Information Supply for Business Processes: Coupling Workflow with Document Analysis and Information Retrieval

Andreas Abecker, Ansgar Bernardi, Heiko Maus, Michael Sintek, Claudia Wenzel

German Research Center for Artificial Intelligence (DFKI) GmbH

P.O.Box 2080, D-67608 Kaiserslautern, Germany
Phone:+49 631 205 3582, Fax:+49 631 205 3210

{aabecker,bernardi,maus,sintek,wenzel}@dfki.uni-kl.de

Abstract

Explicit modeling of business processes and their enactment in workflow systems have proved to be valuable in increasing the efficiency of work in organizations. We argue that enacted business processes - that is: workflow management systems - form a solid basis for adequate information support in complex and knowledge-intensive business processes. To support this claim we demonstrate results from two different projects:
The *VirtualOffice* approach employs workflow-context information to support high-precision document analysis and understanding in standard office settings; the combination of workflow context and document analysis allow for the automatic handling of incoming paper mail with respect to the appropriate workflows.
The *KnowMore* approach focuses on the support of people who work on knowledge-intensive tasks by automatic delivery of relevant and goal-specific information. To this end, workflow context, an extended process model, and a detailed modeling of information sources are combined.
Both approaches show ways to proceed from workflow systems towards IT support for active knowledge management.

Introduction

Workflow Management is a widespread technology for automating structured business processes. Today it is mainly used to coordinate complex processes where many activities must be scheduled and dispatched among many possible agents. Further support comes from an integrated handling of application programs used in the process chain and a streamlined passing of application data and electronic documents flowing between different process steps.

As complex business processes rely on intensive information exchange with the company's environment, they are document-driven by nature: Employees deal with and react to information and knowledge transferred by and embedded in all kinds of documents, including forms, letters, books, manuals, records, either electronic or paper-based.

Typically, it is not represented in a WfMS, which knowledge and information needs must be applied in order to perform the activities in a given process step. Nevertheless, this knowledge is accessible somewhere in the company's information space, enabling the employees to do their jobs.

Consequently, one would like the WfMS to automatically offer access to relevant knowledge sources, or to even directly 'pump' information items extracted from incoming documents to the appropriate places in the data models of the actual workflow instance. This vision requires an extensive exchange of information items *and suitable semantic annotations* between workflow, application, and information space. To realize this, the WfMS should possess interfaces for exchanging knowledge items with the surrounding support environment, able to bridge between different conceptualizations and data models.

Such considerations are not subject of today's standard WfMS approaches ((Georgakopoulos, Hornick, & Sheth 1995), (Alonso *et al.* 1997)). Contemporary systems exhibit only a "very thin interface" for data exchange with other office applications such as text processing applications or corporate data bases.

In order to overcome this limitation, this paper will present two different solutions which describe how a true knowledge transfer between business processes and their surrounding information space can be established. Both approaches focus on process-embedded information delivery from documents and fit into a common description frame which is sketched in Figure 1.

The WfMS represents running business processes by workflow instances and executes them by means of a workflow engine. The second system is a mediator system which we call *information provider*. The information provider gets an information request and some additional context information from the WfMS. It accesses the documents by some kind of document index. This index may consist of an inverted index file as required for information retrieval tasks, but it might be any more sophisticated document model as well. The retrieved information is handed over to the WfMS. So, the interface between the WfMS and the information provider comprises three different kinds of information, the in-

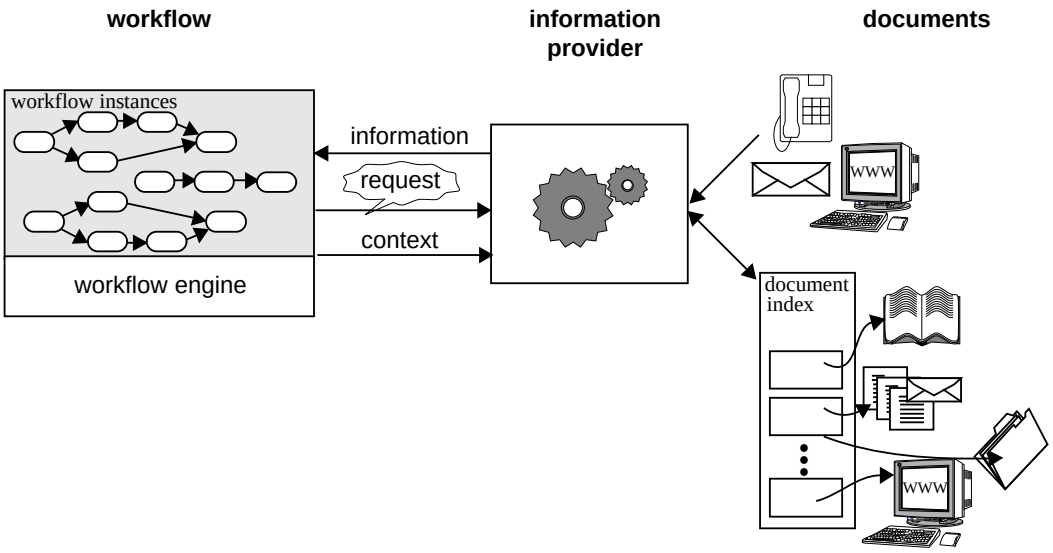

Figure 1: *Information Support for Workflows: A Common Description Frame*

formation request plus supporting context information on one hand, and the retrieved information results supplied to the WfMS on the other hand.

Central to our considerations is the notion of *context* which allows the information provider to perform his tasks proactive, more effective, and more precisely. What context information can be delivered from the WfMS, and, even more, which context information is required to support the information provider's job, highly depends on the actual request to be fulfilled by the information provider. So, the abstract, general description frame of Figure 1 does not help to refine the notion of context in the given scenario. Hence, we consider two concrete instantiations of the general description frame:

The *VirtualOffice* project (Wenzel 1998) presented in the following section integrates paper-based information into arbitrary workflow activities, while the *KnowMore* project (Abecker *et al.* 1998), described in the subsequent section, aims at supporting so-called knowledge-intensive tasks by proactive document delivery.

After presenting these two projects, we review some related work and conclude the paper with a unifying view which shows the commonalities and differences of the two approaches, as different instantiations of the same, then technically refined version of the overall framework above. This unified view also leads to suggestions for the design of future WfMSs.

The VirtualOffice Scenario: Information Support by Paper Documents

Although the paperless office has been a buzzword for many years now, it still has not come into reach. On the contrary, the enactment of business processes by administrative workflows has even complicated the integration of paper documents since such workflows require electronic representations of all documents involved.

Looking a little bit closer, such workflows are characterized by manifold documents which belong to one common process and arrive in a chronological order. Typical examples can be found in insurance companies where initial applications for contracts, changes in the policies, annual invoices, and damage claims, etc., are dealt with. Another example are business trips where the traveller has to fill out an application, the application must be confirmed, some invoices, e.g., for plane tickets, have to be payed in advance and, finally, several receipts must be accounted for. In this paper we will use the purchasing process in a company as the basis for our examples.

The *VirtualOffice* project, conducted at DFKI Kaiserslautern, integrates paper-based information into workflows by the use of a document analysis and understanding (DAU) system (Baumann *et al.* 1997b) as a workflow application. The DAU system in turn benefits from this integration since context information from the workflow system can improve the analysis of business letters and is therefore a crucial point of our research (Baumann *et al.* 1997a).

Figure 2 sketches typical analysis steps and results in DAU. Typically, DAU systems are divided into two parts: The analysis front-end of the system works as provider of uniform information about the characters appearing in a scanned document along with additional logical attributes. This information is interpreted by the system's document understanding components which form the back-end.

