on the collaboration process model, the system provides tools and data that will be required by the group to accomplish the diagnosis. For example, a collaboration procedure of the diagnosis process could be to discuss the sensor data of the machine using a video conference call. Therefore, the system interprets the process description and sets up a videoconference room with the referenced experts. Furthermore, the system filter relevant sensor data from the DSMS and provides them as a basis for discussion (refer to requirement R3).

During the whole collaboration process, the system gathers collaboration data being exchanged by users, and constantly analyzes these data. If the system identifies that the group is collaborating in a different way than expected based on the original collaboration process design, the system adapts the process to incorporate new features (refer to requirement R3). For example, if the collaboration process expects the group to reduce the number of ideas generated during a brainstorming session (to identify equipment failure modes) but instead the group evaluates the ideas, the system uses this process variation as an input to adapt the collaboration process for that situation.

5. Conclusions

This paper proposes an architecture that intends to overcome problems that hinder collaboration in dynamic situations, such as diagnosis processes. As we discussed in the paper, collaboration is an important activity to be conducted whenever a group has to handle complex problems. Therefore, the group needs collaboration support to achieve their goal more effectively and efficiently.

Our approach aims at transferring group facilitation knowledge to the system, in order to allow the system to control the execution of collaboration sessions. The idea is that the system contains models of collaboration processes and by those, it guides the collaboration of a team. The difference of our approach is that our system adapts to the course of collaboration of the team. The architecture of the system supports it to learn from the collaboration and adapt the pre-specified collaboration processes. This way, we believe that the system can constantly evolve in an emergent way.

Acknowledgments. This work has been partially supported by the FP7 EU Large-scale Integrating Project SMART VORTEX (Scalable Semantic Product Data Stream Management for Collaboration and Decision Making in Engineering) co-financed by the European Union. For more details, visit http://www.smartvortex.eu/

References

Abramovici, M. (2007). Future Trends in Product Lifecycle Management (PLM). In F.-L. Krause (Ed.), The Future of Product Development (pp. 665-674). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-69820-3_64

Boughzala, I., Assar, S., & Romano Jr., N. C. (2010). An E-government Field Study of Process Virtualization Modeling. In G. J. de Vreede (Ed.), Group Decision and Negotiation 2010 (p. 154).

Dey, A., Abowd, G., & Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction, 16(2), 97-166. doi:10.1207/S15327051HCI16234_02

Hayes, C. C., Goel, A. K., Tumer, I. Y., Agogino, A. M., & Regli, W. C. (2011). Intelligent Support for Product Design: Looking Backward, Looking Forward. Journal of Computing and Information Science in Engineering, 11(2), 9 pages. ASME. doi:10.1115/1.3593410

Knoll, S. W., Plumbaum, T., Hoffmann, J. L., & de Luca, E. W. (2010). Collaboration Ontology: Applying Collaboration Knowledge to a Generic Group Support System. In G. J. de Vreede (Ed.), Proceedings of the Group Decision and Negotiation Meeting 2010 (GDN) (p. 37).

Kolfschoten, G. L., & de Vreede, G. J. (2009). A design approach for collaboration processes: a multi-method design science study in collaboration engineering. Journal of management information systems, 2009(26 - 1), 225-256.

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. Armonk, NY, USA: M. E. Sharpe, Inc. doi:10.2753/mis0742-1222260109

Levrat, E., Iung, B., & Crespo. (2008). E-maintenance: review and conceptual framework. Production Planning & Control, 19(4), 408-429. Taylor & Francis. doi:10.1080/09537280802062571

Lukosch, S. G., & Kolfschoten, G. L. (2011). Towards effective collaboration design and engineering. In T. Hussein, S. Lukosch, H. Paulheim, J. Ziegler, & G. Calvary (Eds.), Proceeding of the IUI workshop on Semantic Models for Adaptive Interactive Systems (SEMAIS).

Ming, X. G., Yan, J. Q., Wang, X. H., Li, S. N., Lu, W. F., Peng, Q. J., & Ma, Y.-S. (2008). Collaborative process planning and manufacturing in product lifecycle management. Computers in Industry, 59(2-3), 154-166. doi:10.1016/j.compind.2007.06.012

Nunamaker Jr., J. F., Briggs, R. O., Mittleman, D. D., Vogel, D. R., & Balthazard, P. A. (1996). Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings. Journal of Management Information Systems, 13(3), 163-207. Retrieved from http://portal.acm.org/citation.cfm?id=1189548.1189557

Pattberg, J., & Fluegge, M. (2007). Towards an Ontology of Collaboration Patterns. Proceedings of Challenges in Collaborative Engineering 07.

Rajsiri, V., Lorré, J.-P., Bénaben, F., & Pingaud, H. (2008). Collaborative Process Definition Using An Ontology-Based Approach. Pervasive Collaborative Networks, 283, 205-212. doi:10.1007/978-0-387-84837-2_21

Romano Jr, N. C., Briggs, R. O., Nunamaker Jr, J. F., & Mittleman, D. D. (1999). Distributed GSS Facilitation and Participation: Field Action Research. Hawaii International Conference on System Sciences (Vol. 1). Los Alamitos, CA, USA: IEEE Computer Society. doi:http://doi.ieeecomputersociety.org/10.1109/HICSS.1999.772737

SmartVortex. (2011). SmartVortex. Retrieved from http://www.smartvortex.eu/?q=Project

Stark, J. (2004). Product Lifecycle Management: 21st century Paradigm for Product Realisation (p. 400). Springer. Retrieved from http://www.springerlink.com/content/978-1-85233-810-7/

Terveen, L. G. (1995). Overview of human-computer collaboration. Knowledge-Based Systems, 8(2-3), 67-81. doi:doi: 10.1016/0950-7051(95)98369-h

Wood, D. J., & Gray, B. (1991). Toward a Comprehensive Theory of Collaboration. The Journal of Applied Behavioral Science, 27(2), 139-162. doi:10.1177/0021886391272001