Organizational Memory Supported Workflow Management

Rob van Kaathoven
Swiss Life, Zürich (Rob.vanKaathoven@swisslife.ch)

Manfred A. Jeusfeld
KUB Tilburg University, NL (jeusfeld@kub.nl)

Martin Staudt
Swiss Life, Zürich (martin.staudt@swisslife.ch)

Ulrich Reimer
Swiss Life, Zürich (ulrich.reimer@swisslife.ch)

Contents

Electronic Business Engineering / 4. Internationale Tagung Wirtschaftsinformatik 1999.
Hrsg.: August-Wilhelm Scheer; Markus Nüttgens. – Heidelberg: Physica-Verlag, 1999

Abstract

Business processes are by nature information-intensive and require IT support. Database systems solve the basic need for secure and efficient data storage and access. As such they are well-understood and widely applied. Workflow management systems, currently being added to support and optimize formal business processes, distribute jobs among employees. Recently, researchers and practioners started to promote the idea to explicitly represent the knowledge of the enterprise as so-called "Organizational Memory".

Such knowledge includes the goals of the enterprise, its tasks, its rules and its resources. This article investigates the interrelationship and interplay of organizational memory systems with workflow management systems. We describe experiences gained from a concrete integration project at Swiss Life, an insurance company mainly engaged in the private life insurance and pension scheme management business. Result of the study is a formal model of the relationship of information handled in the two systems and a specification how such systems can interoperate to provide knowledge-based workflow management.

1 Introduction

Knowledge in modern organizations is a valuable resource that has to be managed properly. As counterpart to data (base) management systems, so called *Knowledge Management Systems* and environments are intended to collect the typically very heterogeneous, in part informal structures describing this knowledge, and to offer an integrated view and flexible retrieval facilities. The kernel of such systems is often covered by the notion of *Organizational Memory Systems* (OMS) which serve as a repository for the *Organizational Memory* (OM). The information contained in an OM comprises enterprise goals, organizational structure, tasks and rules up to resource information (e.g. knowledge maps of employees' skills etc.). On the other hand organizations are switching from functional to process-oriented environments to be able to react quickly on rapidly changing customer-demands, to be able to introduce innovative products and services easier, and thus to improve the overall quality. Automation of processes in these companies is supported by *Workflow Management Systems* which transport data and distribute jobs between nodes acting in the process chain. An interesting question arises with the combination of Organizational Memories and Workflow Management Systems.

Currently an Organizational Memory ‘in the small’[1] is being implemented at Swiss Life (Reimer 1998), one of the leading European companies in the private life insurance and old-age pension schemes sector. The system, called EULE2 (Reimer et al. 1997), has the objective to assist office-clerks in doing their job. It has the knowledge and capability to deliver detailed assistance and to guide an office-clerk through an Office Task that has to be accomplished. Its use is not obligatory, so a user asks for support when needed. A Workflow Management System (WFMS) is not operational yet but a definitive employment is assumed to take place in the near future. Anyhow, even EULE2 already has some WFMS functionality built in and it is expected that more WFMS functionality is needed by or in cooperation with EULE2. Shortly said, EULE2 operates from a single-user perspective in a detailed and intelligent way that cannot be found these days in a commercial WFMS. A WFMS on the other hand is specialized in coordinating tasks that are to be performed by a group of people.

This paper elaborates an integration concept for workflow management and organizational memory. The practical experiments were undertaken with the WFMS Staffware®. Here, we concentrate on the conceptual level and discuss integration scenarios. Since OM systems are rarely found in practice, we use the EULE2 system as an example. Given this restriction on generality, the study proposes answers to the following integration questions:

What knowledge is modeled on the OM side and what on the WFMS side?

Section 3 proposes a process-oriented meta model that captures the common terms and is used to represent the different viewpoints that OM and WFMS have on the activities of an enterprise. The meta model also exhibits how the two viewpoints are conceptually related.

Which models, notations, functionality, and architecture can be used to form an integration concept?

We propose in Section 4 to use so-called flow chunks guarded by pre- and post-conditions to specify the interoperation of both systems.

2 State of the Art and Motivation

In order to lay down the basis for the proposed integration we proceed with looking at relevant material in the literature, draw a short résumé and then give an example for illustrating our application in the insurance business.

[1] An ‘Organizational Memory in the Small’ is a term introduced in (Ackerman 1994) and stands for an organizational memory that is directed towards specific organizational tasks. The current EULE2 prototype can be seen as a task-based organizational memory.

2.1 Relevant Aspects in the Literature

We explored several areas of work for our integration task. Besides results and proposals linked to OMS and WFMS in particular, the general topic of modeling processes is of interest.

Organizational Memory

A lot of incoherent definitions of Organizational Memory can be found in the literature (Wargetitsch et al. 1997). One of the most cited definitions is from Walsh and Ungson (Walsh/Ungson 1991): *"Organizational Memory refers to stored information from an organization's history that can be brought in bear on present decisions"*. This definition was extended by (Stein 1995): *"OM lead to an higher effectiveness of an organization, under some circumstances also to a lower one"*. While OM is a conceptual term, an OMS is aimed at supporting this concept with information technology. The task of an OMS is to help enterprises organize their knowledge and experience and using it in business processes (Wargetitsch et al. 1998). The OM dimension helps to answer questions like the *why* and *how* concerning a certain procedure within a firm or specific group within that firm, for example. Without these aspects being involved, workflows could be executed but the system cannot deliver (intelligent) support during workflow execution. It would not be possible to supply assistance on how to deal with certain working steps, when the current workflow is relatively new to the user. Furthermore, the user would not learn why certain steps should be executed. This reduces insight and overview of the user wrt. the whole process.

