Attentive Tasks: Process-Driven Document Analysis for Multichannel Documents

Kristin Stamm*†, Andreas Dengel*†
*German Research Center for Artificial Intelligence
†Knowledge-Based Systems Group, Department of Computer Science, University of Kaiserslautern
Kaiserslautern, Germany
Email: {firstname.lastname}@dfki.de

***Abstract*—The increasing amount of email data has led many companies to new challenges with their employees now having to deal with information overload while managing multiple communication channels at the same time, e.g., email, mail, and phone. Moreover, emails can contain attachments, i.e., files with additional information. Most existing approaches for reducing email processing time require significant domain specific customization efforts to achieve good performance and lack attachment handling. We aim at providing a more domain independent approach by integrating the process context and using the information expectations of a process to guide the document analysis (DA) schedule for emails and their attachments. We rely on the concepts of *Attentive Tasks (ATs)* and *Specialist Board (SB)*. *ATs* are templates that describe all relevant and expected information about a process currently waiting for input. The *SB* provides a machine readable description of DA methods, so-called *specialists*, that extract all relevant information for further processes. We present our approach and demonstrate the benefits for a domain specific application, i.e., a financial institution.**

***Index Terms*—process context; multichannel document analysis systems; information extraction; applications**

I. Introduction

Since the introduction of email communication in 1975, enterprises have been registering tremendous challenges caused by this additional communication channel. One of the biggest issues, in our opinion, is the work overload involved in manually processing incoming emails and managing multichannels.

According to Belotti et al., workers' overload is caused by the increasing number of incoming emails and the complexity of email related tasks [1]. Radicati [2] forecasts a doubling of emails sent in 2013 compared to 2009 and estimates for workers an average 25% of daily work time for email processing. Since emails are still processed manually in many enterprises, they are increasing enterprise's costs.

Email, as a new communication channel to the existing channels, also increases the complexity in customer care. Since, each channel is usually handled by a dedicated system that the worker has to manage separately, enterprises seek an integrated multichannel system. To overcome the challenges of increasing email processing costs, it is necessary to provide users with more support in terms of (semi-)automation in email understanding and processing, and more flexibility to the communication channel. Some approaches aim at "understanding" emails by using text analysis on the content of the email but often lack efficiency or quality. For email management, we also reproach the lack of domain independence, applicability and multichannel integration.

We believe that input channels in enterprises should be analyzed holistically to support employees in managing dependencies. Requests in organizations arrive simultaneously through different input channels: mail, fax, email, phone, and eDocs. Depending on the channel, they provide information in the following formats: metadata, text, and document images, e.g., in PDF. In our work, we develop a document analysis (DA) approach applicable to all input channels, focusing on the implementation for the email channel. For this purpose it is necessary not only to consider the email's header and body but also its attachments, that also contain business relevant information. For example, 69% of enterprises use email to exchange electronic invoices or bank data [3]. We expect our approach to be applicable to multiple channels.

We aim at developing an approach that gives enough flexibility to handle the problems caused by email and multichannel communication. Since requests in enterprises often invoke or relate to a task or a process, we suggest combining the idea of task-oriented email management with traditional DA by using the process context to guide DA. Our hypotheses are that (h1) enterprise processes have information expectations towards incoming requests and (h2) analysis results and runtime costs can be improved by integrating a request's process context. We suggest to apply two concepts: (1) the usage of the requests' process context by introducing *Attentive Task (AT)* templates to better define analysis goals and to guide DA. (2) A *Specialist Board (SB)* based on the work of Dengel and Hinkelmann [4] that allows to automatically generate DA programs employing the description and evaluation of specialist methods.

In the following section, we shortly present the related work. We then describe the building blocks of our approach focusing on the concepts of *ATs* and *SB*. In a next step, we explain the results of a preliminary study within an enterprise revealing the need for process-driven DA support. Based on this study, we implemented a prototype and conducted some first evaluations on analysis results and execution time. Finally, we summarize our results and conclude with tasks for future work.

