A Knowledge-Based Approach to Support Business Processes

Ulrich Reimer, Andreas Margelisch, Martin Staudt
Swiss Life
Information Systems Research Group
Postfach
CH–8022 Zürich, Switzerland
⟨first name⟩.⟨last name⟩@swisslife.ch

Abstract

Office tasks related to the processing of contracts in the insurance business are complex and highly dependent on legal and company-specific regulations. Furthermore, due to increasing competition on the market there is a strong pressure to increase efficiency and quality of office task performance. The only way to meet these manifold requirements is to provide a computer-based guidance and interactive support for office workers. At Swiss Life, we have developed the EULE system which fulfills these requirements. EULE's functionality is in the triangle of Knowledge Representation, Business Process Modeling, and Knowledge Management – the latter because EULE encodes and stores knowledge which is crucial for the company. The system relies on a knowledge representation language which covers data and process aspects as well as the relevant legislation and company regulations.

Introduction

Some time ago, Swiss Life, as many other companies, reorganized its customer support. Office workers are no longer specialists dealing with certain kinds of office tasks only, but meantime must be generalists who deal with all kinds of tasks. The number of different kinds of office tasks (like changing the beneficiary, increasing the risk sum, surrender of a contract) is rather high (about 60), and many of them only occur very sporadically. An office task has an attached process description which specifies how it should be performed. These processes are typically quite complex because there are many laws and company regulations which take influence on how they are to be performed properly. As a consequence, the work of this new generation of office workers is quite demanding and urgently calls for an appropriate support in order to meet Swiss Life's high quality standards.

The following observations can be made with respect to the characteristics of the office work:

- The experience of the office workers ranges from novices who have just joined the company to experts with many years of training.
- About 80% of the office work consists of standard cases with medium complexity. The remaining part consists of very complicated tasks that require a lot of experience.
- There are about 60 different kinds of office tasks. Most of them occur very seldom so that an office worker does not have the opportunity to acquire certain routine.
- It is the idea that new employees get a training first when they join the company. However, in order to have enough participants, the training courses actually take place only two or three times a year so that most of the training is nevertheless on the job and not before.
- The way a certain office task is properly executed changes from time to time due to new products, new company regulations, or new legal restriction.

The situation characterized above shows that there is a potential for increasing the quality of office work and decreasing the average time needed per office task by providing the office workers with an appropriate support system for executing its process steps. Such a system must show the following functionality:

Support of Office Work
The system supports all kinds of office workers, from the novice to the expert. The novice gets his training while working with the system and can properly execute the processes attached to office tasks from the very beginning due to the active support and guidance of the system. The expert profits less from the system's guidance but from getting relieved from routine tasks, like writing letters and sending memos. He can request support from the system whenever the need arises (e.g. when dealing with an exotic case).

During the specification phase for such a system (which took many cycles), it became evident to us that additional requirements will have to be met by the system to become a success:

Just-in-time knowledge delivery
A user is always given exactly that kind of information he needs in a specific situation so that he never needs to ask for it (also called *just-in-time knowledge delivery* (Cole, Fischer, & Saltzman 1997)).

Adaptation
Users with more experience need less guidance than novices. The system offers exactly that much guidance as needed by the user.

Maintainability
New legislation, and new company regulations (e.g., due to new products) make it necessary to adapt the office task descriptions regularly. As office task descriptions must be up-to-date to ensure proper office work it is of paramount importance that necessary changes can be done without much effort. Ideally, this is the case when the updates are made by the insurance experts themselves without involving people from the computing department (except maybe for special tasks like defining links to database fields).

Just-in-time knowledge delivery unifies business process support with the central knowledge management issue of supplying people with the knowledge they need to do their work properly. The adaptation functionality originates from the requirement to support novices as well as experts. Maintainability ensures that keeping the system alive will be feasible w.r.t. time and money.

