2012 10th IAPR International Workshop on Document Analysis Systems

Towards understandable explanations for Document Analysis Systems

Björn Forcher, Stefan Agne, Andreas Dengel
Knowledge Management Deptartment,
German Research Center for Artificial Intelligence DFKI GmbH,
Kaiserslautern, Germany
Email: {firstname.lastname}@dfki.de

Michael Gillmann
Insiders Technologies GmbH,
Kaiserslautern, Germany
Email: m.gillmann @insiders-technologies.de

Thomas Roth-Berghofer
Centre for Model-based Software Eng. and Explanation-aware Computing,
School of Computing and Technology
University of West London, United Kingdom
Email: thomas.roth-berghofer@uwl.ac.uk

***Abstract*—smartFIX is a product portfolio for knowledge-based extraction of data from any document format. The system automatically determines the document type and extracts all relevant data for the respective business process. Data that is unreliably recognized is forwarded to a verification workplace for manual checking. In general, users have no difficulties to interpret the document data and wonder why the system needs additional input. For that reason, we implemented an explanation component that is used to justify extraction results, thus, increasing confidence of users. The component is using a semantic log making it possible to provide understandable explanations. We illustrate the benefits of that kind of technology in contrast to the current smartFIX LogViewer by means of a preliminary user experiment.**

***Keywords*-explanation; understandability; information extraction; semantic technology; DAS**

I. Introduction

In smartFIX documents are classified automatically on the basis of free form and forms-analysis methods [2]. Relevant data is extracted using different methods for each document type and is validated and valuated via database matching and other sophisticated knowledge-based methods. Due to mathematical and logical checks data quality is enhanced. Data that is accurately recognized is released for direct export. In contrast, unreliably recognized data is forwarded to a verification workplace for manual checking.

In many cases, users have no difficulties to read data on a document. Consequently, they often do not understand the difficulties of smartFIX during the extraction process. Making the system more transparent, smartFIX already creates a log in a proprietary format that can be accessed by the so called *LogViewer*. However, the log is very detailed and even trained users often cannot utilize the LogViewer to understand the system's behavior. Actually, users have to consult customer support to solve extraction problems.

For that reason, we developed an explanation component—Prof. Smart—that is used to justify reliable as well as unreliable extraction results of the smartFIX system that is the implementation of our conceptual work as presented in [10]. The goal of justifying extraction results is to increase customer satisfaction and to reduce the effort of the customer support. A prerequisite for this is an intuitive method to specify the explanation need and customized explanations depending on the users' expertise.

In order to achieve this goal, explanation generation is based on semantic technologies, i.e., a *semantic log* that contains all process relevant information of the smartFIX system such as process type, results and application domain. This enables the explanation component to understand the users explanation problem and allows to generate customized explanations. More precisely, we rely on a process ontology language to encode the semantic log. Further, we use sophisticated semantic search technology enabling an intuitive access to the logging information. Finally, we use ontology transformation to adapt the logging information to real explanation information. In this context, we describe how certain ontology characteristics such as inheritance and (transitive) relations can be utilized to adapt logging information and to generate understandable explanations.

This paper is structured as follows. The next section gives a short overview about relevant research on explanations. Sect. III presents the smartFIX system and motivates its explanation need by an intuitive example. Additionally, we describe the smartFIX LogViewer currently integrated in the smartFIX retail version. Sect. IV presents our explanation generation approach whereas the following describes a preliminary user experiment that illustrates the benefits of our approach in contrast to the LogViewer. We conclude the paper with a brief summary and outlook.

II. Related Work on Explanations

Wick and Thompson [7] developed the expert system REX, which implements the concept of *reconstructive explanations*. REX transforms a trace, a line of reasoning, into a plausible explanation story, a line of explanation. The transformation is an active, complex problem-solving process using additional domain knowledge. The degree of coupling between the trace and the explanation is controlled by a filter that can be set to one of four states regulating

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.92

the transparency of the filter. The more information of the trace is let through the filter, the more closely the line of explanation follows the line of reasoning. In this work, we describe how semantic technologies can be applied to (re-) construct explanations.

In [10] we described our conceptual work on explaining smartFIX. In our explanation scenario (Fig. 1) we distinguish three main participants: the *user* who is corresponding with the software system via its user interface (UI), the *originator*, the tool that provides the functionality for the original task of the software and the *explainer*. Originator—smartFIX—and explainer—Prof. Smart—need to be coupled in order to provide the necessary knowledge about the inner workings of the originator for the explainer. In (rule-based) expert systems looking at the rule trace was the only way of accessing the originator's actions. Given that the inference mechanism is fixed in those systems the trace was all the explainer needed.

