A Model-Based Software Architecture to Support Decentral Product Development Processes

Patrick D. Stiefel and Jörg P. Müller

Clausthal University of Technology, Department of Informatics, 38678 Clausthal-Zellerfeld, Germany
{patrick.stiefel,joerg.mueller}@tu-clausthal.de

Abstract. The cross-organizational, collaborative product development needs adaptive and open platforms for integration that scale well and provide an intelligent environment especially in regard to definition and design. Based on these requirements our research focuses on how decentral information technology can further the processes especially needed for collaborative product development (CPD). We therefore follow a model driven development (MDD) approach, but it is still an open research question how requirements for that approach in a decentral development process should be reconciled with the potential of decentralized IT systems (especially in P2P environments). This paper presents methods and models which have been hardly taken into consideration by existing centralized collaborative systems. We aim to provide the basics for the next generation of loosely coupled collaborative systems for cross-organizational CPD.

Keywords: Decentral and collaborative product development (CPD), Peer-To-Peer(P2P)-based collaboration, model driven development (MDD) of decentral organized information systems, loosely coupled collaboration platform.

1 Introduction

Collaboration platforms are a modern approach to support cross-organizational knowledge processing in product engineering. The resulting cooperative product data management networks not only support worldwide access to all data concerning the product life cycle but also contribute to the-long term challenges of digital product engineering processes.

According to projections of [1] there is going to be a revolution in key industries within the automobile economy. Increasing pressure in cost and innovation is expected to force manufactures into a "productivity squeeze". This is expected be the third industry changing revolution after the assembly line introduced by Henry Ford and the introduction of Lean Production by Toyota. Key changes affect the structure of products in general, the in-house production depth, relocation of innovation and revenue drivers leading to moving away from OEM based production models towards system suppliers massively influencing the end product.

Collaborative product development (CPD) is the most important part of this development. It can be divided into synchronous ("same time – different place") and

R. Sharman, T.S. Raghu, and H.R. Rao (Eds.): WEB 2009, LNBIP 52, pp. 147–159, 2010.

asynchronous ("different time – different place") collaboration. The first mode is mainly used upon real-time communication or data transmission while the latter focuses on workflow management [2]. During multi party cross-organizational engineering projects CPD platforms provide an added value at definition and execution stages, as the main motivation is usually to combine distributed core competencies in design, production, and process know-how. CPD platforms that we are talking about focus on design-by-feature technologies that are mainly used in product design and development. These systems provide distributive and collaboration-enabling support for global design collaboration. One of its major challenges is to share large data volume design models and design changes among a working team efficiently [3].

Based on these requirements our research focuses on developing decentral information technologies to support CPD processes ([4],[6]). A result is the Product Collaboration Platform (PCP). The PCP is an experimental peer-to-peer(P2P)-based software platform to support decentral organized product development processes. We follow a model driven development (MDD) approach to design information systems for decentral and collaborative product development (DeCoP). Based on Computation Independent Models (CIM) we develop in an iterative process different abstraction of IT models: starting from IT architecture models (platform independent models, PIM) over platform specific models (PSM) to concrete software artifacts.

It is still an open research question how requirements for a MDD approach in a decentral development process should be reconciled with the potential of decentralized IT systems (especially in P2P environments). This paper presents methods and models which have been hardly taken into consideration by existing centralized collaborative systems. We propose approaches at different levels of MDD, which can be used to generate decentral architecture models and in the end software artifacts – automated on some levels. We aim to provide the basics for the next generation of loosely coupled collaborative systems for cross-organizational product development.

Section 2 of this paper deals with fundamentals of DeCoP. In Section 3 we discuss the current state-of-the-art of model driven engineering and the specific requirements during the processes of the MDD approach for the development of decentralized collaboration platforms. Section 4 presents models of the different MDD levels and section 5 concludes with an evaluation of the presented approach with view to real world scenarios.

2 Background

There are actually four buzzwords used as descriptors for what people need to do to work together: ***Communication, Coordination, Cooperation*** and ***Collaboration.*** People tend to use these words interchangeably.