Current Workflow Management systems which also aim at automating processes, fail in these concerns due to the limitations of their modeling languages. These languages, e.g. do not allow to represent detailed legal aspects in a declarative way, nor do they support any associated inferences. However, both would be needed to support an office worker in making the right decisions. Instead, workflow modeling languages concentrate on the procedural flow between the different participants of a process. This is a *macro level* view of business processes while the requirements mentioned above require a support on a *micro level.*

Therefore, a system with a functionality that meets the above requirements was developed by the Information Systems Research Group of Swiss Life. It is called EULE and is situated in the triangle of Artificial Intelligence (AI), Knowledge Management (KM), and Business Process Modeling (BPM). It is an AI system because it is realized as a knowledge-based system. It contributes to the KM efforts of a company because its knowledge base encodes knowledge which is crucial for the company, and thus preserves it and helps to make it available where needed. EULE has an impact on BPM because it provides (formal) models of processes for performing office tasks in a much more detailed view than it is usually the case with the models resulting from a BPM approach. The level of detail of the EULE models is indeed needed for the active decision support the system must provide.

Attempts to bridge BP models and whole workflows are actually hard to find in practice. In so far our approach is also a way of operationalizing (very detailed) BP models so that they can be executed. The above aspects are discussed in more detail below.

EULE: A Knowledge-Based, Cooperative System for Supporting Office Work

In order to build a tool like EULE which is able to support the direct execution of arbitrary office tasks, a model-based approach is most appropriate: For each office task we formulate a model of all involved data objects, the process steps and the relevant regulation contexts. This model is then compiled to process descriptions which run on top of the shell-like EULE system components. In the following we summarize the main features of EULE both from the user and the modeller perspectives and add some remarks concerning the implementation.

The User's View of EULE

EULE visualizes the process attached to an office task as a graph (cf. Fig.1). Its nodes contain a sequence of activities the user must perform, while its links are associated with conditions that must be fulfilled for the activities in the subsequent nodes to become relevant. The conditions result from federal law and company regulations. An office worker starts working with EULE by selecting an office task and entering task-specific data as requested by the system (EULE takes most of the data needed from various data bases and only asks the user data which is not available elsewhere). For each activity encountered the user can request an explanation, why it is necessary and why it is to be performed in the requested way. The explanations also give references to the underlying legislation and company regulations.

A node may contain a graph again, instead of a sequence of activities only. This nesting to an arbitrary depth allows to give even complex office tasks a comprehensible structure. While being guided through the graph (and its sub-graphs) the user sees only those activities which are actually relevant for the current office task instance. However, the user is free to browse through the whole graph and all the activities in its nodes, including the explanations attached to them, in order to get an overall view of the office task. This is especially helpful when the user is not familiar with that office task and wants to learn more about it.

Usually, during the execution of an office task process several letters (e.g. to policy holders, to beneficiaries, or to federal agencies) need to be written. As the EULE system already knows all the task-specific data it is able to automatically generate the required letters. In this way, the time needed for an office task is considerably reduced. Thus, EULE not only increases the correctness and quality of office work but also helps in accelerating it and is therefore helpful for experts, too.

