Modelling inter-organizational processes with process model fragments

Frank Lindert

Wolfgang Deiters

Fraunhofer Institute Software and Systems Engineering

Joseph-von-Fraunhofer-Str. 20

44227 Dortmund

{deiters,lindert}@do.isst.fhg.de

Abstract

Today most of the workflow management approaches focus on centralized business processes that are being carried out within one orgaization.However, business practice demand that more and more inter-organizational processes have to be considered. Therefore we developed an approach for a decentralized process management basing on the concept of process model fragments. The goal of the approach is especially to consider the autonomicy of the organizations that participate in inter-organizational processes. A process model fragment is related to an organization and describes the part of the process the organization (resp. its human agents) is responsible for. By interconnecting process model fragments the information exchange between the organizations is described. The concept that is proposed in this paper is generic an can be adopted to many workflow management approaches.

1. Introduction

Within the last years workflow management has become a technology that is being more and more used in order to support business processes. Based on an enactable description of the processes, so called workflow model, the processes are being supported by workflow systems that usually interpret the processes, assign to the various people involved in the process the tasks they have to perform, and, provide the tools and objects that are needed to perform the tasks. Thus workflow systems drive and monitor the business processes.

Various approaches have been developed for managing business processes. What is common to these approaches is that they nearly all focus on the management of what we call centralized business processes, processes that run in one organization mainly at one geographical location. However, there are several arguments showing that a centralized approach turns out to be inappropriate for managing many of the larger scale industrial processes:

- Processes run across different geographical locations
  Various processes are performed in companies that are spread over different geographical locations. Coordinating these processes from one central point makes the complete workflow management application dependent on the connections between the organizations. A breakdown of the central system or the telecommunication lines between the locations would turn down the complete process in all locations.
- Processes run across different organizational units or even different cooperating partners
  In different application areas, such as for example in the automotive industry, business processes are spread over different organizations or even completely independent companies that cooperate in order to produce a common product or service. These partners operate independent from each other on their own „local“ process parts, furthermore they have agreed on process interfaces on a contractual basis. Since (partial) autonomicy is

an important goal for the different partners a central process description would not be accepted among them. Furthermore, autonomicy is very important in virtual organizations and in the area of e-commerce especially in business-to business scenarios.

- Heterogeneous workflow systems are being used
  Different partners that work together in a common process partially have their individual workflow system installed and running. Thus an approach that enables an interoperation of heterogeneous workflow applications (i.e. applications coming from different vendors) is needed.

Deriving from these requirements we developed a decentralized process management approach. This approach starts from the assumption not to concentrate the process information of the various cooperating partners in one model but rather to develop individual process models for the different cooperating partners. We will call these individual process models process model fragments in the following. The fragments are the process definitions of the (partially) autonomously operating partners that have to be fitted with interfaces in order to arrange the overall coordination.

Within this paper we are going to briefly sketch the concept of process model fragments and the interconnection of these fragments towards decentralized process models. This generic concept is part of the concept of fractal process management which is described in [Lind99]. Beside the process modelling the whole concept considers also organizational aspects, a fractal process life cycle and a system architecture which considers the components of a fractal process management including security issues.

2. Process model fragments

The previous section gave a brief motivation for the development of a decentralized process management approach dealing with inter-organizational processes. To support these processes we need a concept capable to deal with organizational frontiers, autonomous organizations[1] that participate in the process, and, heterogeneous environments. The main idea behind that is to extend process management towards managing cooperating partners each of which staying independent from the others as much as possible (e.g. in virtual communities). Thus the aspects of autonomy of partners and coordination among partners have to be balanced well. The notion of autonomy that we use here is derived from [Warn95] addressing vitual, self-contained, self-organizing units that pursue own goals.

The basic idea of our approach is to build up process fragments that are as independent as possible from each other. A process fragment has interfaces that are used to connect the fragments together. These connected process fragments represent the inter-organizational process. Each process fragment is related to an organizational unit that is responsible for the fragment. The fragment contains all activities the organizational unit has to perform during enaction of the process. The organizational unit can autonomously describe the fragment and enact the fragment.