II. Related work

The challenge of email management has been approached in different ways, for example, with task-centric email man-

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.21

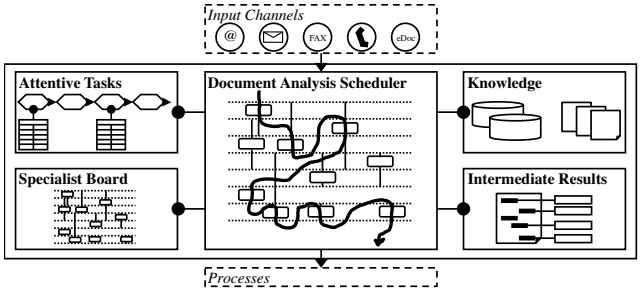

Fig. 1: Building blocks of our process-driven document analysis approach.

TABLE I: Example showing the information slots for the *Attentive Task* LoanRequest while waiting for a process.

Descriptor	Value	Type	Constraints	Slot
SenderName	Ina Mueller	Person	isCustomer	Ident.
SenderEmail	ina@mueller.org	EmailAddress	Related(senderName)	Ident.
LoanType	Dispo	LoanType	{dispo, longTerm}	Other
LoanAmount	?	Money	LargerThan(0)	New
StartDate	?	Date	After(today)	New
EndDate	?	Date	After(startDate)	New

?: New value expected; Ident.: Indentifying

agement or with document analysis (DA) management.

Task-centric email management builds on the fact that most incoming emails within an enterprise environment trigger a task. It is therefore necessary to find the underlying task for each incoming email. Approaches in this research direction relate emails to tasks and use the task context to support the worker [1], [5], [6]. Unfortunately, these approaches have a lot of drawbacks. First, they are often domain specific and require significant customization efforts to support new domains. They further do not rely on task context to extract relevant information, are often focused on email, and do not discuss the integration of the other communication channels.

Another research direction is the generation of DA programs. In the last decades, numerous specific DA methods have been created. DA methods need to be organized sequentially in order to reach the analysis goal. DA programs are often manually designed or taught through machine learning techniques [7]. Researchers in this field are still challenged by high costs and complexity for program design and long execution times. The use of general DA frameworks, like the General Architecture for Text Engineering (GATE), does not perform sufficiently in all domains [8].

Baumgartner et al. [9] and Krishnamurthy et al. [10] address these challenges by extending declarative database languages to the DA domain. By applying database optimization techniques design phases are decreased and program execution time is improved. But since the programs still need to be designed by hand, supporting additional domains remains too expensive and complex.

III. Process-driven document analysis

To solve the challenges of enterprise multichannel management, we extend the document analysis (DA) approach of the *Specialist Board (SB)* introduced by Dengel and Hinkelmann [4]. The main goal of the original *SB* is to enable the automatic generation of an optimized DA plan by describing all available DA methods – the specialists – in a formalized, machine readable way. We extend the *SB* approach as follows: (1) We include information expectations from the process instances towards incoming documents. We formulate these expectations as *Attentive Tasks (ATs)* to create a more precise analysis goal. (2) We use continuous planning to allow the adaptation of the analysis plan, based on intermediate results and relevant *ATs*. (3) We apply this concept to the domain of email and multichannel management. The building blocks of our process-driven DA approach are depicted in Fig. 1. The system processes incoming documents from all input channels, maps them to *ATs*, extracts process-relevant information, and provides the analysis result to the process. The DA scheduler, the system's core element, generates an analysis plan to reach the analysis goal. During iterative planning and analysis execution, the scheduler interacts with the remaining independent blocks: the process context in form of *ATs* that is created independently in the processes, the *SB* containing formalized information about available DA methods, and enterprise knowledge for analysis and planning decisions, as well as a storage for document's intermediate DA results. In the following subsections, we discuss the role and functionalities of each block in detail.

