Continuous Partial Order Planning for Multichannel Document Analysis: A Process-Driven Approach

Kristin Stamm*†, Marcus Liwicki*† and Andreas Dengel*†
* German Reseach Center for Artificial Intelligence, Kaiserslautern, Germany
† University of Kaiserslautern, Department of Computer Science, Knowledge Management Group, Kaiserslautern, Germany
Email: *firstname.lastname*@dfki.de

***Abstract*—With the rise of email communication, enterprises strive to manage incoming documents from all input channels for achieving customer satisfaction. Their overall goal is to reduce request processing time and to increase processing quality. Previously, we proposed the approach of process-driven document analysis (DA) using the concepts of Attentive Tasks (ATs) and the Specialist Board (SB). The ATs formalize information expectations of the processes toward an incoming document, whereas the SB describes all available DA methods. Here, we propose to apply continuous partial order planning (CPOP) from machine learning for guiding DA with the goal optimal extraction accuracy and runtime. To our knowledge, this approach provides a novel method to integrate knowledge management with DA, in particular for processes. Since planning has not been applied to this field yet, we explore learning the suitability function (SF) and the adaptation of the DA plan. For SF optimization we propose: (1) Suitability measures of runtime, accuracy, and their combination and (2) offline, online, as well as off- + online suitability learning. For planning adaptation strategies we examine: (1) one-time goal setting, (2) continuous current state, and (3) continuous goal adaptation. First evaluations indicate the applicability of the approach and preferences for calibration.**

I. Motivation

The growing use of email communication causes an overload for enterprises in terms of quantity and quality of incoming requests [1]. Enterprise's service employees are challenged by the task of managing an increasing quantity of customer requests arriving through multiple input channels simultaneously. Instead of supporting employees by automating the understanding and processing of incoming requests, todays IT systems lack the integration of the communication process into internal processes and remain fragmented for each channel. Researchers address parts of the problem by proposing document to task mapping approaches or improving document analysis (DA), but no one provided a domain independent approach covering process integration and multichannel handling.

In our previous work, we proposed the approach of process-driven DA for enabling domain independent multichannel management [2]. The approach maps incoming documents to the related task instance and uses the context knowledge of the task, formalized as *Attentive Task (AT)*, to dynamically guide DA. We showed that the approach can improve DA quality and runtime costs. The detailed approach of a document to task mapping in [3] showed that selective use of DA results as input features leads to high quality mapping results. It remains open a detailed understanding and solution for efficiently guiding the DA of incoming documents based on the task's information expectations. For managing DA, we intend to use the *Specialist Board (SB)* introduced by Dengel and Hinkelmann [4]. The SB provides formal DA *specialist* descriptions making them available for DA planning. In this way, it reduces manual program design effort when transferring DA solutions to new domains.

This work aims at addressing the challenges of DA optimization and the potential of adapting DA. We apply the method of continuous partial order planning (CPOP) from machine learning to the field of DA [5]. To our knowledge, this is a new approach to integrate enterprise knowledge management (KM) into DA. We examine, how DA can be transformed into a mapping problem including the definition of states, goals, and actions. For the optimization antagonism, we then explore Suitability Functions (SFs) learning: (1) Suitability measures including *accuracy*, *runtime*, and their *combination*, as well as (2) *offline*, *online*, and *off- + online* learning of the SFs. Regarding the interdependence of mapping and DA, three strategies are examined: (1) One-time goal setting, (2) continuous current state, and (3) continuous goal adaptation. All strategies are evaluated on a corpus of a financial institution toward DA performance measures.

In the following, we embed our approach into related work. Then, we present the overall system of process-driven DA , propose the transfer of DA to the planning domain and detail the planning algorithm. Then, we propose strategies for SF learning and plan adaptation. Finally, we provide evaluations on all concepts and draw conclusions leading to future work.

II. Related Research

This work aims at automating multichannel management by meeting enterprise requirements. The approach should automatically provide all information within an incoming document relevant for further processing. For enterprises, it is necessary to meet three major requirements: (1) low manual DA design effort, (2) input channel and domain independence, and (3) optimized quality and runtime of DA. Researchers often focus on one particular aspect of the problem.