There exist several approaches that focus on the subdivision of a given process into pieces. For example, in the Exotica/FMQM project [AMGA95] the subdivision of a process is based on a formerly completely described process model. The basic idea there is to build sets of activities that have to be performed at the same “place” (place is the server, the responsible users are connected with). Each set of activities is transferred to the appropriate server. In [GrGr95] a process model is subdivided into pieces that afterwards can be extended. In [NSH98] the subdivision is based on the hierarchical decomposition of a process. An activity of a process can be refined by a another process. During enaction each process is enacted on one of the available enaction servers. [BaDa97] focuses on the distribution of the enaction of a process to optimize the available resources. [GAHM98] introduces a collaborative editing approach to describe distributed software processes. All these approaches do not consider the “autonomy” of the organization that participate in the process. [LuWh99] use gateways to support cross-organizational workflow management. This approach considers the privacy of the organizations which is similar to the autonomy we want to support but they focus on the system architecture but does not consider synchronization aspects, complex documents types and the description of interfaces in the process model.

To support inter-organizational processes the process fragment approach has to consider the following requirements:

1. Each participating organizational unit has to be able to model, analyse and enact its part of the process autonomously. Then the organizations can hide their processes from other organizations. In virtual environments the

1. In the following we use the term organization to describe both, organizational units in large organizations and organizations as a whole.

internal business processes are one key competence of the organizations they want to preserve from the other organizations.[1]

2. Each process fragment interacts with other fragments. Like a process a process fragment needs a set of input documents it works on and produces a set of output documents. The input documents are provided by other process fragments and the output documents are used by other process fragments. However the passing of inputs and outputs between fragments is not restricted towards a „procedure interface“. We rather have to distinguish between start-up inputs and termination outputs, i.e. documents that are necessary inputs for starting a fragments and documents that are produced as result of the fragment and intermediate inputs / outputs, documents that are being exchanged during the fragments operation.
3. Each participating organization uses its own process management approach. Therefore the process fragment approach must not focus on one specific process management approach. It should be generic to be implemented in various existing process management approaches.

.

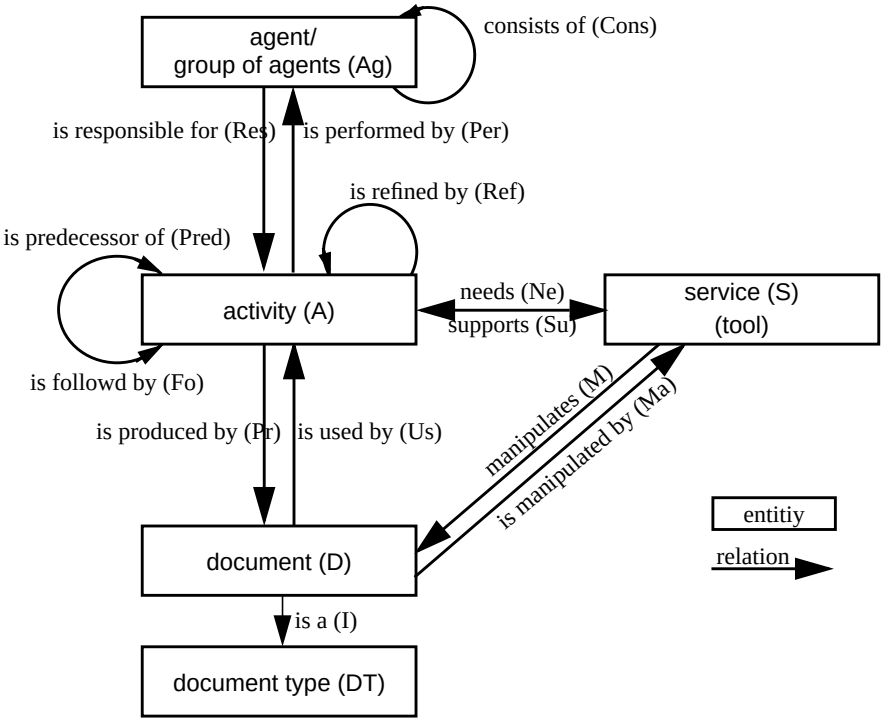

figure 1: General metamodel for process models

To fulfil theses requirements we introduce the concept of process fragments. They are described by extended process models. The extension describes on the one hand the interface, the process fragment uses to interact with other fragments and on the other hand provides additional information for the analysis and enaction of the process fragments. The interface description is public. It is used by other organizations to connect process fragments. The sequence of activities of a process fragment is private to the organization. It is not published to other fragments.