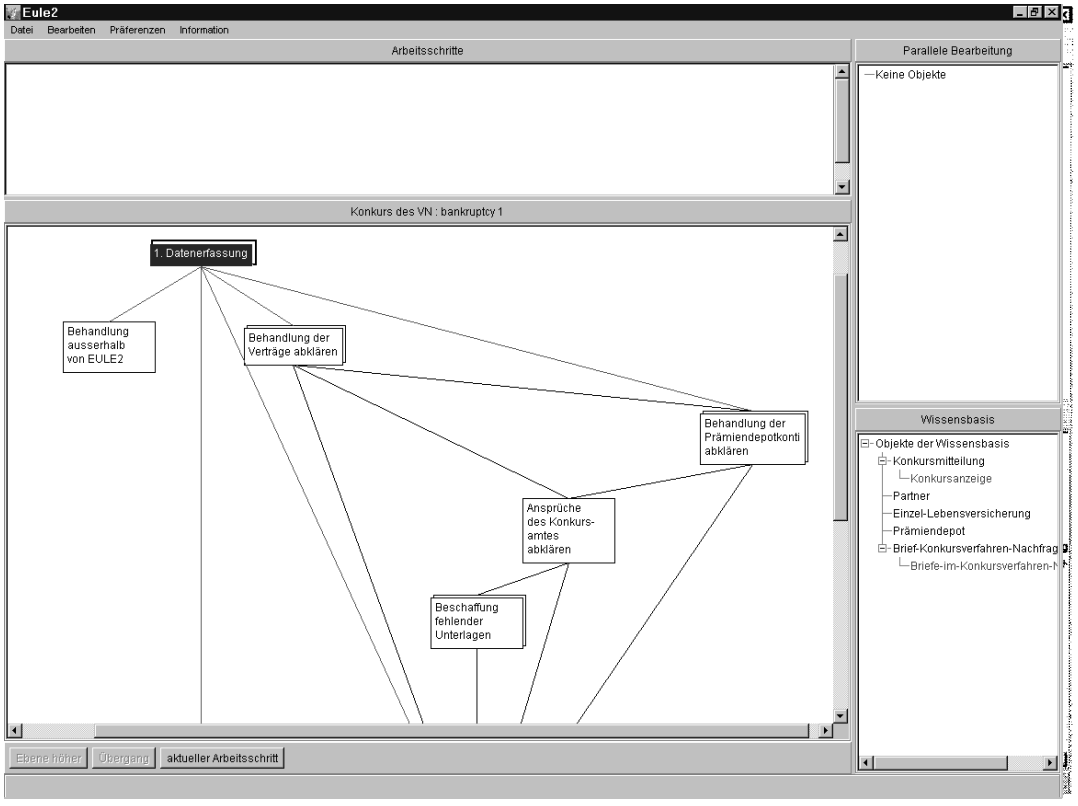

Figure 1: The Top-Most Level of the Office Task 'policy-holder-goes-bankrupt'

To be able to provide a less experienced user more detailed information, EULE can ask so-called *novice mode questions* which allow the user not only to enter 'yes' or 'no' but also 'I do not know'. Depending on his experience a user may already know the right answer (e.g., if a contract is legally valid or not) and can directly enter 'yes' or 'no'. If the user is not sure he can answer with 'I do not know' and (only then) gets a couple of additional questions which help him to find out what is the case. Of course, novice mode questions can be nested. An office task is completed when a terminal node in its top-most graph is reached.

Office Task Example: "Policy holder goes bankrupt"

To illustrate how strongly an office task may be influenced by law and company regulations let us discuss the case of a policy holder going bankrupt. A fraction of the top-most level of the corresponding office task graph is illustrated in Figure 1. The Bankruptcy Office publicly announces all cases of bankruptcy. Everybody who holds properties of the bankrupt is obliged by law to report them to the Bankruptcy Office. The properties will be sold by auction to pay off at least a part of the bankrupt's debts. The property an insurance company may hold of a bankrupt is an insurance where the bankrupt is policy holder. Whenever an office clerk is informed of an opened bankrupt proceeding, he must check if such an insurance exists (cf. node "data entry" in Fig.1). When the bankrupt is not policy holder of an insurance the office clerk destroys the notification of bankrupt and closes the case (cf. node "close case"). Otherwise, the office clerk must report the insurance to the Bankruptcy Office, unless an exception is defined by law. This is done inside the the node "clarify treatment of contracts" which contains a whole subgraph. One exception states that a provident insurance is exempt from seizure. Thus, provident insurances cannot be sold by auction and therefore need not be reported to the Bankruptcy Office. Instead, the office clerk informs the bankrupt that he remains policy holder. An insurance where the spouse, the children or both are the only beneficiaries is also exempt from seizure. By law such an insurance is transferred to the beneficiaries as soon as the policy holder goes bankrupt. The office clerk informs the beneficiaries that they are the new policy holders (this and the following activities correspond to nodes not shown in Fig.1). The beneficiaries have the right to accept or reject the transfer of the insurance. They accept the transfer by sending a certificate issued by the Debt Collection Office to the insurance company. The reception of the certificate is a necessary precondition for Swiss Life to issue a new policy and to send copies to the new policy holders. By sending a note to the insurance company the beneficiaries can reject

the transfer of the insurance. When an office clerk receives a note of rejection he reports the insurance to the Bankruptcy Office because it is up to the Bankruptcy Office to decide what to do with the insurance.