The analysis of a scanned document starts with low-level image preprocessing such as skew angle adjustment and upside-down detection. Afterwards, segmen-

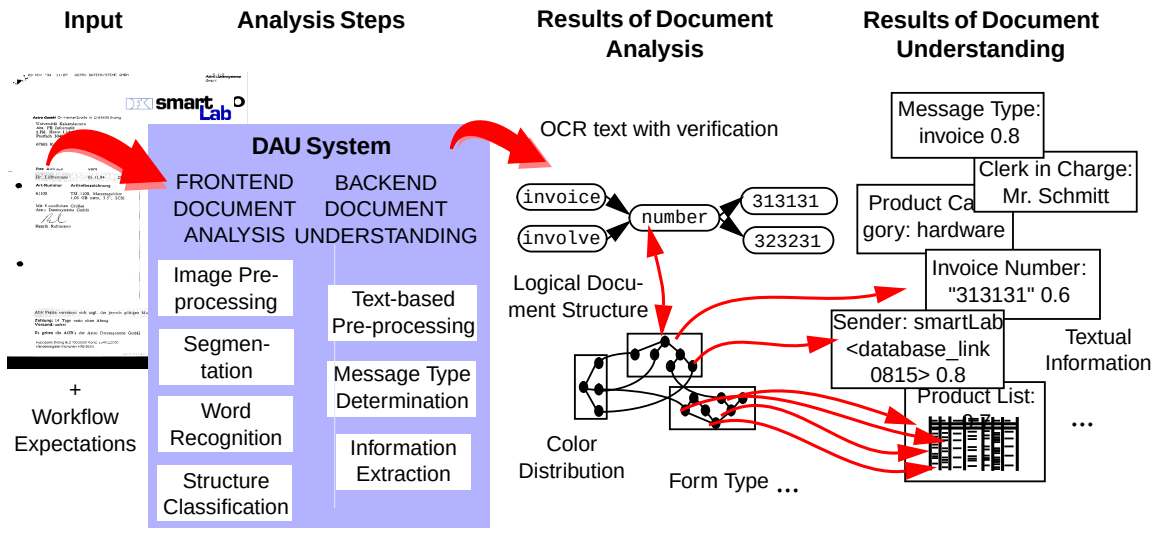

Figure 2: *Document Analysis and Understanding: Steps and Results.*

tation divides the document into geometrically connected components and identifies segments of characters, words, lines, and blocks. Then, text recognition explores the captured text segments, generates character hypotheses, and merges them into word hypotheses. Structure classification takes this given geometric structure to hypothesize the so-called logical objects of a document, e.g., title, author, chapter, etc.

Within *document understanding*, the generated word hypotheses are validated by dictionary look-up in the text-based preprocessing step. Now, the content-based part of analysis is invoked. First, the message type of a document is derived (e.g., confirmation of order by supplier xyz). This information is used to start a more in-depth analysis for the extraction of reference data, products, the date of the letter and so on. For more information on DAU, please refer to (Baumann *et al.* 1997b).

The basic scenario of our integration of DAU into WFMSs fits well into the general architecture presented in Figure 1: The DAU system represents an information provider and a company's mail box represents the collection of documents involved as the relevant part of the company's information space. The task which is requested by a workflow instance represents an information need of this particular instance. In order to support DAU in answering this request, the workflow instance transfers context information to the DAU system. Having finished the task, the DAU system hands over the data requested thus satisfying the information need of the workflow instance.

Within *VirtualOffice*, two different kinds of tasks are distinguished:

1. The task of *process identification* is stated whenever workflows inform DAU that they are waiting for particular documents. In this case, DAU analyses incoming documents in order to determine the correct workflow instance to which the document shall be assigned. This scenario is discussed in the following section and shown in Figure 3.
2. The task of *information extraction* is stated when a particular workflow needs specific information contained in particular documents. Then, DAU analyses the documents in order to identify and deliver exactly the information and data requested. This scenario is subject to the section after next and displayed in Figure 5.

After discussing these two tasks, this chapter concludes with an overview of the design and implementation extensions which must be accomplished for integrating a DAU system as an information provider in the overall scenario.

Task 1: Satisfying an Information Need by Process Identification

Workflow activities within a running process are sometimes triggered by events occuring outside of the workflow. In such a case, the process is in a waiting state until this certain event occurs. Imagine for example that an order has been written to a supplier. Afterwards, the corresponding workflow is waiting for a confirmation of this order by the supplier. Thus, this workflow has an information need which can be satisfied by observing incoming information and by assigning it to the waiting workflow. The DAU performs this task by process identification for incoming documents.

Basically, process identification is a match of the information contained in documents with corresponding data available in the workflow instances. To achieve this, the relevant data in the workflow instances are collected: They specify the analysis task for the DAU system since they define which items the DAU has to search for. Further, these workflow data are important

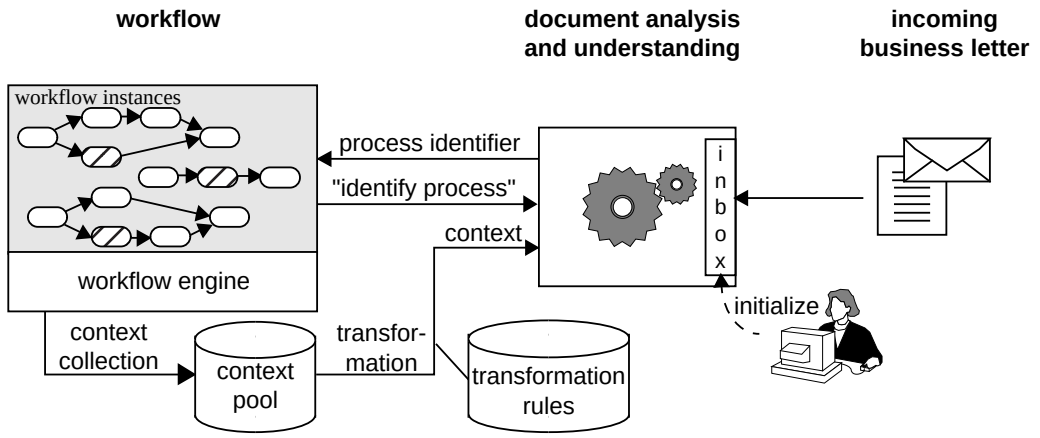

Figure 3: *Information Support by Process Identification for Paper Documents.*

for the result quality of the instance match. If there is a lot of open instances, and a new document has to be assigned, more information about the workflow instances and the expected documents (e.g, document type, sender, products mentioned and even possible references to other events within the same instance) allow for a more accurate matching.

Figure 3 shows process identification as an instantiation of the general scenario presented above: Several activities within different workflow instances state the current context of these instances whenever new information is available. The workflow instances deliver two different kinds of context information:

- One kind of context is delivered by the WfMS and the running workflow execution itself and therefore called *workflow context*. It consists of:
  - workflow-relevant data (e.g., the process number),
  - data coming from the workflow's audit trail (e.g., reference data to preceeding documents), and
  - data from the workflow models (e.g., the name of the next event which provides the return address for the workflow).
- The other kind of context stems from applications invoked from the WfMS (e.g., from the text processing application for writing an order) during the execution of specific workflow activities, and is thus called *task context* (e.g., the supplier's address).

Those context data useful for the DAU analysis task are transferred to the *context pool*, a database located within the DAU system. In the context pool, context information is held until an event occurs in a workflow instance which implies a response by a document (shown by hatched oval activities), i.e., the workflow has an information need which will be satisfied by an incoming document related to the event or workflow. This information need is stated by handing over the request of "process identification" to the DAU system and by formulating its context. This whole collection is called *expectation*. It describes (cp. Figure fig:ExpecGen):

- content and meaning of the expected document,
- a refined specification of the information need (typically a list of data which shall be extracted), and
- some administrative, technical data in order to contact the workflow instance in the case of successful identification.

In order to allow the DAU to interpret such expectations, it is related to the DAU's document ontology (see (Lichter *et al.* 2000)) which describes structure and content of the documents of the domain under consideration.

Expectations are generated by an inference engine which uses the context pool together with a set of *transformation rules*. These rules relate the workflow-context information stored in the context pool to possible contents of an expected document. Hence they transform the data schema used in the workflow and its attached applications into the concepts of the DAU domain ontology. For instance, a rule may state that the receiver of an (already known) order will be the sender of the (expected) corresponding invoice.

All expectations are stored at the information provider's (i.e., the DAU system's) side in the so-called *expectation set*. Since the expectation set stores all expectations of all processes, it contains the whole workflow context relevant from the DAU point of view.