The development of the approach is based on the general metamodel in figure 1. This metamodel has been derived from the models presented in [Deit93] and [Jabl95]. Additionally we consider services and agents. So opposite to the general process description languages like WPDL [WfMC97], PSL [MSID98], PIF [LGJM96], CPR [PeCa97] this metamodel does not describe all aspects of a process model, e.g. transition rules or complex organizational structures of the agents.

The following process model fragment definition is based on the view based FUNSOFT [DeGr98] approach and the aspect based approach MOBILE [JB96]. These approaches subdivide the description of a process model into

1. The assumption for the model development is as follows: The cooperating partners meet in an initial design phase and thereby agree upon the overall process model goal, the main interfaces, etc. Then, the model definition of the individual fragments takes place autonomously. Interfaces are checked and made fitting. Thus, the meaningfulness of the whole process model is achieved by the process model development lifecycle (see [Lind99]).

different views (aspects). Each view focuses on one aspect of the description. Examples of views are structural view (functional aspect, behavioural aspect, information aspect), service view (operational aspect) or the project management view (organizational aspect).

We define a process model as follows:

Definition 1: Process model[1]: $PM = (StV, SV, IV, OV)$ with

$StV = (A, Pred, Fo, Ref, Pr, Us)$ is the structural view,

$SV = (S, Ne, Su, M, Ma)$ is the service view,

$IV = (D, DT, I)$ is the information view and

$OV = (Ag, Cons, Per, Res)$ is the organizational view.

The subdivision of the process into process fragments that we focus on within this paper is related to the control flow and the data flow. For this we have extended the structural view in order to describe the interfaces of a process fragment. The control flow describes the sequence of activities and the data flow describes the documents the activities consume and produce. In an inter-organizational process we have inter-fragment control flow and inter-fragment data flow. Both have to be described in the process model fragment.

To describe the inter-fragment control flow we introduce a new entity in the metamodel called events. This is necessary because the control flow ((Fo) and (Pred) in figure 1 are relations between activities. Since we want to hide the activities of a process fragment a description serving as a substitution is needed. An event can be produced or consumed by activities. In figure 2 the description of inter-fragment control flow is shown in an example.

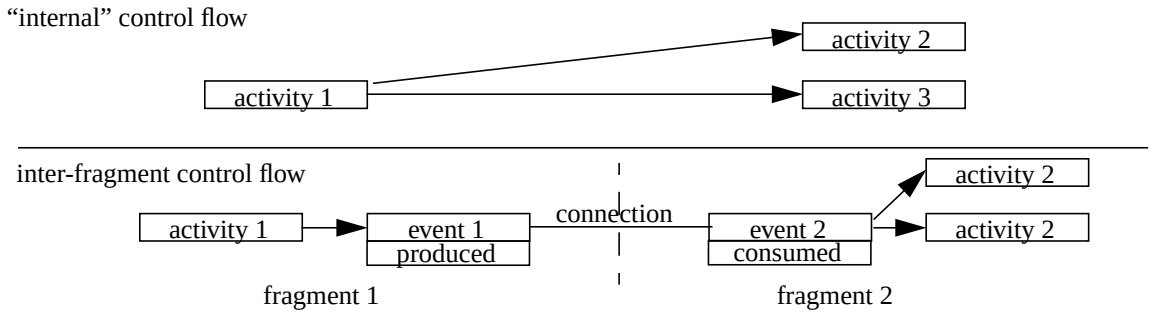

figure 2: description of inter-fragment control flow with events

To integrate the events in the metamodel we define a new entity and two new relations between the entity "event" and the entity "activity".

To describe the inter-fragment data flow no new entities are needed. There is rather the task to handle the proper transfer of documents (incl. document type mapping) between the different fragments[2]. Therefore, the documents have to be described in detail in the interface description in order to ensure that the receiver of a documents gets the document he expected. The documents that are passed between activities in the process fragment have no influence on the inter fragment data flow. The main problem is to describe the structure and the semantics of a document on both sides. The sender has to describe what documents he will provide and the receiver has to describe what documents he needs. In several application domains there exists standards that describe documents, e.g. CDIF, STEP, EDIFACT. These standards describe the structure and the semantics of documents. In other application domains no standards exist.