According to the 3C model of Teufel et al. ***communication*** describes the way on how information (especially human experiences) is transferred in organizations [7]. Without ***coordination*** different units create overlap, redundancy and/ or separation. Unlike *communication*, *coordination* looks to inform each unit or part of the whole as to how and when it must act to achieve efficiency in the group. What's still missing is a connection between inputs and outcome to tell about the consequence dependent on context. ***Cooperation*** is the strongest style of teamwork among groups and needs a strong compliance of targets: the group as a whole needs to be in charge of the result ("get with the group").

Collaboration is distinct from each of the C words mentioned before. Unlike *communication* it is not about exchanging information, it is about using information to create something new. Unlike *coordination*, it has a desire for spontaneity not structural harmony. And unlike *cooperation*, it allows disagreement, dissent and even conflict. *Collaborations* are established to solve problems, to develop new understandings and to design new products. The following steps are needed to create *collaboration*: Define the challenge (achievement point), define the collaborators, create a space (blackboard/ shared screen), allow the time and harness the result like a prototype e.g. ([8],[9]).

Our research describes different shapes of *collaboration* dealing with the challenges of cross-organizational product engineering aiming to facilitate the engineering of a product model proposal which satisfies the requirements set out by all parties involved in said process. A *product model* is defined as "[...] a general product structure for a certain individual product. It contains information on an individual product, recorded and arranged to (a corresponding) *product information model*. For example, the individual product units', product models or product structures for two similar but customized products might differ, even though the products are alike at a generic, product information model level" [10].

According to Lindemann business processes in cross-organizational product engineering reflect a certain development process and work with a specific product model shape [11]:

- ***Target model:*** gathering, structuring and documenting of desired system properties.
- ***Problem model:*** to improve the understanding of problems and challenges of the product in regard to existing or future properties (i.e. strain, feasibility, material characteristics).
- ***Development model:*** to aid in the specifications regarding the structure and the geometric and material condition of the product to be developed.
- ***Verification model:*** compilation and analysis of essential properties which matter for the evaluation with view to expected quality and requirements.

Each DeCoP process describes a distributed solution of a given product engineering problem (***specification***). Our approach, as shown in Figure 1 is based on the Distributed Problem Solving paradigm solving paradigm developed for multi-agent systems ([12],[13]). The DeCoP process provides a synthesis of the distributed partial solutions of the participants (***proposals***) to an overall solution satisfying the initial requirements of the initiator.

Nowadays decentral organization of distributed systems is a known paradigm for managing global and dynamic networks [14]. A decentral architecture based on P2P technology provides the needed flexible and scalable approach for a collaborative product engineering platforms:

- Support of ad-hoc interconnections of world-wide distributed partners that usually did not cooperate in the past.
- Efficient distribution of product models among participating engineers either for load balancing reasons or for the reasons of task distribution resulting in exploitation of net locality to facilitate lower latency.

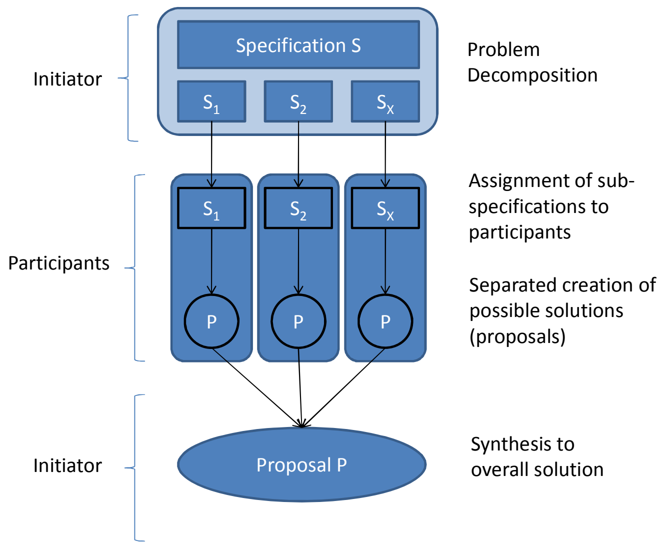

Fig. 1. Decentral and Collaborative Product Development (DeCoP)