While searching for relevant theory it seemed very hard to find any usable material that concerned the integration of Organizational Memory with Workflow Management specifically. A reason for this could be that this is a rather new area. In (Wargetitsch et al. 1997; Wargetitsch et al. 1998) the development of an evolutionary WFMS by using an Organizational Memory approach is discussed. This approach is however mainly focusing on learning aspects and so called ad hoc workflows. That is, the workflows are composed by the users themselves, while using historical business cases and template building blocks together to build the actual workflows. In the insurance business, a more standardized type of workflow is used, the so called production workflows (Georgakopoulos/ Rusinkiewicz 1997). Most other literature on Organizational Memory is (still) concerned with conceptual issues and emphasizes building an OM without considering an integration with existing systems. An example is (Abecker et al. 1997), where a conceptual technology is suggested for building Organizational Memories which emphasizes the integration of knowledge 'inside' the OM and not with systems like a WFMS. Concerning WFMS-based integration themes, approaches generally deal with the integration of legacy systems and WFMS (Beltman 1997a; Beltman 1997b). An important difference to our case is that the activity defined on the *OM side* (EULE2) has to be integrated with the ones on the WFMS side. How to link one WFMS to another is discussed in the literature in several aspect in (Jablonski et al. 1997). Although interfacing standards are defined in (Lawrence 1997), in practice most WFMS products still have their own standards and proprietary application interfaces.

Workflows

Elementary workflow issues, terminology, workflow characterization, and workflow state-of-the-art are presented in (Österle/Vogler 1996; Leymann/Roller 1997; Verhoef/Joosten 1997; Georgakopoulos/Rusinkiewicz 1997). Workflow selection criteria and workflow trends are found in (CW 1998a; CW 1998b; Lange 1998). Besides this, the Workflow Management Coalition published material that also covers standardization (Lawrence 1997). Other attempts on standardization can be found in (Schulze et al. 1998). We can look at WFMS from various viewpoints. For example, build-time (concerning workflow modeling) and run-time aspects (concerning the execution of workflows) can be distinguished. Also a separation on modeling levels and phases can take place (business process (meta) models, workflow (meta) models, procedures etc.). Another division can be by implementing workflow systems, designing workflow systems and using workflow systems. These approaches are all explained in (Jablonski et al. 1997; Österle/Vogler 1996).

Process (oriented) Modeling

Much literature presents overviews of known modeling techniques per domain (Fox/Gruninger 1997) or stick at an abstract, conceptual, or terminological level (Uschold et al. 1996). A domain that is smaller (but still huge) and more directly related to workflow is process modeling and workflow modeling itself (van der Alst/van Hee 1997; van den Berg/Pottjewijdt 1997; Lee et al. 1996; Stanford 1997; Casati et al. 1995). An interesting development is that business process modeling is related more often to workflow modeling. Steps towards integration are studied and described in (Galler 1997; Amberg 1996). These days numerous approaches and methodologies do exist that offer ways to model (business) processes (Jablonski et al. 1997; Hess/Brecht 1995). A well-known and broadly used approach is the so-called Event-driven Process Chain (EPC) that is used by commercial products like ARIS from IDS-Scheer[2] and SAP R/3[3]. Here the point is reached that workflow touches the domain of business process redesign (Hess/ Brecht 1995) and where a permanent connection between these parts can be established (Scheer 1996; Scheer et al. 1995). Interesting initiatives can be found that emphasize on creating a single platform for process definition and translation (Lee et al. 1996; Stanford 1997). Another area is concerned with modeling itself and with modeling the models, where a technique is used that is called meta modeling (Nissen et al. 1996). Meta models can be applied to reveal and judge expressiveness of (workflow) languages (Jablonski et al. 1997; Kradolfer/Geppert 1997) or models used for business process redesign for

[2] IDS-Scheer: http://www.ids-scheer.de

[3] SAP: http://www.sap.com

example (Hess/Brecht 1995). A theme related to meta modeling is conceptual modeling that is a broader view on modeling on different levels and with different perspectives (Nissen 1997).

Résumé

The short review shows that the integration of OM with WFMS has not been systematically investigated. We claim that the integration is beneficial for the following reasons. A WFMS concentrates on the timely execution of a set of tasks by a limited group of office workers. An office worker is responsible for a set of tasks that are defined in order to maximize the throughput. The context of the task cannot easily be recovered since it can only be understood when looking at related activities and company rules. On the other hand, OM systems like EULE2 do not care about limited resources and efficiency. They define activities as if a single person would execute them. Thus, an OM system can be regarded as the knowledge base of a WFMS.

2.2 Example: Application for Insurance Contracts

To make the difference between an OM and a WFM view on activities more concrete, we consider a typical example on how to handle insurance applications (compare activity diagram in Figure 1). The client is informed by an insurance agent. After this, the client will supply its decision.