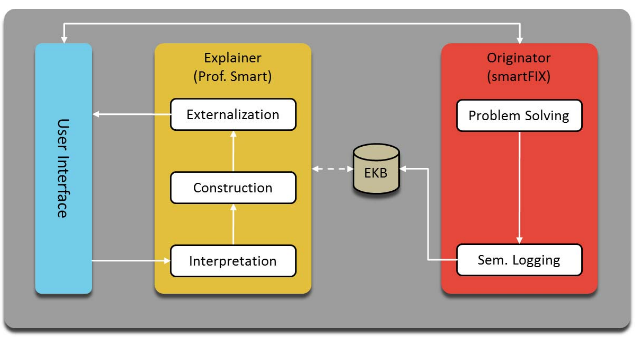


Figure 1. Transformation processes in explanation scenario, cf. [10]

The mentioned scenario implies that the originator has to provide detailed information about its behavior and solutions. Therefore it is necessary that the originator prepares some kind of log representing the initial starting point for the explainer to generate explanations. Regarding user questions, this information is step-by-step being transformed into an adequate explanation. Thus, a multi-layered explanation model is constructed, whereas each step contributes a layer to the model, *i. e.*, the transformation result.

Depending on the coupling, originator and explainer share information that is required for problem solving and for explanation generation as well. In Fig. 1, this information is contained in the explanation knowledge base (EKB). The originator may have access to information which is hidden from the explainer and vice versa. At least, they have to share the *semantic log*. As its name implies, the *logging process* collects all information with respect to the behavior of the originator for building the log.

Users communicate their explanation needs by keywords or in natural language. As the formal language of originator and explainer is often completely different from the user's language an *interpretation process* is necessary. In simplified terms, relevant parts of the semantic log and EKB must be identified and the exact explanation needs of the user must be determined. The result of the interpretation process is called *translation layer*.

The translation layer does not necessarily represent adequate explanation information. Until this stage, the explainer is only aware of the users' explanation problem concerning, for instance, an incomprehensible result of the originator. However, the information that solves the users' explanation problem has not been derived. The explanation generation process is called *construction process* which is similar to the concept of reconstructive explanations. The result of that process is called *content layer* representing useful explanation information. As understandability is a very important aspect of explanation [5], [6], it does not contain too much or too confusing information.

Explanation is information that is communicated by text, charts, tables, *etc.* Each communication form has different application possibilities in an explanation scenario. Text can describe complicated conceptions whereas charts can reveal qualitative connections between concepts in a simple way [8]. The *externalization process* transforms the content layer into a formal description for communicating explanations, namely the *externalization layer*. In this work, we put a special emphasis on semantic networks based on mathematical graphs for depicting explanations. However, this layer does not include layout and style information. Rendering the externalization layer is a task of the UI.

III. SMARTFIX

smartFIX extracts data from paper documents as well as from many electronic document formats (e.g., faxes, e-mails, MS Office, PDF, HTML, XML, etc.). Regardless of document format and structure, smartFIX recognizes the document type and any other important information during processing. Basic image processing such as binarization, despeckling, rotation and skew correction is performed on each page image. If desired, smartFIX automatically merges individual pages into documents and creates processes from individual documents. For each document, the document class and thus the business process to be triggered in the company is implicitly determined. smartFIX subsequently identifies all relevant data contained in the documents and related to the respective business process. In this step, smartFIX can use customer relation and enterprise resource planning data (ERP data) provided by a matching database to increase the detection rate. A special search strategy searches for all entries from the customer's vendor database on the document. The procedure works independently of the location, layout and completeness of the data on the document. Within smartFIX this strategy is called "Top Down Search". Moreover, smartFIX provides self-teaching mechanisms as a highly successful method for increasing recognition rates. Both general and sender-specific rules are applied. An automatic quality check is then performed on all recognized values. Beside others, Constraint Solving [1] and Transfer Learning methods [3] are used. Values that are accurately and unambiguously recognized are released for direct export; uncertain [4] values are forwarded to a verification workplace for manual checking and verification. The quality-controlled data is then exported to the desired downstream systems, e.g., an enterprise resource planning system like SAP for further processing. An overview of the system architecture is also presented in [10].

Let us illustrate exemplary an actual scenario that currently results in support calls and internal research and clarification effort by experts. Often, several subcompanies of the same trust are resident at the same location or even in the same building. If one smartFIX system has to analyze, for instance, invoices of more than one of those companies, very similar database entries can be found in the customer's master database.