In the area of email management, there are several researchers focusing on the mapping of documents to related tasks, e.g., Bellotti et al. [1] and Kushmerick et al. [6]. The methods range from simple heuristics, such as threads, to sophisticated classification algorithms. All these approaches do not exploit the potential for automating DA.

In the field of DA and Information Extraction (IE), research reviews as from Sarawagi show that there exists a variety of DA methods for each step and domain [7]. Usually, these methods need to be configured or implemented manually and also tied together to one program for one domain. Apart from the high manual effort for program design, optimization is often applied to each method independently. Therefore freely available frameworks were introduced, e.g., GATE [8] provides a modular architecture and a basic IE toolset resulting in lower initial design effort and overarching optimization. Nevertheless, the frameworks requires additional configuration and extension. Our previous work [2] showed that GATE, for example, generates more information results than necessary and does not cover extraction items for specific domains.

Regarding optimization, researchers are progressing by modeling DA programs with declarative expressions and applying optimization techniques from database query optimization, e.g., Krishnamurthy et al. [9]. Despite the promising optimization results, the declarative approaches cover mainly cost optimization and program design remains manual. We conclude, that recent research does not comprise all requirements from enterprise multichannel DA.

III. PROCESS-DRIVEN DOCUMENT ANALYSIS

We briefly present the process-driven document analysis (DA) approach developed in [2] and originated by the work of Maus [10]. We detail the concept of Attentive Tasks (ATs) and AT search, and present the *Specialist Board (SB)*.

A. Overall Algorithm

The system receives documents from different input channels: email, mail, fax and telephone, as well as eDocs. Documents arriving in the system are related to a process instance and all relevant information is extracted with DA methods. The core of the system consists of three main modules that need to managed by a controlling unit: (1) a DA planner, (2) a DA executer, and (3) an AT search module. For each incoming document, we initially generate a DA plan for a set of best performing evidences. Based on this plan, DA is executed and the extraction results are used to generate evidences for AT search. The search module ranks all ATs and returns the best fitting AT. This AT is used for adopting the DA plan and repeating the procedure until all available information has been extracted. Finally, the extraction results are transferred to the related internal process that continues processing the request. The following sections detail the major concepts.

B. Attentive Tasks and Search

Attentive Tasks (ATs) are the formalized information expectations of a process instance toward an incoming document. When a process instance stops and waits for input, an AT is generated, filled with existing information, and added to the pool of AT. An AT consists of a collection of slots where each slot consists of a descriptor, an information type, as well as a value or alternatively constraints that describe the expected value. Table I depicts an exemplary AT representing a change of owner process instance waiting for more information from the customer. It contains specific information of this instance, for example, information about the sender and the name of the new owner. It expects additional information for the process, such as the new owner's date of birth (DoB) and the responsible admission office.

TABLE I: Example of an Attentive Task [3].

Descriptor	Value	Type	Constraints
SenderEmail	anna@blue.org	EmailAddress	in(customer.email)
SenderName	Anna Blue	Person	in(customer.name)
RequestClass	ChangeOfOwner	Class	in(requestClasses)
NewOwnerName	Klaus Mustermann	Person	-
NewOwnerDoB	?	Date	DD.MM.YYYY
AdmissionOffice	?	Organization	in(organizations)

?: New value expected

For searching the corresponding AT, we previously proposed a prioritization algorithm [3]. We use Dempster-Shafer theory to assign a degree of belief to each AT depending on the current DA results. We examined that search performance highly depends on the selection of evidences. Therefore we proposed a structured use of DA results for achieving good reliable search results in a minimum number of search steps. For further details see [3].

C. The Specialist Board for Multichannel Document Analysis

The major challenges of multichannel management are the quantity and complexity of the incoming requests. Furthermore, customers use multiple channels which urges the integration of all channels. One reason for the channel specific IT systems today lies in the differences between the arriving documents including the format as well as their content. A static DA program is difficult and complex in design. Despite the differences, we belief that there are also commonalities between the channels, especially regarding the *document format*: Image, text, and metadata. The input channels mail, telephone, and eDoc provide one document format (image, text from a telephone scriptor, or metadata). Fax, instead provides the image and basic metadata about sender number and reception time. The most complex channel is email providing all three formats: metadata in the header, text in the body, and images from all kinds of attachments.