3 Model-Driven Decentral and Collaborative Product Development

The definition of models on different levels of abstraction and therefore the modeling of different aspects is one of the main ideas of MDD [15]. CIM describes the functionalities of the collaborative platform on the functional level and is expressed in a language suitable for the audience (i.e. engineers). It is meant to facilitate the collaboration among the stakeholders (i.e. software developers and engineers) in the DeCoP. In our research we ended up with the Business Process Modelling Notation (BMPN) after starting off with Aris Event-driven Process Chains (EPC). BPMN is a notation for business processes as well as technical models (i.e. loops, exception handling, and transactions). Workflows expressed in BPMN get interpreted and executed with the process engine of a Business Process Management System (BPMS). Business Process Diagrams (BPDs) provide a visual representation of BPMN models.

The platform concept in MDD characterizes a closed software component or technology with access through interfaces. Accessing components need not to know about the implementation of concrete platform functionalities. The platform provides technical services that are necessary to provide an expected software behavior. In correlation to platform the concept of a *Platform Independent Model (PIM)* is used. A PIM is a model for the functionalities of components which is independent of the platform. The *Platform Specific Model (PSM)* in contrast knows about services compositions, DRM, and overlay specifications. The PSM and its services enable the PIM on the platform.

3.1 Concept

MMD concepts are already applied in the field of product lifecycle management. Néjib et. al present case studies about its application to develop software applications that ensure the interoperability between given heterogeneous information systems [5]. As shown in Figure 2, we follow this top-down idea and start with functional descriptions of DeCoP business processes on CIM level to end up in decentral architectures at PSM level [6]. Our architecture approaches are described in three PSM layers: a SOA framework, a Decentral Resource Management (DRM) layer and a specific P2P overlay.

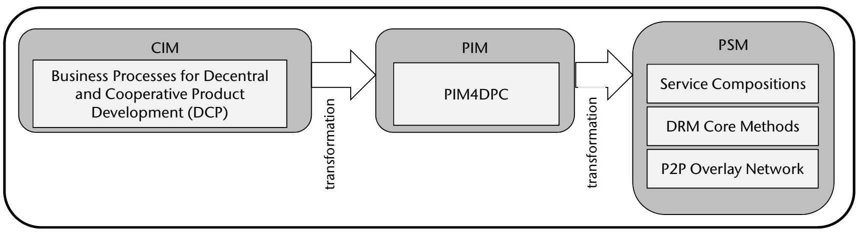

Fig. 2. MDD process for DeCoP

The DRM is an abstraction layer of specific P2P overlay functions especially enhanced for decentral product model handling (publish, search, subscribe, notify). Generic collaborative services are based on these DRM core methods (i.e. the management of a product model structure). They are part of the service layer and get executed from local service engines.

3.2 Requirements

During the model generation at MDD process we were made aware that – for decentral organized information systems – all layers of the model deal with requirements concerning the network topology. We provide two examples in the *CIM layer* to illustrate our findings: First a goal of cooperation could be to facilitate a design for the product structure based on a *development model*. Second could be a CIM workflow dealing with the exchange of ideas aiming to find valid states for the *target model*. Both business processes differ – independent from being decentral – on how they make use of generic product development processes to represent a DeCoP process. In the first case several participants work on isolated partial problems and send their results to an initiator. The initiator reviews the results, comes to a conclusion, and decides on how to proceed (cp BPD at Figure 3, left hand). The second example does not really deal with a centrally managed process as there are several initiators involved working on several iterative produced results in a parallel manner. Both collaborative scenarios are decentrally organized collaborations but with totally different requirements for their participants in regard to their level of trust concerning their contributions to the product models (cp Section 4.1). During the model transformation the requirements influence an outcome of the models in the *PIM layer*.

Decentralism becomes more of an issue once we descend down to the architecture in the *PIM*. It develops more towards questions regarding the requirements resulting from the mappings in CIM and how those affect specific properties of the P2P overlay. The decision whether a structured or hybrid approach is made during that stage (independent from how it is going to be implemented later). In addition how roles are going to be assigned to which peers during the collaboration process and how these roles lead to the meeting of specific requirements set for those roles and task in question. It results in specific architectures and services (cp Section 4.4).

