2011 International Conference on Document Analysis and Recognition

# Semantic Logging: Towards Explanation-Aware DAS

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

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

Thomas Roth-Berghofer
*Explanation-aware Computing Systems, Institute of Computer Science University of Hildesheim, Germany*
*Email: rothberg@uni-hildesheim.de*

***Abstract*—smartFIX is a product portfolio for knowledge-based extraction of data from any document format. smartFIX automatically determines the document type and extracts all relevant data for the respective business process. Data that is uncertainly 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 will integrate an explanation component that will be used to justify uncertain extraction results, thus, increasing confidence of users. The component will be based on semantic technologies in general and on a semantic log in particular. The log will contain all process relevant information enabling the explanation facility to generate customized and understandable explanations. In this paper, we will discuss the benefits of that kind of technology with reference to DAS.**

***Keywords*-DAS, explanation, information extraction, semantic technology, semantic logging**

## I. Introduction

In smartFIX [2] documents are classified automatically on the basis of free form and forms-analysis methods. 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, uncertainly recognized data is forwarded to a verification workplace for manual checking (see Sect. III).

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

For that reason, we will integrate an explanation component into smartFIX that will be used to justify uncertain extraction results. 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 will be based on semantic technologies, in general, and on a *semantic log*, in particular. The log contains all process relevant information of the smartFIX system such as process type, results and intermediate results. 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 will use sophisticated semantic search technology enabling an intuitive access to the logging information. Finally, we will use ontology transformation to adapt the logging information to real explanation information.

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 Sect. IV motivates its explanation need by an intuitive example. Sect. V contains our conceptual work on explanation whereas the following section describes how the example problem can be solved with semantic technologies. 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 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. We take up the theme of (re-)constructing explanations in this work.

In [8], Ducassé and Noyé describe a stratified model for user-oriented program analysis including explanations. It contains three steps to explain solutions of logic programs, namely *Extraction*, *Analysis*, and *Visualizations*. The model uses several sources of input such as source codes, execution

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.230

information and other specifications which are integrated by the extraction step. The result of the extraction is the trace that is analyzed and abstracted in the following step. The resulting data of the second step is presented to the user. Ishizaka and Lusti [9] extend that model by defining a trace model as first step.

## 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 [4] are used. Values that are accurately and unambiguously recognized are released for direct export; uncertain [5] 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 shown in Fig. 1.

## IV. smartFIX example

Let us illustrate exemplary a realistic scenario that, today, results in support calls and internal research and clarification effort by experts. Later, in Sect. VI we will show how explanations and, in particular, the presented semantic logging approach can be applied.

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

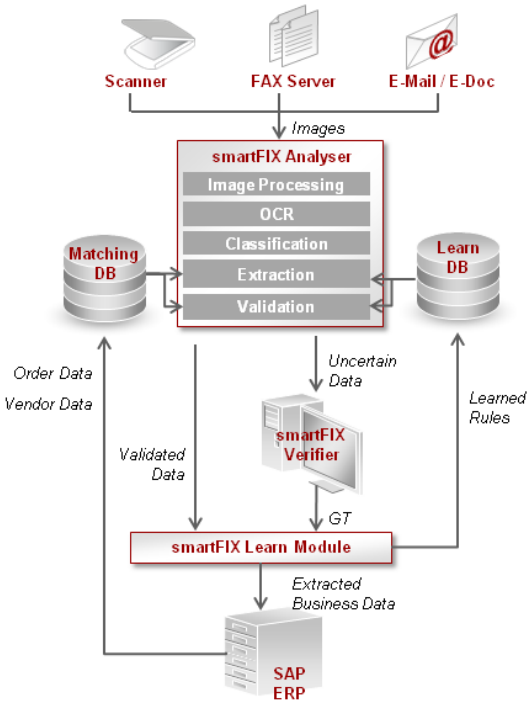


Figure 1. smartFIX system architecture

entries can be found in the customer's master database as exemplary shown in Fig. 2.

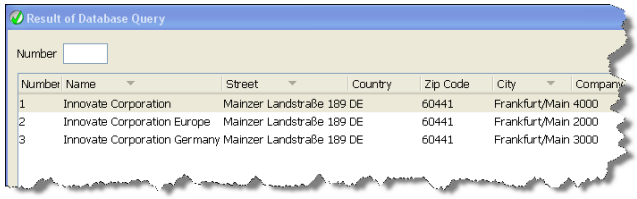


Figure 2. look into recipient's master data

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 (see Fig. 3) 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.

Of course, smartFIX extracts the most reliable result due to other extraction rules. In this case it does not valuate that result as certain but as a suggestion [5]. Fig. 4 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.