In order to describe the documents in the inter-fragment data flow we introduce an external document type description language (DTDL) which is based on the OMG IDL. We removed the method descriptions of IDL and inserted a mechanism to consider the existing standards. Due to the limited space in this paper we only give a short example

1. The abbreviations in the definition can be taken from figure 1.
2. Of course, document types also have to be described within the fragments. However, this is being done in the languages of the approaches that model the resp. fragments and, thus, no subject of discussion for the scope of this paper.

of a DTDL document description:

```
documenttype order: business_transaction{ // Inheritence

   typedef struct {
      EDIFACT::Bill
      IMC:vcard            customer;
      Date                 delivery_data<annotation2>;
   } ordering_documents;

   ...
   typedef sequence <EDIFACT::Delivery> single_deliveries
   ...
   attribut ordering_documents          MyBill;
   ...
}
```

Reference to a standard DT

figure 3: Example of a document type definition with DTDL

Similar to the document type problem there are other aspects to be described in order to define what a fragment expects and what a fragment offers:

- Number of documents.
  Whenever more than one document, e.g. a set of documents, has to be transferred, a minimum and a maximum number of documents has to be described. Some approaches use containers of documents to describe the data flow, e.g. the Petri-net based FUNSOFT [DeGr98] approach. Other approaches describe each document with one entity in the process model like LinkWorks.
- Copy or move.
  In some cases it is necessary to specify that a copy of a document has to be transferred and the original document remains in the sending fragment.
- Optional or required.
  Sometimes not all documents are expected. Some may be optional, e.g. for a credit request in case of a large credit amount some optional reports have to be transferred.
- Fragment identifying document.
  Sometimes it is necessary to determine the receiving fragment (see below). In this case a fragment identifying document can help to do this. For example, an application can identify the appropriate process fragment with the application number.
- Informal annotation.
  Due to the fact that the semantics of a document cannot be described any time we consider an informal, multimedia annotation to describe further information of a document.

Summing up the general metamodel has to be extended by new entities, relations and by attributes in order to cope with process fragments. The extended metamodel is shown in figure 4:

.