Dengel and Hinkelmann discovered earlier that DA specialists can be viewed as the transformation from one format to another [4]. Additional to the document formats they add intermediate DA results as layout and logical structure, index terms, document type, as well as entities. We extend this concept including the description of the multiple input channels in Figure 1. Based on the method descriptions, we claim to automatically generate a DA plan that leads to an extraction result. The next section details the choice of planning algorithm as well as the description of the DA problem as planning problem.

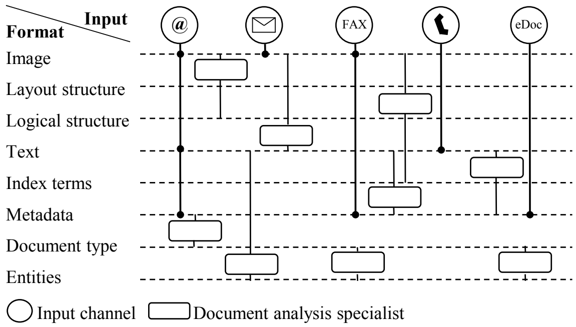

Fig. 1: Specialist Board for multichannel document analysis [4].

IV. Planning in Document Analysis

Planning comprises the generation of a action sequence to transform the current state into the goal state. In multichannel management, this means to transform the current knowledge about a request into the expected knowledge by applying document analysis (DA) methods. We select an appropriate planning algorithm, present a knowledge formalization for states and DA methods, and present strategies for suitability learning and plan adaptation.

A. Planning Algorithms

AI and machine learning provide a broad choice of planning algorithms [5]. For the DA we chose the *partial order planning (POP)* algorithm since it breaks down the planning problem and allows parallel execution of actions if they are independent as appearing for DA specialists. During document analysis (DA) the initially generated plan can become obsolete when the current state develops unexpectedly or the goal state is changed. To handle these uncertainties, we use *continuous partial order planning (CPOP)* that allows to adopt states and the plan. The overall algorithm is outlined in the following:

Algorithm 1 Analyze documents with CPOP.

function ANALYZEDOCUMENT(Document doc, List aTasks, List initialEvids, List methods)
  $initialState \leftarrow generateInitialState(doc)$
  $goalState \leftarrow generateGoalState(initialEvids, null)$
  while initialState != goalState AND planChanges() **do**
    $plan \leftarrow CPOP(plan, initialState, goalState, methods)$
    $plan.executeNext(doc)$
    $initialState \leftarrow generateCurrentState(doc)$
    $aTask \leftarrow searchATs(doc.getEvidences(), aTasks)$
    $goalState \leftarrow generateGoalState(initialEvids, aTask)$
  end while
  $aTask.fillWithResults(doc)$
  return $aTask$
end function

The initial state *initialState* is generated based on the current knowledge about the document *doc*. The goal state *goalState* is initialized with a list of predefined evidences for search. Based on the states and the available methods, CPOP adopts a *plan*. The first method in the plan is executed. The initial state is updated to the current state of the document knowledge. This knowledge is used as input for the AT search [3]. The goal state is adapted according to the *aTask*. The plan is adapted according to the changes in initial and goal state until both match or no more methods can be added.

B. State Representation

The description of states and methods is crucial for applying planning to a domain. For sufficient expressiveness, we chose the Action Description Language (ADL) over the Stanford Research Institute Problem Solver (STRIPS) language. ADL allows to use variables for unknown values and allows knowledge (open world). (see [5])

In DA the current state describes the actual document knowledge including metadata and the DA annotations. Initially, only metadata is available. For example, if a customer sends an email to request a change of contract owner, the current state looks as follows:

```
InputChannel("Email":Channel, d:Doc)
Available("Metadata":Format, d:Doc)
Available("Text":Format, d:Doc)
```

After performing request classification and sender recognition is extended to:

```
InputChannel("Email":Channel, d:Doc)
Available("Metadata":Format, d:Doc)
Available("Text":Format, d:Doc)
Annot("ChangeOfOwner":Class, "Class":type, d:Doc)
Annot("Ina Mueller":Person, "SenderName":type, d:Doc)
```

