Towards a Functional Integration of Document Analysis and Understanding in Workflow Management Systems

Heiko Maus

German Research Center for Artificial Intelligence (DFKI GmbH)
Erwin-Schroedinger-Strasse
P.O. Box 2080
67608 Kaiserslautern
Germany
Heiko.Maus@dfki.de

Abstract. The integration of business applications into workflow management systems (WfMSs) is a recent topic in workflow management. Today, most efforts are concerned mainly with the integration itself. However, considering the information need of knowledge intensive applications, new possibilities arise when using information available in workflows. With the example of document analysis and understanding (DAU) as knowledge intensive business application, this paper introduces a concept for providing context information from workflows – so-called workflow context - to DAU in order to enhance both, namely application and workflow. This concept can be seen as a first step towards a functional integration of these two systems which focuses on the satisfaction of DAU's information need. The paper defines the content of a workflow context, states how it can be arranged in workflows and represented to DAU. The representation uses the concept of expectations which have to be stated by workflows requesting DAU tasks. These expectations place the workflow context at the application's disposal and serve as a link between workflow and DAU. The application of expectations in WfMSs is demonstrated by explaining the necessary modelling efforts in the workflow definition and the execution of an exemplary workflow.

1 Introduction

Nowadays, many solutions can be found for implementing document-based business processes with workflow management systems (WfMSs). Due to the immense amount of paper-based information used in such processes, the electronic representation of paper documents involved, especially a company's mail box, is an important issue. Therefore, the automatic analysis of business letters has developed into an significant business application. Here, document analysis and understanding

(DAU) techniques are used to retrieve information contained in documents[1] in order to provide support in filtering, sorting, and extraction of relevant information. Such a DAU application may be considered as knowledge intensive because the overall analysis task is solved by a complex sequence of single components requiring a lot of knowledge about the documents' domain.

As stated in [1] a WfMS is the set of tools used to design and define workflow processes, the environment in which these processes are executed, and the set of interfaces to users and applications involved in a workflow process. Therefore, a WfMS constitutes a perfect system to support other applications. Now, if document-based business processes are automated by means of workflow management, the integration of DAU bridges the gap between the still paper-based and the automated process steps. However, a simple juxtaposing of DAU and workflow management systems is unsatisfying because the resulting system will not go beyond former applications such as an automatic routing of documents to the corresponding recipient respectively mail box. Instead, with the goal of a functional integration a more powerful solution arises, namely the assignment of incoming documents to the corresponding workflow instances according to information available in the workflows. And as stated in [2] the efficiency of DAU will be significantly increased by using context from business processes. Therefore, workflow context plays an important role in a functional integration. Unfortunately, today's WfMSs do not offer satisfying solutions for this since they only allow for the explicit representation of workflow control data.

To overcome this drawback, this paper presents a concept for the supply of DAU with workflow context in order to allow a first step towards a functional integration and to enable DAU to analyse documents under consideration of the appropriate business processes. Although the paper restricts to DAU as application, the concept presented can also be adapted to other knowledge intensive applications.

The paper starts providing the basis for the concept, then explains the concept itself and concludes with the implementation in WfMSs. The structure in detail: After a short overview of related work, Chapter 2 provides the basis for the paper by introducing to DAU, the features of the DAU system used in the *VirtualOffice* project and the resulting benefits with respect to the intended functional integration. Furthermore, it explains workflow context as a means for this integration. Then, Chapter 3 addresses the representation and arrangement of workflow context and afterwards Chapter 4 deals with the implementation of the presented concept in WfMSs. It describes how to model such a workflow and concludes with the workflow execution showing a purchasing process as example. A conclusion finishes the paper in Chapter 5.

[1] The documents under consideration are traditionally paper documents or fax images. However, electronic documents such as e-mails are also taken into account. This saves low level document analysis steps such as OCR.

1.1 Related Work

Related work covers two different areas: The first one deals with other approaches of integrating DAU into WfMSs, whereas the second one covers approaches which use context information from workflows for integrating applications.