With this picture on screen, the user wonders why the system asks for interaction (here, pressing the Return key to confirm the correct extraction results) although she can


Figure 3. Address area of the analyzed invoice

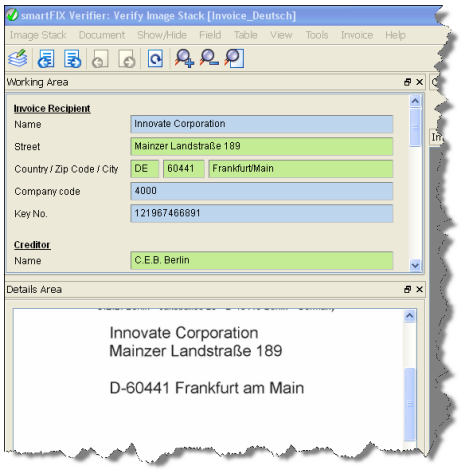


Figure 4. Analysis results presented in smartFIX Verifier

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.

## V. Semantic Logging

In a general explanation scenario (Fig. 5) we distinguish three main participants [3]: the *user* (not shown here) who is corresponding with the software system via its user interface, the *originator*, the tool that provides the functionality for the original task of the software, and the *explainer*. Originator and explainer need to be coupled in order to provide the necessary knowledge about the inner workings of the originator. 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.

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

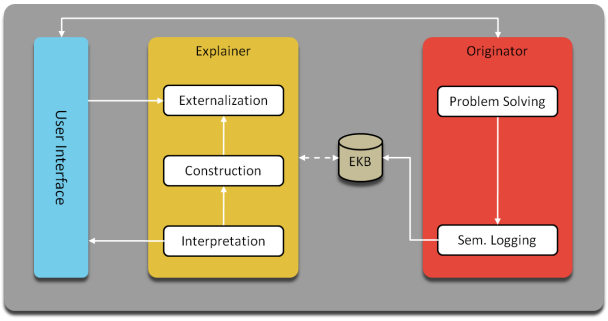


Figure 5. Transformation processes in explanation scenario

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. 5, 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 as presented in Sect. II. The result of that process is called *content layer* representing an understandable explanation that 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 [10]. 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 user interface.

The main difference between this and existing approaches as presented in Sect. II is the providence of a *mediation layer* that is orthogonal to the others. For being *explanation-aware* the explainer has to log its own transformation process including the results of the single process steps. Consequently, the explainer is aware of how explanation needs and explanation externalization correspond to each other. In addition, it is aware of why an explanation is given in a certain explanation scenario enabling explanatory dialogs and analysis of given explanations.

So far, we described our conceptual work on explanation generation. In the following, we present different Semantic Technologies and tools that will be used in the explanation component of smartFIX. For encoding the semantic log we rely on the OWL-S ontology[1]. OWL-S is based on the ontology language OWL and provides a set of representation primitives capable of representing features and capabilities of Web services in unambiguous, machine-interpretable form. The ontology comprises three modules: profile module, grounding module and process module. The profile module is used to describe what the service does, the grounding profile can be used to describe how to access information and the process model allows to describe how the service works. In smartFIX OWL-S ontology can be used in two ways. First, the process model can be used as upper level ontology for describing smartFIX specific processes. Here, we will not use the complete expressiveness of OWL-S in order to conform with RDFS. Second, the resulting representational constructs can be leveraged to encode or to instantiate specific logging information: the SemLog.

For interpreting the user's explanation needs we will use the semantic search engine KOIOS [11], a keyword-based (or natural language) search on graph-shaped RDF data. KOIOS first computes a set of ranked SPARQL queries allowing users to select the most appropriate ones. Consequently, a selected query is used to search in a respective knowledge base. The main advantage is that users do not need explicit knowledge about the query syntax or the underlying ontologies. For computing queries, the keywords are mapped to elements of input RDF data in a first step. Based on these *keyword elements*, a search is performed on the RDF graph data to determine a *connecting element*. As its name implies, it is a particular graph element connecting the keyword elements. The paths between the keyword elements and the connecting elements are analyzed to create a *matching subgraph*. For each subgraph, a conjunctive query is constructed by mapping the graph elements to query elements. To achieve a scalable search, the input data is pre-processed obtaining two data indices. An *inverted keyword index* is used to realize the keyword mapping. To summarize, KOIOS will enable a simple access to logging information even for non-trained smartFIX users. The highest ranked SPARQL queries and subgraphs, respectively, represents a part of the semantic log or the translation layer in our conceptualization.

As explained, KOIOS selects a certain part of the log that describes a certain extraction process and results of smartFIX. Potentially, this extract is still not understandable for non-expert smartFIX users. In the next step, the extract of the log is adapted to the users needs. As the result is some kind of RDF graph, an RDF transformation language such as QPL [11] can be used to adapt the log extract to the corresponding user. As a result (construction layer), the log does only contain as much information as necessary to solve the users explanation problem.

As a first step, the RDF graph of the construction layer can be interpreted as semantic network and thus, the RDF graph also represents the externalization layer.

