Advancing Adaptivity in Enterprise Collaboration

Norbert Jastroch
MET Communications
Bad Homburg, Germany
norbert.jastroch@metcommunications.de

Abstract

Enterprise interoperability and collaborative work are two aspects to involve in knowledge oriented enterprise collaboration. The first one implies a rather systemic view, the second one a rather activity related view on collaboration of enterprises within their ecosystems. For example, manufacturing or distribution are often dealt with by applying an input-output perspective. Product development or supply maintenance are increasingly organized as a set of networked activities performed by different actors inside and outside the enterprise, and therefore are to be dealt with by applying the perspective of composite practices.

Information technology is challenged to support both aspects. However, the predominant approaches in modelling and engineering of systems still permit instantiations with low adaptivity. This interferes with easy take-up in business practice. On this background, the paper suggests enhancing existing structure with additional features to ease adaptivity of enterprise collaboration systems. It reflects work that was done on the subjects of enterprise interoperability and collaborative working environments.

Keywords: Enterprise collaboration, interoperability, adaptivity, composite practice, composite systems, cross-organizational processes, knowledge management.

1 Introduction

The Enterprise Interoperability Research Roadmap of the European Commission [3] considers knowledge oriented collaboration one of the four grand research challenges in the field of Enterprise Interoperability; web technologies, interoperability service utility, and the creation of a science base for enterprise interoperability are the others. The two fundamental aspects to involve in knowledge oriented enterprise collaboration are enterprise interoperability and collaborative work.

Enterprise interoperability, as discussed in [4], is affected by interoperability barriers in the three categories: conceptual, technological, and organizational. While the problem of mere technical connection of IT systems has been resolved by the global Internet infrastructure, the need for ICT systems supporting business processes across enterprise networks is rising. Design, engineering and development of such systems pose specific challenges that need to be overcome.

The space of concern of ICT supporting enterprise collaboration today primarily comprises the mapping of business processes. From a business engineering point of view, this means the operation of shared processes as well as the employment of common knowledge objects. From a systems engineering and development point of view, it provides a mechanism to deal with the two dimensions of interoperability and adaptivity. In this paper we address in particular these issues.

We start by reflecting on collaborative software development as a manifestation of enterprise collaboration. It is considered as a composite practice based upon the sharing of knowledge while aiming at joint activities performed within different though linked business processes. The issues of interoperability and adaptivity can be viewed and instantiated in the context and within the space of complexity of resulting systems. We describe engineering approaches to enterprise collaboration that keep the potential to enable adaptivity. In subsequent chapters we suggest the application of knowledge management based concepts for collaborative software development, in particular the use of interfaces set out to advance the adaptivity of enterprise collaboration systems.

2 Enterprise collaboration: A composite practice

In inter-enterprise collaboration significant complexity arises from the business processes involved and their mapping, and from the semantics that need to be taken into account. One of the issues raised by the linking of core competencies of collaborative network members is quality management. In [14], the need for quality-bridging within the temporary value chain of a collaboration is being pointed out.

Independent from these quality issues, the mapping of business processes in any inter-enterprise setting becomes a question of organizing complexity. And so does semantics, which is necessary to establish a common understandability of messages that control knowledge sharing activities.

Collaborative manufacturing or distribution, for example, are often dealt with by applying an input-output perspective. Supporting ICT systems mostly focus on the interchange of production data and information. Object oriented engineering approaches like those of the OMG [12] offer a comprehensive set of instruments for these purposes. For the engineering of such business processes the concept of Business Process Management (BPM) has been established, comprising standards, tools and technologies for the modelling and execution of processes. A number of most prominent business process modelling methods have been discussed by Lippe et al [10], especially as to their ability to facilitate cross-organizational processes. Their findings well apply to enterprise collaboration as this implies particular forms of cross-organizational processes.