DAU and Integration into WfMSs. 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 document's recipient. Furthermore, the context used for analysing a document is static, e.g., given by predefined keywords. The usage of dynamic context as provided by business processes is - besides our own project *VirtualOffice* (currently conducted at the DFKI Kaiserslautern, see also [3]) - not a subject of research.
Although commercial solutions provide an integration of document analysis and imaging into WfMSs (e.g., FormsRec AIDA solution from ICR [4]), 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 [5] classifies documents according to a set of predefined keywords. Afterwards, the documents are routed to the corresponding WfMS worklists to which the keywords are assigned.

Workflow Context. An approach which uses context information from workflows for the integration of an application can be found in [6] and [7]. Here, the *WoTel* (Workflow and Telecooperation) project integrates a synchronous CSCW application with the help of context information from the workflow instance such as conference participants, agenda, and workflow history.
A different approach for using context information from workflows can be found in [8] where workflow activities are extended in the process definition by so-called *KIT-descriptions* (knowledge intensive task). These descriptions enrich documents handled in workflows with context information such as metaproperties (e.g., author, type, availability), context (e.g., creation process and department), and content (e.g., product, supplier). If a knowledge intensive task is due during a workflow execution and the user needs support, an information provider is requested for relevant documents. The information provider searches with the help of the activity's KIT-description for relevant documents. This is done by comparing the context information in the KIT-description with the documents' attached KIT information from former workflow executions.

2 DAU from an Integration Point of View

Before the concept for representing workflow context to DAU is explained, this Chapter addresses the necessary basics. Therefore, a short overview on DAU as application to integrate is provided and the benefits of the integration are pointed out. Then workflow context is explained and defined.

2.1 Document Analysis and Understanding

Document analysis and understanding is a well known research area with high application impacts [9]. As already mentioned, it has the goal of retrieving information contained in documents. Therefore, the term "DAU" covers two sequential activities: The *document analysis* part starts with a *structure analysis* which produces a layout structure and logical objects such as subject, sender or signature from a document image. The layout structure forms the input for *text recognition*. In this step an OCR (*optical character recognition*) transfers the binary image data into machine readable ASCII text. In order to achieve better results, sophisticated post-processing steps can be applied such as a *dictionary look-up* to compare word hypotheses provided by OCR with (domain-specific) vocabulary to choose the best fit.

The result of document analysis is a well-structured document representation according to a document model. This forms the input for *document understanding* which also consists of different steps such as *morphological text-analysis* (reducing inflected words to their word stems), *message type classification*, *parsing*, and *pattern matching*. These steps are applied to retrieve the document's information and meaning or meaning of its parts' respectively, e.g., the document's message type or facts retrieved from the included free-form text.

DAU System. The DAU system presented in [10] carries out aforementioned steps by specific components which are combined by a DAU control. This control maps DAU tasks to designated plans which will be executed to accomplish the tasks. The system is an ideal basis for the functional integration into WfMSs since it provides different granularities of DAU tasks ranging from a complete analysis of business letters to requests for single information. Therefore, the workflow designer is able to select the necessary integration degree of DAU into a workflow.

Benefits of an Integration. In general, providing DAU with context information from workflows, enables it to work with additional information, i.e., a priori knowledge, more detailed information, or pre-given directions. This results in a reduction of the search space and in an overall enhancement of the recognition rate. Further, the workflow context enables DAU to classify documents according to the provided context and to assign them to the corresponding workflow instances by matching workflow context with document content. Here, workflows benefit from the replacement of the manual assignment of documents and the resulting reduction of transport and idle times. Moreover, instead of transferring required data contained in the documents manually, DAU extracts exactly the data required by a workflow because the information need is declared in the workflow context. Only the verification of extracted data is left to the clerk. Last, the tight application integration allows the workflow to state information extraction requests to DAU and also to obtain requested data throughout its execution. This also supports ad-hoc workflows for which information about required data or documents involved is not available at build time, e.g., suppose an additional invoice check is triggered by a reminder.

Given a DAU system and having the ability to state DAU tasks, it seems that all prerequisites are available for an integration. But as already mentioned, the information need of DAU is not yet satisfied. In order to accomplish this, the remainder of this paper deals with the usage of workflow context as the missing link between DAU and WfMS.

