Semi-automatic knowledge extraction from semi-structured and unstructured data within the OMAHA project

Pascal Reuss[1][2], Klaus-Dieter Althoff[1][2], Wolfram Henkel[3], Matthias Pfeiffer[3], Oliver Hankel[4], and Roland Pick[4]

[1] German Research Center for Artificial Intelligence
Kaiserslautern, Germany
http://www.dfki.de

[2] Institute of Computer Science, Intelligent Information Systems Lab
University of Hildesheim, Hildesheim, Germany
http://www.uni-hildesheim.de

[3] Airbus
Kreetslag 10 21129 Hamburg, Germany

[4] Lufthansa Industry Solutions, Norderstedt, Germany

Abstract. This paper describes a workflow for semi-automatic knowledge extraction for case-based diagnosis in the aircraft domain. There are different types of data sources: structured, semi-structured and unstructured source. Because of the high number of data sources available and necessary, a semi-automatic extraction and transformation of the knowledge is required to support the knowledge engineers. This support shall be performed by a part of our multi-agent system for aircraft diagnosis. First we describe our multi-agent system to show the context of the knowledge extraction. Then we describe our idea of the workflow with its single tasks and substeps. At last the current implementation, and evaluation of our system is described.

1 Introduction

This paper describes the concept of a semi-automatic knowledge extraction workflow, which is developed for a distributed decision support system for aircraft diagnosis. The system will be realized as a multi-agent-system. It is based on the SEASALT architecture and includes several case-based agents for various tasks. The knowledge extraction workflow will be realized using several agents within the decision support system. In the next section we give an overview of the OMAHA (Overall Management Architecture For Health Analysis) project, the SEASALT architecture and the application domain. In Section 3.1 we describe the instantiation of our decision support system based on SEASALT. Section 3.2 contains the initial concept for the knowledge extraction workflow, while 3.3 describes the current implementation status of the workflow. The Section 3.4 shows the evaluation setup and the evaluation results and Section 4 contains the related work. Finally, Section 5 gives a short summary of the paper and an outlook on future work.

2 OMAHA project

The OMAHA project is supported by the Federal Ministry of Economy and Technology in the context of the fifth civilian aeronautics research program [6]. The high-level goal of the OMAHA project is to develop an integrated overall architecture for health management of civilian aircraft. The project covers several topics like diagnosis and prognosis of flight control systems, innovative maintenance concepts and effective methods of data processing and transmission. A special challenge of the OMAHA project is to outreach the aircraft and its subsystems and integrating systems and processes in the ground segment like manufacturers, maintenance facilities, and service partners. Several enterprises and academic and industrial research institutes take part in the OMAHA project: the aircraft manufacturer Airbus (Airbus Operations, Airbus Defense & Space, Airbus Group Innovations), the system and equipment manufacturers Diehl Aerospace and Nord-Micro, the aviation software solutions provider Linova and IT service provider Lufthansa Systems as well as the German Research Center for Artificial Intelligence and the German Center for Aviation and Space. In addition, several universities are included as subcontractors.

The OMAHA project has several different sub-projects. Our work focuses on a sub-project to develop a cross-system integrated system health monitoring (ISHM). The main goal is to improve the existing diagnostic approach with a multi-agent system (MAS) with several case-based agents to integrate experience into the diagnostic process and provide more precise diagnoses and maintenance suggestions.

2.1 SEASALT

The SEASALT (Shared Experience using an Agent-based System Architecture Layout) architecture is a domain-independent architecture for extracting, analyzing, sharing, and providing experiences [4]. The architecture is based on the Collaborative Multi-Expert-System approach [1][2] and combines several software engineering and artificial intelligence technologies to identify relevant information, process the experience and provide them via an user interface. The knowledge modularization allows the compilation of comprehensive solutions and offers the ability of reusing partial case information in form of snippets. Figure 1 gives an overview over the SEASALT architecture.