The company's master data is an important knowledge source used by Top Down Search during the analysis step of smartFIX. When smartFIX analyzes an invoice sent to such a subcompany it may be unable to identify a clear and unambiguous extraction result due to the high degree of similarity of the master data entries. So, smartFIX has to regard all the subcompanies as possible hits.

smartFIX extracts the most reliable result based on extraction rules. Here, it does not valuate that result as reliable but as a suggestion [4]. Fig. 2 presents a look into the smartFIX Verifier in that case. You see that the recipient's name and identifier are correctly extracted but the values are marked blue which means "uncertain" in the smartFIX context.

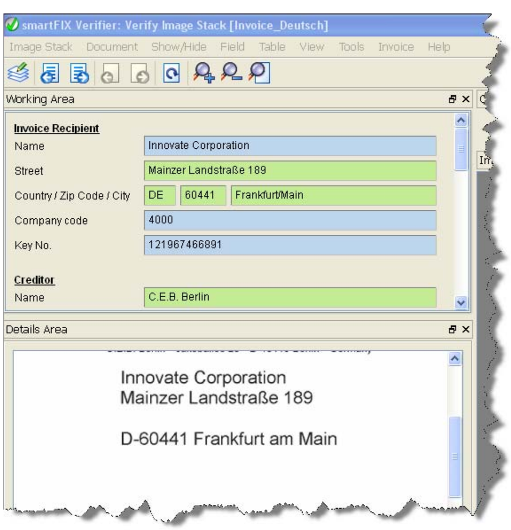


Figure 2. Analysis results presented in smartFIX Verifier

With this picture on screen, the user wonders why the system asks for interaction (here, for pressing the Return key to confirm the correct extraction results) although she can clearly and easily read the full recipient's address on the invoice. This scenario holds, too, and becomes more intransparent the more extraction rules and sophisticated extraction and valuation methods come into operation.

As explained before, smartFIX already creates a log in proprietary format that is hard to read for non-trained smartFIX users. The conventional log for one processed document page counts more than 25,000 lines; unmanageable for human users without computer-aided support. For this purpose, the smartFIX system includes a tool called LogViewer that can read the log and visualize the log entries in a tree structure. The nodes represent processes, warnings, errors, and results to which filtering is possible. The LogViewer offers a (conjunctive) keyword-based search to find relevant log entries. In case the keywords are contained in multiple log entries, the viewer allows to navigate through the search results. It unfolds the tree if necessary and jumps to the respective log entry of the current search result. If the log entry concerns a certain area, the viewer renders the respective document and highlights the area with a blue frame. Despite all these efforts the location of explanatory log entries is very difficult because there are often too many entries visible at the same time. In addition, the interpretation of the entries is not intuitive due to much technical information. It turned out the log viewer can not always provide explanations even for trained smartFIX users.

IV. Explanation Generation

For realizing *Semantic Logging* we developed the *smartFIX Process Ontology* (sFPO), an extension of the OWL-S ontology[1]. OWL-S provides a set of representation primitives capable of describing features and capabilities of Web services in unambiguous, machine-interpretable form. This includes, among other things, the possibility to describe how the service works. OWL-S comprises general constructs to represent processes, results and intermediate results. sFPO extends the OWL-S ontology with smartFIX specific concepts and explanation relevant constructs. This allows not only to describe the behavior of smartFIX in an abstract way, but also to instantiate a concrete log with respect to the *Semantic Logging* step. The advantage of ontologies is that ontological entities can be easily extended with labels and comments and that both the instantiated log and the sFPO as well can be interpreted as mathematical graph. Both aspects are import regarding the generation next step.

For interpreting the user's explanation needs we integrated a keyword-based semantic search engine into Prof. Smart that is based on the work of Tran *et al.* [9]. The search engine maps keywords to elements of the log and searches for a connections between elements. As mentioned above, a log can be interpreted as directed graph. The search engine looks for subgraphs representing basic explanation information.

[1] http://www.w3.org/Submission/OWL-S/

Regarding the example as presented in Sect. III a user may use the keywords *analyser*, *recipient* and *uncertain result* in order to find out why the recipient is not identified correctly. A subgraph that connects the corresponding elements of the log builds some kind of bridge enabling users to understand why the result is unreliable. In general, users do not use the same keywords to specify their explanation need depending on the user's level of expertise. For that reason, we enrich the sFPO ontology with synonyms taken from the WordNet thesaurus[2]. As a result, users can also use *unsure result* instead of *uncertain result*. However, keywords generally map on several elements of the log. Hence, subgraphs are ranked depending on the weighting of the graph elements which represent possible explanation alternatives. To summarize, one advantage of ontologies is the possibility to enrich the concepts with synonyms so that users can use their own words to specify their explanation needs. The other advantage is that we can formulate the explanation problem as a graph search problem applying various graph search algorithms and weighting strategies.