Creating and Maintaining Office Task Process Descriptions

As has been stated in Section the guidelines of how to properly execute an office task change whenever new products are introduced, or legislation or company regulations change. The latter case happens quite often. Therefore, it is important that the knowledge represented in EULE can easily be modified. Maintenance should be done directly by the insurance experts as far as possible – without a major involvement of the computing department. According to our experience this can only be achieved when the knowledge is represented in a high-level language that is especially tailored to the application domain and the view insurance experts have on their domain.

The high-level language we have developed for EULE is called HLL. Its syntax does not need to take computational efficiency issues into account (with respect to reasoning and integrity checking) because a representation in it is compiled down into efficiently executable code (but, of course, HLL must be computationally tractable). Thus, HLL can be tailored towards understandability without any compromise. In the following, we give an overview of HLL by discussing a few examples of how knowledge is represented in it.

Activities:
EULE distinguishes between three kinds of activities:

- A *working step* is an atomic activity which occurs as an element inside a node of an office task graph. It consists of a textual description to explain the user what to do, and either requires an acknowledgement of completion of the activity, or contains a request to enter certain attribute values into a concept instance in the knowledge base of EULE (e.g. the name of a beneficiary, or the premium amount). For the latter purpose the user has an especially designed object editor available.
- A *simple activity* is a sequence of working steps. Each working step can be associated with a condition when it is to be performed. A simple activity corresponds to a node in an office task graph that does not again contain a graph.
- A *composite activity* is a set of partially ordered simple and/or composite activities, i.e. it forms an acyclic graph. A composite activity either corresponds to a whole office task or to a node of an office task graph which again contains a graph.

An example of a simple activity as it is defined in HLL is given in Fig.2.

Laws and regulations:
An important requirement for HLL is to support the

```
Activity
  Name:
      obtain-bankruptcy-announcement
  Description:
      "Enter required data about the bankruptcy
       announcement"
  Arguments:
      K: bankruptcy-announcement
  Type:
      simple
  Working Steps:
      ( ObjectEditor, "Enter Data about Bankruptcy
                       Announcement",
          ( Action: Edit K
              ( Text: "Where does the information
                       originate?"
                Fields: information-source ⟨force⟩
              );
              ( Text: "Specify the status of the
                       proceedings"
                Fields: status-proceedings ⟨force⟩
              );
          EndAction)
      );
End
```

Figure 2: Example of a Simple Activity

VVG 81
Right of pre-emption of the spouse and the descendants

⋮

2 The beneficiaries are obliged to notify the insurer of the transfer of the insurance by presenting a certificate issued by the Debt Collection Office or the Official Receiver. If there are several beneficiaries, they must appoint a representative to accept the notifications incumbent upon the insurer.

Figure 3: Fragment of a Law

representation of law and regulations that influence how an office task is properly executed. A law defines a right or an obligation and specifies when it holds. HLL therefore knows of two predefined concepts 'right' and 'obligation'. A certain (sub)section of a law is then represented by introducing an appropriate specialization of 'right' or 'obligation' and by describing the conditions under which a certain instance of it exists. This instance is automatically generated when its existence condition holds. A generated right or obligation instance may then cause the precondition of some activity to become fulfilled. An example is depicted in Figure 4 which represents the law given in Figure 3. Figure 4 also shows examples of concept definitions.

Preconditions for activities:
Any kind of activity can have a precondition that must be fulfilled for the activity to be permitted. Precondi-

```
Concept
   Name:
      obligation
   Type:
      ontological
   Properties:
      is-obliged (object, [1,*], natural-or-juristic-person);
      holds-against (object, [1,*],
                     natural-or-juristic-person);
      is-fulfilled (boolean)
End

Concept
   Name:
      obligation-to-show-certificate
   Type:
      derived
   SuperConcepts:
      obligation
   Properties:
      is-obliged (object, [1,*], natural-person);
      holds-against (object, [1,*], natural-person);
      concerned-insurance (object, [1,1], life-insurance)
End

Law: VVG, Section: 81, Subsection: 2
InstanceDerivation
   For Derived Concept:
      obligation-to-show-certificate
   As:
      foreach i: life-insurance and
      foreach b: publication-of-bankruptcy
      if
         i.policy-holder = b.bankrupt and
         b.status = 'open' and
         i.beneficiary-clause = 'spouse-descendant' and
         exists-value(i.beneficiaries)
      then derive the Instance with:
          is-obliged = i.beneficiaries;
          holds-against = i.policy-holder;
          concerned-insurance = i
      end
End
```

Figure 4: Representing Laws

tions are derived from company regulations and often refer to the existence of a right or obligation which is the cause to perform the activity. An example of a precondition is given in Figure 5.