Example: Generating Expectations for Process Determination

Figure 4 illustrates the generation of an expectation for process determination / identification in some more detail. Here, context is collected throughout the execution of workflow instance *Purchase-189*. For instance, after file number generation the number is stored in the context pool as context information about the process. Furthermore, during order arrangement, all available information about the order's content is stored (e.g., the order's recipient). Afterwards, the order is sent and, since an invoice is expected from now on, the fol-

lowing activity states this expectation. Therefore, this activity stores initial data for expectation arrangement in the context pool (section *expecInit3*). This data comprises information about:

- which event is the basis for this expectation (*basedUpon*),
- which references may be taken into account during expectation generation (*references*),
- which tasks the DAU has to perform (*processDetermination*),
- some additional information needs (*get(..)*), and, finally,
- which action to perform in the workflow engine after document arrival (namely, trigger event *InvoiceArrived* in workflow instance *Purchase-189*).

Then the inference engine is invoked with this initial section, loads these data as initial facts in its fact base, and starts with the inference. The rule base used is specialized for the business letter domain in order to arrange a proper expectation for DAU purposes. Therefore, the rules contain domain knowledge such as 'the recipient of an order is the sender of the corresponding invoice'. The expectation arranged is stored in the expectation set and from now on available to DAU.

Access to the expectation set is triggered by any incoming document for which DAU has to perform a process identification. To achieve this, an event listener hands over the new document and the task of process identification. Note that the identification of the corresponding process takes into account all expectations of all workflow instances. Consequently, this task can be seen as inherent to the business letter domain. Technically, these "built-in" information needs are hard-coded in the DAU software. Conceptually, we represented them in Figure 3 by the manual initialization of the "inbox" event listener of the DAU system. The result of process identification is a unique process identifier and, of course, the name of the document which is assigned to this process. The expectation of the correponding instance is deleted after the process identification has been verified within the workflow instance.

This concludes the process identification scenario. Data kept within thecontext pool are deleted when the corresponding process instance has come to an end.

Task 2: Satisfying an Information Need by Information Extraction

Information supply using incoming paper documents is not necessarily finished with handing over a document to the corresponding workflow instance. Rather it is often wished to transfer the relevant information contained in the document into an electronic representation directly accesible by the workflow applications. This goal is addressed by the *information extraction* task. Such an information extraction task may not only be required at process identification time, but a workflow may frequently have additional information needs from a given document after this document has already been assigned to it.

Figure 5 provides a conceptual view on the information extraction scenario: Here, only one workflow instance is of further interest. All documents related to this workflow instance, as well as the already extracted, electronic information and intermediary analysis results available for specific documents, can be accessed via an index connecting workflow instances and document folders. Within the workflow instance considered, the context pool is still updated after process identification whenever new context information is available. If an activity states an information extraction request, a new expectation is generated as described above. The request now includes a list of identifiers (according to the business letter ontology) which shall be extracted from the attached business letter. In contrast to the process identification scenario, the DAU now has not only the raw paper image at its disposal, but it can also access previous analysis results for this document, such as OCR and process identification results.

Implementation Issues

One goal of the *VirtualOffice* project was to build upon commercial WfMSs. [1] Therefore, we have to face some conceptual restrictions, e.g., it is impossible to extend the workflow model in order to provide the necessary context information. Thus, our solution uses the interfaces as provided by the WfMS and adds the workflow context information by a "work-around" which: (i) holds and processes all context information in system parts outside the WfMS; (ii) explicitly models the export of data to be fed into these external databases as parts of the workflow; (iii) adds "semantics" to this data by explicitly relating them to the DAU domain ontology in the transformation rules; and (iv) again, explicitly models the cooperation between "conventional" workflow and DAU using events and workflow-application calls for the requests to the information provider system. This allows us to incorporate any WfMS which is based on the WfMC standard for Interfaces 2/3 (Workflow Management Coalition 1998).

In order to extend a conventional workflow to be able to get an automatic information supply from paper documents, the following working steps must be performed:

- At the *workflow side*, the collecting of context must be modeled at process definition time by including actions which retrieve context data and store them in the context pool. Furthermore, the workflow has to be extended with activities which state expectations and handle documents after their assignment.
- At the *information provider's side*, the ontology for the specification of DAU tasks and the domain ontology for documents must exist. Furthermore, existing

[1] Actually we use the commercial workflow management system of the company Staffware which is one of the market leaders:`http://www.staffware.com`

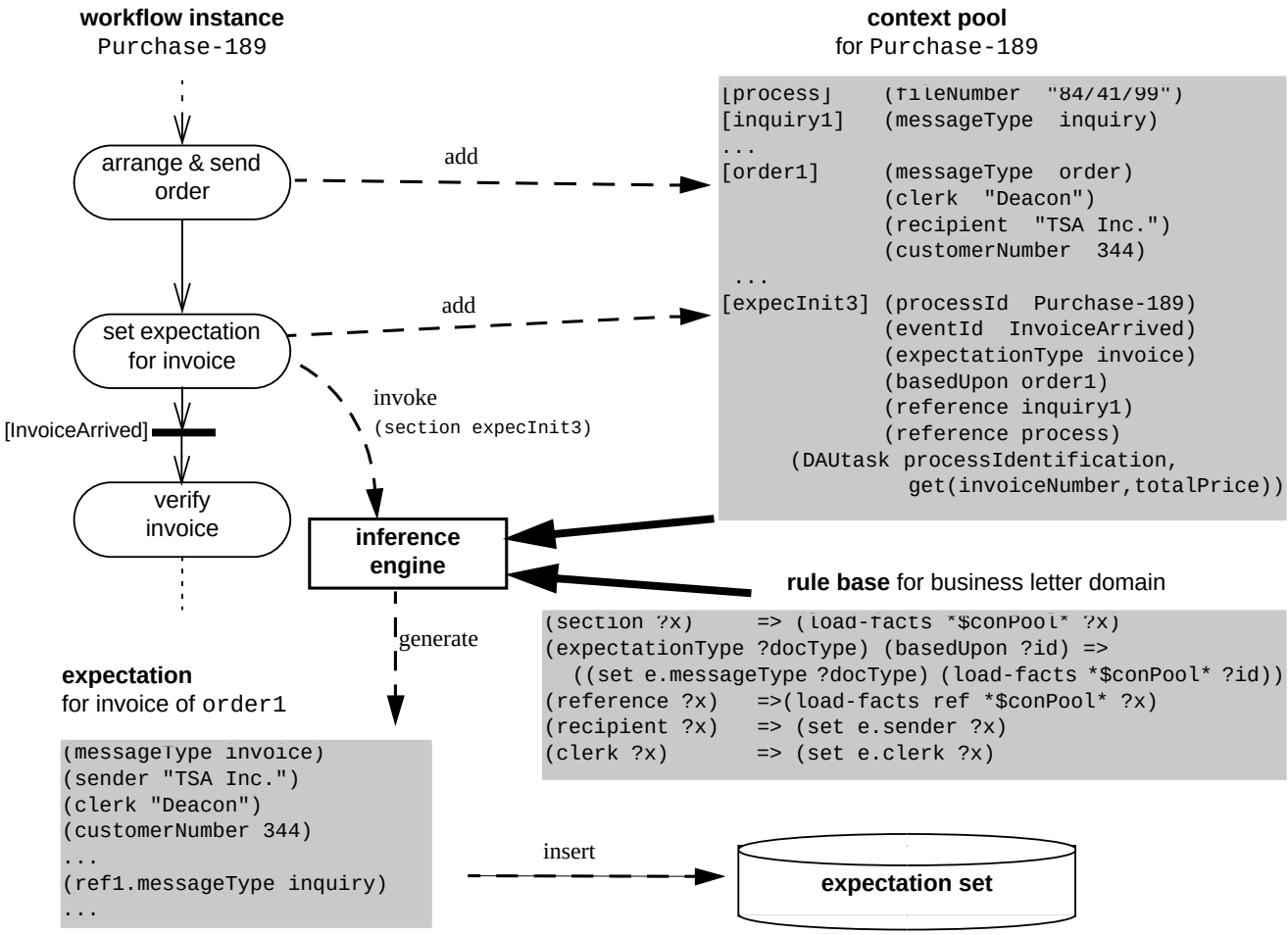

Figure 4: *Expectation Generation.*

DAU algorithms need an interface to use the information stored in the expectation set. Further, the transformation rules between the two system data schemata / ontologies must be specified. These steps are done at system set-up time.

The KnowMore Scenario: Active Information Support by Workflow Integration

The *VirtualOffice* scenario emphasized the use of workflow-context information for improved document analysis and information extraction. The *KnowMore* approach targets on supporting a person working on some knowledge-intensive task by *actively* (i.e., without an explicit, detailed request by the employee) delivering context-sensitive and relevant information.