A. Attentive Tasks

The purpose of our *Attentive Task (AT)* approach is to formalize information expectations in the process towards an incoming document. In enterprises, processes represent a sequence of activities necessary to achieve a certain goal. A process is often instantiated by new customer requests or interrupted due to customer interaction, especially in transactional units with high customer interaction. For example, when a customer writes an email to apply for a new loan from his bank, a "new loan" process is invoked. The service employee asks the customer to provide additional information on his request, leading to an interruption of the process execution until the missing information can be found in the customer's response. In both steps, the employee has expectations about which information should be contained in the request - first general information about the loan and then more specific additional information. Based on this, we can create a template describing the expectations. In order to release the worker from actively waiting, these templates are collected and remain active, i.e., "attentive", while waiting for the right incoming request.

We define an *AT* as a schema formalizing knowledge and information expectations by the means of slots. Some examples for such slots are given in Table I. Each slot is build by a descriptor that implies the relation to the process or the

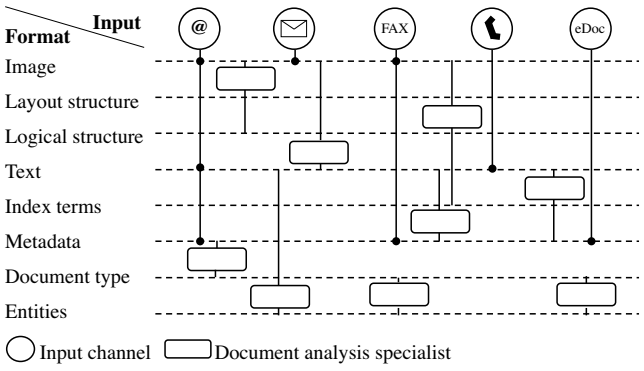

Fig. 2: The Specialist Board.

incoming document, the value of a defined information type, and a set of known constraints about the expected value. We differentiate three different kinds of slots:

- *New information slot* is empty and used to describe expectations towards new information in the document later needed in the process.
- *Identifying slot* contains data to help identifying the process instance, such as, "SenderName=Ina Mueller".
- *Other context slot* contains a value that is not likely to appear in the incoming document.

ATs are generated independently from the DA process and all active instances are collectively available. The scheduler can search them given the currently available DA results.

B. Specialist Board

The *Specialist Board (SB)* aims at making DA specialists methods available in a machine readable way. During DA, the document is transformed in different formats, e.g, from text to layout structure as illustrated in Fig. 2. The approach categorizes specialists according to the transformation they perform. We consider the following structures: image, layout structure, logical structure, text, index terms, metadata, document type, and entities. More details are given in [4].

Additionally to the original approach, we consider that some specialists are better suitable to different input channels than others, e.g., due to the information structure that they provide. For example, mail is provided as image whereas emails additionally provide text and metadata, thus requiring different DA methods. Channels can also vary between similar structures, e.g., fax documents have much lower quality than scanned mail documents. These criteria will therefore be included in the specialist's description.

Table II contains an example of a formal description framework for a specialist. It describes the *accessibility* to the specialist, information to enable automatic *planning*, input and return *parameters* to use the method correctly, as well as information about the *suitability* under given circumstances. To access a specialist, we need name, type, and path to the method. For automatic planning purposes, we need to define which pre- and postconditions need to be met before and after executing the method. This includes especially information about available information structures and channel dependent properties. Further, we need to define which input parameters need to be transferred to the method and which variables will be returned. The suitability measure function is crucial for selecting one from several similar methods. We consider assumptions about the analysis quality, such as precision and recall, towards the defined analysis goal and efficiency measures, like expected runtime or used storage, depending on the goals of the system.

TABLE II: Example of the formal method description for the customer database match specialist.