Each HLL construct has a reference to the knowledge source from where the knowledge being represented originates (see Figs. 4, 5). The reference can be used to give a user an explanation why a certain activity must be performed, and is used to show a knowledge engineer all constructs belonging to a specific source.

Implementational Aspects

From an implementation point of view the EULE system consists of four main components: the HLL compiler, the EULE KBMS, the EULE execution controller and the EULE GUI.

```
Regulation: GEP, No 29, 08/03/1993
Precondition
   Name:
      condition-to-clarify-treatment-of-vontracts
   For Activities:
      clarify-treatment-of-contracts
   Condition:
      exists b: bankruptcy-announcement and
      exists i: life-insurance
      such that
         b.bankrupt = i.policy-holder
      end
End
```

Figure 5: Activity Precondition

The EULE compiler transforms HLL descriptions into acyclic, nested graphs and prepares the office-task-specific data strucures. Its output consists of code fragments in Java and in a Prolog-style language which are interpreted by the controller and the KBMS respectively. The controller governs the graph traversal at runtime, initiates GUI actions, e.g. editing of objects, and invokes the KBMS for data retrieval, deduction tasks and validity checks. Conditions attached to the links in a graph act as integrity constraints formulated as Datalog$^{\neg}$ clauses (Ceri, Gottlob, & Tanca 1990) and are tested on demand for being satisfied or violated. Knowledge about concepts and instances is formulated in the terminological and assertional components of a description logic like subsystem of the EULE KBMS (Woods & Schmolze 1992). Knowledge about law and regulations is encoded in Datalog$^{\neg}$ clauses where we distinguish integrity constraints that must not be violated, and deduction rules which derive new attribute values or whole concept instances when their condition part is fulfilled. With this combination EULE competes with the few implemented hybrid inference systems coupling description logic and deductive database technology (Levy & Rousset 1998). However, opposed to earlier versions, the current EULE KBMS implementation relies on a deductive engine only, and limited terminological reasoning is exploited by the compiler and modeling support environment. For more details on EULE see (Reimer *et al.* 1998; Reimer, Margelisch, & Novotny 1999; Kietz & Staudt 1999).

The Impact of EULE on Knowledge Management and Business Process Modeling

The EULE knowledge base captures knowledge which is important to Swiss Life. Via EULE this knowledge is made available to office workers in such a way that they always get exactly that knowledge which is relevant in the current situation. Besides for supporting office work, EULE's knowledge is useful for other people and for other purposes as well, e.g., for tutoring

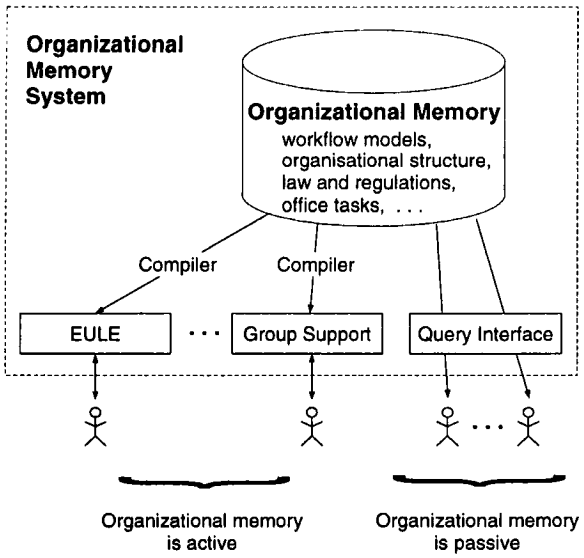

Figure 6: Integrating EULE into an OM System

new employees, for inquiring about the effects of certain company regulations on office tasks, or for finding out how often a certain regulation took effect in an office task in the past. As a consequence, a system like EULE must not be considered isolated from other issues of knowledge organization and business process modeling. In the following, Section discusses the integration of EULE in an organizational memory system, while Section deals with the integration of EULE within a framework for business process modeling.

