Modelling and Handling of Dataflows in Crossorganizational Processes

Markus Bon
AG DBIS
Dept. of Computer Science
University of Kaiserslautern
PO-Box 3049
D-67653 Kaiserslautern
bon@informatik.uni-kl.de

Norbert Ritter
VSIS Group
Dept. of Informatics
University of Hamburg
Vogt-Kölln-Strasse 30
D-22527 Hamburg
ritter@informatik.uni-hamburg.de

Hans-Peter Steiert
Research &Technology
RIC/ED
DaimlerChrysler AG
PO-Box 2360
D-89013 Ulm
hans-peter.steiert@daimlerchrysler.com

Abstract: Nowadays, workflow management is a well known domain, at least if only the possibilities of single workflow management systems (wfms) are concerned. In contrast, an automated control of dependencies between processes which are controlled by heterogenous wfms is widely unexplored. Nevertheless, this kind of control is extremly desirable for cross-enterprise processes, as a great decrease of effort can be expected. Therefore, this paper performs a further step in exploring this field by describing a first approach for the automated handling of dataflows between heterogenous workflows. The basic aspects, which are relevant for the development of a integration component, will be discussed. Furthermore, a first architecture based on existing EAI-technology will be outlined.

1 Motivation

Processes existing in big enterprises are numerous and complex. As new products shall be developed in as short time as possible, the development not only takes place at the enterprise itself, but also at many smaller supplying firms exploiting special know-how. Especially for controlling such complex, cross-organizational processes workflow management promises a high potential.

In the COW-project (cross-organizational workflows [KRS01]), first research towards the description of island-spanning processes and support of global flow of control has been performed. Here, “island” refers to the collection of systems used by a company, for example workflow management systems (wfms) and product data management systems (pdms), the existing process and workflow types, and finally the available resources like applications and organizational units. Obviously, island-spanning dataflow plays an important role in this context [BHR01, BHR02].

For reasonable cooperation it is essential to share data between the participating groups. As an example, consider engineers constructing a new automotive vehicle. To examine the accuracy of the fittings, the CAD geometries have to be exchanged between them. After this, a kind of digital mock up (dmu) may be performed. Our goal is the automated support of such island-spanning data flows.

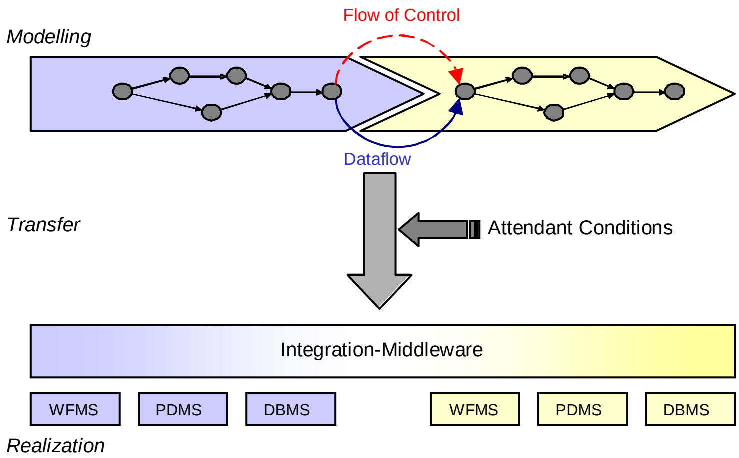

Figure 1: Modelling and Realization of Dataflow

To reach this goal, we have to deal with the following problems:

- The modelling of island-spanning processes (Figure 1, top) must be supported to allow the description of data flow dependencies (DFD) between islands along with their associated attributes. Therefore, we have to identify categories of DFDs.
- For each category different kinds of realization exist depending on the attendant conditions. To examine a suitable solution for a given collection of systems and according organizational instances requires to describe these attendant conditions. Furthermore, implementation guidelines have to be developed.
- At last, a middleware offering an integrated view to enterprise-spanning workflows is needed. In our case, this means that the dataflow concerning crossorganizational processes is processed in an automated way (Figure 1, buttom). As we belief, it is very important to minimize the changes of the originally given infrastructure.

In this paper, we will briefly discuss the essential aspects of island-spanning dataflow. Furthermore, we will derive a set of categories, which are called “integration patterns”. After this, we will discuss, why different possibilities for the realization of such an integration pattern have to be considered, as this realization strongly depends on the given infrastructure. Further on, we will show how existing wfms- and eai-tools enhanced by additional components can be used to build the required middleware. At last, we will sum up our results and give an outlook to our future plans.