The KnowMore Scenario in an Example

To illustrate this approach we consider a snippet from a rather simple process: managing a contact to a potential customer at our research institute.[2] After the initial contact (e.g., a telephone call), the relevant topics of interest are identified (for instance, specific technologies or tools potentially useful for the customer, or former projects dealing with similar issues as the customer's problems) and appropriate information material is selected (for example, a technology whitepaper, a brochure about a specific tool, or a project flyer). Having done this, an information package can be sent to the potential customer, whose reaction will then determine the further steps (e.g., arranging a meeting, or terminating the process due to lack of interest). Most of these activities can be considered knowledge-intensive. It is easy to imagine useful support for these activities: When selecting the information material to be sent, active suggestions from the system would be helpful, supposed that the system takes into account the information from the activities done so far, e.g., the selected topics. An automatic, context-aware archiving of the results is also useful when a similar process is started

[2]This example is explained in more detail in (Abecker, Bernardi, & Sintek 2000).

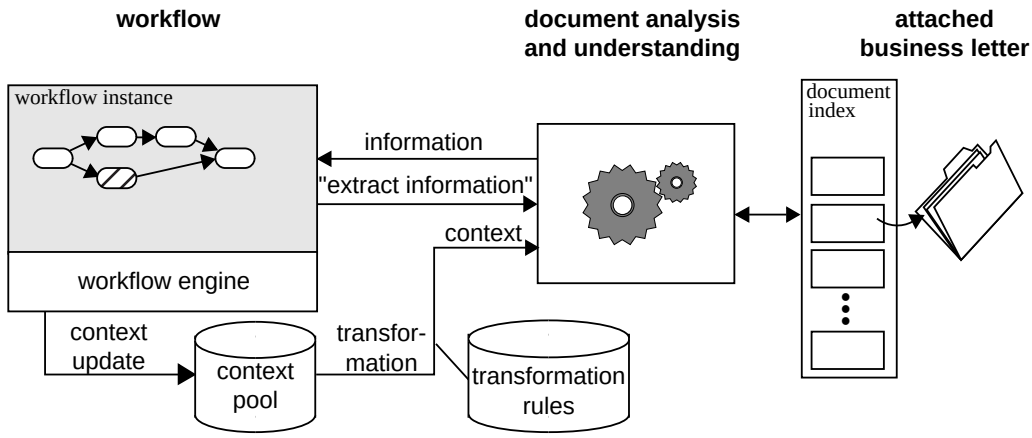

Figure 5: *Information Support by Information Extraction from Paper Documents.*

at a different time and/or location: The initial contact will then profit from information about the earlier contacts to the same company or even the same employee in this company, or about similar cases.

Figure 6 shows a screenshot of our experimental system prototype. On the left, in the background, we see an editor window of the workflow application used to specify relevant information material. To do this, it is necessary to fill the text fields in the input mask. The *KnowMore* system provides support in the following way:

When the workflow engine starts this activity, the system takes the information needs associated to the activity and finds out whether some element in the company's information space[3] is relevant for this job (i.e., whether there is some material which is relevant wrt. the topics identified before). If interesting material has been found, it can be *suggested* for the mail to the potential customer by automatically inserting it into the user's input mask as proposed decision values. In the example, the selected topics were "knowledge management", "KnowMore", a DFKI corporate memory project, and "ESB", another corporate memory project. The suggestions ("Recommendations") comprise (i) general information about the DFKI and the research department for Information Management and Document Analysis, which is always sent in a first contact, and (ii) project flyers for the KnowMore (Corporate Memory) and the ESB projects because these topics could be found directly as indices in the document descriptions. These materials are directly inserted as suggestions in the user's application, because the retrieval agent has some knowledge about different types of retrieval requests and information material and can infer that this kind of concise, public-relations oriented documents is highly relevant for inclusion in initial customer information packages.

Furthermore, the system offers additional material, for example a scientific paper about knowledge management and intellectual capital, because concept-based information retrieval revealed the potential relevance of this document for the selected topic list. However, it is not offered as a recommended value, because only the highest ranked retrieval results above a given relevance treshold shall be recommended automatically.

The user is free to accept or dismiss the recommendations, or to select different material according to personal knowledge. Whatever the choice, the system keeps track of the solutions and the workflow, and records the results automatically, together with the relevant context information. If, sometime later, further material is to be selected because of the customer contact is continued and a refined information need occurs, the system remembers the results of the earlier steps and modifies the suggestions accordingly. Figure 7 shows the support after some material has already been sent: Not only is the list of automatically computed suggestions shortened by the already-used elements, there is also an additional link to the letter which has been written earlier.

Thus the system actively offers supply to the user by providing context-specific relevant information.

Task 3: Active and Context-Aware Information Supply

In order to provide the services described above, the *KnowMore* approach combines an *extended workflow model* with sophisticated *information agents*. As illustrated in Figure 8, the workflow actively asks for retrieving relevant documents. To this end, several modeling activities are needed at process definition time, which facilitate the adequate enactment at runtime.

Process Definition Time.

- **Model business processes.** Model the overall business process with a conventional BPM or work-

[3]In the following, we refer to the entirety of explicitly represented and electronically accessible information in a company or organization also as the *Corporate Memory*, or *Organizational Memory*, shortly OM.

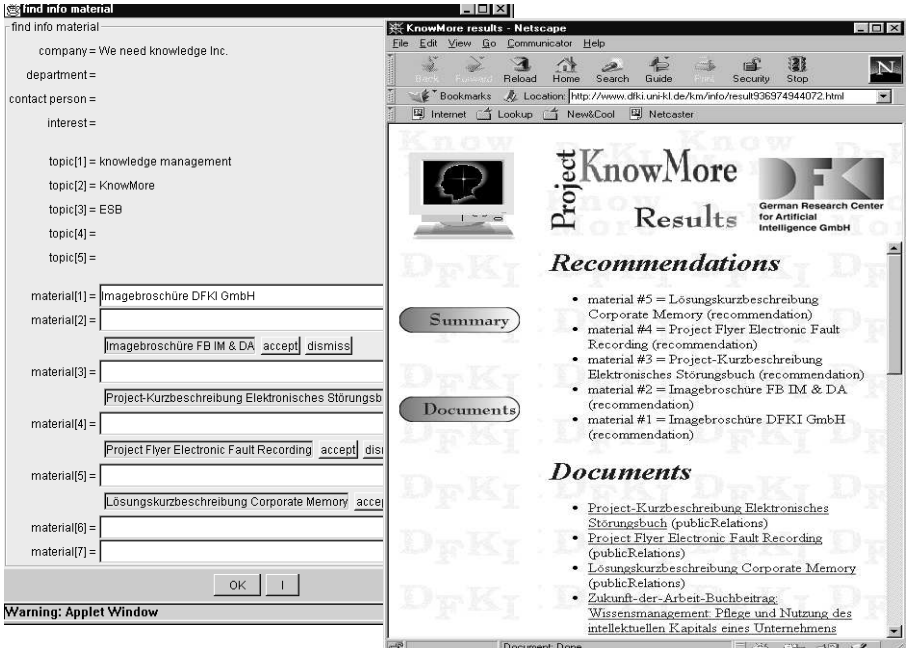

Figure 6: *Active Support by Suggesting Relevant Information Material.*

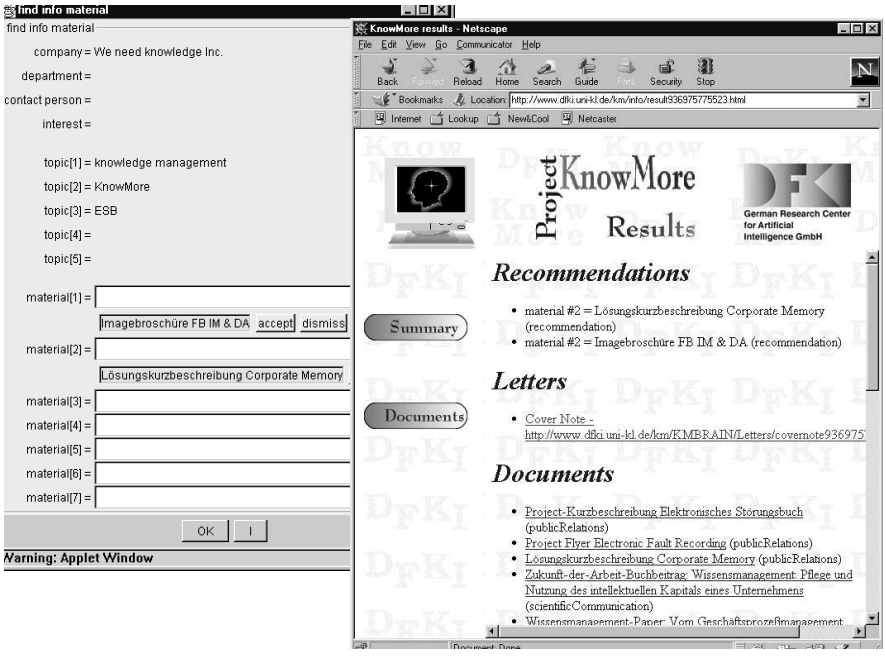

Figure 7: *Context-Aware Support: Information From Earlier Activities is Taken Into Account.*