EULE as Part of an Organizational Memory System

At the core of all knowledge management activities in an organization is the organizational or corporate memory (not necessarily (fully) computerized (Stein 1995)). It is the central repository of all the knowledge relevant for an organization. Building up such organizational memories (OM) and making them available to people and application systems with quite converging needs is a big challenge which can only be met by an integration of approaches from various fields of computer science as well as business and administration science (cf. (Abecker *et al.* 1998) and (van Heijst, van der Spek, & Kruizinga 1996)).

There are two major roles an organizational memory can in principle play (cf. Fig.6). In one role it has a more *passive* function and acts as a container of knowledge relevant for the organization (including meta-knowledge like knowledge about knowledge sources). It can be queried by a user who has a specific information need.

The second role an OM can adopt is as an *active* system that disseminates knowledge to people wherever they need it for their work. This second functionality is not just mere luxury but of considerable importance as users often do not know that an OM may contain knowledge currently helpful to them. Furthermore, querying an OM whenever the user thinks it might be possible that the OM contains relevant knowledge is not practical because the user does not always think of querying the OM when it might actually be helpful and because it would be too time consuming (as it interrupts the user's primary work and takes time for searching and browsing the OM).

For the OM to be able to actively provide the user with the appropriate knowledge it needs to know what the user is currently doing. EULE is an example of a system that solves this kind of problem. Such systems have been called *electronic performance support systems* in the literature (cf. (Cole, Fischer, & Saltzman 1997)). A system like EULE plays an important role in utilizing the knowledge in an OM and should be a part of the organizational memory system. This involves the integration of the knowledge formalized in EULE with other formal knowledge in the OM. For example, the terminology in the EULE knowledge base should be consistent with all the terminologies (or ontologies) developed in various other contexts, e.g. as part of the business meta data for the data warehouse, or for a search facilitator in the intranet information system. This alone is a major unification and integration effort. Another example are the EULE office tasks representations which need to be integrated with other kinds of formal process representations, as for example workflow descriptions.

Furthermore, it is necessary to integrate the formalized knowledge in EULE with informal one (i.e., textual or semi-structured, like SGML, HTML). For example, the knowledge represented in EULE stems from written documents, like company regulations, memos, minutes of meetings. Often, those documents contain more knowledge than has been formalized in EULE. Thus, a user of EULE may want to see the source of a company regulation that has an influence on the office task he is currently dealing with. The written document gives him additional background information which helps him to better understand the rationales for the regulation. For that reason, the knowledge represented in EULE has links to the sources which a user can follow to become more competent.

The Information Systems Research Group of Swiss Life is currently concerned with integration aspects of the above mentioned kind, aiming at building an OM from already existing but isolated knowledge sources in the company (see also (Reimer 1998)).

Embedding EULE in a Business Process Modeling Framework

As a support system for office tasks, EULE contains quite detailed task repesentations. Thus, EULE is a major component of the busines process modeling framework in the company and must be well settled within it. Process descriptions typically evolve in a top-down fashion. They start with a high-level structure of business processes that span various organizational

units, and then get broken down into more and more local views which at the same time become more detailed, until at the most detailed level EULE office task representations are obtained. The high-level descriptions are completely informal and are given in a textul and/or graphical form. More detailed levels show more structure, typically accompanied by the utilization of a process modeling tool which knows predefined modeling patterns, or introduce whole reference architectures (see e.g. (Scheer 1999)). Even more detailed levels finally require formalized models. Consequently, EULE representations typically evolve from already existing, informal or semi-formal process descriptions by adding additional detail. New insights gained when building an EULE model may lead to a feedback and a revision of models on a higher level. A proper integration of all process modeling activities is therefore needed to allow the evolution and interaction over all levels. The idea is to come up with a layered modeling approach which allows to adopt the view with that granularity which is currently needed, while ensuring that changes made on one layer will automatically be propagated to those other layers where they are relevant, too.