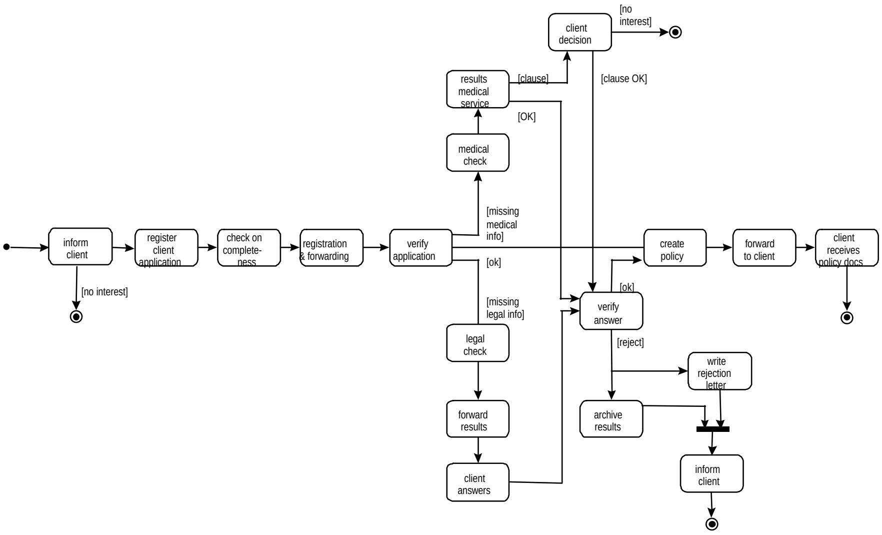

Figure 1: "New Application" Process

If the decision is positive then the client submits the new application (forms) to the agent. The agent will forward the application to the administration. The administration will then check the application for completeness. If this is the case then the next step consists of the verification of the application wrt. legal and medical details. If one of them is incomplete or causes problems, then additional information has to be requested or specific actions have to be taken. If legal information is missing then the agent has to get additional information from the client. If medical information is missing or if problems with the existing medical information arise, a medical check has to be executed (by public health service for example). A result of this action can be that a condition clause in the contract is suggested, which means that the new application can be accepted only under certain restrictions. The client has to tell whether he agrees with these extra conditions for his insurance. The answers (from the legal as well as medical checks) are returned and verified. When an agreement is reached the policy is created and the client receives a confirmation together with the policy via his agent. When there are unrecoverable problems the request for a new application is rejected. The client will be informed and the case data will be archived.

An OM system like EULE2 represents the whole activity graph to explain how an insurance application is handled. In contrast, a WFMS aggregates some activities to larger activities when handled by the same worker in sequence. Activities to be performed by different worker are separated in the WFMS. For example, the medical check is done by a physician whereas further processing of its results is done by the insurance agent. The activity graph of a WFMS is the implementation of an OM activity graph under limited resources.

3 Integration of Concepts

We now turn to the meta-model based integration of concepts used in OMS and WFMS. This is done by looking at four different perspectives: the functional perspective (definition of activities), the behavioral perspective (sequencing of activities), the informational perspective (data elements and data flow) and the organizational perspective (departments, working groups).

3.1 Meta Modeling

A common technique for integrating different languages is meta modelling, i.e. identifying their main building blocks and constructs, describing them in the same formal framework (meta-level) and derive semantic relationships or mappings between them. Continuing our previous application example, a workflow specification and a specification in the EULE2-High Level Language (HLL) lead to meta models as sketched in Figure 2. In the following, we use UML as a representation language for the meta models.

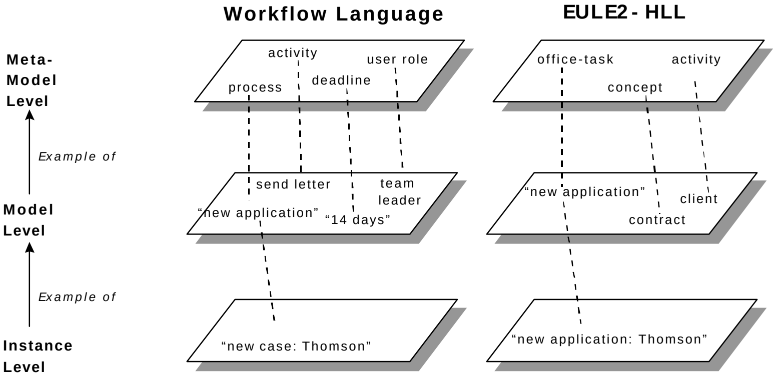

Figure 2: Meta Modeling Example

3.2 Functional Perspective

In Figure 3 the most common constructs are shown that are necessary to express the functional structure of workflows. The constructs in this figure are based on the terminology used in (Lawrence 1997). A process can contain several sub-processes which on their own can consist of (blocks of) activities. An activity block is a group of activities that have a logical or behavioral relationship (at workflow execution time). (Sub)Processes used in process hierarchies are also known as sub(work)flows. A hierarchy is used here to keep a clear overview and connection to a business process view. Decomposing activities over multiple levels ends at the specification of the 'leafs', which are the primitive activities (sometimes also called 'normal' activities), i.e. they are the most detailed ones.

When defining workflows it is important to be able to specify the relationships between (sub)activities. To do so, constructs are needed to model hierarchies of activities, so that it becomes clear which subactivities or sub-steps belong to more abstract activities (or blocks of activities) or which activities belong to a certain process. In (Jablonski et al. 1997) it is indicated that there is a coherence between shape (structure) and content (function). Nevertheless, it is claimed that an eye should also be kept on the differences of these aspects. In EULE2, there are extensive ways to specify functional structures (see Figure 4). At the highest level an Office Task can be specified. An Office Task is a task that is modeled from the perspective of a single user or office clerk (a 'single user' workflow). It states what (sub)activities and steps have to be executed by this single person to complete the accompanying EULE2 Office Task.

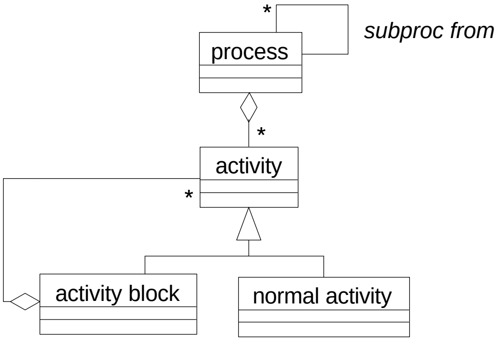

Figure 3: Meta Model of Functional Perspective in WFMS

For each Office Task one single start activity is specified usually of the category 'composed' activity consisting of several sub-activities or other composed activities. A simple activity contains the actions or work steps that have to be executed. An example is the simple activity ‘enter client data’ where the actual work steps are to ask the user for filling the single fields. EULE2 supports more activity types than discussed here.