The SEASALT architecture consists of five components: the *knowledge sources*, the *knowledge formalization*, the *knowledge provision*, the *knowledge representation*, and the *individualized knowledge*. The *knowledge sources* component is responsible for extracting knowledge from external knowledge sources like databases or web pages and especially Web 2.0 platforms, like forums and social media plattforms. These knowledge sources are analyzed by so-called Collector Agents, which are assigned to specific Topic Agents. The Collector Agents collect all contributions that are relevant for the respective Topic Agent's topic [4]. The *knowledge formalization* component is responsible for formalizing the

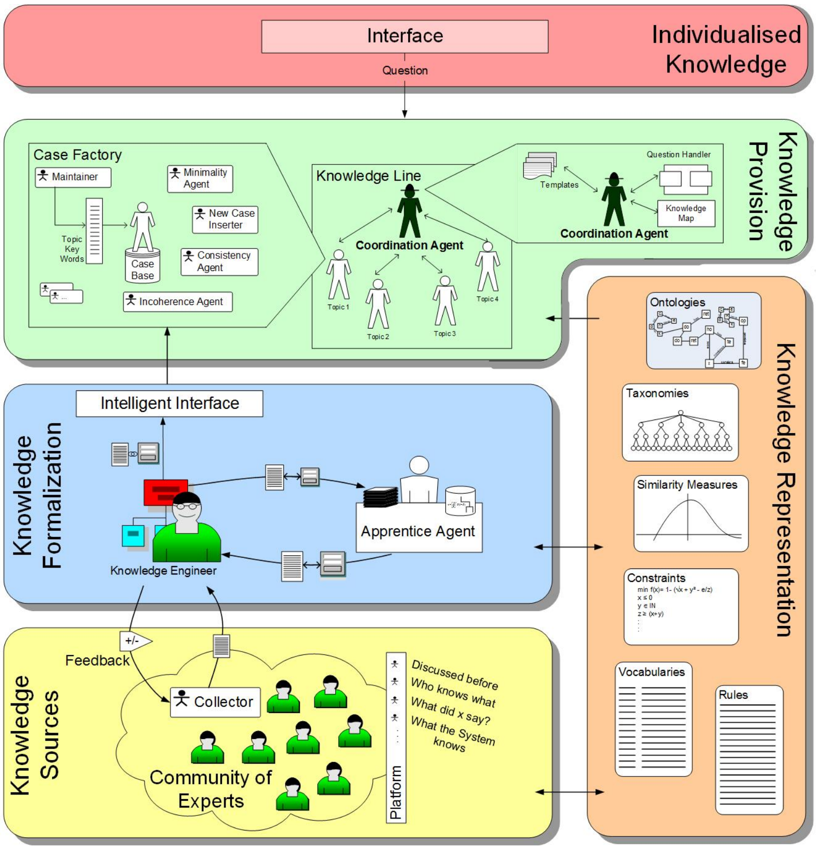

Fig. 1. Overview of the SEASALT architecture

extracted knowledge from the Collector Agents into a modular, structural representation. This formalization is done by a knowledge engineer with the help of a so-called Apprentice Agent. This agent is trained by the knowledge engineer and can reduce the workload for the knowledge engineer [4]. The *knowledge provision* component contains the so called Knowledge Line. The basic idea is a modularization of knowledge analogous to the modularization of software in product lines. The modularization is done among the individual topics that are represented within the knowledge domain. In this component a Coordination Agent is responsible for dividing a given query into several sub queries and pass them to the according Topic Agent. The agent combines the individual solutions to an overall solution, which is presented to the user. The Topic Agents can be any kind of information system or service. If a Topic Agent has a CBR system as knowledge source, the SEASALT architecture provides a Case Factory for the individual case maintenance [4][3]. The *knowledge representation* component contains the underlying knowledge models of the different agents and knowledge sources. The synchronization and matching of the individualized knowledge models improves the knowledge maintenance and the interoperability between the components. The *individualized knowledge* component contains the web-based user interfaces to enter a query and present the solution to the user [4].

2.2 Application domain

The domain of our application is aircraft fault diagnostic. An aircraft is a highly complex machine and an occurring fault cannot be easily tracked to its root cause. The smallest unit, which can cause a fault, is called Line Replacement Unit (LRU). While a fault can be caused by a single LRU, it also can be caused by the interaction of several LRUs or by the communication line between the LRUs. The data about the fault is in some cases very well structured (e.g., aircraft type, ATA chapter), but in other cases semi-structured (e.g., displayed fault message, references) or unstructured (e.g., fault description, electronic logbook entries, recommendations). These data have to be transformed into vocabulary, similarity measures, and cases.

The application is a first prototype demonstrator with several CBR systems. The systems represent different data sources and subsystems of an aircraft. The data sources are service information letters (SIL) and in-service reports (ISR) and we focus on the subsystems hydraulic and ventilation system. Service information letters contain exceptions to the usual maintenance procedure. These exceptions are described with information like the aircraft type and model, failure code, ATA chapter, displayed message, fault description, recommendations, actual work performed, and references to manuals. In-service reports are failure reports from airlines and contain partially overlapping information with the SIL like aircraft type, ATA chapter, fault description, but contain additional information like starting and landing airport, engine type, and the flight phase in which the fault occurred.