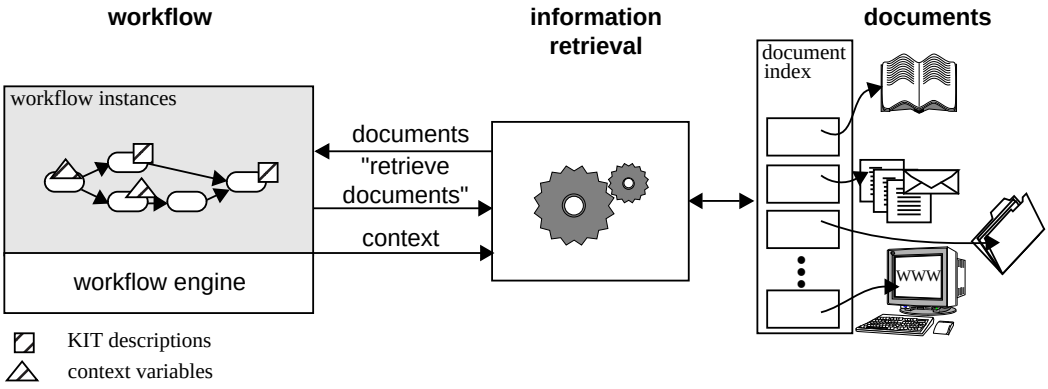

Figure 8: *Information Support by Information Retrieval for Knowledge Intensive Tasks.*

flow tool.

- **Extended modeling of knowledge-intensive tasks**.

  The knowledge-intensive tasks (KIT) among the activities of a particular process require special attention. In order to enable the intended active information support, the KIT representation — illustrated in Figure 8 as a hatched square —extends the conventional description of a workflow activity by a *support specification* mainly describing the respective *information needs* as generic queries or query schemata, together with the information agent responsible for their processing. At runtime, the agent processes the instantiated queries thus delivering relevant information which helps to execute the activity in question.

  The so-defined information needs are in some sense analogue to the information needs described above for the information extraction task. However, instead of exact *data* which are required for specific computations and can be found directly in the documents analysed, here we have a vague human *information* need. The user shall perform the task at hand, and the system may retrieve support material which helps the user, but normally not the problem solution. Supporting documents (except for the recommendations above which are indeed solution suggestions) are typically relevant only to a certain extent, which makes necessary sophisticated IR algorithms, whereas in the *VirtualOffice* information extraction scenario the required data can be found accurately. Another complication compared to the information extraction scenario comes from the fact that there the data to be extracted are themselves a part of the data schema used in the workflow, since they are required for subsequent workflow applications; here, however, workflow-task context may refer to things the user does and works with which are not necessarily represented in the WfMS. Thus, the WfMS data schema must be extended which is described in the following.

- **Attach extended information flow.** In order to instantiate the generic queries at runtime, thus exploiting situation-specific knowledge and context parameters, the retrieval process must have access to the workflow parameters. These parameters are represented in variables which are handled by the workflow environment. As they go typically beyond what is modeled in a conventional workflow specification we talk about KIT variables.[4] They simply describe the information flow between tasks in the workflow. So, they constitute a communication channel between WfMS and information retrieval agents. In Figure 8, hatched triangles indicate additional KIT variables.

  To enable the necessary reasoning for intelligent retrieval, the KIT variables must be embedded into a domain ontology (essentially, this means that their values must be of a type defined as an ontology concept).

  In summary, the KIT variables partially model the data flow in the process and represent the relevant context of the knowledge-intensive activities at runtime.

- **Model information sources.** Typically, an OM contains a variety of knowledge and information sources to be searched and retrieved for active task support. These sources are of different nature, resulting in different structures, access methods, and contents. To enable precise-content retrieval from heterogeneous sources, a representation scheme for uniform knowledge descriptions is needed. To this end, structure and metadata, information content and information context are modeled on the basis of formal ontologies (Abecker *et al.* 1998). The *document index* in Figure 8 is thus realized as a set of descriptions modeling the information sources and facilitating an ontology-based access and retrieval.

Process Enactment Time.

- **Enact knowledge service processes.** Whenever a knowledge-intensive activity is reached during work-

[4]Note that there is no conceptual difference between them and the conventional workflow variables.

flow enactment, the workflow engine not only starts this activity, but also calls an appropriate *information agent* (which is specified in the KIT description).

The information agent performs the actual retrieval of relevant information from the information sources. It relies on domain knowledge – available in a domain ontology – to realize an extended, ontology-based information retrieval, and utilizes the context information from the ongoing workflow — found in the instantiated KIT variables — in order to determine relevant information.

Traversing the formal ontologies according to specified search heuristics (Liao *et al.* 1999), the information agent is able to extend and refine the given queries and to reason about the relevance of available information items.

- **Realize context-aware information storage.** Whenever a workflow activity results in the creation of an information item worth preserving, an appropriate information agent takes into account the current context of the process — available via KIT variables and access to the workflow control data. Thus the information is automatically linked to its creation context and can be retrieved accordingly, should the need arise. [5]

Implementation Issues

Figure 9 illustrates that the realization of the *KnowMore* system fits into the standard architecture of a WfMS as promoted by the workflow management coalition (Workflow Management Coalition 1999): The KIT descriptions are an extension of the workflow relevant data handled by a WfMC-conform workflow engine. The worklist handlers start the relevant activities and trigger in addition the information agents specified in the KIT descriptions. These agents then access the context information in the extended workflow relevant data, evaluate parameters and search heuristics to find relevant information, and offer the results to the user. In spite of the fact that the KnowMore prototype uses a self-developed workflow engine, the results are thus compatible with commercially available workflow implementations.

Unified View

A synoptic view of the scenarios of *VirtualOffice* and *KnowMore*, respectively, allows us to illustrate the key ideas of our approaches for bringing knowledge to business processes. The overall objective tackled in both projects is to support knowledge-intensive activities by providing adequate, automatic access to relevant information, thus realizing a central idea of knowledge management: *knowledge is information made actionable.* This breaks into several roles played by distinguished modules which realize and detail the common description frame presented in this paper. This refined view is presented in Figure 10

The knowledge-intensive activities at hand are performed by suitable **applications**. The exact realization of these applications may cover the whole spectrum from a totally manual activity, e.g., a human decision, to a totally automatic one, e.g., some number-crunching program. In the examples presented here, all activities concern tasks from the office world and deal with tasks like writing letters, identifying objects, deciding about materials described in databases, and so on. Thus, the applications relevant in these scenarios will certainly include word processors, spreadsheets, databases and similar office-type programs. As the intended communication between the application and the other components in our scenario goes beyond what might be available in standard programs, especially given the wish for continous observation of and reaction to user interaction, we note that suitable wrappers will usually be necessary to realize this communication. Although such wrappers may be very sophisticated to realize in some cases, in principle they are a small technical detail.

Each knowledge-intensive activity belongs to some comprehensive business process, thus being integrated into the ongoing work of the enterprise. These business processes are explicitly modeled and enacted by some **workflow management system** (WfMS). The WfMS controls the workflow, initiates activities and starts the respective applications, and provides input data to the applications where needed. Beyond the data offered by the WfMS (which is called **workflow-relevant data** in the WfMC's standard glossary, denoting data necessary for the applications which is accessible and managed by the WfMS) each application might handle its own data resources beyond the realm of the WfMS (conveniently called **application data** by the WfMC). The internal data structures of the WfMS encompass the so-called **workflow control data**, and especially the **modeled business processes**, and the continuous **audit trail** which records the states of all processes handled by the WfMS.