Based on the architecture and services in the PIM layer the *PSM* implements corresponding service oriented architectures (SOA) and specific P2P overlay implementations to match the requirements of the DRM layer. As the PIM describes how the processes are distributed the PSM maps needed Service Choreographies and Compositions (cp Section 4.2 and 4.3). It takes care of the details on how a service is distributed and executed through responsible peers and for example whether or not there is someone coordinating the process?

4 Models of a MDD Process

4.1 DeCoP Business Processes at CIM Layer

A basic business process of DeCoP is shown in the BPD on the left hand in Figure 3. Creating a specification starts with the thoughts of the initiator leading to an initial idea about a possible shape and properties ending up in a first sketch of a development model and parameters called *specification*. Corresponding to that artifact a data element *proposal* is getting published by a participant. The initiator receives and checks the proposal. Product model collaborations derived from the basic business process have the following dimensions: Product Model(PM-)Distribution, Hierarchy, and Iteration. Figure 3 (right hand) illustrates the introduced dimensions of the product development scenarios introduced in this paper. E.g., a simple scenario is the combination: no iterations/ central model repository / no hierarchy.

PM-Distribution: The BPD explicitly assigns specifications and proposals to their respective owners. The initiator decides how the data is distributed among participants in the engineering process – trust can be one of the criteria. There are three different strategies: central, hybrid, and decentral PM-Storage.

In case of a *central PM-Storage* one single peer stores the specification and the corresponding proposals, default peer is the initiator. The BPD reflects this with an assignment of the artifacts *specification* and *proposal* to the initiator´s pool (cp Figure 3, left hand). In addition Figure 4 shows on the left hand the case of a cooperative network with central PM-Storage. The notation Vz(Ty).x next to the product models reflects the state of the element as version x of proposal z made by participant y.

In case of a *hybrid PM-Storage* all participants organize the storage themselves. Accordingly the BPD shows the artifact specification at the initiator and the proposals at the corresponding participants. The option to distribute proposals over several participants remains. That would result in a collaboration of a subset among the participants which would be the case when partial results from other peers are needed to submit a proposal (cp Figure 4, right hand).

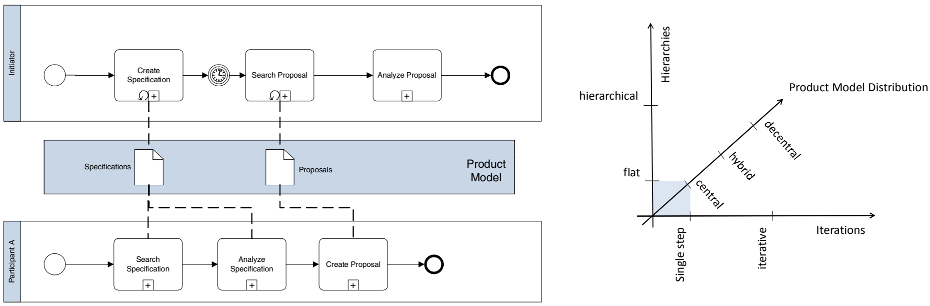

Fig. 3. DeCoP basic business process, designed as BPD (left hand) and DeCoP Dimensions (right hand)

In case of decentral PM-Storage the specifications and proposals get stored "arbitrarily" at peers in the network (decentral method). For that reason neither specification nor proposals get assigned to a specific pool in the BPD.

Hierarchies: A participant has the option to further split and distribute the given sub-problem from the initiator to participants of his choice (i.e. in a separate network). A BPD illustrates this with a set of sub specifications following the earlier mentioned notation. In general there is no limit for additional participants or hierarchy level s. Hierarchies are possible in all PM-Storage options.

Iterations: Collaboration processes without iterations are unlikely in reality. In the decentral development of a product model, numerous iterations should be needed in reality. For instance an initiator notices the need for an update to his specification after receiving the first proposals. The option to request an update to specific proposals exists as well. For that reason two new processes get introduced ("update specification" and "update proposal").