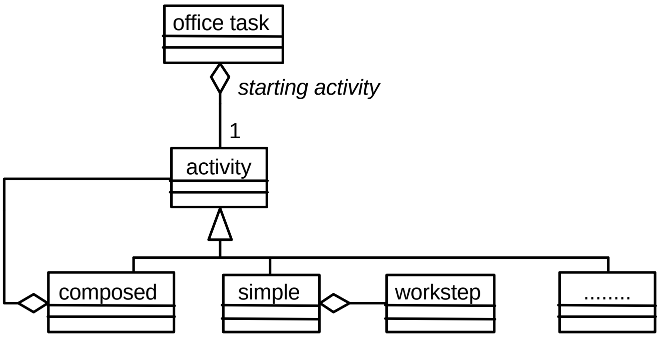

Figure 4: Meta Model of Functional Perspective in EULE2

3.3 Behavioral Perspective

Besides the structure of activities, the sequence and behavior of activities play a very important role. Constructs that can be used to model time and logical aspects belong to the so-called behavioral perspective. A distinction is often made between *prescriptive* and *descriptive* control flow elements. Prescriptive flow elements describe exactly one possible execution order of activities (for example ‘activity n follows activity m’). Descriptive flow elements on the other hand specify multiple equivalent execution orders of activities (Jablonski/Stein 1995).

First we take a look a the behavioral perspective for WFMS in general, see Figure 5 based on the terminology in (Lawrence 1997). With the *Sequence* flow element, it is possible to link two activities sequentially (‘b follows a’ for example). The *if-then-else* as well as the *iteration* flow elements can be used to specify flow constructions as known in elementary programming. More interesting are the *split-join* constructions that allow a workflow path to split (itself) into multiple parallel branches. It can be specified that such parallel branches all have to be executed at the same time (*and-split*), that only one (*xor-split*) or some (*or-split*) of these branches have to be executed. Furthermore, it has to be mentioned that a distinction does exist between the use of these flow elements in combination with atomic activities and the use of them in combination with grouped (block of) activities. Switching a block of activities in parallel causes multiple sub(work)-flows to be executed at the same time. These sub-flows can then be executed and accessed independently from each other until the moment when they are synchronized again after all the parallel branches have been completed.

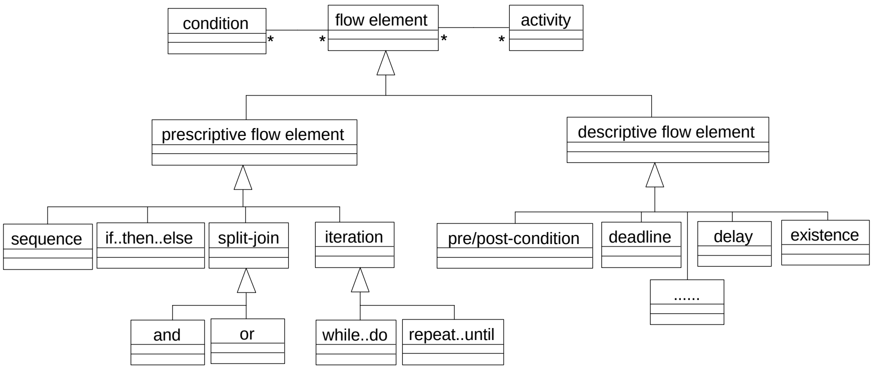

Figure 5: Meta Model of Behavioral Perspective in WFMS

All flow elements are connected to one or more conditions that have to be tested in order to select a certain flow direction. This is also the case for flow elements and activities; some flow control elements influence or describe the behavior of one or more activities. In the figure above the associations between the *flow element construct* and both the *condition* and *activity construct* are shown in a simplified form only. To keep the model comprehensible the exact relations are thus not shown in detail here. A *if(c)-then(a1)-else(a2)* construct for example can be associated with one condition *(c)* and two activities (or activity blocks) *a1* and *a2*.

Some of the *descriptive flow elements* can be built with the primitive prescriptive elements that were discussed above. For example a *delay* construct possibly could be implemented by combining *if* and *iteration* constructs. Nevertheless they are shown separately here because their ‘composed’ functionality will be applied frequently. For a *deadline* a certain point in time can be specified (possibly on the basis of a given expression). When this point of time is reached or a given period is expired a certain activity can be started - for example an activity that informs a team leader about certain delays within a team. Other frequently used flow elements are pre- and post conditions that can guard the entry or exit of an activity. As an alternative the pre- and post constructs are sometimes combined and replaced by a *transition* construct. These constructs have a functionality comparable to *if-then-else*, however the latter uses an explicit control flow logic instead of an implicit one, i.e. the condition test is modeled as an explicit step here (Jablonski et al. 1997). Furthermore, constructs like *delay* and *existence* (Jablonski/Stein 1995) can be useful to specify that certain activities have to wait for others or that an activity will only be executed if a certain other activity has been executed before (that is, the instance of that activity does exist).

For EULE2, constructs are available on the activity level as well as on the work step level to model sequence and behavior (see Figure 6). The order of activities can be specified by using the *connection* construct. Due to the fact that activities in EULE2 belong to an activity graph these connections are always formulated in one direction and no steps back like iterations are supported here. However, it is important to notice that this does not mean that the user cannot undo a certain step. For guarding the execution of certain activities *preconditions* can be used. This is the implicit variant of the *if-then* construct that specifies decision points explicitly. Typical within EULE2 are the *split-join* constructs. It is important not to confuse them with splits and joins as found in workflow terminology (Lawrence 1997). For the latter, a split-join construction stands for a transition from an activity x to multiple other branches (activities or activity blocks) that come together at a *join* afterwards. On the other hand, a split-join construct in EULE2 means that the current (sub)flow *instance* itself is spliced up into multiple independent instances (but based on the same block of (sub)activities) that are also joined again later on. A typical EULE2 split-join construction is used for example to apply the same task or treatment on multiple contracts in parallel. Finally, *entry-exit* constructs are more important for internal reasons (for example to ensure that a subgraph with activities has only one end-activity) and will not be discussed here in detail.