As explained above, the *Interpretation* step determines an extract of the log (subgraph). Potentially, this extract is still not understandable for non-expert smartFIX users because it may contain too much unknown information. For this, the *Construction* step adapts the extraction of the log to the users needs. After initially experimenting with explanations of smartFIX it turned out that the extract of the log contains too much information. For that reason, we focused in particular on shortening the determined extract of the log. For shortening the extract two characteristics of ontologies can be applied, namely inheritance and (transitive) relations that do not affect the truth content of the explanation. Both concepts are illustrated by the following examples.

In general, a smartFIX log contains a root process which starts several subprocesses which in turn start further subprocesses and so on. As a result, an extract of a log may contain a long path of processes as depicted in Fig. 3. In this example, the *Root Process* leads to the process *TopDown search Recipient*. The edges labeled with *is subprocess of* correspond to a the transitive relation *sFPO:hasSubProcess*, which is part of the sFPO. Thus, the process path in Fig. 3 can be shortened to the path as depicted in Fig. 4. However, it is essential, that the removed nodes do not have an (exclusive) connection to graph elements that were determined by the keyword-element-mapping.

Inheritance in an ontological context means that attributes of an upper concept are inherited by a subconcept. That means that instances of the subconcept are always instances of the upper concept too. That relationship can also be used to shorten an explanation path. Consider the two nodes in the top left corner of Fig. 3. With respect to the example in Sect. III the keyword *smartFIX component*, is used to find out why the recipient is not identified correctly. In this case, the respective analyzer instance is not certain about the correct recipient which is also an instance of the class *smartFIX component*. However, it is not essential to inform the user that an analyzer instance has generated the result that is also an instance of the class *smartFIX component* as depicted in Fig. 3. The inheritance characteristics of ontologies allows us to shorten the graph as illustrated in Fig. 4.

In the ideal case the result of the *Construction* step represents complete explanation information that, in turn, must be externalized by chart or a text. In general, it is advisable to have different forms of externalization as some users prefer text to chart. As instantiated log and the sFPO can be interpreted as graph and simply be depicted as semantic network (cf. Figures 3 and 4).

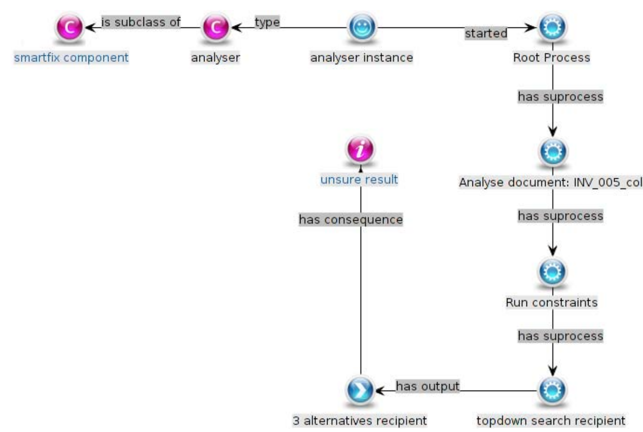


Figure 3. Extracted subgraph of the log

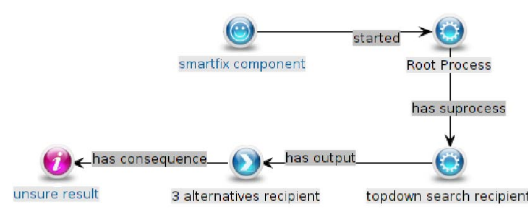


Figure 4. Shortened Explanation

V. USER EXPERIMENT

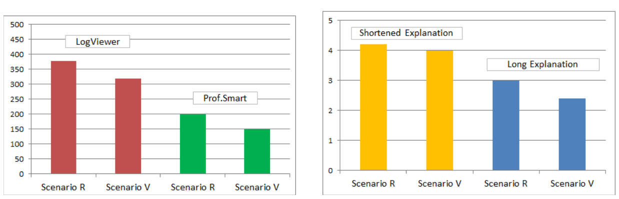
In this section we present a preliminary user experiment that has two objectives. The first objective is to illustrate that the current explanation component is a more useful instrument for understanding system behavior in contrast to the conventional LogViewer. The second objective is too find out whether the shortening is a useful mean to generate more understandable explanations.