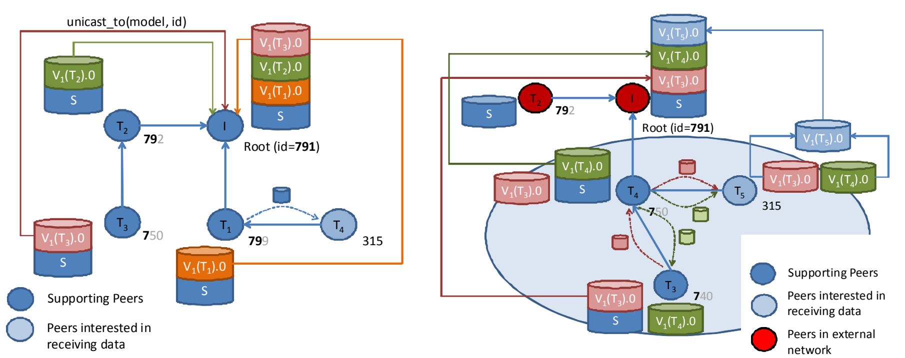

Fig. 4. Product Model Distributions: Central PM-Storage (left hand) and hybrid PM-Storage (right hand)

4.2 DeCoP Architecture Styles

According to Roser we distinguish between three archetypes of architectures for the execution of cross-organizational business processes: brokerless, central broker, and decentral broker architecture [16].

Brokerless architecture: The brokerless architecture uses either message passing between services or observer services on a virtual product model to establish a certain level of control flow logic. The needed interaction between nodes can be achieved with choreography while a peer's local task services (TS) are simply orchestrated [16].

Central broker architecture: A central broker runs a controller service (CS) to achieve the needed control flow logic of the collaborative business process. The idea is following the paradigm of an orchestration.

Decentral broker architecture: The central controller service is split among peers acting as controller service over a subset of the network. These peers run a service to distribute information to achieve the same orchestration and control flow logic as if there were a central broker.

The choice of one of these archetypes depends on the concrete service distribution. The distribution describes which roles (initiator/participant) provide services and who is allowed to use them. This decision is directly related to the product model distribution set by the developer of the business process (cp section 4.1). Which strategy is going to be used during the collaboration is decided during runtime and is based on the participating users and the trust among them.

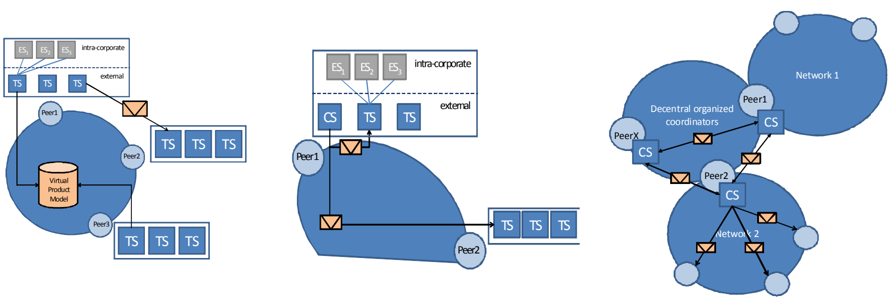

Fig. 5. Archetypes of architectures for the execution of DeCoP business processes (Roser 2008)

It has to be distinguished between the following *Service-Distributions*:

1. In case of a central PM-Storage on a fixed set of nodes the *brokerless architecture* is the best choice. Every node in the network ends up with all collaboration services (= *central availability of the services*).
2. In case of a hybrid PM-Storage all proposals get stored on nodes of a trusted subset of the network. Peers in a team are therefore able to share services among them. The *central broker architecture* is a good fit. The initiator assumes the role as CS and the TS roles are assumed by selected participants (= *hybrid availability of services*).

3. In case of decentral PM-Storage (i.e. load balancing) product models can be “arbitrarily” distributed among the nodes. In this case *decentral broker architecture* provides a good fit. The CS uses P2P technologies i.e. DHT technologies of structured P2P networks *(= decentral availability of services*).