The knowledge-intensive activities are to be supported based on the available information space, which contains information sources of various types and characteristics together with suitable access structures. Details vary between the two scenarios, but we will emphasize some topics below.

At the heart of the scenario is the **intelligent assistant** which bridges between the information space and the knowledge-intensive activities by offering the intended informational support. The goal of the intelligent assistant is to satisfy the knowledge-intensive activity's actual information need - making the information need a key player worth of a more detailed inspection. Depending on the situation, the intelligent assistant has to act on triggers from the respective activities in the workflow or to react on incoming elements

[5] Please note that, essentially, context-aware storage is the fourth task to be discussed. However, we subsume it here under task 3 (information supply) since it is tackled by the same technological provisions.)

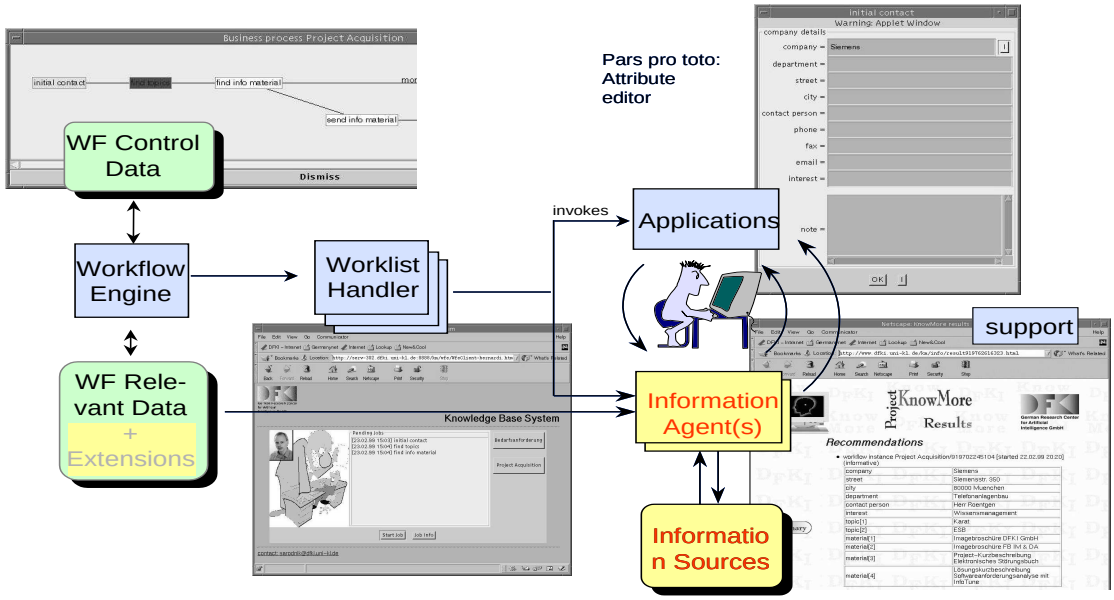

Figure 9: *The KnowMore extensions fit well into the WfMC standard workflow architecture.*

in the information space. While the former can be seen as a straight-forward extension of an activation of an application by the WfMS, the latter requires a specific event listener functionality as part of the intelligent assistant.

Information Need

If the intelligent assistant is to be able to satisfy an application's information need, this need has to be stated in a clear and comprehensive way. We start by restating the notion of an information need as follows: *an information need represents a demand for information necessary for the support and accomplishment of a knowledge-intensive task.*

Obviously, the information need depends on particular tasks. The information necessary for a knowledge-intensive activity comprises the data needed by the application as well as any information which is relevant for the human user. We demonstrated in both scenarios that information needs can be formulated for single process steps in the workflow. The modelling of these task-dependent information needs should be done during process modelling time. Conceptually, process modelling is extended to cover not only *what is to be done* but also *what is needed to do it.*

In addition to the task-dependent information needs the *VirtualOffice* scenario illustrated the existence of task-encompassing, permanent information needs: The system needs to know certain details of any incoming business letter, regardless of the state of active processes. It is worth repeating that these permanent information needs are typically domain-specific, although independent of specific business processes. Thus the modelling of permanent information needs results in a domain-specific configuration of the general system. For example, the definition of the basic reaction patterns for the intelligent assistant's event listener creates a system specific for the *VirtualOffice* scenario.

From the representational point of view it is important to note that an information need can be represented as some kind of query—possibly a sophisticated one. It has to be stated

- What information is needed, i.e. which variables have to be filled?
- How to obtain the information, i.e. which sources to consider and which search heuristics to use?
- How the concrete information need at runtime depends on the context of the actual application and the state of the business process?

Context

The last item in the previous list mentioned another key concept of our approach: At runtime, any information need is interpreted in the actual **context** of the activity at hand. This context comprises task-specific, workflow-specific, and domain-specific aspects.

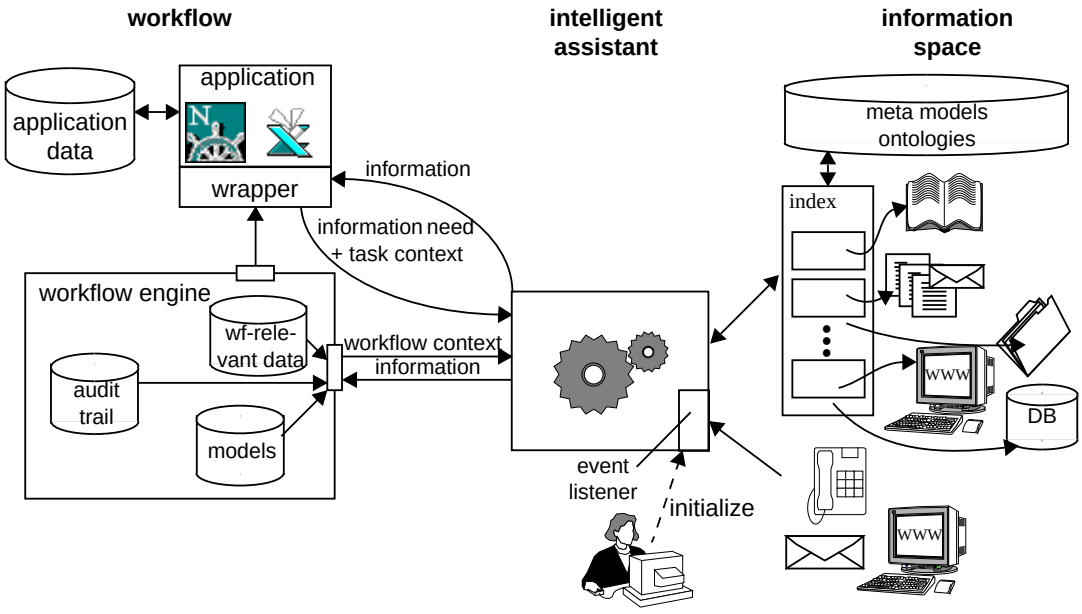

Figure 10: *Role of Context, Information Need, and Ontologies within the Common Description Frame.*

The *task context* is composed of the data available to the application at hand, e.g. the parameters handed over during invocation by the workflow instance, the relevant application data, but also the user interaction observed by the wrapper.

In addition to the task context, the information accessible to the WfMS forms the *workflow context* of the activity. This comprises the business process models or workflow process definitions, and the current state of all active workflow instances (in the workflow control data) as well as their history (found in the audit trail). According to the WfMC standards, these data are accessible via standardized interfaces. Today's commercially available WfMSs, however, usually have their own, more restricted interfaces offering limited access. The *context pool* of the *VirtualOffice* example shows a way to overcome these difficulties by a separate management of the interesting workflow context.

Based on the actual task and workflow context, the information need of a particular activity can be instantiated and passed on to the intelligent assistant which then purchases the necessary information. The *KnowMore* example shows how instantiated information needs are directly used for an ontology-based, heuristic information retrieval from available information sources. The *VirtualOffice* scenario demonstrates the transformation of instantiated information needs into expectations for the DAU components which then extract the necessary information from the documents at hand. The context information helps in restricting the search space by defining the relevance of available information, and selects between different possible interpretations of the instantiated information needs by selecting appropriate conceptualizations/ontologies. For example, if the information need asks for a document about LATEX, only context information enables the intelligent assistant to derive whether to search for Donald Knuth's LaTeX text processing system, or for some elastic materials.