2.2 Requirements for a Functional Integration

From the WfMS's point of view, a DAU system must meet the following requirements in order to establish a functional integration:

- A workflow should be able to state DAU tasks throughout its execution,
- DAU tasks must be definable in every granularity, in order to satisfy the different information needs of workflow instances (e.g., extracting all data in a document or only an invoice number),
- extracted information has to be transferred to the requesting workflow,
- and documents have to be assigned to the corresponding workflow instance.

With a DAU system fulfilling these requirements, it is possible to break down document analysis and understanding into single steps invoked by the WfMS, a crucial point in establishing a functional integration (as proposed in [11] and [12]), i.e., allowing the WfMS to take control over the DAU system. However, problems arise because the DAU control has many tasks and requires knowledge about components, domain and current analysis tasks. Further, in order to accomplish the analysis tasks efficiently, DAU should be dynamic, be able to flexibly exchange components, and be able to choose between alternative analysis paths. Because of this, such a fully functional integration would be beyond the scope of our work. Instead, we focus on satisfying the information need of DAU as a knowledge intensive business application. However, this concept provides a solid basis for a functional integration of DAU in WfMSs.

In order to accomplish such a first step, the above mentioned requirements have to be put in concrete terms:

- Because DAU is knowledge intensive, all information with respect to the intended application domain is valuable. Hence, the *workflows' context* information has to be placed at DAU's disposal.
- The information need of workflows states the *actual tasks* to be performed by DAU, therefore it has to be made known to DAU.
- In order to assign incoming documents to the corresponding workflow, *information about expected documents* has to be stated by workflows.

Throughout this paper these three information pieces required from the workflow side are called *workflow context*. The paper explains the transformation of workflow context to DAU by introducing the concept of expectations and describes a basic concept of the integration of DAU into WfMSs with focus on the workflow management part. For a detailed discussion of the concept with respect to DAU see [13].

2.3 A Definition of Workflow Context

As already mentioned the efficiency of DAU will be significantly increased by using context information from business processes in which the documents to be analysed occur. Because information is contained in and communicated by documents, several statements and assumptions can be drawn by examining information available in business processes and affiliated documents. For instance, it is possible to get the illegible address of a document's sender by extracting the included file number and referring to the corresponding process data. Furthermore, it is possible to state regularities such that an inquiry results in a response by an offer, or assuming a document's structure by identifying the contained company logo.

Furthermore, business processes deal with various data ranging from databases with core data of business cases to information about states of processes. Such information with relations to documents is also a valuable context for DAU. With the automation of business processes through workflows, these information and data sources are available within workflows. Also, workflows can be set in relation to documents by matching the extracted data with available context information from workflows. Therefore, workflow context includes both context information about the documents to be analysed as well as workflow related information such as workflow identification or its information need. Considering this, we get following definition for workflow context:

Workflow context includes all data related to a document with relevance to DAU and all data required for the integration into the corresponding workflow.

As this definition shows, the workflow context is defined depending on DAU needs. Surely, considering knowledge intensive applications in general, this definition is not sufficient and has to be extended. However, this is not intended here. Instead, this definition should give an impression of the content of workflow context for a specific application.

Now we are able to identify workflow context in the workflow.

3 Representation and Arrangement of Workflow Context

The *VirtualOffice* project deals with the question how workflow context from business processes implemented by WfMS can be used for DAU. With respect to WfMSs, this question divides into two problems with relevance to the scope of this paper, namely how it has to be presented and from where workflow context can be obtained. The following paragraphs explain the 'how' question, whereas the remaining question will be addressed in Chapter 4.

Representation of Workflow Context. In order to represent workflow context, we introduce the concept of *expectations*. An expectation is a collection of data retrieved from a workflow instance. It indicates possible content and meaning of a document expected by the workflow, declares the information need, and identifies the workflow. Content and meaning can be retrieved by exploring context information available in

the related workflow. For instance, in order to assume the content of an expected invoice the workflow has to be examined in which the corresponding offer was arranged. Hence, various information is known in advance, such as product list, customer number, and sender of the invoice.