2 Classification Criteria

Obviously, to define an island-spanning dataflow requires to identify criteria describing the properties of the dataflow. For this, we have to answer the following questions:

- In which form is the data administrated before the start of the flow?
- What effects has the dataflow on source and target?
- How are ownership and possession effected?
- In which form is the data offered and how can it be accessed at the target side?

For a first examination, we use a simplified scenario. We assume that only two islands are connected by a DFD. At each island resides a wfms to support an automated execution of local processes. Especially, these wfmss handle those workflows acting as source and as target for global DFDs. Additionally, there are applications at each island performing the actual process-related work. Usually, the data used during process execution is stored persistently and therefore, some kind of data-storage (DS) is available at each island, too. Furthermore, a component for the integration of island-spanning workflows exists as superordinated tier ("WFI"). The WFI represents all mechanisms necessary to monitor and resolve defined DFDs. Further on, it is responsible for the transfer and the provision of data in an adequate way. Therefore, it offers all the functionality which becomes necessary besides the islands' own possibilities.

All the discussion given in the following refers to this scenario. We assume that a process-step is activated by the source-wfms $WFMS_S$ and processed by application A_S, producing data which has to be transfered from $island_1$ to $island_2$ according to a defined DFD. We call this data "cooperation data".

2.1 Data Management

If cooperation data is stored in a persistent way after its creation, we call it "DS-managed". In this case, access to the data-storage DS_S becomes necessary during the processing of the DFD.

In contrast, if the cooperation data is not application data (like in the first case), but workflow-relevant data, which the application hands over to the wfms for further management, we are speaking of "wfms-managed" data. This proceeding is supported by most current wfms.

2.2 Effects of Dataflow

Now that we have described the management of data on the source side, we will have a closer look at the handling of global dataflows. Taking in account the dataflow's effect to the source island, we distinguish between two kinds of dataflow:

- *source-conserving*: the data is still available on the source island after completing the dataflow,
- *source-consumptive*: the data is removed from the source island.

If we look at possible combinations of data-managers (DS or WFMS) and effects to the source, the following forms for transfering cooperation data emerge:

1. *Replication*: source-conserving from DS_S to DS_T,
2. *Copy*: source-conserving from DS_S to $WfMS_T$,
3. *Pass*: source-consumptive from DS_S to DS_T,
4. *Release*: source-consumptive from DS_S to $WfMS_T$,
5. *Spread*: source-conserving from $WfMS_S$ to $WfMS_T$,
6. *Deposit*: source-conserving from $WfMS_S$ to DS_T,
7. *Travel*: source-consumptive from $WfMS_S$ to $WfMS_T$,
8. *Deliver*: source-consumptive from $WfMS_S$ to DS_T.

2.3 Ownership and Possession

One more distinctive feature is the question of ownership and possession of data. Possessor of data is everyone being able to access it in any form. In contrast, the owner has the contol, how and by whom the data may be processed. If multiple versions are available, he has to decide which one is valid. Furthermore, he is responsiple for the validity and the consistency of the data. Handing over data does not necessarily imply the loss of ownership immediately. Therefore, it is possible to be the owner of certain data without possessing it for a while. Table 1 shows the situations which may be observed during the execution of dataflows. Here, ‘O’ means owner, ‘P’ stands for possessor.

	Before		After		
	Island 1	Island 2	Island 1	Island 2	Description
1	O,P	-	O,P	P	copy (C)
2	O,P	-	P	O,P	copy with shift of ownership (CSO)
3	O,P	-	O,P	O,P	copy with grant of ownership (CGO)
4	O,P	-	O	P	transfer (T)
5	O,P	-	-	O,P	transfer with shift of ownership (TSO)
6	P	O	-	O,P	return (R)
7	P	O	P	O,P	return with retention of a copy (RRC)

Table 1 : possible combinations of ownership and possession

2.4 Supply-Mode

We can think of two possible modes to supply the target island with the cooperation data. *Materialized* means that the cooperation data is provided physically at the target island for local access. Naturally, we have to consider that especially data used in the CAX-domain is huge in size and a physical transfer may be expensive. Therefore, materialized supply only makes sense, if the cooperation data is really accessed at the target island.