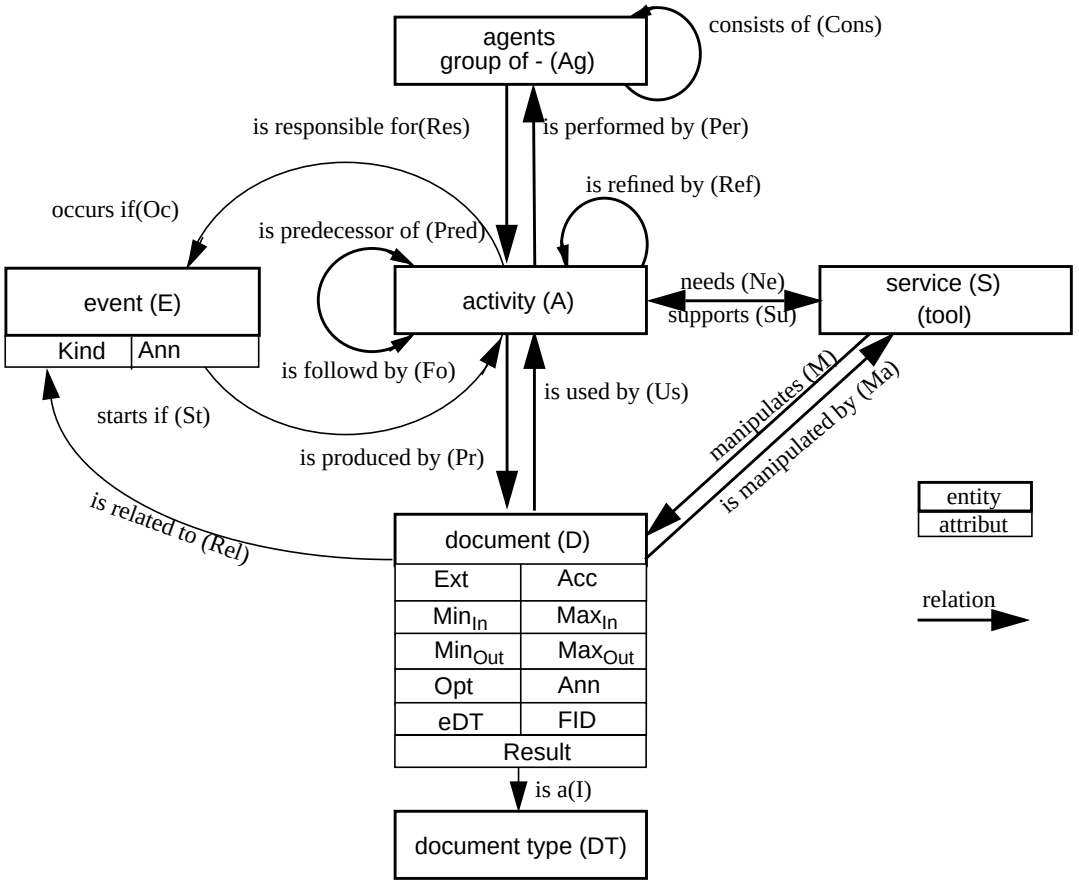

figure 4: Extended general metamodel

The process model fragment is defined as follows:

Definition 2: Process model fragment: $PMF_{OE} = (PM', ER, IV, Con)$ with

$PM' = (PV', SV, DV, OV)$ is a process model that consists of an extended process view PV' (extended by the relations (St) and (Oc)), a service view SV, a data view DV and a organizational view OV,

$IV = (E, Kind, Rel, Ext, Acc, Min_{In}, Max_{In}, Min_{Out}, Max_{Out}, Opt, Ann, eDT, Hor)$ is the new interface view that describes the interface of the process fragment[1], ER is the external representation of the process model fragment that describes the process model PM' in a simplified representation. With this external representation the organization can decide how much details about the internal activities are to be published. Furthermore during enaction the state of the process fragment can be queried. The external representation is completely independent from the process model PM'. Con is a structured set of connection points of the process model fragment. These connection points are used to connect the process model fragments. Each connection point consists of a set of events and external documents.

An inter-organizational process consists of several process fragments. So the description of the whole process consists of a set of process model fragments that have to be connected in order to describe the control flow and data flow between the fragments. Such a complete description does not need to exist at process model build time because sometimes not all fragment information (or even the fragments at all) are known at that point in time. Consider e.g. the complex process of developing and producing a new car. Such a process lasts a long time, sometimes several years. Even the organizations that participate in the production of the car are not known all at the beginning

1. The attributes FID (fragment identifying document) an Result (specifies a special kind of output document) is part of Hor, the description of horizontal connection points.

of the process. So is has to be possible to complete the process description even during its enaction. Therefore it is possible to leave interfaces unconnected at the start of the enaction. During run-time the process engine has to consider the interfaces and to notify the process modeller in case of an unconnected interface. Due to the fact that a description of the whole process often does not exist in advance only limited possibilities for analysis exist, e.g. deadlocks that are spread over several process fragments can't be computed[1]. This can't be avoided because otherwise the autonomy of the participants will be violated. Taking into account this autonomy requirement partially consistency and correctness has to be assured by introducing organizational rules in the process management lifecycle [Lind99].

There exist two major types of connections between interfaces of process fragments. On the one hand we consider vertical connections. This type of connection allows the hierarchical refinement of activities of a so called father process fragment by child fragments. It is a connection between an activity in the father fragment and a child fragment. This type of connection is similar to the approach in [NSH98] and to the hierarchical connection scenario of the WfMC [WfMC96]. We extend these approaches by the multi-vertical connections that allows one process fragment to be connected by more than one vertical connection (cf. figure 5, in this example we do not use events to describe the inter-fragment control flow)