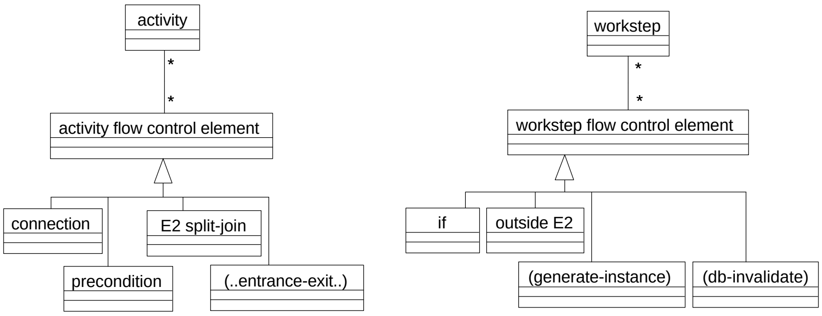

Figure 6: Meta Model of Behavioral Perspective in EULE2

With regard to work steps there are also flow control elements available. These are the *if* (with the usual semantics) and *outside E2* constructs (for parts of tasks that cannot be executed with EULE2 assistance or that are simply not modeled for the current Office Task). *Generate-instance* and *db-invalidate* are more for internal use.

3.4 Informational Perspective

This perspective focuses on what data variables and what kind of data flows are used within process oriented models. In contrast to the previous perspectives that were discussed, no common WFMS meta model is shown, but only the Staffware case is considered. One reason for this is that not much detailed information was found in general. On the other hand, the meta model of EULE2 could be seen as a well-equipped set of constructs to record the informational aspects of a common WFMS as well.

One of the most useful characteristics of the EULE2 modeling language in defining data structures is the support for (multiple) inheritance when specifying concepts (see Figure 7). This makes defined concepts reusable and flexible for later extensions. A concept can be used in association with an activity. It will be available as a local variable in such a case and can be passed to a subactivity as a parameter.

A *primitive* concept definition is a 'normal' data variable definition where the properties (attributes) of a concept are defined (for example concept 'person' can have 'name' and 'address' as properties). Concepts can be connected to database fields (*db*) so that values are read from a database and used as property values of these concepts.

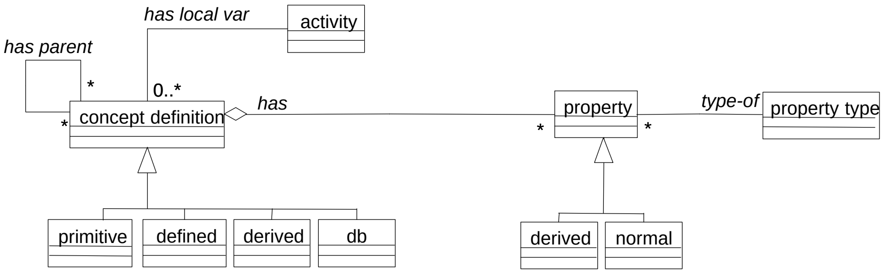

Figure 7: Meta Model of Informational Perspective in EULE2

The other subtypes of concept (defined and derived) as well as the derived property type will not be discussed here as they are used in more complex situation that are not relevant here. In Staffware, fields are defined globally within the whole procedure, that is within all activities of the workflow all defined fields can be accessed or modified and no parameters are used to transfer local variables, because they do not exist (see Figure 8). This situation is of course unfavorable considering hidden data dependencies that can be a source for major maintenance problems in complex workflows. A Staffware *field* can be compared with a *property* in EULE2. In Staffware, the fields do not belong to a class or a data structure that is defined at a higher abstraction level, like a EULE2 property that can be interpreted as a 'feature' or an attribute of a concept.

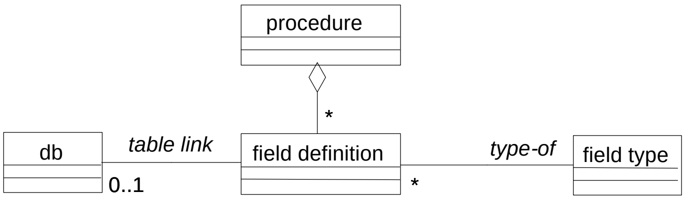

Figure 8: Meta Model of Informational Perspective in Staffware

3.5 Organizational Perspective

Often WFMS have a rather fixed meta model for specifying organizational structures. The organizational structure is needed to specify who has to do what at a certain moment. In Staffware the organizational and functional aspects are therefore connected by linking an *organization entity* with a *step*. The organizational meta model is shown in Figure 9. An *organizational entity* can be a *group* or a *user* that can play a certain *role*. For *groups* and *users* the same set of *attributes* are available, however, they can be extended by specifying new attributes. For EULE2 a meta-model for the organizational structure does not exist. This means that organizational aspects cannot be modeled within EULE2. Since its main focus lies on modeling Office Tasks to be executed by an individual user, this is not needed.

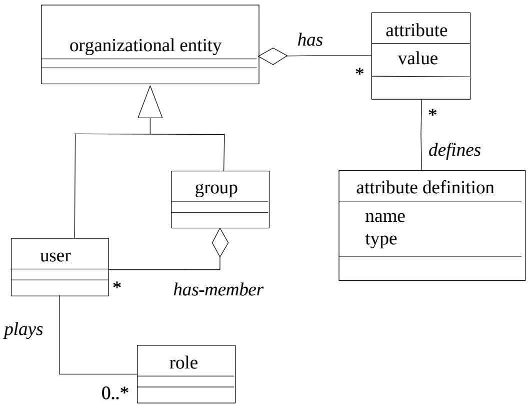

Figure 9: Organizational Perspective in Staffware

In summary, the concept of activities is central to both kinds of systems. Thus, an integration should focus on it. In the next section, we use flow chunks as a facility for integration.

4 Flow Chunks for Interoperation

4.1 Flow Chunks

Flow chunks link the activities of OM and WFMS without the need to change the definition of activities in the two systems. From the viewpoint of the OM system, a flow chunk is a subgraph of the overall activity graph. Such a subgraph is linked to an activity of the WFMS (compare Figure 10). The dangling links then induce the necessity of pre- and postconditions: An incoming flow is interpreted as a precondition (some data elements must exist before the execution of a workflow activity), an outgoing flow is interpreted as a postcondition of a workflow activity (some data elements exist after execution).