In contrast, the mode *referenced* provides only the supply of a reference. If data is effectively accessed this reference can be used to get the physical data on demand. Thus, the possibility of parameterizing the request exists, too. Therefore, only the data really needed can be selected and transfered. Obviously, this makes sense, if the set of cooperation data cannot completely be specified in advance.

2.5 Modelling Cross-Enterprise Dataflows

After the analysis of essential characteristics of island-spanning dataflow in sections 2.1 - 2.4, we now consider reasonable combinations. Afterwards, we identify the categories of dataflow a middleware being used as realization has to support.

From the elements described by now, many possible types of dataflow can be derived. If we take into account the data manager at the source island, the supplying mode, the effects of the dataflow and the question of ownership, than obviously not every posssible combination is realistic. If a copy is transmitted, the source has to remain owner. In contrast, if the data is transferred, then the source will no longer retain ownership. A referenced transmission of data managed by a wfms at the source island obviously does not make any sense, as in general wfms only support the local use of workflow-relevant data. Table 2 shows the resulting relevant *integration patterns* (IP).

Data-Manager (Source)	Supply Mode	Effect	Ownership/Possesion-Constellation
DS	materialized, Referenced	replicate, copy	C, CSO, CGO, RRC
		pass, release	T, TSO, R
WFMS	Materialized	spread, deposit	C, CSO, CGO, RRC
		Travel, deliver	T, TSO, R

Table 2 : integration patterns

3 Realization Aspects

To develop some kind of middleware handling dataflows in an automatic way requires the discussion of the major influencing factors as done in the previous chapter. Nevertheless, for actually executing a dataflow we have to consider the technical and organizational circumstances. Obviously, they determine the necessary actions.

Therefore, a middleware must be capable of considering the actual attendant conditions in a very flexible way.

We want to demonstrate this using the integration pattern “materialized pass with shift of ownership” (DS, materialized, pass, TSO) shown in Figure 2. First, the activity A_1 finishes work and stores the results into DS_1 (1). Next, according to the workflow’s definition a reference is handed over to $WFMS_1$ (2) and from there forwarded to the WFI (3). Hence, the WFI has to provide proper functionality to support reading and deleting the data from DS_1 (4). Furthermore, the WFI has to handle the transport of the data towards the target island. There, it has to store the data into the local system DS_2 (5) and adjust the changed rights of ownership (6). For the further local workflow procession the WFI supplies $WFMS_2$ with the data’s reference (7). After this, the workflow activity A_2 is started by $WFMS_2$ (8) and may access the data now available in DS_2 (9).

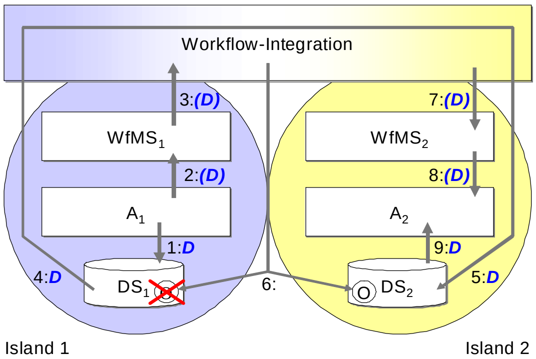

Figure 2: szenario “materialized pass with shift of ownership”

Connecting the WFI with $WFMS_S$

Current wfms have the ability to support local dataflows [LR00]. But, if the local environment’s borders are exceeded, an additional interface to the outside becomes necessary. Indeed, there are two possibilities to gain this goal: First, the local workflows may be expanded with an additional activity, which is responsible for the communication with the WFI. The second way is non-invasive and provides some mechanism for the WFI to monitor the progress of the workflow processed by $WFMS_S$. In the first case, special new activities are inserted into the local workflow-types at the position, where the global dataflow has to take place. These activities call the WFI, hand over some kind of reference to the actual data and trigger the dataflow. In the second case, the WFI monitors the current state of all dataflow-relavant workflow-instances. If an activity ends, which produces data that is to be transferred, the WFI recognizes this, gets the data’s reference from the wfms and transfers the data.

Data Access

Basically, there are two possibilities for accessing the cooperation data stored in DS_Q. For direct reading access the API of DS_Q may be used. Yet, many systems currently also support some kind of check-out mechanisms. This can be useful to extract the data in a well manageable format like XML, for example. Afterwards, the data is sent to the target island.