An expectation has to be stated in a workflow every time an event occurs which causes a response by a document (e.g., a letter, an e-mail, or even a internet online form). Such an event could be the sending of an order to a supplier who may answer by using three different document types, namely commitment of order, invoice, and delivery note. This results in three expectations, one for each expected document. See Figure 1 for an example of an expectation.

Content Data

```
MessageType   = invoice
sender        = TSA Inc.
fileNumber    = 84/41/99
productList   = 100,coffee pot 'Java';
                1,coffee brewer 'saeco Pro'
customerNumber= 344
clerk = Deacon
...
```

Reference Data

```
1.messageType = inquiry              2.messageType = offer
  mediaType   = mail                   mediaType   = fax
  date        = 1999-05-11             date        = 1999-05-03
  documentNumber = 34-2                documentNumber = 34-3
  fileNumber = 84/41/99                fileNumber = 84/41/99
  ...
```

Administrative Data

```
ProcessId = CanteenPurchase_34
eventId   = InvoiceArrived
DAUtask   = processDetermination, get(invoiceNumber,totalPrice)
```

Fig. 1. *Exemplary expectation*

An expectation consists of three parts: First, *content data* list data potentially occurring in the expected document. Second, *reference data* list related data such as references[2] to previous documents. Third, *administrative data* identify the originating process and its information need declared by additional information extraction requests (`DAUtask`), i.e., they specify required data which have to be extracted and handed over to the process.

All data are listed as identifier-value pairs. The identifiers used are defined in the logical document structure of the DAU document model, e.g., message type, sender, clerk responsible (see [13] for more details). This representation guarantees the compatibility between information in the workflow and information used by DAU.

[2] Reference data are kept in a further database table, so that in the actual expectations only reference ids are listed.

Expectation Arrangement. An expectation is based on the workflow context. But, what does that mean in concrete terms? Workflow context is contained as data in databases, files, forms, documents and applications. Hence, the first task in expectation arrangement is the collection of all relevant data, in other words, retrieving a value, assigning it to the respective identifier, and storing this information in a designated place. This place is the so-called *context pool*, a database where all context information for one workflow instance is collected. It is divided into several sections, one for each event resulting in one or more expectation (see Figure 2) and each section is identified by an unique event name. A section contains all information related to this particular event. For example, in section `order1` all information is stored related to this specific order.

Having collected the data in the context pool, the next step takes place: The actual generation is done by an inference engine which uses the context pool as a fact base and produces the expectation. The inference engine uses production rules which relate information pieces retrieved from a workflow to contents of an expected document. For instance, a rule states that the sender of an expected invoice is the receiver of the corresponding order, or that the responsible clerk named in an order will be addressed in the greeting phrase of the corresponding invoice. These rules are formalised and represented in a rule base. Here are some exemplary rules for the domain of business letters:

```
(recipient ?x) => (set contentData.sender ?x)
(clerk ?x)  => (set referenceData.clerk ?x)
(expectationType ?doctype ?basesUpon) =>
   ((set contentData.messageType ?doctype)
    (load-facts *$contextPool* ?basesUpon))
```

Thus, an expectation is arranged by filling the context pool and invoking an inference engine which uses production rules to generate a description of the expectation.

Finally, expectations are stored in a designated database, the so-called *expectation set*, which acts as a knowledge base for DAU components. Hence, the expectation set represents the actual state of all workflow instances expecting documents and therefore provides the necessary workflow context to DAU.

4 Using Workflow Context for an Integration of DAU into WfMSs

The previous Chapters revealed the need for workflow context for a proper integration of DAU into WfMSs and introduced the concept of expectations which represent workflow context for DAU purposes.

Now, an environment is available which enables the retrieval of desired information using functionality provided by WfMSs. Generally speaking, DAU integration profits from the workflow management paradigm: ‘Get the right data to the right tool at the right time and for the right people’ [14]. This Chapter addresses implementation aspects, answers the question from where workflow context can be obtained, and explains how an expectation is arranged in a workflow.

The examples used come from an analysis of different WfMSs. Our actual prototype uses the WfMS from Staffware [15].

4.1 Workflow Modelling

In order to integrate DAU into a workflow, we start with the workflow definition of a business process.