Ontologies

The third key player in our approach is the explicit specification of the conceptualizations used in the various modules by defining suitable ontologies. According to (Chandrasekaran, Josephson, & Benjamins 1999) **ontologies** are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specific domain of knowledge. Thus they provide the terms for describing the information available in a specific domain.

The objective of using ontologies in our scenario is twofold: By referring to modeled ontologies the system components are enabled to translate the expressions created by one component into the realm of another component. For example, an automatic mapping of an information need instantiated in the context of a particular workflow instance onto the terms describing an applicable information source requires access to the ontologies used in both areas. The transformation rules of *VirtualOffice* are the condensation of an ontology-mapping process and perform this mapping at runtime.

Second, the organization of the domain-relevant concepts in suitable ontologies facilitate reasoning about these concepts. Together with the index structures which facilitate the access of the available information sources it is possible to perform a semantically sound, extended information retrieval on this basis: If the content of some information item is described by a set of

concepts from the ontology, and the information need refers to some other concepts which, however, are related to the former concepts in the ontology, a suitable reasoning algorithm will result in finding the appropriate information item. The *KnowMore* example demonstrated how this can be used to augment particular information needs by specific search heuristics which navigate on the ontology in order to retrieve relevant concepts (Liao *et al.* 1999).

Thus, all elements in our scenario, that is, the concepts used in the **document index** for describing the various information items with respect to structure and contents, the terms describing the process models and their instances, and finally all data elements handled by the applications, must be linked to suitable ontologies. Consequently this requires the use of semantically enriched data structures within the workflows. Today's systems do not support this aspect, so we are restricted to modeling conventions and encoding tricks.

Related Work

The idea of coupling a user observation / task management system with a sophisticated information retrieval tool for proactive and context-sensitive information support is becoming more accepted in the last few years. (Budzik & Hammond 2000) present an architecture for *Information Management Assistants (IMAs)* which shall observe users interact with everyday applications and then anticipate their information needs using a model of the task at hand. The idea of assessing the user's work context for starting active delivery and for enhancing information retrieval quality is very close to the *KnowMore* approach, though being restricted to retrieval and not addressing the possibility of context-aware information *storage*. Nevertheless, the *KnowMore* system can be seen as an IMA in the authors' sense. Unfortunately, the ANTICIPATOR module of their system which shall anticipate the user's future information needs on the basis of the actual work context and the stored task model, is not discussed in depth in their published work. However, it seems that they mainly build upon rather shallow, static context models which roughly determine the area of work in order to dissolve linguistic ambiguities in text retrieval. Natural-language aspects play a more important role in their work than worked-out task models as formal objects of consideration. Their WATSON system (Budzik & Hammond 1999) observes interactions with everyday applications like word processors and web browsers, and mainly try to find out linguistic context of occupation. This approach is focused on the *personal* context rather than the dynamic *task* or workflow context and is thus related to user profiling and personal information agents. Thus, it is rather a complementary element to our work than a direct "competitor".

The main focus of our work is on modeling the formal task and workflow context in the sense of *just-in-time knowledge delivery* as suggested by (Cole, Fischer, & Saltzman 1997): 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. Systems aiming at this goal by maintaining a deep understanding of the task at hand are called electronic performance support system. An excellent example is the EULE2 system for knowledge-based assistance in insurance office work (Reimer *et al.* 1998). This system builds upon a deep, formally represented knowledge-base about all relevant concepts to be known and dealt with when working on the insurance office tasks considered. Systems building upon such a deep task model can be called *vertical OM application* in the sense of (Benjamins, Fensel, & Gomez-Perez 1998) which means that they give far-reaching support for crisply defined situations, similar to classic expert system services.

The authors also identified the goal which motivated us to our developments, namely coupling such sophisticated, vertical services with a more or less conventional WfMS in order to have more comprehensive overall support and to get sort of a horizontal system, since EULE2's services can be accessed and linked into a number of freely configurable workflows running in the company.

In (Margelisch *et al.* 1999) the ongoing efforts for realizing such a coupling are described. To this end, the concept of *flow chunks* is introduced, a synchronization mechanism for the conventional WfMS and the concurrently running EULE2 tool assisting in enacting specific activities. The two tools are complementary because the WfMS cares about schedule and coordination of tasks and transport of data and documents,while EULE2 offers single-user support within specific tasks which are modeled in a very fine-grained manner.

Another interesting approach for OM-based support for running workflows, is presented by (Staab & Schnurr 1999). Their ideas are also very close in spirit to the *KnowMore* approach, but do not care about context-aware storage, or information extraction tasks from paper documents. Compared to our system, they explore in more depth the inferential power of ontology-based retrieval on top of the Ontobroker software (Fensel *et al.* 1998) and introduce the notion of context-based views for coupling workflow and retrieval which is the analogue to our information needs. The approach presented seems to be more developed in some respect than ours, for instance concerning XML-based document representations, or a deeper conceptual integration of workflow and service process. They also build on a conventional workflow modeling paradigm extending the well-known Petri Nets approach.

Related work in the area of document analysis and understanding has reached a point where the resulting systems are successfully put on the market. But these DAU systems are mostly standalone applications with generally only one task, e.g., extracting all data from a form or extracting a documents recipient. Furthermore, the context used for analysing a document is static, e.g., given by predefined keywords. Up to now, the use of dy-

namic context as provided by business processes is still a unique feature of the *VirtualOffice* project (Wenzel 1998). Although commercial solutions provide an integration of document analysis and imaging into WfMSs (e.g., FormsRec AIDA solution from ICR[6]), they only offer an isolated analysis of documents, i.e., they do not consider any open workflow instances waiting for a particular document. For example, the COI IntelliDoc solution[7] classifies documents according to a set of predefined keywords. Afterwards, the documents are routed to corresponding WfMS worklists to which the keywords are assigned.

Conclusion: A "Better WfMS World" for Intelligent Assistants

We demonstrated that the combination of business processes and their enactment in WfMSs with the information space of an organizational memory can lead to effective user support. Two different approaches have been presented to illustrate this claim:

The *VirtualOffice* project basically aims at improving the company's information logistics by a smoother integration of paper-based documents into administrative workflows and transfer of information contained in paper documents into the electronic workflows. Thus its service can be applied virtually throughout the whole process and in each process which has interfaces to the external environment of the company where media breaks occur. It directly supports the "object level" of the business process where paper documents are an integral element.

The *KnowMore* project is not so much focused on the object level optimization of whole, and arbitrary workflows. Instead, it concentrates on knowledge-intensive core processes of a business, and knowledge-intensive tasks at the heart of those processes. Not the processes themselves are improved, but a meta level is added, an additional knowledge-service process which helps the user working on her tasks.

Both examples emphasize the use of the business process models and their enactment in the WfMS as indispensable source of context information. Looking in some detail at this common goal, we can extract the following observations:

Successful coupling of workflow and knowledge-intensive application requires some formal semantics as the basis of communication. That is, we need a mapping from the context information modeled in the workflow to the ontology of the information provider. The examples illustrate two principal approaches to this end:

- When the workflow modeling is extended especially with respect to the support of knowledge-intensive activities, it might be possible to consider the available ontology already at process modeling time. The resulting KIT representation is then well-grounded in the ontology of the information provider.
- On the other hand, if the already-defined workflow is to be left untouched, it is possible to add the necessary semantics by defining transformation rules which map the terms already used in the workflow model to the ontology of the information provider.

To effectively access the workflow context information it seems useful to treat the workflow instances as first order citizens in the information world. Two approaches have been presented to realize this goal:

- The explicit modeling of context information in KIT variables extends the data flow model in the workflow. This is suitable if the workflow engine controls when and what is transferred outside.
- The handling of a separate context pool is suitable to deal with situations where a large amount of workflow instances provide volatile context information which is required by activities outside of the workflow's influence. The continuous collection of context information and its organization according to the needs of the supporting application (e.g. the DAU component) even allows to handle context information which the WfMS only creates temporarily and which otherwise might not be available when needed.