The goal state defines the desired DA results. In DA, the goal state has to be initialized without knowledge about the corresponding AT. With the first DA results, we prioritize the ATs and use the matching AT for creating a more specific goal state. For initializing the goal state, we propose to use as list of well performing evidence types [3]. In the example, the request class and the sender's name are the best performing:

```
Annot(c:Class, "Class":type, d:Doc)
Annot(p:Person, "SenderName":type, d:Doc)
```

Assuming the AT search returns the AT example depicted in Table I. The corresponding goal state is:

```
Annot("anna@blue.org":EmailAddress, "SenderEmail":type, d:Doc)
Annot("Anna Blue":Person, "SenderName": type, d:Doc)
Annot("ChangeOfOwner":Class, "Class": type, d:Doc)
Annot("Klaus Mustermann":Person, "NewOwnerName":type, d:Doc)
Annot(a:Date, "NewOwnerDoB":type, d:Doc)
Annot(o:Organization, "AdmissionOffice":type, d:Doc)
```

Note that variables are applied for expected informations.

C. Method Description and Suitability Learning

The DA methods or *specialists* are described for planning. We build on the description categories proposed by Dengel and Hinkelmann for the Specialist Board (SB): Accessibility, parameters, planning (pre- and postconditions), as well as suitability [4]. Table II depicts an exemplary description of the specialist *CustomerDatabaseMatch* that extracts customer

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

Accessibility	Name	*CustomerDatabaseMatch*
	Path	*Specialists.DBMatchMethod*
Parameters	Input	*text:String* *doc:Document* *customerDB:FileName*
	Output	*doc:Document*
Planning	Precond.	*Available("Text":format, d:Doc)* *Available("Metadata":format, d:Doc)*
	Postcond.	*Annot(n:Person, "SenderName":type, d:doc)* *Annot(e:Email, "SenderEmail":type, d:doc)* *Annot(s:Street, "SenderStreet":type, d:doc)* *Annot(c:City, "SenderCity":type, d:doc)* *Annot(o:Country, "SenderCountry":type, d:doc)*
Suitability	Quality	$SF_{acc}(doc) = avgAccuracy$
	Costs	$SF_{runtime}(doc) = avgRuntime$

data by matching text against a database. Dynamic access and parameter setting of a method can be easily implemented in programming languages, e.g., Java's *ClassLoader*. According to the SB a specialist transforms one document format into another concerning one or several channels. Therefore, we include this information in the pre- and postconditions. A review of DA methods, such as in GATE [8], shows that on entity level more differentiation is necessary to effectively use the methods. The pre- and postconditions should contain which entity types the method requires and extracts.

The suitability function aims at the optimization of the plan towards a measure. This measure can be a quality measure, such as *accuracy*, or a cost measure, such as *runtime*. Often enterprises want to achieve good quality extractions in a minimum amount of time. We belief that for some methods both measures can be orthogonal and a trade off needs to be found. We define three types of the suitability function:

1) **Runtime.** We use the average runtime *run* of the method m over all training samples S. We set the suitability in relation to the maximum runtime over all methods *maxRun* and substract this value from 1:
$$SF_{run}(m) = 1 - \frac{\sum_{s \in S} run(m,s)/|S|}{maxRun} \tag{1}$$
2) **Accuracy.** We use the average accuracy *acc* of a method m concerning the predicted postconditions *post*:
$$SF_{acc}(m) = \sum_{s \in S} acc(m,s,post)/|S| \tag{2}$$
The initial calculation of the accuracy suitability is costly since it requires the manual annotation of the test set S.
3) **Combined.** When optimizing toward accuracy and runtime, we weight both suitability functions with w_c:
$$SF_{combine}(m) = w_c SF_{run}(m) + (1 - w_c) SF_{acc}(m) \tag{3}$$

The suitability is only calculated, when at least one of the postconditions of the method meets preconditions of the goal.