For the experiment we consider two explanation scenarios R and V. In scenario R the test persons should find out why

[2]http://wordnet.princeton.edu/

the recipient is not identified unambiguously. In scenario V the probands should find a justification why there is no uncertainty concerning the vendor. Based on that, we constructed two tests A and B. In test A scenario R must be solved with the explanation component and scenario V must be solved with the LogViewer. Vice versa in Test B: scenario R must be solved with the LogViewer and scenario V must be solved with the explanation component. After a short introduction, a test person only need to do one Test (A or B). Altogether, four values are collected. The first one concerns the time the test persons need to solve the explanation problem, whereby each scenario has a limit of seven minutes. If a test person quits the time is set to seven minutes. In case the explanation component must be used, the probands have to assess the quality of the correct shortened explanation on a five-point scale (one means 'very bad'). In case the LogViewer is used, test persons must additionally asses the quality of the correct not shortened long explanation.

Ten persons took part in the experiment, all having a computer science background, but they do not have a deeper understanding of the smartFIX system. The results are presented in Fig. 5. The bar chart on the left illustrates the average time of test persons using the LogViewer (bars on the left) and explanation facility (bars on the right) and . It is easy to see that the use of the explanation component brings a major time advantage to find a solution for the explanation problem. The right bar chart illustrates an average valuation for shortened explanations (bars on the left) and complete explanations (bars on the right). It is obvious that test persons prefer shortened explanations in both scenarios.



(a) Average Time (b) Average Rating

Figure 5. Experiment Result

VI. Conclusion and Outline

In this paper, we presented our current approach on explaining the smartFIX system. We described an intuitive explanation problem in document analysis and illustrated how semantic technology can be utilized to generate understandable explanations. The approach is based on an extension of the OWL-S ontology that is used to construct a semantic log describing smartFIX processes and results. Keyword-based search technology is applied to find relevant parts of that log. In a preliminary user experiment we illustrated that ontology constructs such as inheritance and transitive relations can be utilized to adapt these parts of the log and generate understandable explanations respectively. In addition, we showed the explanation is a more useful tool in contrast the current LogViewer.

In a future version of smartFIX the explanation component will not only be able to justify extraction results but also to give practical hints to avoid low quality extraction results. In addition, we provide further forms of explanation externalization such as text and semantic networks combined with text and further experiments.

Acknowledgment

This work was funded by the German Federal Ministry of Education and Research (BMBF) in the EMERGENT project under grant nr. 01IC10S01.

References

[1] A. Fordan, *Constraint Solving over OCR Graphs*, 14th International Conference on Applications of Prolog (INAP), Tokyo, Japan, 2001.

[2] B. Klein, A. Dengel, and A. Fordan, *smartFIX: An Adaptive System for Document Analysis and Understanding*, in: A. Dengel, M. Junker, A. Weissbecker (Eds.), *Reading and Learning - Adaptive Content Recognition*, LNCS 2956, Springer, 2004.

[3] F. Schulz, M. Ebbecke, M. Gillmann, B. Adrian, S. Agne, and A. Dengel, *Seizing the Treasure: Transferring Layout Knowledge in Invoice Analysis*, 10th International Conference on Document Analysis and Recognition (ICDAR), Barcelona, Spain, 2009.

[4] B. Seidler, M. Ebbecke, and M. Gillmann, *smartFIX Statistics – Towards Systematic Document Analysis Performance Evaluation and Optimization*, 9th IAPR International Workshop on Document Analysis Systems (DAS), Boston, MA, USA, 2010.

[5] W. R. Swartout and S. W. Smoliar. Explanation: A source of guidance for knowledge representation. In Katharina Morik, editor, *Knowledge Representation and Organization in Machine Learning*, volume 347 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 1987.

[6] W. R. Swartout, C. Paris, and J. D. Moore. Explanations in knowledge systems: Design for explainable expert systems. *IEEE Expert*, 6(3):58–64, 1991.

[7] M. R. Wick and W. B. Thompson. Reconstructive expert system explanation. *Artif. Intell.*, 54(1-2):33–70, 1992.

[8] P. Wright and F. Reid. Written information: Some alternatives to prose for expressing the outcomes of complex contingencies. Journal of Applied Psychology **57 (2)** (1973) 160–166

[9] D. T. Tran, H. Wang, S. Rudolph and P. Cimiano. Top-k Exploration of Query Candidates for Efficient Keyword Search on Graph-Shaped (RDF) Data Proc. of the 25th Intern. Conference on Data Engineering (ICDE'09)

[10] B. Forcher, S. Agne, A. Dengel, M. Gillmann, and T. Roth-Berghofer. Semantic Logging: Towards Explanation-Aware DAS Int. Conf. on Document Analysis and Recognition (ICDAR-2011), Beijing, China