As discussed by Allert & Richter [1], in the case of cooperative work, however, object oriented approaches fall short in describing the social and mediated nature of the related practices. Collaborative product development or supply maintenance, for instance, are often organized as a set of networked activities performed by different actors in different enterprises. Collaboration here includes contextual joint actions, and therefore requires to apply the concept of composite practice. In contrast to a particular activity a practice represents a recurrent pattern which can be filled out by various activities actualizing the practice, or to put it differently, a practice denotes the prototypical characteristics of a set of activities including forms of bodily and mental activity, tools and their usage, as well as certain forms of knowledge (cf. [1]).

This concept suggests a hierarchical structure of practice → activity → action, where actions are embedded in activities and a practice is a compound set of activities. When modelling an activity or action, role features shall be added in order to allow modelling a natural type object to take different roles according to the activity system it is embedded in. In the case of enterprise collaboration, applying this concept will help overcoming the loss of adaptivity which is inherent to strictly formalized systems modelling practices. It enables the switching betweeen the formal systems perspective and that of a composite practice which is very much subject to flexibility, i.e. the capability to adapt to change.

The sharing and transfer of knowledge in enterprise collaboration, in addition, raises the issue of codification and management of knowledge. Only codified knowledge can be transferred between different organizational entities, as discussed in [6]. Codification of knowledge therefore has become a major research subject dealt with in the field of semantic computing and has fostered ongoing development work on semantic technologies.

Collaborative software development surely is one of the most knowledge intense practices of enterprise collaboration. And it is one of growing importance for as well as impact on globalizing business environments. The management of knowledge flows between different teams working together to develop software becomes a critical issue in particular with regard to adaptivity, which is the potential to adapt to changing business environments.

3 Enterprise collaboration: A composite systems approach

Interoperability in the general context of enterprise collaboration is the ability of a system to work with another system by making use of a common information base. The IEEE definition stresses explicitly the information flow aspect: Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged.

In the case of enterprise collaboration this means seamless integration across organizational boundaries between dispersed operational units of a number of organizational entities. The result are highly distributed system architectures. In particular, such systems need to be able to cope with change even after deployment during their run-time.

Major research into the relating interoperability issues (cf [2]) has shown that current systems architecture and engineering concepts fall short in combining interoperability and adaptivity of integrated systems for enterprise collaboration. In discussing these and putting it into the more general context of complexity, in [5] the characteristics of coupled inter-enterprise systems have been summarized as to their level of coupling as follows (exh. 1).

Exh. 1: Coupled systems - space of complexity

	Complexity	Interoperability	Adaptability
• Loosely coupled systems	low	low	high
• Composite systems	moderate	moderate	high
• Highly integrated systems	high	high	low

Based on the reflections offered there, we suggest denoting as composite systems those which combine the concept of component based systems with that of dynamic configuration of components as part of the architectural design. A composite system then consists of *components*, which represent activities within a practice, of *interfaces* by which the flow of information between components is realized, and their *configuration.*

According to [1] the concept of roles shall be applied to components giving a component its role within a specified context. Interfaces then will have to include information relating to roles in a meaningful sense.

Both components and interfaces of a composite system are considered as artifacts as discussed in [9]. There it is argued that in the case of novel or innovative problems the nature of these may not be fully specifiable once and in advance, but should leave room for corrective and adaptive modification even after implementation. Of course, this requires ensuring appropriate traceability [cf 11]. The artifact flow through and dependency control model described in [9] consequently calls for means to anchor instead of to fix interfaces. We follow this by taking up the suggestion of decomposing interfaces into kernel, shell and supporting services.

The configuration of a composite system is required to become subject to change before implementation as well as during its use. A given instantiation of a composite system can have to be modified in order to address changes of functional requirements in any phase, again be those corrective or adaptive. In particular adaptive modification will take into view changes in the observed characteristic – the activity pattern – of business activities that are supported by the system, rather than taking the perspective from the system design pattern toward the buiness avtivity [cf 13].

This different perspective is the central feature of the dynamic configuration as described in [5]. It aims at preparing a composite system for adaptation through modification of its structure. Such modification can comprise in particular the aggregation (integrate several components to a new one) and segmentation (split a component into several new ones) of components. Interfaces, taken as the means to realize the knowledge flow between components, will have to support that modification by making it propagating and traceable. Consequently, its them which must bear the information necessary to guide and control the changes which are applied to the composite system.