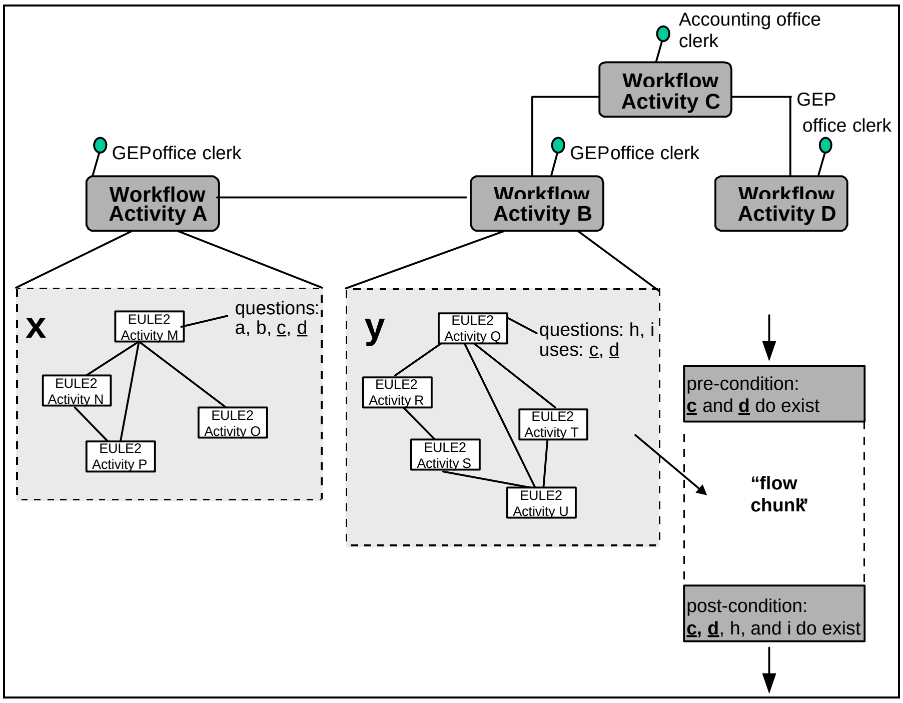

Figure 10: Flow Chunks

The OM activities are spread over multiple participants (and multiple departments). On the other hand, the EULE2 ‘flow’, the Office Task, consists of a graph that contains EULE2 activities. A flow chunk is related to a single workflow activity. In other words, these EULE2 activities are actually more detailed (sub)activities of the corresponding workflow activity. Data entry takes place and questions are answered at the EULE2 activities. Because of this and because flow chunks represent parts of the EULE2 main graph, dependencies between such flow chunks exist. The normal situation is that the EULE2 activities of an office task are walked though from the beginning (of the office task). However, when the user wants assistance at an arbitrary workflow activity, it might be the case that this is the first ‘contact’ with the corresponding EULE2 office task. A jump has to be made from the current workflow activity to the related location (somewhere) in the office task. This means that the answers on questions and other information that is normally available at this location in the Office Task would not be known. Therefore, such missing information must be identified and requested afterwards. This is the reason why preconditions (and post-conditions) are introduced for flow chunks. Figure 10 illustrates that within the first flow chunk some questions (a, b, c and d) have to be answered by the user. In the next flow chunk, besides the fact that some new questions are asked (h and i), a part of ‘old’ information is needed, too (namely the answers on questions c and d). To specify this dependency formally, the precondition *”c and d do exist”* is used for the second flow chunk. As a post-condition it is specified that *c, d, h,* and *i* will exist after this last flow chunk is completed.

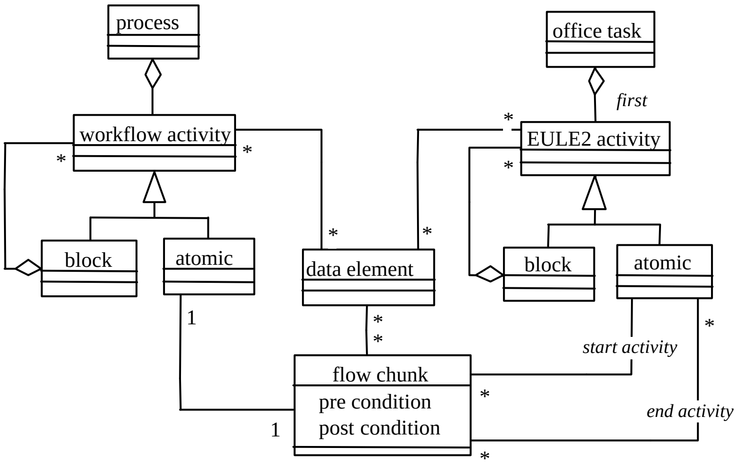

Figure 11: Meta Model with Flow Chunk as a Construct

To be able to explicitly specify flow chunks we suggest to introduce a so-called *Flow Chunk construct*. It can be used as a principle for integrating languages, specifying reference mechanisms (or referencing languages) or for translating between languages as well. Figure 11 shows that the flow chunk construct associates atomic activities (also called primitive, normal activities before) of the workflow side with one start activity and one or more ending activities at the EULE2 side. The construct *data element* is included to show that some kind of mechanism is needed for data exchange (to pass answers on questions for example, that are stored in certain parameters). In the figure both sides are separated quite strongly. This is a consequence of the difference in 'meaning' of the activities (caused e.g. by the multiple vs. single participant perspective) and detail of activities on both sides (i.e. the differences in granularity).