Identifying Expectations. During build time knowledge is available about control and data flow, document flow, necessary applications, and occurring information. This facilitates our task to expand the workflow definition with the additional activities of information collection, expectation arrangement (for each expected document), and verification of the assigned document and extracted data. As mentioned, all types of documents which are to be processed are known at build time. Incoming documents have been identified up to a certain level of abstraction (e.g., invoice, delivery note) including their possible contents, their relations to process information and where and how they will be processed in the workflow. Hence, by using knowledge about a workflow‘s document flow all positions can be determined where expectations have to be stated so that the workflow definition can be expanded.

Assignment Handling. We also have to model how a workflow reacts to the assignment of the expected document. This depends on the expressiveness of the workflow model used. For instance, an activity can be suspended after an expectation has been stated and resumed after a document has been assigned (e.g., activity suspend/resume in IBM FlowMark [16]) or the WfMS supports an event handling such as Staffware [14] or COSA Workflow [17]. Similarly, modelling the incorporation of the extracted data in the workflow is also necessary. Since the DAU system defines all possible DAU tasks with their results and data types (e.g., result of the DAU task `get(sender)` is an address data structure), the requested data structure is known at build time and enables therefore a well-defined interface. Here, concepts such as input containers in FlowMark are useful. After this design decision for assignment handling, a verification tool must be inserted. This tool has to be invoked for verifying document assignment and extracted data before incorporating them in the workflow.

Transaction Support. In case an assignment of a document to a workflow instance has been done and during verification it turns out to be an incorrect one, we need some kind of transaction support (e.g., COSA allows to define cancel actions for each action declared in an activity) respectively model constructs which allow a simulation of this behaviour (e.g., loops). Then the DAU system has to be informed and the workflow instance has to be set back to the state where the expectation has been stated.

Data Flow. Since a WfMS is no general database management system as stated in [18], it differentiates between several types of data flow. According to [19], these are *workflow control data* such as state information, *workflow relevant data*[3] such as transition conditions, and *application data* which is application specific and not accessible by WfMSs. In other words, not all necessary data is directly accessible in the workflow itself. This drawback has consequences for our system: The collection of required context information from application data has to be done explicitly by invoking helper applications (e.g. accessing application data in a database) or some kind of agents collecting relevant application data during workflow execution.

However, workflow relevant and workflow control data can be accessed by interfaces provided by WfMSs, especially by interfaces 2/3 as proposed in [20]. Although current commercial WfMSs do not (yet) implement these interfaces, nearly every vendor supports most of the requested functionality in a certain way.

Filling the Context Pool. The context pool for a workflow instance may either be stored in a designated database or available in a file belonging to the respective workflow instance. In order to fill the context pool during workflow execution, we have to identify the required context information and model the filling, e.g., by helper applications. This task is much easier if the WfMS provides any kind of support such as the scripting language of Staffware which allows, among others, direct output of workflow attributes to a file. Furthermore, because we can already make assumptions and statements about expected documents, some values of entries in the context pool are known in advance, e.g., after sending an inquiry the expected document will be an offer referencing a given process number. Such entries can be prepared during modelling. The section is offer1, the message type is order, and the process number is a variable which is instantiated during runtime before storing the entry in the pool (e.g. COSA allows such variable instantiating in command line strings). See Figure 2 for an example of a context pool.

Administrative Information. The workflow's data flow and the process to model already reveal the information need of the workflow during workflow design. Hence, we can declare which document data has to be extracted and handed over to the workflow instance. Furthermore, as mentioned earlier, the workflow's behaviour after a document's assignment has to be modelled. For example, if events are used, the event name to trigger has to be added. Then, the event an expectation is based upon (e.g. order1) and the type of expectation (e.g. invoice) must be stated. During execution, this administrative data has to be added to the context pool in a uniquely named section, for instance, a section name could be a combination of process and activity id. This section name has to be handed over to a specific broker application, the so-called *DAU/WfMS broker*, which has to be invoked by the workflow in order to state an expectation. A more detailed description of the technical architecture is given in [21].

---

[3] defined in [19] as „data used by the WfMS to determine the state transitions of a workflow instance, for example within pre- or post-conditions, transition conditions or workflow participant assignment." Can be manipulated both by workflow applications and WfMS.