Another possible integration requirement is EULE's interplay with a workflow management system (WFMS). The functionality of EULE does not much overlap with the functionality of a WFMS but is in fact rather complementary: EULE supports a single user, while a WFMS is specialized in coordinating tasks that are to be performed by a group of people. It knows which subtasks must be performed by which (kind of) people and manages the flow of control and information between them. Additionally, a workflow description is procedural while the office task descriptions in EULE are declarative (since EULE is knowledge-based). As already pointed out before, EULE would not be maintainable within acceptable time and cost limits if office task descriptions were given in a procedural language.

Finally, a working step in a wokflow is atomic but typically corresponds to a task graph in EULE. Therefore, the main idea for coupling a WFMS with EULE is to associate subgraphs in an EULE office task with working steps of a workflow. A user will then start a new office task by selecting a workflow or by pickung up a task in the in-box of his WFMS. When reaching a working step that is associated with an EULE subgraph the user can request EULE assistance. When entering the EULE subgraph the user is prompted for the information missing at that specific point. When the EULE subgraph is terminated EULE reports back to the WFMS that the current working step is finished. The WFMS selects the next working step according to the workflow description. If that working step has an EULE subgraph, too, the user may enter it again, or is transferred into it automatically when he indicates that he wishes to stay in the EULE support mode.

The integration of EULE with a WFMS is difficult because knowledge in the knowledge base of EULE is relevant for the workflow system and vice versa. A coupling of both systems requires a (partly) sharing of of their knowledge bases. A more detailed discussion of the coupling can be found in (Margelisch *et al.* 1999) and (van Kaathoven *et al.* 1999).

Experience and Related Work

There exist only a few other systems with a functionality roughly comparable to EULE. Like EULE, they aim at supporting business processes by offering the user (access to) that information which he or she needs to successfully proceed with the current office task (see (Abecker, Bernardi, & Sintek 1999; Staab & Schnurr 1999)). Unlike EULE, those systems are not oriented towards detailed and completely described office tasks but go for weakly structured workflows instead, as they are typical for tasks which comprise a great degree of unforeseeable variations.

A slightly different approach is presented in (Wargitsch, Wewers, & Theisinger 1998). A user who wishes to perform a certain office task and does not exactly know how to do it right is presented cases of former task instances which are similar to the task at hand. For this, the user must first give the system a characterization of the current task and then gets task instances presented by the system from which the one which appears most appropriate is selected. Subsequently, that instance is manually adapted to the current needs.

While the approaches described in (Abecker, Bernardi, & Sintek 1999; Staab & Schnurr 1999; Wargitsch, Wewers, & Theisinger 1998) are to a large extent underspecified, the processes supported by EULE are strongly constrained by Swiss Federal Law and internal company regulations.

The experience we have gathered with EULE during a field study has been extremely positive so that the system is meanwhile deployed. The field study was conducted during August 1998 with the participation of seven office workers that had quite different experience and background. They used EULE to execute 311 office task instances. The field study was concluded with an interview of all participants and their team head. Those five office workers who had considerably less experience than the remaining two said that they could extend their expertise during their work with EULE, and that they needed much less support from colleagues or their team head while using EULE. Six of the seven office workers said that EULE pointed out to them aspects to consider which they would otherwise have missed. Thus, EULE indeed helped to avoid incorrectly executed tasks.

The team heads of the participating office workers noticed a considerable relief from support they usually need to give their team members whenever they encounter a situation they do not know how to deal with. Furthermore, the necessity for the team heads to check the work done is reduced, additionally relieving them from unproductive tasks. Generally speaking, the overall quality of office work has considerably increased (less complaints, faster completion).

Our modeling experience showed that in order to fully enable insurance experts to create and maintain office task descriptions, a graphical modeling tool is additionally needed. The tool must hide most of the actual syntax of the representation language and must allow several views on the knowledge base simultaneously, e.g. via a terminology browser, via a tool for editing and displaying office task graphs, and via a window which shows the various pieces of knowledge not by construct type but according to the knowledge source from which they originate. Thus, one can for example see all the concepts, preconditions and derivation rules that belong to one specific company regulation. Such a modeling tool is currently being developed at Swiss Life.