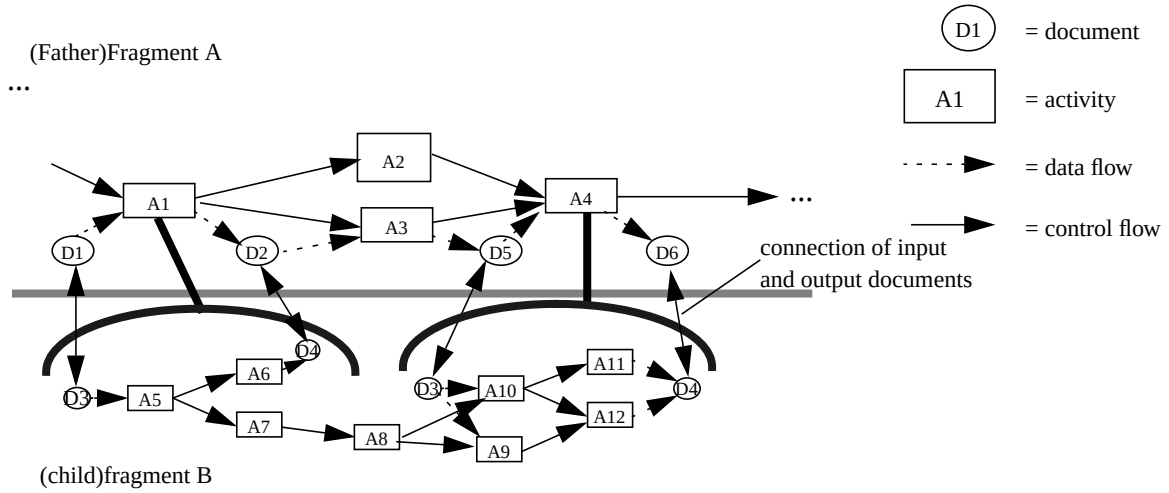

figure 5: Multiple vertical connection

The other type of connection is the horizontal connection. This type connects documents and events. By that the passing over of documents and events can be described. This kind of connection is similar to the discrete connection of the WfMC. In figure 6 an example of a horizontal description is shown.

1. In [Aals98] the verification of interorganizational workflow is discussed. This approach is based on petri nets. All process fragments have to be described by petri nets and the verification uses the whole petri nets so the autonomy requirement is not assured.

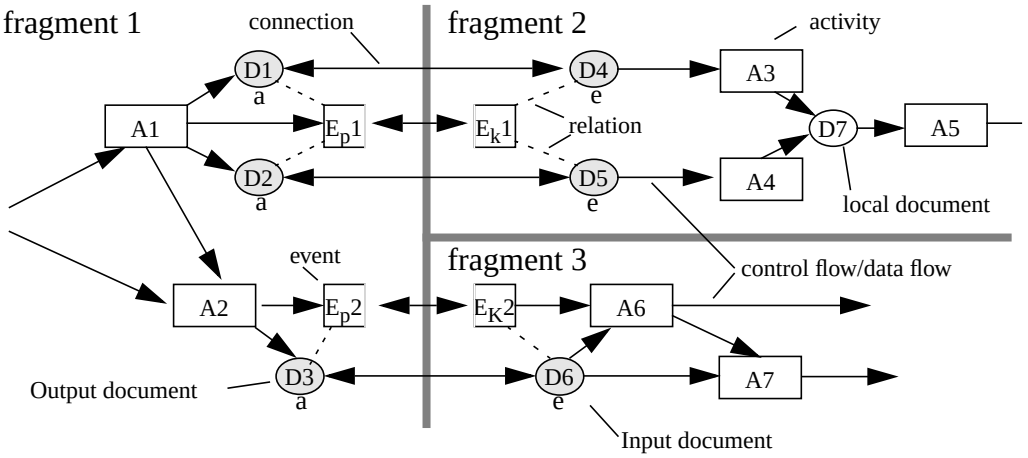

figure 6: Horizontal connection of three fragments

In the process model fragment we define the description of offered and expected connections using the relation Con. This relation is a set of alternative connections a process model fragment offers resp. expects. Due to the limited space in this paper we cannot describe all features of the connection process in more detail. However, the connection of the process model fragments is a negotiation process. The organizations that want to connect their process models have to ensure that the corresponding interfaces fit together. This negotiation process is supported by the attributes we introduced in the interface description. Based on these attributes we defined a set of consistency rules that have to be considered in order to specify valid connections. Each organization stores the connections locally. This is necessary to fulfil the autonomy requirements.

With the process model fragments each organization can autonomously describe its part of a inter-organizational process independent from the other participating organizations. Only the interface descriptions are published. So we fulfil the requirements we listed at the beginning of this section. The general metamodel is independent from any existing process model approaches, thus it becomes possible to implement it in different workflow systems.