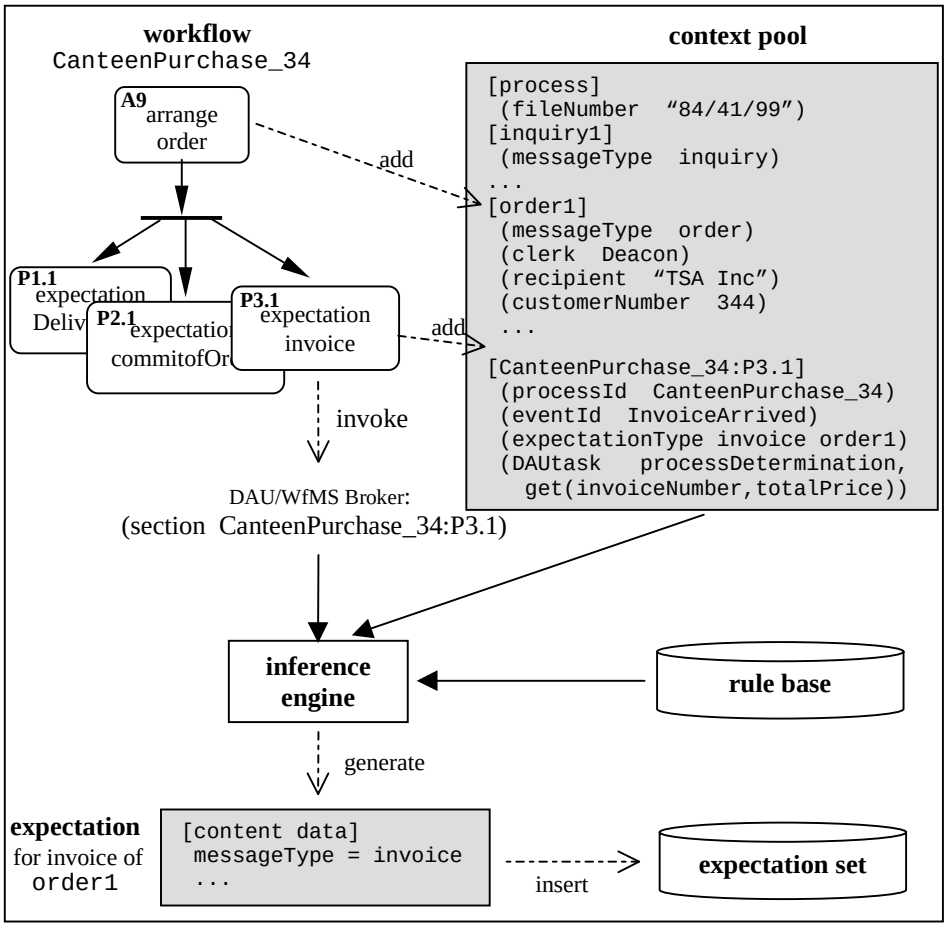

Fig. 2. *Expectation generation*

Having done these preparations during workflow design, the workflow is ready to be executed.

4.2 Workflow Execution

This section explains the actual execution of a workflow under consideration of expectations. Our exemplary workflow is a purchasing process which is started by the arrival of a demands announcement. Then, a clerk has to send requests for offers to different suppliers. Returning offers have to be collected and after a certain time limit the best offer is chosen. The documents involved in this process are forms, offers, requests, invoices, delivery notes, reminders, etc. Figure 3 shows the sequence of phases which will be explained in the following paragraphs.

The Purchasing Process. We enter the workflow at the point where a clerk places an order based upon the best offer he received. The clerk enters all required data for an offer in a form mask or text processing application. Some fields in the form are already filled such as offer number and product list, because some data could be retrieved from the corresponding offer which has already been analysed. The clerk adds further information such as clerk[4] responsible, payment conditions, and maybe some remarks to the supplier. Having finished this, the order is arranged automatically and then sent via a fax server to the supplier's address. From now on, an arrival of a response is possible. This could be a confirmation of order, a delivery note, or an invoice. Therefore, the workflow instance now states this expectation.