4.3 Platform-Specific Models for DeCoP (PSM4DeCoP)

As part of our research we started with basic modeling and evaluations of the brokerless architecture in the PSM (cp Figure 6). Each peer has a set of functional services which is developed locally through a participant. For instance the service “Create Specification” could result in the creation of a specification with data extracted through a product data management system (i.e. Teamcenter Engineering, TCE). In addition there are a number of technical services like “Publish”, “Search”, “Subscribe”, and “Notify” which – according to the architecture model – are implemented the same way in all peers [4]. A workflow is assigned to each development partner and includes a plan for the local services to make sure the business process is executable.

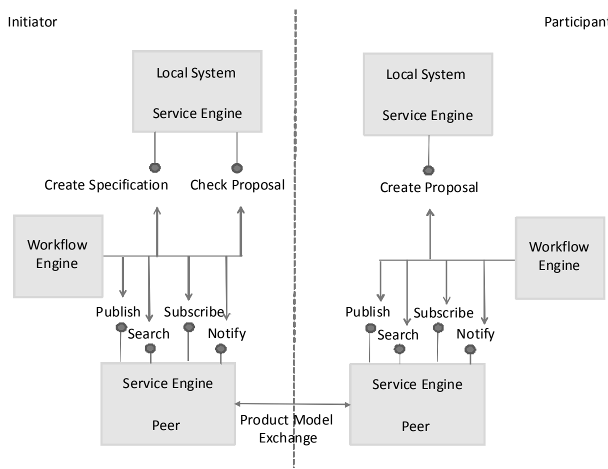

Fig. 6. Model of a brokerless architecture for DeCoP

4.4 Platform Independent Model for DeCoP (PIM4DeCoP)

Sections 4.1-4.3 show that the models at CIM, PIM and PSM level are not static. Some parameters reflect this and are needed to set up for a top-down approach based on the business process designer on the one hand and for a bottom-up approach based on the IT expert on the other hand. We give an example for the top-down approach: the business process designer creates the following two scenarios for DeCoP (A and B) and sets the parameters for the given dimensions in Table 1.

Table 1. DeCoP dimensions

DeCoP Dimension	Development Models	Target Models
Maximum number of participants (including sub initiators)	100 (10)	1000 (1)
Number of specifications	10	1000
Number of proposals per specification and participant (Iterations)	10	100
Permitted collaboration-independent sub-developments	Yes	No
Participant behavior	Confidential product models/ cooperation only with initiator	Non-confidential PM/ Cooperative behavior

The challenge for PIM4DeCoP is based on the evaluation of adjusted parameters (cp Table 1) to make an automated transformation and to come up with a recommendation for the architecture (cp section 4.2). We would expect the following: DeCoP scenario A should end up in a *brokerless architecture* while scenario B is a candidate for a *decentral broker architecture*. PIM4DeCoP´s metamodel describes the following four aspects: *organization, information, process, and overlay* (cp Figure 7; the model was simplified due to space constraints).

The *organization´s metamodel* describes the collaboration structure among the *actors*. Actors can only take over *processes* they are suited for according to their *privileges*, they were assigned to over their *team role*.

The *information´s metamodel* contains all the necessary information for the product development. An *InformationObject(IO)* represents the structure and design of a