4.2 Flow Chunks at Execution Time

The work lists of the two systems have to be linked also, that is the WFMS *in-box* with activities that have to be performed by the user in the WFMS have to be connected to the detailed activities and working steps at the EULE2 side. In Figure 12 it is illustrated how both systems will cooperate with regard to the *Flow Chunk* principle. In the normal situation the user has started the WFMS and selected a certain workflow that has to be processed (i.e., a new instance of such a workflow occurs). In the background EULE2 will be initiated also, so that these systems will not be loaded, started and closed again all the time. When reaching a certain workflow activity, the user can request EULE2 assistance. Most of the time this will not be at the beginning of the office task so the corresponding EULE2 activity has to be located by means of the flow chunk that is associated with the current workflow activity. When EULE2 is at the right location in the office task, a mechanism must be started that prompts for missing information at that specific point, based on the defined *pre-condition* of the current flow chunk.

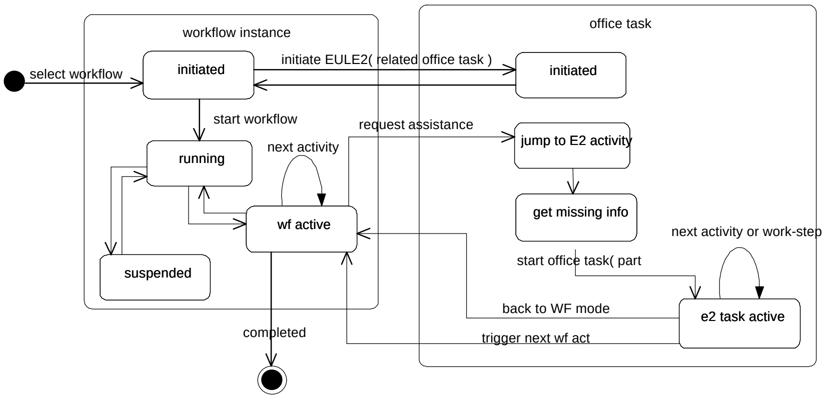

Figure 12: State Transition Diagram for Run-Time Integration

When the user is working within EULE2 the work list link is used to trigger activities on the WFMS side at the moment when the *ending activity* of the current Flow Chunk is reached in EULE2. In other words, EULE2 will trigger the work list of the WFMS so that the current activity is confirmed and the next one will appear in the WFMS work list. The WFMS has to notice that EULE2 is still active (i.e., the user has chosen to stay in the 'EULE2-assistance-mode') and that the (EULE2-) *start activity* of the next flow chunk (the one that belongs to the current, just triggered, workflow activity) will be started now. If the user decides not to stay in the EULE2-mode he can go back to the WFMS and complete the whole workflow there (or ask again for assistance at a certain workflow activity later on).

4.3 Linking Data Sources

On the side of the WFMS, different kinds of data stores can be distinguished (Lawrence 1997). First there is the *application data* that is specific to the application that is cooperating with the WFMS. This data is not accessible by the WFMS. Secondly, there is the *workflow relevant data* that is used to determine the state transition of a workflow instance (pre- and post-conditions for example). Also case data like addressees used in a workflow instance, belongs to workflow-relevant data and may be manipulated by the workflow (enabled) applications. Third, there is the workflow control data that is tightly connected to the workflow engine. This mainly concerns internal data and is not accessible to applications.

In the case of the EULE2-WFMS link the most interesting kind of data is the *workflow relevant data*. This is exactly the kind of information that is used by the flow chunk principle, for example with regard to the answers on questions (parameters) that have to be exchanged. Whenever EULE2 gives the control back to the WFMS such a data buffer is needed. This happens for example in the case of split-join constructions in EULE2 where addressees are checked and all letters are sent together afterwards. During an office task or even during office tasks that are started in parallel by separate users the addressee data can be stored in a so-called ROWM (read-once-write-many) memory. At a certain point the WFMS is called and all letters to the collected addressees that are found in this shared data base are sent together.

5 Summary and Outlook

The article represents the outcome of an on-going project to integrate an OM system with a WFMS for the insurance business. The conceptual analysis of both types of systems revealed that though both manage activities, they do have rather different viewpoints: an OM system represents the definition of activities, a WFMS distributes chunks of activities to a limited number of workers.

The next major task is to implement the integration. Within the next months, the EULE2 system will be deployed as a standard tool which the office clerks have available on their workstations. Then, a WFMS with the ability to notify the OM system about enactment of workflow activities has to be selected. We are planning to use a state-transition machine to implement the event notification between the two systems. As the EULE2 system supports an undo of activities, we should assume such a facility in the WFMS as well. Due to the different granularity of activities, an undo will probably have to be defined in terms of single workflow activities. Results on nested transaction in databases and transactional workflows may help here. The integration has also implications for the OM system. Currently, EULE2 does not support loops in activity graphs. Most WFMS however do support this construct.

Within the insurance case study, an office worker will typically have to interact with the WFMS when performing an activity. Usually, the OM system will only be invoked when the office worker needs specific help for an activity. It will guide through the individual steps, display applicable legal and company rules, and will feed already known data elements into the input forms.

This paper contains some specifications about a future integration of OM and WFMS. One may argue that only an implementation and its use in an enterprise can reveal whether the claims in this paper do hold. However, the specifications in the paper are not just speculations but the result of extensive discussions with users at the insurance company and with people from the development teams. It is a specification based on a concrete test-bed.

Acknowledgement

We like to thank Staffware GmbH for providing us with a test version of their system and excellent training and support during this case study.

References

Abecker, A./Bernardi, A./Hinkelmann, K./Kühn, O./Sintek, M. (1997): Towards a Well-Founded Technology For Organizational Memories. In: Proceedings AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, Stanford University, March 1997. http://ksi.cpsc.ucalgary.ca/AIKM97/abecker/OM.html.

Ackerman, M. S. (1994): Definitional and Contextual Issues in Organizational and Group Memories. Information Technology and People, 9(1996)1, pp. 10-24.

van der Aalst, W./van Hee, K. (1997): Workflow Management: modellen, methoden en systemen. Academic Services.