4 Advancing adaptivity

Whenever enterprise collaboration is being set up in a specific case, two things to be organized cannot be kept separated: interoperability, i.e. the exchange of knowledge between components, and collaborative work, i.e. the linking of activities of components. In general, this integration can be achieved by introducing managerial function. Supporting systems, however, most frequently are limited by addressing the interoperability part only. Their focus is on a pre-configured flow model the design of which is derived from established systems practices. Thus they tend to limit adaptivity, as changing practice is difficult to incorporate.

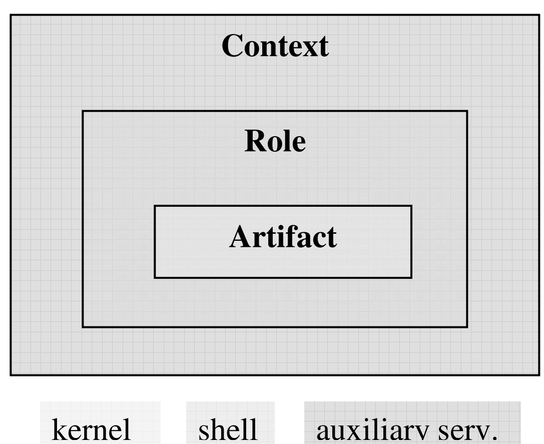
The considerations made in previous chapters suggest improving this shortfall by adding a composite practice perspective and supporting this by a composite systems approach. Central elements taken into view are the component interfaces within a component based system. Their functional restriction of serving as exchange vehicles only, transporting the output of one source component as input to another target component, can be mitigated by introducing role features. In addition, interfaces are to get enriched by incorporating contextual knowledge which permits the switching between the systemic and the composite practice perspective.

Role features help enabling the translation, abstraction, and pattern recognition of an output of one component by the target component. Contextual knowledge facilitates the filtering, authentication and knowledge protection of every output exchanged.

In a formal, component based system every component output is represented as an artifact, i.e. a formal object. In general, its first hand function is to serve as input to a target component, i.e. to entail a specified action. The second hand function is to support functionality of the activity system it is part of. Illustratively, since such component output will be used as input by a target component with source and target component being part of the composite system, it has to be considered that any interface is subject to the fulfillment of the component functionality – generating the specified action, but also to that of appropriately supporting the system it is embedded in – facilitating the activity system, and to its usability within the overall composite system supporting the composite practice. Hence interfaces that link components have to bear features beyond the pure formal function of the artifacts transported.

Our approach makes use of the tripartite interface definition by Kirova & Marlowe [9] and its segmentation into interface *kernel*, *shell*, and *auxiliary services* (Exh. 2). In brief, the kernel bears the core functional content being characteristically fixed. It represents the artifact to be exchanged between components.

Exh. 2: Interface decomposition



The kernel is extended by a number of aspects of embedding and perspective which provide information about the role it has within the system. They are useful for abstraction, translation and interpretation of the kernel artifact. Auxiliary services comprise information about the context in which the kernel artifact is embedded. These can be used for authentication or intellectual property purposes and enable filtering of content according to its relevance for the functional characteristics of the target component. Both shell and auxiliary services enable evolutionary adaptation in the case of changing practice. The kernel artifact thus is embedded within the interface as shown in the illustration above.

In a first step toward realization of this approach, in the subsequent chapter we describe how it applies to the collaborative development of software, an area of growing relevance in the information technology sector.

5 Implications for collaborative software development

Software development these days increasingly becomes distributed and cross-organizational. It is subject to collaboration of locally spread and timely separated teams of developers. Collaborative development thus requires appropriate organization of knowledge flow and exchange between dispersed developers. Moreover, in the case of complex applications with life-cycles of many years, adaptivity becomes a major challenge. The business environment is supposed to change constantly, which in turn impacts related practices and raises the issue of adapting an existing application which was designed through a process of collaboration. The knowledge flow organization must take care of this need and prepare for appropriate support to ongoing adaptation.