For learning the suitability functions, one can differentiate between *offline*, *online*, and *off- + online* learning strategies. Offline learning describes the pre-evaluation of each method on a test corpus. Online learning comprises the adaptation of the method's suitability function during runtime. Off- + online learning combines of both. It is necessary to investigate the choice of suitability function and learning strategy.

D. Planning Adaptation Strategies

Choosing continuous planning algorithm enables flexibility to changes in the current state and the goal. The current state differs from planning expectations if the applied method is unable to extract the required information or the information is not contained in the document. The planning goal evolves during DA and AT search if the matching AT is changed. We propose three adaptation strategies:

1) **One-time goal.** The initial goal is used to extract the search features. The goal is adapted one time based on the information expectations of the matching AT.
2) **Continuous current state.** 1) and the current state is updated after each method execution.
3) **Continuous goal state.** 2) and search is performed after each method execution. In case of a new matching AT the goal is adapted.

We belief that 2) and 3) increase the robustness of the DA system. It needs to be evaluated which impact they have on DA performance and costs.

V. EVALUATION OF PLANNING

We conduct evaluations with the planning approach focusing on suitability learning and planning adaptation strategies.

A. Experimental Setup

We evaluate our approach on a corpus generated from two business processes of a financial institution. The corpus comprises 49 emails in 19 communication threads from probands that conducted requests toward a bank. The corpus was annotated and ATs have been generated accordingly. We focus on email communication here since it entails all document formats, but we emphasize that the approach can be extended for any channel. For evaluation, a prototype performs search, planning, and document analysis (DA). For DA we provide 23 specialist methods, where 9 extract unique information and 14 provide overlapping functionalities for 6 entity groups varying in quality and runtime. With the goal of DA optimization and robustness, the evaluations focus on two areas

Suitability Learning Strategies. We evaluate *offline*, *online*, and *off- + online* suitability learning with the target of optimizing runtime, accuracy, and their combination. We compare the results to the case for not applying suitability learning (*none*). As evaluation measures are used runtime, accuracy, precision, and recall. For the *combination*, we vary the weight w_c.

Planning Adaptation Strategies. The *one-time goal*, *continuous current*, and *continuous goal state* planning strategies are evaluated in random AT setups and repeated 1,000 times toward runtime, accuracy, precision, and recall. We compare methods with combined and without suitability.

TABLE III: Evaluation of offline, online, and off- + online learning.

Optimization	Offline				Online				Off- + online			
measure	*run*[1]	*acc*	*prec*	*rec*	*run*[1]	*acc*	*prec*	*rec*	*run*[1]	*acc*	*prec*	*rec*
none	2,234	0.87	0.97	0.89	-	-	-	-	-	-	-	-
runtime	164	0.90	0.98	0.92	218	0.84	0.97	0.87	161	0.90	0.98	0.92
accuracy	2,228	0.93	0.97	0.96	1,145	0.91	0.98	0.93	866	0.91	0.97	0.93

1: in ms

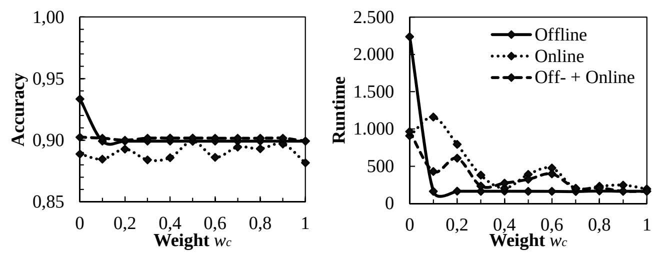

Fig. 2: Combination of accuracy and runtime with different weights.

B. Suitability Learning Strategy

Table III depicts the results for runtime and accuracy optimization and Figure 2 for combined optimization:

Optimization potential. Learning can improve results clearly. In comparison to *none* learning, runtime is reduced to 7% (from 2,234 to 161 ms) and accuracy is improved by 4 - 6 percentage points (from 0.86 to 0.91-0.93). Since precision is relatively stable here, recall is the major driver for accuracy.