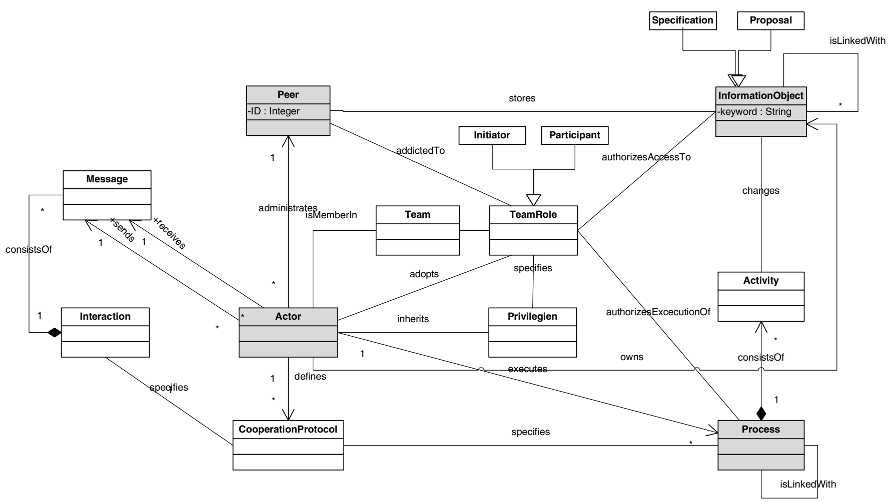

Fig. 7. PIM4DeCoP metamodel

product model and connections between *specifications* and corresponding *proposals*. Each *IO* gets assigned to an *actor* based on its *TeamRole* and is stored on an associated *peer*.

The *process´s metamodel* maps *activities* in a graph to represent collaborative business processes. *Activites* are assigned to *actors* and correspond to specific *IOs*. Several *activities* result in a collaborative business *process* which is further specified by a *CollaborationProtocol*.

The *overlay´s metamodel* is one source of information for the identification of a P2P overlay in the PSM. Core components of the metamodel are *peers* and their corresponding *actors* and *IOs*.

5 Evaluation and Summary

Our paper indicates how to develop a platform for DeCoP with the aid of existing MDD process models. We introduced approaches to the different levels of abstraction. We are going to focus on two aspects for our concluding evaluation of the MDD approach:

1. To what extent is the model driven process suited for our approach of developing a decentral organized collaboration platform. Therefore we are going to focus on the final product quality to measure whether the introduced models in the CIM, PIM, and PSM layer suffice. The quality is measured based on two criteria: a) acts the software as expected by the modelers and b) are the models flexible enough to support a decentral infrastructure?
2. How well do the introduced processes do in the decentral environment when compared to traditional client server collaborative platforms?

Aspect 1a: Tests and simulations based on our implemented prototypes indicate whether the software produces the expected results of the modeler. The following criteria help to determine that:

- Was the goal of the collaboration achieved? If it was not, is the business process correctly designed?
- Does the achieved product model distribution match the model?
- If all workflows are correct: Does the achieved collaboration course match the expected process?

Aspect 1b: It is difficult to answer whether the models are flexible enough for the decentral case or not. The resulting dynamic of the system has to be taken into account during design time as the success of the collaboration at run time depends on the level of model details. Because peers act autonomous it cannot be said whether a peer is available all the time which makes it kind of difficult to make deterministic assumptions during design time.

The dynamic system behavior – characteristic of a P2P network – has to be taken into account during all modeling stages of the MDD. Generally we account that by generating independent model parts at all MDD levels – This could for example be a self-contained local part of a cooperative business process or an adapted local workflow part executed by a single peer. The interaction between several separated activities is realized through the product model itself. Therefore we normally talk about model-centric processes. Furthermore we model dynamic system behavior with expected behavior at PIM (Quality of Service, QoS). This allows us to select relevant P2P overlay structures one the one hand and to optimize needed workflows and service distributions on the other hand. The following OoS parameters are relevant:

- Availability of local product models
    - Definition of groups which replicate each other´s data
    - Inheritance of roles within teams
    - Queuing theory for buffering data during temporary non availability
- Scalability
    - Efficient product model storage in the network matching the dependencies of the company structure in the DeCoP
    - Content-delivery strategies / data chunks for large files
- Search
    - Efficient search for peers and product models (i.e. wildcards, exact match, range queries) independent of the amount of peers in the network
- Trustworthiness
    - Private key infrastructures for sensitive product models
    - Possibility for keyword-based subnets

Metrics allow the evaluation on how good an implementation of the overlay is fulfilling the QoS requirements. Metrics measure adaptivity (scalability/ stability/ flexibility), efficiency, validity and trustworthiness (reliability/ security) [18].

Aspect 2: In the end the acceptance of DeCoP in companies depends on a lot of elements which have to be taken into account during the development. The following paragraph mentions a few of them.

Comparing the P2P approach to the traditional client-/server approach reveals advantages in regard to scalability, flexibility, data storage, and response time. Even though all of these advantages never lead to a P2P based implementation of a collaboration platforms used in companies. After we presented our prototype to companies the feedback provided us with two main reasons why this was neglected so far:

- The product engineers are afraid of losing the control of their product data and
- Companies are not sufficiently aware of the need to disclose their processes for more efficient collaboration.