Accessibility	Name	*CustomerDatabaseMatch*
	Type	*Specialist*
	Path	*Document.analysis.DBMatchMethod*
Planning Information	Precondition 1	*isText(span)*
	Precondition 2	*DatabaseAvailable(db_customer)*
	Postcondition	*CustomerIdentified(span)*
Parameters	Input 1	*FieldList:customerFields*
	Input 2	*String:span*
	Input 3	*List:weightOfCols*
	Output	*FieldList:customerField*
Suitability	Quality	*f(precision, recall)*
	Costs	*f(runtime, storageSpace)*

C. Knowledge

Similar to the worker realizing request processing manually, we need to access additional enterprise knowledge about customers, business partners, and contracts during the DA execution. We use this knowledge to verify DA results and to decide about the further analysis steps. This knowledge can be made available in databases, documents or via interfaces to other enterprise systems.

D. Intermediate results

Intermediate DA results for each document need to be stored and made available to the scheduler and to specialists currently executing. One possible format is the document structure used by the GATE system [8]. A document is stored in its original format and each DA result is stored as annotation to a text span. Additionally, we need to track all information about the current state and the goal state. The DA result is similar to the *AT* structure and consists of a set of slots. For each of these slots we maintain a confidence index about the extracted value.

E. Document Analysis Scheduler

The scheduler uses all available context information in order to plan, execute, and replan DA until all information required in the following process has been extracted. The algorithm breaks down into the following steps:

1) Get the document. Initialize current state and initial empty analysis plan.
2) Prioritize available *ATs* with *Dempster Shafer's Rule* according to the existing evidences.
3) If all *AT* priorities are below threshold, select new *AT* template and instantiate.

TABLE III: Exemplary analysis of 48 documented processes.

Expected Information	Invoke	Inv./Wait	Wait	Other	Sum (%)
Identifying (exist)	–	26	2	–	28 (58%)
Identifying (new)	14	26	–	–	40 (83%)
New	6	18	1	–	25 (52%)
Overall	14	26	2	6	48
(%)	(29%)	(54%)	(4%)	(13%)	(100%)

4) Define DA goal based on current knowledge, i.e., highest prioritized *ATs*, other available information about input channel.
5) Generate DA plan by using *Continuous Partial Order Planning* and the *SB*.
6) Execute next DA method.
7) If DA goal reached, return results, else goto (2).

IV. Application in enterprises

Since our main goal is applicability in enterprises, we like to discuss the relevancy of our approach in enterprises and evaluate the approach in an enterprise motivated test environment. We therefore conducted as preliminary study a process review in a financial institution to examine information expectations in processes. Afterwards, we generated a corpus with test persons to conduct first evaluations with a prototype.

A. Information expectations in business processes

The preliminary study helps understanding the relevancy of our *AT* approach and was carried on a large financial institution with over 5,000 employees world wide. We focused on a service center, where employees mainly interact with customers. We analyzed 48 of the organization's processes that were already documented and available to the employees via intranet. For each process, we identified if it was invoked by an external request (Invoke), if there were activities waiting for external response (Wait), or both (Invoke/Wait). We further examined the information types expected in the incoming request similar to the *AT* slots, i.e., new or existing *identifying* information and *new* information for the enterprise. The main findings of the process analysis are summarized in Table III:

- *Input channels as trigger.* Input channels are the main trigger of processes in this unit. 83% (29%+54%) of the processes are invoked by a request from an external input channel and 58% (54%+4%) have at least one activity that is waiting for a response through an input channel.
- *Information expectations in processes.* Most processes (83%) expect new identifying information at the process instantiation and still 58% use this information later to identify the process instance. 52% of the processes expect new, unknown information from incoming requests.
- *Relevancy of multichannel management.* An additional analysis of requests per communication channel shows that currently telephone (54%) and mail (37%) are the main input channels whereas email is used with only 9% and fax with 1%. In interviews, employees predicted an increase in the email channel and complained about having to deal with channel-specific systems.

We conclude that the process-driven DA approach would be helpful for this kind of organization, allowing the integration of external communication into the internal processes. We have seen information expectations in the process descriptions towards incoming documents. In the next sections, we describe and evaluate a first implementation of our concepts.