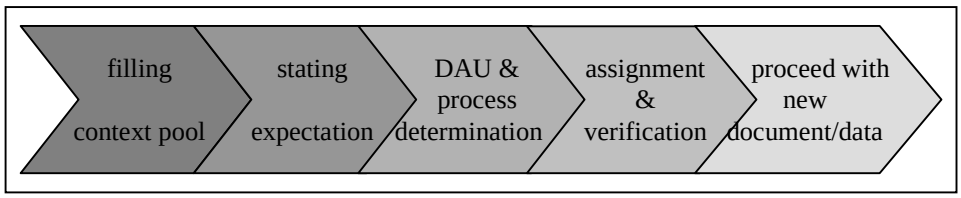

Fig. 3. *Phases for DAU integration*

Stating an Expectation. Throughout the workflow execution the context pool has been filled with information such as the process number or the section for the inquiry. After the order arrangement, a section `order1` is added and filled with data from the order. Afterwards, the activity is executed which will state the expectation. This activity (named `P3.1` in Figure 2) stores its administrative data and information need in the context pool in section `CanteenPurchase_34:P3.1.` Then the activity invokes the broker with the section name as argument. The broker calls the inference engine which produces the actual expectation from the context pool starting with the given section as initial facts by using given rules.

As shown in Figure 1, content data in the new expectation is based on data collected from the order[5]. Because an expectation has been arranged for the supplier's offer as well, these content data are added as reference data. Furthermore, the id of the workflow instance (*CanteenPurchase_34*) and the event to trigger (*InvoiceArrived*) are included as administrative data. Finally, tasks which have to be performed by DAU are specified. First of all, process determination is required in order to obtain the expected document in the workflow instance. For further processing in the workflow, the invoice number and the total price of the invoice are required (information need of the workflow).

Expectations are stated for all expected incoming documents within the workflow instance. In the case of our purchasing process, an order may also be answered by more document types such as a confirmation of order and a delivery note. Therefore, the workflow splits into three branches each coping with one of these documents. Thus, there is one designated activity in each branch which will invoke the

[4] if the current clerk is responsible, this could be done by inserting the current workflow participant automatically

[5] for the sake of simplicity, further data is not mentioned such as sending date or probable medium for the expected document (mail, fax, e-mail, Internet).

DAU/WfMS Broker. After having stated the expectations, the three branches are resumed and wait for events indicating an assignment of an expected document.

Assignment and Verification. Now, assume that the expected invoice arrives. The document is analysed with the help of workflow context residing in the expectation set. Remember that every expectation is a link between an expected incoming document and the corresponding workflow instance. Hence, DAU carries out a process determination by default, in other words, it tries to assign the document to an expectation by matching a document's content with information given in each expectation. Having found a match, a unique document id (e.g., given by a document management system) is inserted in the respective expectation, any listed information extraction requests are accomplished, and the results are stored in a designated database from where the verification tool gets the extracted data. After that, the DAU/WfMS broker is informed by handing over the id of the satisfied expectation. The broker initiates the required action in the workflow instance determined in the expectation. Depending on the workflow design decision, this can be accomplished by resuming the suspended activity (in case an activity id is given) or triggering an event (in case an event name is given).

Then the workflow invokes a verification tool providing the document image, the expectation assigned, and further information from the workflow instance. A clerk checks if the document is the one expected and verifies any requested data by comparing the extracted results with the document. Having accomplished this, the document and its extracted data are transferred to the workflow instance and can be further processed.

In case the document has been assigned to a wrong workflow instance, the document is given back and the workflow is set back to the state after the expectation was stated. But what happens with the document? The verification tool allows to assign the document manually to an expectation. Then the broker assigns the document again. If the clerk is not able to do a manual assignment, a default workflow definition is instantiated whose only task is to assign the document for further investigation, e.g., by routing it to the accounting department if the document type is invoice.

5. Conclusion

This paper presented a concept which provides important means for a functional integration of document analysis and understanding in workflow management systems. The concept supports DAU with context information from workflows and enables workflows to incorporate incoming documents automatically and to retrieve information from them.

It was also shown that both - DAU and WfMS - benefit from a functional integration. As a result, this work leads to a smooth integration of paper-based business processes into workflow enabled ones and also contains the potential to increase the efficiency of business processes.