Conclusions

Office tasks related to processing of contracts in the insurance business are complex and highly dependent on legal and company-specific formal regulations. On the other hand, the increasing competition strongly demands for higher efficiency as well as higher quality standards. The EULE system developed at Swiss Life is designed to meet these demands by providing computer-based guidance and interactive support for office tasks dealing with private life insurances. The system relies on a knowledge representation language which covers data and process aspects as well as the relevant legislation and company regulations. We also sketched the embedding of EULE into a general organizational modeling framework, in particular w.r.t. organizational memories, workflow management and business process modeling in general.

A field study showed that EULE indeed fulfills the high expectations we have in it. The system was set productive in mid-1999. The major future activities concerning EULE is the development of a tool that supports the modeling of knowledge needed for EULE. The ultimate goal for this tool is to enable insurance experts to do the modeling themselves, without the help of people from the computing department. Furthermore, we continue our work on integrating EULE into all the modeling activities going on at Swiss Life, aiming at an organizational memory as a central repository containing all the knowledge relevant for the company.

Acknowledgements. We thank our colleagues Jörg Junger, Jörg-Uwe Kietz, Claude Marksteiner, Bernd Novotny and Thomas Vetterli, for their efforts to bring EULE into existence and to make it a success.

References

Abecker, A.; Bernardi, A.; and Sintek, M. 1999. Enterprise information infrastructures for active, context-sensitive knowledge delivery. In *ECIS'99 Procs. of the 7th European Conf. on Information Systems.*

Abecker, A.; Bernardi, A.; Hinkelmann, K.; Kühn, O.; and Sintek, M. 1998. Toward a technology for organizational memories. *Intelligent Systems* 13(3):40–48.

Ceri, S.; Gottlob, G.; and Tanca, L. 1990. *Logic Programming and Databases.* Springer.

Cole, K.; Fischer, O.; and Saltzman, P. 1997. Just-in-time knowledge delivery. *ACM Comm.* 40(7):49–53.

Kietz, J.-U., and Staudt, M. 1999. *Access to Heterogeneous Data Sources for Supporting Business Process Execution.* Swiss Life, Information Systems Research.

Levy, A., and Rousset, M.-C. 1998. Combining horn rules and description logics in CARIN. *Artificial Intelligence* 104(1-2):165–209.

Margelisch, A.; Reimer, U.; Staudt, M.; and Vetterli, T. 1999. Cooperative support for office work in the insurance business. In *Procs. of the Int. Conf. on Cooperative Information Systems, Edinburgh, Scotland.*

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

Reimer, U.; Margelisch, A.; and Novotny, B. 1999. Making knowledge-based systems more managable: A hybrid integration approach to knowledge about actions and their legality. In R., P., and Fronhöfer, B., eds., *Dynamic Worlds: From the Frame Problem to Knowledge Management.* Kluwer. 247–282.

Reimer, U. 1998. Knowledge integration for building organizational memories. In *Proc. of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop.*

Scheer, A.-W. 1999. *ARIS - Business Process Modelling.* Springer.

Staab, S., and Schnurr, H. 1999. Knowledge and Business Processes: Approaching an Integration. In *OM99 - Proceedings of the International Workshop on Knowledge Management and Organizational Memory (at Int. Joint Conf. on Artificial Intelligence, 1999).*

Stein, E. 1995. Organizational memory: Review of concepts and recommendations for management. *Int. Journal of Information Management* 15(2):17–32.

van Heijst, G.; van der Spek, R.; and Kruizinga, E. 1996. Organizing corporate memories. In *Proc. of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop.*

van Kaathoven, R.; Jeusfeld, M.; Staudt, M.; and Reimer, U. 1999. Organizational memory supported workflow management. In Scheer, A., ed., *Electronic Business Engineering*, 543–563. Physica.

Wargitsch, C.; Wewers, T.; and Theisinger, F. 1998. Organizational-Memory-Based Approach for an Evolutionary Workflow Management System - Concepts and Implementation. In Nunamaker, J., ed., *Procs. of the 31st Hawaii Int. Conf. on System Sciences*, volume 1, 174–183.

Woods, W., and Schmolze, J. 1992. The KL-ONE family. In Lehmann, F., ed., *Semantic Networks in Artificial Intelligence.* Pergamon Press. 133–178.