B. Corpus

Although our goal is to address multichannel management, we focused in our experiments on the email channel to decrease complexity. However, we aim at extending our approach in a next step to the remaining channels.

A cohesive corpus including email communication threads between customers and a company, and *ATs* expressing the expected information in the related process are required to evaluate our approach. Since no such corpus is currently available, we generated in a first step a test corpus under real world circumstances. We selected two processes from our financial institution partner and asked participants to play the role of ten customers and two service employees. All participants had personal experience with financial institutions and email communication.

The customers had to perform two tasks: (1) Change the owner of their contract. (2) Postpone payments to a new deadline. Based on a brief task description and some fictive contract information, customers had to send email requests to the service center. The service employees reacted to incoming emails according to two process descriptions using answer templates – both based on our case study partner's processes. During two weeks, customers created a corpus of 48 emails: 19 process invokes and 29 with additional information for process instances. 9 emails contained a form as attachment. Due to the open task description, first emails lacked in most cases information to further proceed. The customers, therefore, needed to provide additional information in the next emails. We generated *ATs* based on both - provided and missing information during the conversations.

C. Prototype

We implemented a first prototype in Java to evaluate our process-driven approach. We defined *ATs* and *SB* descriptions in XML format. The DA specialists are a set of standard ANNIE (A Nearly New Information Extraction System) methods from GATE [8] and self-implemented DA methods. Active *ATs* are generated and stored manually in a central folder. The prototype can process emails and their attachments stored in a central inbox folder. The DA schedule can be predefined as a fix pipeline or be generated dynamically during runtime according to the corresponding *ATs*.

D. Experimental setup

The goal of our first evaluations is to better understand how process-driven DA influences performance. We compare our

approach with a general framework and brute force analysis, i.e., execution of all available methods.

The experiments have been conducted on the corpus described previously with four DA scheduling methods:

- *GATE (fix):* Fixed execution of the standard ANNIE pipeline including Default Tokeniser, Default Gazetteer, Sentence Splitter, Part of Speech Tagger, Transducer, OrthoMatcher, and Coreferencer. This pipeline represents a domain independent framework and should reveal how well standard frameworks do perform in new domains.
- *GATE + specialists (fix):* Since we expected the extracted information of the GATE pipeline to be insufficient, we extended the GATE methods with specialists, e.g., regular expression extractor, database matcher, and classifier. This fix order pipeline represents a completes tool set to extract all relevant information.
- *Specialists (fix):* Execution of all domain specialists in fixed order.
- *Specialists (dynamic):* Dynamic DA with specialists based on slots in the corresponding *AT* to evaluate quality and cost performance of the process-driven approach.

For the evaluations, we determine precision Pr, recall Re, and f1 measure $F1$, as well as runtime cost C per data size for a DA pipeline p on a document d as follows:

$$Pr_{p,d} = \frac{|A_d^{rel} \cap A_{p,d}^{ex}|}{|A_{p,d}^{ex}|}, Re_{p,d} = \frac{|A_d^{rel} \cap A_{p,d}^{ex}|}{|A_d^{rel}|}$$

$$F1_{p,d} = 2\frac{Pr_{p,d}Re_{p,d}}{Pr_{p,d} + Re_{p,d}}, C(p,d) = \frac{run_{p,d}}{s_d}$$

where A_d^{rel} are all process-relevant annotations in the document, $A_{p,d}^{ex}$ all extracted annotations (excluding intermediate results), run the runtime, and s the size of file.

E. Evaluation results

The evaluation of the different DA schedules on our test corpus has revealed strong differences in performance. Table IV contains the average results for each scheduling method.

DA with the fix GATE pipeline shows a very low precision of 13% caused by a large amount of analysis results not relevant for the corresponding process. The recall (48%) indicates that the available methods in GATE are not sufficient to extract all information for our special domain. We therefore had to implement additional domain specific methods that meet the processes' expectations. The high runtime costs (594 μs/byte) are caused by many unnecessary DA steps.