3. Summary

With the concept of process model fragments inter-organizational processes can be described. The concept considers the autonomicy of the participating organizations. In this paper we described the modelling of inter-organizational processes. Furthermore we developed an architecture that extends existing process management systems in order to model, analyse and enact inter-organizational processes. First components of the architecture have been developed and implemented in the project VORTEL [BDFL96]. In this project we integrated the CORMAN System based on the FUNSOFT approach and the Systems LinkWorks and FlowMark focusing on the enaction of workflows. [Henn98] focused on an architecture for supporting a decentralized process modelling. He introduced various traders (based on the CORBA technology) that allow to find process models of cooperating partners that can be matched, that allow a negotiation on the interface, and, that allow to match object types of the interface of different fragments.

References

[Aals98] Aalst, W.M.P., „Interorganizational Workflows“, in Proc. of PROLAMAT’98, p. 21-43, Trento, 1998

[AMGA95] Alonso, G., Mohan, C., Günthör, R., Agrawal, D., El Abbadi, A., Kamath, M., “Exotica/ FMQM: A Persistent Message Based Architecture for Distributed Workflow Management”, in Proc. IFIP WG 8.1 Working Conference on Information Systems for Decentralized Organizations, Trondheim, 1995

[BaDa97] Bauer, T., Dadam, P., “A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration”, in Proc. Second IFCIS Conf. on Cooperative Information Systems, Charleston, South California, 1997

[BDFLS96] M. Böhm, W. Deiters, M. Friedrich, F. Lindert; “Workflow Management as Teleservice”, in Computer Networks and ISDN Systems, Vol. 28, No. 4, S. 1961-1969, 1996

[Deit93] Deiters W., „A View Based Approach to Software Process Management“, University of Berlin, 1993

[DeGr98] Deiters, W., Gruhn, V., “Process Management in Practice - Applying the FUNSOFT Net Approach to Large-Scale Processes”, in Automated Software Engineering 5, S. 7 - 26, Kluwer Automated Press, 1998

[GAHM98] Grundy, J.C., Apperley M.D., Hosking, J.G., Mugridge, W.B., “A Decentralized Architecture for Software Process Modeling and Enactment”, in IEEE Internet Computing, September October, S. 53-62, 1998

[GrGr95] Graw, G. und Gruhn, V.; “Distributed Modeling and Distributed Enaction of Business Processes”, In Software Engineering ESEC ‘95, 5th European Software Engineering Conference, Sitges, Lecture Notes in Computer Sience 989, Springer Verlag, S. 8-27, 1995

[Henn98] Henning, D., “Eine Systemarchitektur für dezentrales Prozeßmanagement in autonomen Organisationseinheiten”, Diplomarbeit Universität Dortmund, 1998

[JB96] Jablonski S., Bussler C., Workflow Management. Modelling Concepts, Architecture and Implementation, International Thomson, 1996

[LGJM96] Lee, J., Gruninger, M., Jin, Y., Malone, T., Tate, A., Yost, G. et al, “The Process Interchange Format and Framework”, Working Group, Version 1.1, 1996

[LuWh99] Ludwig, H., Whittingham, K., „Virtual Enterprise Co-ordinator - Agreement-driven Gateways for Cross-Organisational Workflow Management“, Proc. of the Int. Joint Conference on Work Activities Coordination and Collaboration (WACC’99), San Francisco, 1999

[Lind99] Lindert, F., „Fraktales Prozeßmanagement“ (in german), Phd. Thesis, Technical University of Berlin, 1999

[MSID98] Manufacturing Systems Integration Division, “The PSL Home Page”, http://www.mel.nist.gov/psl/, 1998

[NSH98] Neeb, J., Schaumberger, R., Schuster, H., „Using distributed Object Middleware to implement scalable Workflow Management Systems“, in Ozsu, T., Dogac, A., Ulusoy, O. (Eds.), Proc of the 3rd Biennial World Conf. on Integrated Design and Process technology, Berlin, 1998

[PeCa97] Peace, R. A., Carrico, T., „The JTF ATD Core Plan Representation: A Progress Report“, Proc. of the AAAI Spring Symposium on Ontological Engineering, 1997

[Warn95] Warnecke, H.-J., „Die fraktale Fabrik - Revolution in der Unternehmenskultur“ (in German), Springer, Heidelberg, 1995

[WfMC96c] Workflow Management Coalition, “Workflow Standard - Interoperability, Abstract Specification”, Document Number WfMC TC-1012, Version 1.0, 1996

[WfMC97] Workflow Management Coalition, “Interface I: Process Definition Interchange”, Document Number WfMC TC-1016, Draft 6.94, 1997