3 Semi-Automatic knowledge extraction

In this section the instantiation of the SEASALT architecture within the OMAHA project is described. The focus is set on the component *knowledge formalization* to show the idea behind the automatic vocabulary building. The current implementation of the knowledge formalization is described as well as the evaluation of the formalization work flow.

3.1 OMAHA multi-agent system

For the multi-agent demonstrator we will instantiate every component of the SEASALT architecture. The core components are the *knowledge provision* and the *knowledge formalization*, but the other components will be instantiated, too. The *individualized knowledge* component contains two interfaces for receiving a query and sending the solution. The first interface is a website to send a query to the multi-agent system and to present the retrieved diagnosis. In addition, a user can browse the entire case base, insert new cases or edit existing cases. The second interface communicates with a data warehouse, which contains data about Post Flight Reports (PFR), aircraft configuration data, and operational parameters. A PFR contains the data about the occurred faults during a flight and is the main query for our system. If additional information is required that is not provided by the data warehouse, it can be added via the website. Figure 2 shows the instantiation of the multi-agent system.

The *knowledge provision* component contains all agents for the diagnostic process. We defined several agent classes for the required tasks during the process: interface agent, output agent, composition agent, analyzer agent, coordination agent, solution agent, and topic agent. Each agent class is instantiated through one or more agents. A PFR and additional data is received by the data warehouse agent and/or the webinterface agent. A PFR contains several items that represent occurred faults. The PFR and the additional data are sent to the composition agent, which correlates the additional data with the individual PFR items. The correlated data are sent to the query analyzer agent and the coordination agent in parallel. The query analyzer agent is responsible for checking the correlated data for new concepts, which are not in the vocabulary, and sending a maintenance request to the Case Factory. The Case Factory checks the maintenance request, derives the required maintenance actions and executes the required actions after confirmation from a knowledge engineer. The coordination agent has two main tasks: sending a correlated PFR item to the right solution agent and integrating the returned diagnoses to an overall diagnosis. To determine the right solution agent, the coordination agent uses a so-called Knowledge Map that contains information about the existing solution and topic agents and their dependencies. The Knowledge Map tasks can be outsourced to an additional agent, the knowledge map agent, to provide more parallel processing. The knowledge map agent has access to the general Knowledge Map and to a CBR system that contains individual retrieval paths from past requests. The knowledge map agent uses the CBR system to determine the required topic agents for solving the query from successful past retrieval paths. After determining the required agents, the coordination agents sends the query to the corresponding solution agents. For each aircraft type (e.g., A320, A350, A380, etc.) an own agent team exists to process the query and retrieve a diagnosis. Each agent team consists of several agents: the solution agent receives the query, decomposes it, and sends the query parts to the required topic agents. One topic agent is used to process the configuration data and determine the configuration class of an aircraft. Because the occurrence of many faults depends on the hard- and software configuration of an aircraft, the configuration class can be used to reduce the number of cases in the retrieval process. The other topic agents are distinguished by the content of the case base and the ATA chapters. We derived cases from SIL and ISR for our prototype, but additional data sources are available. The ATA chapter decomposes an aircraft into several subsystems. By distinguishing the CBR systems this way, we get several smaller CBR systems, which have a smaller case structure and are easier to maintain. Each topic agent performs a retrieval on the underlying CBR systems and sends the solutions to the solution agent. The solution agent ranks the individual solutions and sends a ranked list back to the coordination agent and forwarded to the output agent. Each individual solution represents a possible diagnosis for the occurred fault described in the query. Therefore a combination of solutions is not appropriate. All found solutions above a given threshold have to be displayed to the user. The output agent passes the diagnoses to the web interface and the data warehouse.