TABLE IV: Performance evaluations DA schedule methods.

DA scheduling method	Pr	Re	$F1$	C^1
GATE (fix)	13%	48%	0.19	594
GATE + specialists (fix)	17%	98%	0.27	598
Specialists (fix)	88%	98%	0.92	12
Specialists (dynamic)	90%	98%	0.93	14

1: in μs per byte

Extending the GATE pipeline with domain specific specialists leads to a better recall (98%). Precision (17%) and runtime (598 μs/byte) are not much improved, since the problem of irrelevant DA results has not been addressed. Executing all specialists methods in a pipeline reduces irrelevant DA results tremendously and leads to a better precision (88%) and runtime results (12 μs/byte). The dynamic scheduling of DA methods shows additional improvements in precision (90%).

First evaluation results show that process-driven DA can improve both quality and costs:

1) *Introduction of specialists* help achieving best performance since standard DA frameworks do not suffice.
2) *Reduction of unnecessary DA results* since *ATs* realize specific DA for both the fix and dynamic specialist schedule approaches.
3) *Optimization of DA costs* since using *ATs* optimizes runtime.
4) *Limited use of general frameworks* to overcome lacks in domain specific specialists.

V. Conclusion and future work

We combined the concepts of *Attentive Tasks* and *Specialist Board* to address enterprises' challenges of email overload and multichannel management. Evaluations for a first case study helped us identify information expectations in customer related processes and demonstrate how process-driven dynamic DA planning can improve precision, recall, and runtime costs.

Further investigations are, however, required to validate these first results. We plan to evaluate enterprise's cost reduction as well as quality improvements and repeat the experiments on a larger test corpus in cooperation with our case study partner. At the same time we will extend and evaluate our approach on the remaining input channels.

References

[1] V. Bellotti, N. Ducheneaut, M. Howard, I. Smith, and R. Grinter, "Quality vs. quantity: Email-centric task-management and its relationship with overload," *Human-Computer Interaction*, vol. 20, pp. 1–2, 2005.

[2] Radicati, "Email statistics 2009-2013," The Radicati Group Inc., Tech. Rep., 2009. [Online]. Available: http://www.radicati.com/?p=3237

[3] P. Schmitter, "Rechtliche und technische Fragen im E-Mail-Management," 2005. [Online]. Available: http://www.documanager.de/magazin/artikel_608-print_rechtliche_und_technische_fragen.html

[4] A. Dengel and K. Hinkelmann, "The SPECIALIST BOARD: A Technology Workbench for Document Analysis and Understanding," in *IDPT, 2nd World Conference on Design & Process Technology, Austin, TX, USA*, vol. 2, December 1996, pp. 36–47.

[5] M. Dredze, T. A. Lau, and N. Kushmerick, "Automatically classifying emails into activities," in *IUI*, 2006, pp. 70–77.

[6] N. Kushmerick, T. A. Lau, M. Dredze, and R. Khoussainov, "Activity-Centric Email: A Machine Learning Approach," in *AAAI*. AAAI Press, 2006.

[7] S. Sarawagi, "Information extraction," *Foundations and Trends in Databases*, vol. 1, no. 3, pp. 261–377, 2008.

[8] H. Cunningham *et al.*, *Text Processing with GATE*, version 6 ed., University of Sheffield Department of Computer Science, April 2011. [Online]. Available: http://gate.ac.uk/

[9] R. Baumgartner, S. Flesca, and G. Gottlob, "Declarative information extraction, web crawling, and recursive wrapping with lixto," *Logic Programming and Nonmotonic Reasoning*, pp. 21–41, 2001.

[10] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, and H. Zhu, "SystemT: A System for Declarative Information Extraction," *ACM SIGMOD Record*, vol. 37, no. 4, pp. 7–13, 2009.