Combination weight. For offline learning, the introduction of a weight (> 0.1) leads to runtime improvements (from 2,236 to $\sim$165 ms), but also a deterioration of accuracy (from 0.93 to 0.90). Detailed investigations show that the stability is caused by a stable suitability order within the overlapping specialists. The use of online learning leads to instable results. The combination leads at least to stable accuracy and stable runtime for a weight (> 0.7).

Strategy performance. All strategies perform similar for runtime optimization. Offline learning outperforms the others for accuracy (0.93 vs 0.91). For combination, online learning leads to lower and instable performance. Off- and online learning compensates the instabilities given a calibrated weight.

We conclude that combined optimization is simple to calibrate and can lead to good results when the enterprise accepts minor deteriorations. For learning strategy, we recommend a combination of off- and online training for ensuring adaptability to change. Since offline training is more costly concerning manual annotations, a strategy of offline runtime and online accuracy and runtime learning can be a solution.

C. Planning Adaptation Strategies

Table IV depicts the results for all planning adaptation strategies with and without the application of suitability:

DA quality. For trained methods, the application of current and goal state leads to decreased accuracy (from 0.91 to 0.87) and precision (from 0.98 to 0.92) driven by the extraction of unnecessary information (false positives). For untrained methods instead, accuracy remains stable ($\sim$ 0.89) equilibrated

TABLE IV: Evaluation of planning adaptation strategies.

	One-time goal				Contin. current state				Contin. goal			
Suitability	*run*[1]	*acc*	*prec*	*rec*	*run*[1]	*acc*	*prec*	*rec*	*run*[1]	*acc*	*prec*	*rec*
combined	137	0.91	0.98	0.92	971	0.87	0.92	0.94	740	0.87	0.92	0.94
none	1,763	0.88	0.98	0.90	2,395	0.89	0.92	0.97	2,067	0.89	0.92	0.97

1: in ms

by decreasing precision (from 0.98 to 0.92) and increasing recall (from 0.90 to 0.97). So more information is extracted (true positives) to the price of more unnecessary results.

Runtime. Runtime increases in both cases due to the use of more DA methods in case of adaptation.

We conclude that the continuous adaptation of current and goal state needs to be applied only in selected cases and depending on the enterprises optimization preference. The one-time goal strategy is more efficient and returns high quality results for calibrated methods. Continuous adaptation of the current state can outbalance errors in method suitability when the enterprise prefers higher recall to the cost of precision. The adaptation of the goal seems unnecessary, since search results are equally good or better after one iteration [3].

VI. Conclusion and Future Work

We proposed a novel approach of process-driven document analysis (DA) for multichannel management closing the KM gap between processes and DA. For DA flexibility we built the SB and applied CPOP for DA scheduling. We transfered DA to the domain of planning. Evaluations of suitability learning showed that combined optimization is simple and a combination of off- + online learning recommendable. For planning adaptation, we showed that the flexibility to errors and change costs optimal performance. The choice of strategy depends on the enterprise's preference. In future, further investigation of robustness can improve the over all planning algorithm. Further we will evaluate in different domains on a larger corpus and could extend the approach for filling knowledge gaps, for example, in ontologies.

References

[1] V. Bellotti *et al.*, "Quality vs. quantity: Email-centric task-management and its relationship with overload," *HCI*, vol. 20, 2005.

[2] K. Stamm and A. Dengel, "Attentive Tasks: Process-Driven Document Analysis for Multichannel Documents," *Proceedings of DAS 2012*, 2012.

[3] ——, "Searching Attentive Tasks with Document Analysis Evidences and Dempster-Shafer Theory," *Proceedings of ICPR2012*, 2012.

[4] A. Dengel and K. Hinkelmann, "The SPECIALIST BOARD - A Technology Workbench for Document Analysis and Understanding," in *IDPT*, vol. 2, 1996.

[5] S. Russell and P. Norvig, *Artificial intelligence: a modern approach.* Prentice hall, 2010.

[6] N. Kushmerick *et al.*, "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. Krishnamurthy *et al.*, "SystemT: A System for declarative Information Extraction," *ACM SIGMOD Record*, vol. 37, no. 4, 2009.

[10] H. Maus, "Towards a Functional Integration of Document Analysis in Workflow Management Systems,," *Proceedings of Workflow Management*, 1999.