The *knowledge formalization* component is responsible for transforming the structured, semi-structured, and unstructured data into structured information for the vocabularies, the similarity measures, and the cases itself of the CBR systems. The required maintenance actions for the CBR systems are performed by the Case Factory. For the CBR systems a structural CBR approach was chosen, because almost half of the provided data has the form of attribute value pairs. The other part of the data has to be transformed to be represented as attribute value pairs. The analysis and transformation of the data is done by a so-called case base input analyzer agent. This agent reads the data from different data sources like excel sheets, database result sets, or text documents. Then several information extraction techniques are used to extract keywords and phrases and to find synonyms and hypernyms. In addition, the data is analyzed to find associations within the allowed values of an attribute as well as across different attributes. This way we want to extract Completion rules[5] for query enrichment. The next step in the process is to add the found keywords, their synonyms and phrases to the vocabulary and set an initial similarity between a keyword and its synonyms. Furthermore, taxonomies can be generated or extended using the keywords and their hypernyms. After the vocabulary extension, the cases are generated and stored in the case bases. The last step is the generation or adap-

[5] Completion rules derive attribute values with a certainty factor if the respective condition is fulfilled (a set of attribute values).

tation of the relevance matrices[6] to set or improve the weighting for the problem description attributes. The idea and the top level algorithm of this tool chain and the current implementation status is described in more detail in the following sections.

In the *knowledge sources* component a collector agent is responsible for finding new data in the data warehouse, via web services or in the existing knowledge source of Airbus. New data in the data warehouse could be new configuration data or operational parameters, which have to be integrated into the vocabulary. Web services could be used to update the synonym and hypernym database and from the existing knowledge sources of Airbus new cases can be derived.

The *knowledge representation* component contains the generated vocabulary, the similarity measures and taxonomies, the extracted completion rules, and constraints of the systems to be provided for all agents and CBR systems.

3.2 Initial concept for semi-automatic knowledge extraction

There are more than 100.000 documents and data sets with fault descriptions and exceptions within the Airbus data sources. Every document or data set could contain useful information for our case-based diagnosis or even represent a complete case. This amount of data cannot be reasonably analyzed manually, but semi-automatedly with the help of software agents. The result of the analysis and the transformation has to be checked by a knowledge engineer to get feedback. This feedback can be used to improve the analysis and transformation process.

We designed a workflow with ten tasks for processing the data, extracting the knowledge, extending the knowledge containers, and importing cases. Each task consists of several steps. Figure 3 shows the workflow tasks and the associated steps. The input for the workflow is a set of documents with SIL or ISR content and a mapping document. This can be excel sheets, database result sets, or free text documents. The mapping document contains information to which attributes of a case structure the content of the document should be mapped.

The first task in the workflow is the extraction of keywords. Based on the type of the input document, the individual columns and rows or the entire text are processed. This task starts with the steps stopword elimination and stemming of the remaining words. The next step is to replace all abbreviations with the long form of the word. Therefore a list of used abbreviations within the aircraft domain is used to identify abbreviations. The result of this task is a list of keywords extracted from the document.

The second task in the workflow is to find synonyms and hypernyms for each keyword on the list. For the search we use a synonym database from Wordnet extended with technical terms from the avionics domain. For each found synonym and hypernym a search loop for additional synonyms and hypernyms is performed, too. This loop is repeated until no more new synonyms are found.

[6] A relevance matrix describes the relevance of available attributes concerning available diagnoses (e.g., [9]).