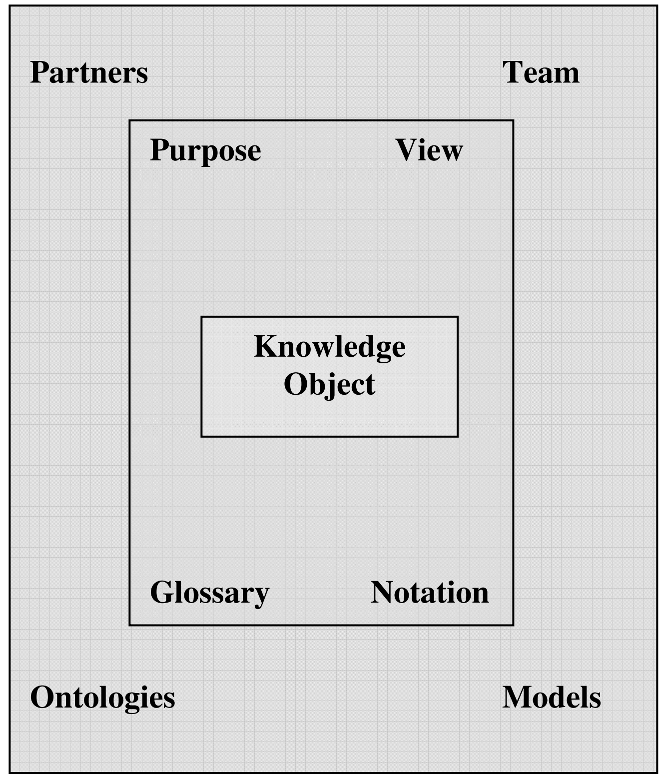
Complex systems being composed of a manyfold of components linked by interfaces may be hard to modify if their formal structure is tight. While components, representing the functional core of such systems, typically are realized based on formal engineering principles, the interfaces are the elements which can become bearer of those features that facilitate adaptivity. The issue therefore is to analyze suitable such features and their localization within interfaces.

As discussed in the previous chapters, our approach is to segment interfaces into *kernel*, *shell*, and *auxiliary services*. The kernel represents the core knowledge content, i.e. the information to be exchanged between components, and is characteristically fixed in its syntax and semantics. We call it the knowledge object transported by the interface.

The kernel is extended by a number of aspects that are useful for abstraction, translation and interpretation of the kernel knowledge and thus enable evolutionary adaptation in the case of changing practice. Our initial analysis of suitable such aspects leads to the denomination of four of these: *glossary*, providing semantic support by clarifying terms used, and *notation*, providing information regarding syntax and structure employed; further, *purpose* and *view*, which both help with abstraction and facilitate switching between the systemic and the collaborative practice perspective.

Auxiliary services comprise information which is relevant for authentication or intellectual property purposes, like *partners* and *team*; they also include references to domain specific data bases, like *ontologies*, and identification of process *models* which can help with model transformation.

The resulting interface decomposition is illustrated in exhibit 3 below.



Exh. 3: Knowledge interface decomposition:

kernel shell auxiliarv serv.

We are taking this frame as a first step towards localizing aspects of knowledge management to component interfaces. It will help separating business strategy and knowledge management functions into component-local, partner-local, domain-local and system-global facets.

We expect this approach to deliver significant contributions to the adaptivity issue relating to collaborative development of complex software systems. In particular, we consider it a major step beyond traditional knowledge management as it is capable of exceeding the separation of knowledge content, being more or less fixed in knowledge objects, from knowledge use, being rather an intangible activity of applying information to function [cf 8].

6 Conclusions and further work

It is worthwhile to note that the described approach does not only ease the reconfiguration of an enterprise collaboration system along its life-cycle of application. In fact, being implemented once, it can be employed as a blueprint for the development of other enterprise collaboration systems. Artifact flow through as well as dependency control [cf 9] will be reconstructable offering a high level of adaptability. Reuse hence becomes an attractive option as instantiation specifics like domain, model, notation, partners etc. are decomposed from the kernel knowledge objects and localized in the shell resp. services segment where they can be substituted easily.