Furthermore, expectations can be seen as a general knowledge transfer and link between WfMS and legacy applications. For instance, a workflow has sent a reminder, now a payment is expected and a booking application directly informs the corresponding workflow if the payment has been made. The link between account, invoice number and workflow is done in the expectation stated after the reminder. Therefore, different legacy applications which require knowledge respectively workflow context from WfMS can be integrated.

If expectations are considered more generally, other kinds of events can be considered which could trigger an expectation such as telephone calls leading to responses per e-mail. This broadens the operational field, for example, an expectation could forward an expected telephone call in a call center to the corresponding clerk.

DAU is one example for a knowledge intensive business application. Extending the concept to other applications and finding new ways for collecting context information are tasks for future work.

Acknowledgements

The work was supported by the German federal ministry of education, science, research and technology under contract number PT-FKZ 01 IW 807.

References

1. Alonso G., Agrawal D., El Abbadi A., Mohan, C.: Functionality and Limitations of Current Workflow Management Systems. IEEE Expert, 1997
2. Baumann S., Malburg M., Meyer auf'm Hofe H., Wenzel C.: From Paper to a Corporate Memory - A First Step. KI-97 Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises, Freiburg, Germany, 1997
3. Wenzel C.: Integrating Information Extraction into Workflow Management Systems. Natural Language and Information Systems Workshop (NLIS/DEXA 98), Vienna, Austria, August 24-28, 1998.
4. www.icr.de
5. www.coi.de
6. Jablonski S., Böhm M., Schulze W. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen; Facetten einer neuen Technologie. dpunkt.verlag, Heidelberg, 1997
7. Schneider G., Maus H., Dietel C., Scheller-Houy A., Schweitzer J.: Concepts for a Flexibilisation of Workflow Management Systems with respect to Task Adaptable Solutions. In D. E. O'Leary, P. Watkins, (eds). AAAI Workshop: AI in business: AI in electronic Commerce and Reengineering, 1996
8. Abecker A., Bernardi A., Hinkelmann K., Kühn O., Sintek M.: Toward a Technology for Organizational Memories. In: IEEE Intelligent Systems, Volume 13(3), 1998
9. International Association for Pattern Recognition: Proceedings of the 4. International Conference on Document Analysis and Recognition, Ulm, Germany, 1997
10. Baumann S., Ben Hadj Ali M., Dengel A., Jäger T., Malburg M., Weigel A., Wenzel C.: Message Extraction from Printed Documents - A Complete Solution. In [9]

11. Leymann F., Roller D.: Workflow-based applications. In: IBM Systems Journal, Vol. 36, No.1–Application Development, 1997
12. Becker J., zur Mühlen M.:Towards a Classification Framework for Application Granularity in Workflow Management Systems. 11. Conference on Advanced Information Systems Engineering (CAISE ‘99), Heidelberg, Germany, 1999
13. Lauer A., Lichter J., Malburg M., Maus H., Meyer auf’m Hofe H., Wenzel C.: Coping with intricate DAU applications by extensive use of context. Submitted to International Journal of Document Analysis and Recognition
14. Reinwald B.: Workflow-Management in verteilten Systemen. Teubner-Texte zur Informatik Band 7, Teubner, 1993
15. Staffware 97 – Graphical Workflow Definer. Manual, Staffware, 1997
16. IBM FlowMark – Modeling Your Flow. 2.1, IBM, 1995
17. COSA Workflow – Administrator Handbuch. COSA Solutions, 1998
18. Jablonski S., Bussler C.: Workflow Management. Modeling Concepts, Architecture and Implementation. International Thomson Computer Press, 1996
19. Workflow Management Coalition: Terminology & Glossary. WFMC-TC-1011, 3.0, 1999
20. Workflow Management Coalition: Interface 2 - Workflow Client Application Programming Interface. WFMC-TC-1009 – Specification, 2.0, July 98
21. Maus H.: Integration of a DAU System in Workflow Management Systems. In: Proceedings of the European Business Process and Workflow Conference. Geneva, Switzerland. GIGA Information Group, 1998