In summary, the extension of the workflow scenario is a suitable way to provide active and extended services and to transform available information into process-oriented actionable knowledge. In order to have better ways for efficiently realizing intelligent services like the ones presented in this paper, without a need for clumsy work-arounds, a number of suggestions for the design of future WfMSs can be derived which are indicated in Figure 11 as an evolution of the unified view presented above:

- We need a *better integration* of business process modeling tools and workflow engines such that all things modeled at the abstract level are also easily reflected by the enactment machinery. In the *KnowMore* project, for example, we had comfortable means to define sophisticated business process meta models in the ADONIS tool[8] in order to cope with KIT variables or information needs, but this is only useful for representation and simulation purposes, and cannot be compiled into operational workflows.

  This point would be a prerequisite for a meaningful realization of the issues following below.
- It would make sense to extend business process meta models by *external starting events* which would be a declarative replacement of the hand-coded domain-inherent requests, creating event listener threads at system set-up time which would wait for certain events in the information space and then start the appropriate workflow.

[6] `http://www.icr.de`
[7] `http://www.coi.de`
[8] The ADONIS BPM tool has been developed by BOC GmbH, Vienna (`http://www/boc-eu.com`).

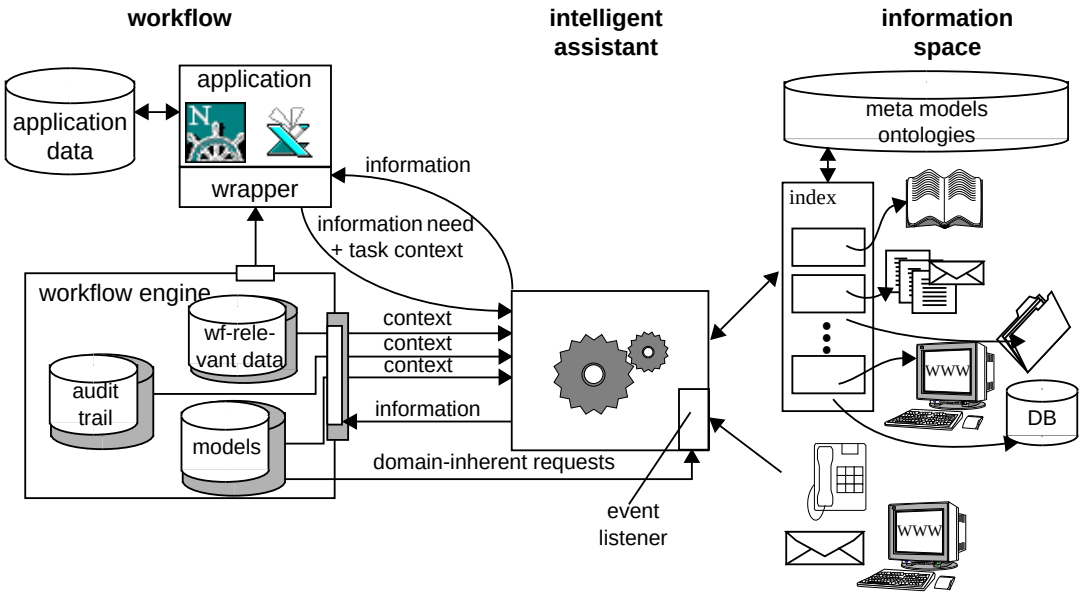

Figure 11: *Coupling Intelligent Assistants with WfMSs: Wishful Thinking.*

- A very important requirement would be to have *more comfortable data structures* (lists, structured objects, e.g., access to arbitrary object types via CORBA) for modeling workflows, i.e., for modeling workflow relevant data (used here for modeling KIT variables) as well as for specifying static call parameters for applications and information assistants. This would ease very much the communication between external assistants and the object level of the WfMS which is now realized by complex export and transformation mechanisms in *VirtualOffice* and by the use of a handmade workflow engine in *KnowMore*.

Acknowledgement The two projects described have been funded by the German Federal Ministry for Education and Research (Bundesministerium für Bildung und Forschung, bmb+f) under grants 01 IW 804 (*KnowMore*) and 01 IW 807 (*VirtualOffice*), respectively.

References

Abecker, A.; Bernardi, A.; and Sintek, M. 2000. Proactive knowledge delivery for enterprise knowledge management. In Ruhe, G., and Bomarius, F., eds., *Learning Software Organizations—Methodology and Applications*. Springer-Verlag. To appear.

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

Alonso, G.; Agrawal, D.; Abbadi, A. E.; and Mohan, C. 1997. Functionality and limitations of current workflow management systems. URL `http://www.inf.ethz.ch/personal/alonso/reference_list.html`.

Baumann, S.; Malburg, M.; auf'm Hofe, H. M.; and Wenzel, C. 1997a. From paper to a corporate memory : A first step. In *KI-97 Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises, Freiburg, Germany*.

Baumann, S.; Ben Hadj Ali, M.; Dengel, A.; Jäger, T.; Malburg, M.; Weigel, A.; and Wenzel, C. 1997b. Message extraction from printed documents - a complete solution. In *Fourth International Conference on Document Analysis and Recognition (ICDAR 97), Ulm, Germany, August 18-20*.

Benjamins, V. R.; Fensel, D.; and Gomez-Perez, A. 1998. Knowledge management through ontologies. In *Proc. of the Second Int. Conference on Practical Aspects of Knowledge Management, PAKM-98, Basel, Switzerland*.

Budzik, J., and Hammond, K. J. 1999. Watson: Anticipating and contextualizing information needs. In *Proceedings of the Sixty-second Annual Meeting of the American Society for Information Science*. Information Today, Inc., Medford, NJ. `http://dent.infolab.nwu.edu/infolab/papers/papersmai`

Budzik, J., and Hammond, K. J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of Intelligent User Interfaces 2000*. ACM Press. `http://dent.infolab.nwu.edu/infolab/papers/papersmai`

Chandrasekaran, B.; Josephson, J.; and Benjamins, V. 1999. What are ontologies, and why do we need them? *IEEE Intelligent Systems* 14(1).

Cole, K.; Fischer, O.; and Saltzman, P. 1997. Just-in-time knowledge delivery. *Communications of the ACM* 40(7).

Fensel, D.; Decker, S.; Erdmann, M.; and Studer, R. 1998. Ontobroker: The very high idea. In *Proc. 11th Int. Florida AI Research Symposium (FLAIRS-98).*

Georgakopoulos, D.; Hornick, M.; and Sheth, A. 1995. An overview of workflow management: From process modeling to workflow automation infrastructure. In *Distributed Databases 3.* Kluwer Academic Publishers, Boston.

Liao, M.; Hinkelmann, K.; Abecker, A.; and Sintek, M. 1999. A competence knowledge base system for the organizational memory. In *XPS-99 - 5. Deutsche Tagung Wissensbasierte Systeme, Würzburg, Germany*, number LNAI 1570 in Lecture Notes in Artificial Intelligence. Berlin, Heidelberg, New York: Springer-Verlag.

Lichter, J.; Maus, H.; Malburg, M.; auf'm Hofe, H. M.; and Wenzel, C. 2000. Coping with intricate applications of document analysis and understanding by extensive use of context. *Int. Journal on Document Analysis and Recognition.* Submitted.

Margelisch, A.; Reimer, U.; Staudt, M.; and Vetterli, T. 1999. Cooperative support for office work in the insurance business. In *Proc. of the 4th International Conference on Cooperative Information Systems (CoopIS'99), Edinburgh, UK.*

Reimer, U.; Margelisch, A.; Novotny, B.; and Vetterli, T. 1998. Eule2: A knowledge-based system for supporting office work. *ACM SIGGROUP Bulletin* 19(1).

Staab, S., and Schnurr, H.-P. 1999. Knowledge and business processes: Approaching an integration. In *OM '99 - Proceedings of the international Workshop on Knowledge Management and Organizational Memory (IJCAI-99). Stockholm, Sweden.*

Wenzel, C. 1998. Integrating information extraction into workflow management systems. In *Natural Language and Information Systems Workshop (NLIS/DEXA 98), Vienna, Austria.*

Workflow Management Coalition. 1998. Workflow Client Application Application Programming Interface (Interface 2 & 3) Specification. WFMC-TC-1009, Version 2.0.

Workflow Management Coalition. 1999. Terminology and Glossary. WFMC-TC-1011, Version 3.0.