Administration of Ownership and Possession

The administration of owners' and possessors' rights must be handled by the WFI as global instance. But solely this is not sufficent at all. If support is offered by other participating components, this should be used of course. Most PDMSs also support computer-aided groupwork. If a PDMS is used as data-store, the internal owner and possessor rights-mechanisms have to be mapped to the underlying mechanisms. This mapping may differ using different systems.

4 Using existing EAI- and Workflow-Technology

During the integration of cross-enterprise processes, we have to consider that all the islands already have an existing infrastructure. Therefore, the WFI must work together with these systems in a compliant way. Furthermore, new components shall only be developed, if no adequate "off-the-shelf" component is available. We choosed to use existing EAI-technologies like IBM's *CrossWorlds* [IBM02] or Microsoft's *BizTalk* [Ms01], for example, to connect existing systems and additional components. The emerging system might look as shown in Figure 3.

At both islands, local WFMSs process workflows according to the appropriate workflow type requirements. We assume that additional, special activities handle the connection to the WFI. These activities are called *dataflow activities.*

As stated before, the WFI consists of commercial EAI-brokers as well as new developed components which add lacking functionality. For example, the ownership/posession-block shown in Figure 3 is such a component.

The EAI is enabled to communicate with system components by using *connectors*. Therefore, connectors are used for three tasks:

- If the EAI requests some kind of service from an integrated component, then the request will be mapped to the interface of the according system.
- If an integration-relevant event occurs inside an integrated component, then this event has to be recognized and passed to the EAI. Event-recognition requires periodical retrieval of the current state. However, this may also be achieved through the components assistance, for example by using some kind of callback-mechanism, direct invocation or, in case of a relational dbms, an appropriate trigger.

- As EAI-broker use neutral data formats for internal communication and different components mostly use different formats, the connectors have to provide some kind of data transformation.

Obviously, we have to specify in which way an integration pattern has to be processed. This is done by defining *collaborations*. If the connector binding the wfms to the EAI signals an event, then inside the broker the according collaboration starts.

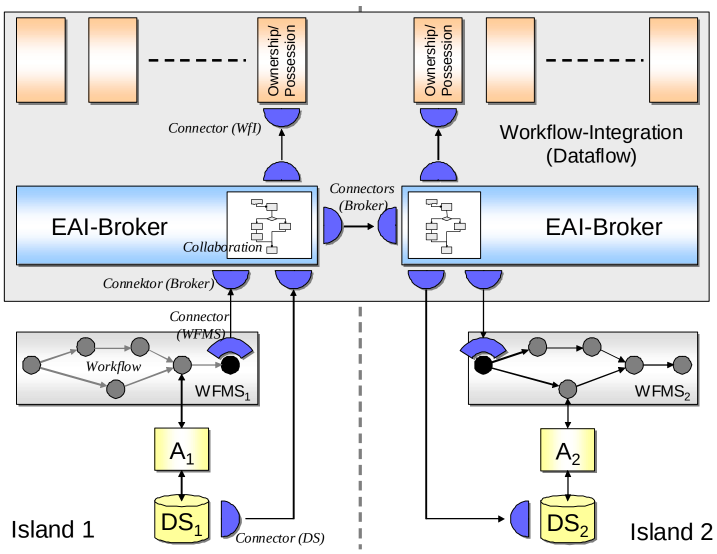

Figure 3: integration architecture for island-spanning dataflow

Using the sketched architecture allows us to describe and realize the categories of dataflow introduced in paragraph 2 as collaborations in an technology independent way. Therefore, the mapping to actual systems is the connectors' task. Additional functionality may be added easily by integrating new system components.

5 Related Work

So far, we described our way of integrating processes by providing some kind of middleware which is based on workflow technology. However, for realizing this central component no global workflow model shall be used. In fact, we want to investigate the basic dependencies between workflow environments occuring in reality and furthermore, we plan to develop an integration component supporting these dependencies in a specific way. To the best of our knowledge, there are no other approaches trying the integration in this way.