We claim that the MDD approach introduced in this paper should help to overcome these obstacles. We have to create an awareness among users that the behavior (and its consequences) of the actors already needs to be taken into account during design time. It is part of our concept to have designers of decentral organized, cross-organizational cooperative product development processes to model business processes on CIM level in their well known environment (i.e. BPDs). This would allow designers to work on their CMP processes without the need to take care of the IT level decisions and get a concrete P2P network supporting the process.

References

1. Frise, P.R.: Integrating design and manufacturing processes in automotive suppliers. In: Bramley, A., NcMahon, C., Coutellier, D., Brissaud, D. (eds.) Advances in Integrated Design and Manufacturing in Mechanical Engineering, Pages, pp. 227–242. Springer, Netherlands (2005)
2. Li, W.D., Qiu, Z.M.: State-of-the-Art Technologies and Methodologies for Collaborative Product Development Systems. International Journal of Production Research 44(13), 2525–2559 (2006)
3. Li, W.D., Ong, S.K., Nee, A.Y.C.: Integrative and Collaborative Product Development Environment. In: Technologies and Implementations. World Scientific Publishing Co. Pte. Ltd., Singapore (2006)
4. Stiefel, P.D., Müller, J.P.: ICT interoperability challenges in decentral, cross-enterprise product engineering. In: Gonçalves Ricardo, J., et al. (eds.) Enterprise Interoperability II: New Challenges and Approaches, pp. 171–182. Springer, London (2007)
5. Néjib, M., Hanène, C., Yacine, O., Frédéric, N., Abdelaziz, B.: Model-Driven Architecture to enhance interoperability between product applications. In: Proceedings of the 5th International Conference on Product Lifecycle Management, Seoul, Korea,
6. Stiefel, P.D., Müller, J.P.: Realizing dynamic product collaboration processes in a model-driven framework: Case study and lessons learnt. In: Thoben, K.-D., Pawar, K.S., Gonçalves, R. (eds.) 14th International Conference on Concurrent Enterprising, Lisbon, Portugal, June 23-25 (2008)
7. Teufel, S., Sauter, C., Mühlherr, T.: Computerunterstützung für die Gruppenarbeit, Oldenbourg (1995)
8. Schrage, M.: The New Technologies of Collaboration, p. 140. Random House, NY (1990)
9. Winer, M., Ray, K.: Collaboration Handbook: Creating, Sustaining, and Enjoying the Journey, Fieldstone Alliance, 1st edn. (1994)
10. Saaksvuori, A., Immonen, A.: Product Lifecycle Management, p. 22. Springer, Heidelberg (2005)
11. Lindemann, U.: Methodische Entwicklung technischer Produkte. In: Methoden flexibel und situationsgerecht anwenden. bearb. Aufl, vol. 2. Springer, Berlin (2007)
12. Müller, J.P.: The Design of Intelligent Agents. LNCS, vol. 1177. Springer, Heidelberg (1996)
13. Smith, R.G.: Frameworks for Cooperation in Distributed Problem Solving. IEEE Transactions on Systems, Man, and Cybernetic 11, 61–70 (1981)
14. Stäber: Service layer components for decentralized applications. Dissertation, TU Clausthal (2008)
15. Miller, J., Mukerji, J.: MDA Guide Version 1.0.1 OMG (2003), `http://www.omg.org/cgi-bin/doc?omg/03-06-01`
16. Roser: Designing and Enacting Cross-Organizational Business Processes, Dissertation, University of Augsburg (2008)
17. Erl, T.: Service-oriented architecture. Concepts, technology, and design. 5th print. Prentice-Hall, Upper Saddle River (2006)
18. QuaP2P: Reports of the QuaP2P research group established by the German Research Society DFG for the "Improvement of the Quality of Peer-to-Peer Systems by Systematically Researching Quality Features and Their Interdependencies" (FOR 733) (2009), `http://www.quap2p.tu-darmstadt.de/`