Amberg M. (1996): The Benefits of Business Process Modeling for Specifying Workflow-Oriented Application Systems. University of Bamberg, Germany.

Beltman, B. (1997a): Workflow-Systemen en de Integratie met bestaande systemen (1); Meer dan de som der delen. Workflow Magazine, 3(1997)7, pp. 11-13.

Beltman, B. (1997b): Workflow-Systemen en de Integratie met bestaande systemen (2); Conflicterende besturing in legacy-systemen. Workflow Magazine, 3(1997)8, pp. 12-17.

van den Berg, A./Pottjewijd, P. (1997): Workflow: Continue verbetering door integraal procesmanagement. Academic Services.

Casati, F. et al. (1995): Conceptual Modeling of Workflows. In: Proc. 14th Object-Oriented and Enterprise-Relationship Approach Int. Conf., GoldCoast, Australia, Springer-Verlag, LNCS pp. 341-354.

CW (1998a): Neun Produkte im Vergleich. Workflow-Studie. Computerwoche, (1998)6, pp.15-16.

CW (1998b): Konsolidierung noch nicht in Sicht. Workflow - enorme Nutzenpotentiale, aber auch reichlich Definitionsprobleme. Computerwoche, (1998)12.

Fox, M.S./Gruninger, M. (1997): Enterprise Modeling. Department of Mechanical and Industrial Engineering, University of Toronto.

Galler, J. (1997): Vom Geschäftsprozeßmodel zum Workflow-Modell, Dissertation, University of Saarland.

Georgakopoulos, D./Rusinkiewicz, M. (1997): Workflow Management: From Business Process Automation to Inter-Organizational Collaboration. Tutorial 23rd International Conference on Very Large Databases, Athens, Greece.

Hess, T./Brecht, L. (1995): State of the Art des Business Process Redesign: Darstellung und Vergleich bestehender Methoden. Gabler-Verlag.

Jablonski, S./Böhm, M./Schulze, W. (1997): Workflow-Management, Entwicklung von Anwendungen und Systemen. dpunkt-verlag.

Jablonski, S./Stein, K. (1995): Die Eignung objektorientierter Analysemethoden für das Workflow Management. In: HMD 185 (1995), pp. 95-115.

Kradolfer, M./Geppert, A. (1997): Modeling Concepts for Workflow Specifications, TRAMs, Joint project of University of Zurich and ETH Zurich.

Lange, B. (1998): Trends: Integration von WWW und Groupware: So flutscht die Arbeit. iX (1998)4.

Leymann, F./Roller, D. (1997): Workflow-based applications. IBM Centre for Advanced Studies, 36(1997)4, pp.102-123. http://www.almaden. ibm.com/ journal/sj/361/leymann.html

Nissen, H. W. (1997): Separierung und Resolution multipler Perspektiven in der konzeptuellen Modellierung. Dissertation, Aachen University of Technology, Germany.

Nissen, H. W./Jeusfeld, M. A./Jarke, M./Zemanek, G. V./Huber, H. (1996): Managing Multiple Requirements Perspectives with Metamodels. IEEE Software, 13(1996)2, pp.37-48.

Österle, H./Vogler, P. (1996): Praxis des Workflow-Managements: Grundlagen, Vorgehen, Beispiele. Vieweg.

Lee, J. et al. (1996): The PIF Process Interchange Format and Framework, MIT Center for Coordination Science, Working Paper #194, 1996. http://ccs.mit.edu/CCSWP194/CCSWP194.html#app6

Stanford University (1997): Summary of the Third PIF Workshop. http://ccs.mit.edu/pif/pifwksh3.htm

Reimer, U. (1998): Knowledge Integration for Building Organizational Memories., In: Proc. 11th Workshop on Knowledge Acquisition, Modeling and Management, Banff, Alberta, Canada. http://ksi.cpsc.ucalgary.ca/ KAW/KAW98/reimer/

Reimer, U./Margelisch, A./Novotny, B./Vetterli, T. (1997): EULE2: A Knowledge-Based System for Supporting Office Work. ACM SIGGROUP Bulletin 19(1998)1, pp.56-61.

Schulze, W./Bussler, C./Meyer-Wegener, K. (1998): Standardizing on Workflow-Management - The OMG Workflow Management Facility. ACM SIGGROUP Bulletin, 19(1998)1, pp.24-30.

Scheer, A.-W. (1996): ARIS - House of Business Engineering -special, Scheer Magazine, October 1996.

Scheer, A.-W./Nüttgens, M./ Zimmermann, V. (1995): Rahmenkonzept für ein integriertes Geschäftsprozeß Management. Wirtschaftsinformatik 37(1995), pp. 426-434.

Stein, E.W. (1995): Organizational Memory: Review of Concepts and Recommendations for Management. International Journal of Information Management, 15(1995)2, pp.17-32.

Uschold, M./King, M./Moralee, S./Zorgios, Y. (1998): The Enterprise Ontology. The Knowledge Engineering Review, 13(1998)1, pp.31-90.

Verhoef, D./Joosten, S. (1997): Een conceptueel raamwerk voor workflow management. Informatie, 38(1997)12, pp. 51-58.

Lawrence, P. ed. (1997): Workflow Handbook 1997. Workflow Management Coalition. John Wiley & Sons Ltd.

Walsh, J.P./Ungson, G.R. (1991): Organizational Memory. The Academy of Management Review 16(1991)1, pp.57-91.

Wargetitsch, W./Wewers, T./Theisinger, F. (1997): Workbrain: Merging Organizational Memory and Workflow Management System. University of Erlangen-Nuremberg, FORWISS.

Wargetitsch, W./Wewers, T./Theisinger, F. (1998): An Organizational-Memory-Based Approach for an Evolutionary Workflow Management System - Concepts and Implementations. In: Proc. of the 31st Annual Hawaii International Conference on System Sciences, Vol. I, Los Alamitos, pp. 174-183.