Nevertheless, there are other approaches concerning company spanning exchange of product-data for the cooperative development of products. The project ANICA tries to support access to distributed CAX-sources by providing an integration-bus. Therefore, not only spatial information can be hidden, but also (as far as possible) the data format by using internal conversion mechanisms [AJSK98]. The project discribed in [AGL98] investigates the possibilities of directly linking heterogenous PDMS. By order of the american Department of Defense (DoD), the “Product Data Markup Language” (PDML) was developed for the exchange of data between heterogenous PDMS. At this, for the purpose of integration a neutral “integration schema” is used [Bur99]. All these projects have in common that the integration problem is solved rather from integrating data sources than integrating processes.

6 Summary and Outlook

With this paper, we take part in the exploration of crossorganizational workflows which are produced through dependencies between heterogenous workflow environments (“islands”). We have chosen an approach allowing automated control of such dependencies on one hand, but also preserving the involved systems as far as possible on the other one.

In this context, the main concern of this paper is the identification of dataflow dependencies. Of course, for every actual application this dependencies have to be identified and modelled, before an automated handling at runtime is possible. This allows a frictionless execution of the global process, respectively the frictionless “cooperation” of heterogenous workflow systems. As essential factors of influence we identified the form of product-data administration at the source system, the effects of the actual dataflow on the source as well as on the target, the supplying mode at the target island and finally the control of ownership and possession according to the changing situation.

Regarding these aspects and the possible combinations of their occurances, a multitude of integration patterns can be derived. Each pattern describes exactly one possible case. Obviously, in view of the mere count of patterns, the supporting system has to be designed in a very flexible way. Therefore, the specific features of connected systems (esp. wfms) can be exploited well and effectually during the automated processing of a dependency. Apparently, these requirements complicate the development of a single solution to solve it all.

The architecture we introduced using existing EAI-technologie still has to be improved to support all possible patterns. But nevertheless, it indicates, how integration patterns may be available in form of collaborations. Therefore, system specific facilities are encapsulated by connectors and enable the system to participate in the integration task. Lacking functionality may be supplied by additional system components.

Further examination and prototypical implementations will have to prove, whether current EAI-technology is capable of supporting still more aspects. Furthermore, the behaviour of such EAI-solutions in case of errors and the possibility to combine different EAI-brokers are still open issues.

7 Literature

[AGL98] Abramovici, M., Gerhard, D., Langenberg, L.: Supporting Distributed Product Development Processes with PDM, in: Krause, Heimann, Raupach.(Hrsg), New Tools and Workflows for Product Development, Proc. CIRP Seminar STC Design, Mai 1998, Berlin, Fraunhofer IRB Verlag, 1998, 1-11

[AJSK98] Arnold F., Janocha A. T., Swienczek B., Kilb T.: "Die CAx-Integrationsarchitektur ANICA und ihre erste Umsetzung in die Praxis", in: Proc. "Workshop Integration heterogener Softwaresysteme (IHS '98), 28. GI-Jahrestagung Informatik'98 - Informatik zwischen Bild und Sprache", Magdeburg, September 1998, 43-54

[BRZ00] Bon, M., Ritter, N., Zimmermann, J.: "Interoperabilität heterogener Workflows", Proc. GI-Workshop "Grundlagen von Datenbanken", 2000, 11-15

[BHR01] Bon, M., Härder, T., Ritter, N.: "Produktdaten-Verwaltung in heterogenen Workflow-Umgebungen", Internal Report, December 2001

[BRH02] Bon, M., Ritter, N., Härder, T.: "Sharing Product Data among Heterogeneous Workflow Environments", in Proc. "Int. Conf. CAD 2002 - Corporate Engineering Research", Dresden, März 2002, 139-149

[Bur99] Burkett, W.: "PDML - Product Data Markup Language - A New Paradigm for Product Data Exchange and Integration", 30.04.1999, www.pdml.org/whitepap.pdf

[IBM02] Technical Introduction to IBM CrossWorlds, IBM Corporation 2002

[KRS01] Kulendik, O., Rothermel, K., Siebert, R.: "Cross-organizational workflow management - General Approaches and their Suitability for Engineering Processes". in: Schmid, B., Stanoevska-Slabeva, K., Tschammer, V. (Hrsg.): Proc. "First IFIP-Conference on E-Commerce, E-Business, E-Government: I3E 2001", Zürich, Schweiz, Oktober 2001

[LR00] Leymann, F., Roller, D.: "Production Workflow: Concepts and Techniques," Prentice Hall PTR (ECS Professional), 2000, ISBN 0-13-021753-0

[Ms01] Microsoft BizTalk Server 2000: Documented, Microsoft Press, 2001