Future work therefore shall be dedicated to further investigation of the benefits of localizing aspects of knowledge management to component interfaces within composite systems. We intend to undertake deeper research into the appropriate decomposition of interfaces according to the field of application and the determination of components and interfaces that will benefit from more specific localization of knowledge management aspects.

References

[1] Allert, H.; Richter, Ch.: *Practices, Systems, and Context Working as Core Concepts in Modeling Socio-Technical Systems.* Proc. of the 5th International Workshop on Philosophy and Informatics, Kaiserslautern, April 2008 http://ceur-ws.org/Vol-332/

[2] ATHENA-IP, *Model-Driven and Adaptive Interoperability Architectures - Model-driven and Adaptable Interoperability Framework.* Public deliverable A6.3, July 2006, http://www.athena-ip.org

[3] European Commission, *Enterprise Interoperability – A Concerted Research Roadmap for Shaping Business Networking in the Knowledge-based Economy.* Luxembourg, 2006

[4] INTEROP-NoE: *Enterprise Interoperability-Framework and knowledge corpus - Advanced report.* Public deliverable DI.2, December 2006, http://interop-vlab.eu

[5] Jastroch, N.: *Towards Adaptive Interenterprise Systems.* Proceedings of the 14th International Conference on Concurrent Enterprising ICE2008, 931 – 938, Lisbon, June 2008

[6] Jastroch, N.: *Wissensmanagement - Darstellung und Transfer von Wissen - Potenziale und Grenzen*. GI Lecture Notes in Informatics, Vol. P-28. Bonn, 2003 and http://ceur-ws.org/Vol-85/

[7] Jastroch, N.: *Adaptive Interenterprise Knowledge Management Systems.* Proc. of the 12th World Multi-Conference on Systemics, Cybernetics and Informatics, June 2008, vol. VII, 230-235, June 2008.

[8] N. Jastroch, S. Lehr, F. Lettice, A. Leslie: *CIKM: Creation of Innovation through Knowledge Management*. Proc. of the eSmart2002 Conference, University of Salford, Manchester 2002

[9] V. Kirova, T. J. Marlowe: *Addressing Change in Collaborative Software Development through Integrated Artifact Flow and Dependence Analysis.* Proc. of the 21st International Conference on Software and Systems Engineering and their Applications, December 2008

[10] Lippe, S.; Greiner, U.; Barros, A.: “A Survey on State of the Art to Facilitate Modelling of Cross-Organisational Business Processes”, p. 7-22, in: *XML4BPM 2005, Proceedings of the 2nd GI Workshop XML4BPM -- XML Interchange Formats for Business Process Management at 11th GI Conference BTW 2005*, M. Nüttgens and J. Mendling (Eds.). Karlsruhe, March 2005, http://wi.wu-wien.ac.at/home/mendling/XML4BPM2005/xml4bpm-2005-proceedings.pdf

[11] T. Marlowe, V. Kirova**:** *Addressing Change in Collaborative Software Development: Process and Product Agility and Automated Traceability.* Proc. of the 12th World Multi-Conference on Systemics, Cybernetics and Informatics, June 2008, 209-215.

[12] Object Management Group*: Business Process Modeling Notation (BPMN)*, Version 1.1. January 2008, http://www.omg.org/spec/BPMN/1.1/PDF

[13] Pieters, C.P.: *Complex Systems and Patterns.* Proc. of the 12th World Multi-Conference on Systemics, Cybernetics and Informatics, June 2008, vol. VII, 268-275, June 2008

[14] Sitek, P.; Seifert, M., Thoben, K.-D.; Rapaccini, M.; Visintin, F.: *Quality Management in Collaborative Networks.* Proceedings of the 14th International Conference on Concurrent Enterprising ICE2008, 203 – 210, Lisbon, June 2008