## VI. SMARTFIX EXAMPLE REVISITED

As explained before, smartFIX already creates a log in proprietary format that is hard to read for non-trained smartFIX users. Regarding the illustrated example of Sect. IV, the conventional log for ten processed document pages counts more than 250,000 lines and thus, the location of explanatory lines is very difficult. In addition, the interpretation of the lines is not intuitive due to encoding issues.

Encoding smartFIX behavior by means of OWL-S the semantic log has a graph-like structure. As described in Sect. V it is a simple directed graph since we limit on RDFS semantics. Graph nodes represent smartFIX entities such as processes, results and intermediate results whereas edges establish connections between the entities. Every graph element is labeled with several expressive keywords enabling keyword-based semantic search on the log or graph. For instance, using KOIOS in the explanation component, it is possible to determine the subgraph of the log that describes the uncertain extraction example from above. Typing the keywords *innovate corporation* and *uncertain data* into the search field of the explanation portal of smartFIX, KOIOS extracts a subgraph that connects the corresponding graph elements. Due to the semantic preparation of the log KOIOS would also extract the same subgraph when using *address* and *uncertain data*. As a result, the user can use (more or less) his own words to access the explanation component depending on the expressiveness of the semantic log. In our example, the translation layer comprises a validation process that is composed of a database matching process returning three companies and an valuation process that valuates the database matches with respect to the input address.

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

In general, it is not necessary to know that the validation process is composed of subprocesses. In addition, the current subgraph (translation layer) may be complex or too complicated and thus, the subgraph must be shortened with respect to the current user and contextual situation. For that purpose, graph-based transformation languages such as QPL can be used to alter the subgraph building the construction layer. With the help of predefined rule sets supporting information can be added and confusing information can be removed. With respect to our example the adapted or constructed explanation may look as presented in Fig. 6 corresponding to the *translation model*. In this case, the user can see that the uncertain data comprises three companies solving the explanation problem. Here, the rule set contains a rule that collapses the complex validation process to a simple process mapping the input address to three valuated database matches.

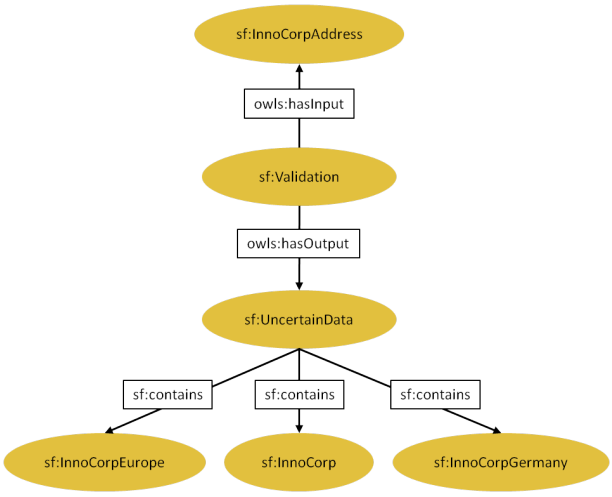


Figure 6. Constructed explanation

Because of the interpretation of RDF models as graphs the externalization as semantic is very simple and thus, the construction model represents also the externalization model. Consequently, the externalization model can be visualized as semantic network including additional interaction possibilities such as browsing the semantic log. Resources, properties and literals of the RDF model correspond to nodes, edges and labels of the semantic network.

## VII. Conclusion and Outline

In this paper we presented our conceptual work to make the smartFIX system explanation-aware. We described an intuitive explanation problem in document analysis and illustrated the benefits of semantic logging based on the web service language OWL-S. The current retail version of smartFIX encodes a log in proprietary language comprising about 25,000 lines for one analyzed document page. In general, even smartFIX experts cannot easily use the log to understand uncertain extraction results, generating high customer support costs. We proposed the use of semantic logging and semantic web technology to generate understandable explanations increasing customer satisfaction and reducing customer support costs. 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.

## Acknowledgment

The work presented in this paper was performed in the context of the Software-Cluster project EMERGENT (www.software-cluster.org). It was funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IC10S01. The authors assume responsibility for the content.

## References

[1] A. Fordan, *Constraint Solving over OCR Graphs*, 14th Int. Conf. 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] T. R. Roth-Berghofer and M. M. Richter. On explanation. *Künstliche Intelligenz*, 22(2):5–7, May 2008.

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

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

[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] M. Ducassé and J. Noyé. Logic programming environments: Dynamic program analysis and debugging. J. Log. Program. **19/20** (1994) 351–384 model of user-oriented program analysis.

[9] A. Ishizaka and M. Lusti How to program a domain independent tracer for explanations. Journal of Interactive Learning Research **17(1)** (2006) 57–69

[10] 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

[11] B. Forcher, M. Sintek, T. Roth-Berghofer, and A. Dengel. Explanation-Aware System Design of the Semantic Search Engine KOIOS Proceedings of the ECAI-10 workshop on Explanation-aware Computing (ExACt 2010)