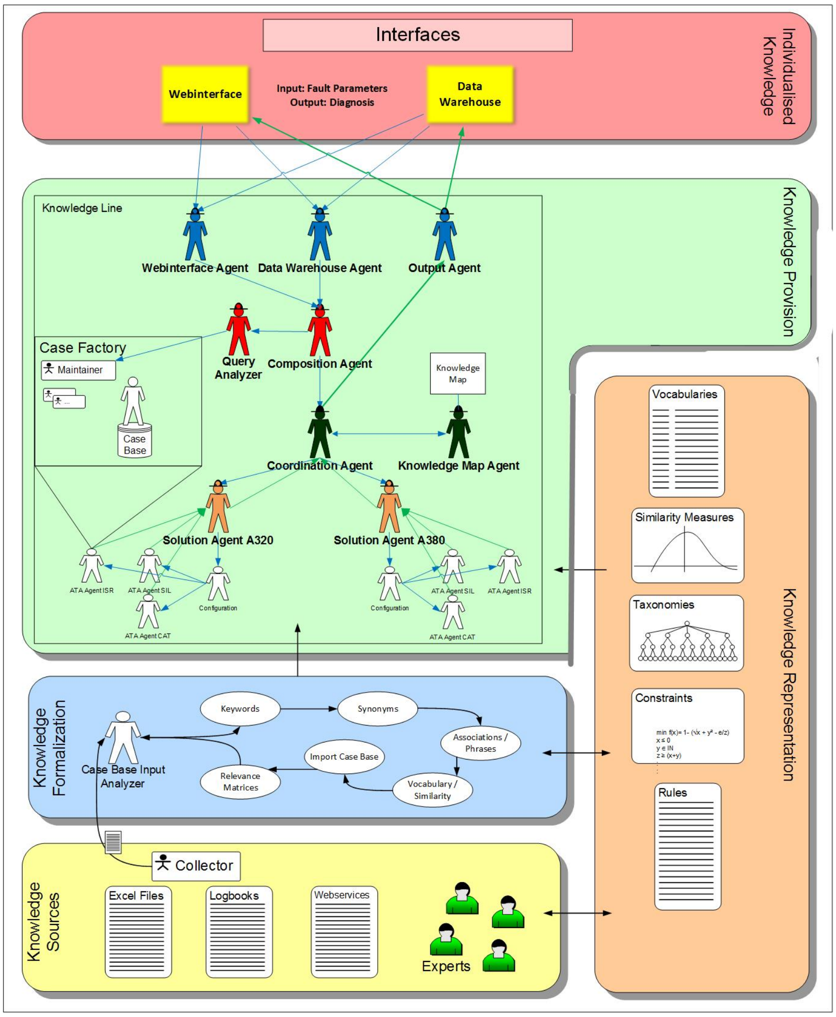

Fig. 2. Instantiation of the SEASALT components within OMAHA

Duplicate synonyms and hypernyms are eliminated and the remaining words are added to the keyword list.

The third task is to identify collocations in addition to the single keywords in the document. While collocations are based on frequently occurring words, the collocation extraction is enhanced by using a vocabulary of technical terms provided by Airbus. This way collocations can be identified even if they occur only a few times, but are relevant to the content. Based on the given technical terms, extracted collocations have a maximum length of five words. All identified collocations are added to a phrase list, while duplicate collocations are removed.

In the next task, all keywords and collocations are added to the vocabulary. The first step is to check the collocations against the keywords, to find combinations of keywords that occurred only as collocation in the given data. The idea is that keywords that do not occur as an individual keyword or as a part of a collocation, but only in the combination of the collocation, will not be added to the vocabulary. This way the growth of the vocabulary can be slowed down.

The fifth task in the workflow contains the setting of initial similarity values between keywords and their synonyms. Due to the fact that words are similar to their synonyms, an initial similarity value of 0.8[7] can be assumed between a word and its synonym. The keywords and synonyms are organized in a matrix. Then the found synonyms and hypernyms are used to build taxonomies for similarity assessments. The hypernyms serve as inner nodes, while the keywords and the synonyms are the leaf nodes. Keywords and their synonyms are sibling nodes if they have the same hypernym. Between sibling nodes a similarity of 0.8 can be assumed. This way existing taxonomies can be extended or new taxonomies can be generated.

Task six is responsible for finding associations between keywords and phrases within a text or between different columns. The idea is to define completion rules based on these associations. An association between keywords or phrases exists, if the combined occurrence frequency exceeds a given threshold. This threshold defines the minimum occurrence of the combination over all analyzed documents and data sets. For example, a combination between two keywords that occurs in more than 70 percent of all analyzed documents, may be used as a completion rule with an appropriate certainty factor. In addition to the occurrence threshold, a threshold for the minimum number of documents to be analyzed during this task has to be defined. This second threshold is required to avoid the generation of rules by analyzing only few documents, but to generate rules with a high significance. Therefore, the second threshold should be more than 1000 documents or data sets. The higher both thresholds are, the more a generated rule is assumed to be significant.

The seventh task is to generate cases from the given documents. The first step uses the mapping document to map the content of the document to a given case structure. The data from the documents are transformed into values for given attributes to fit the structural approach. The generated cases are not added to a

[7] Assuming, here and the further occurrences, that the similarity measures can take values from the [0;1] interval.

single case base, but assigned to several case bases using a cluster algorithm. The idea behind the clustering strategy is to test the scalability of our approach. The idea is to split the cases based on problem description attributes to get smaller case bases for maintenance. Based on the historical data stored at Airbus, a single case base will contain many thousand cases anyway. Generating an abstract case for each case base, a given query can be compared to the abstract cases and this way a preselection of the required case bases is possible.

We assume a homogenous case structure for all cases generated from the documents. The first case is added to a new case base. For the next case, the similarity to the case in the first case base is computed. If the similarity is below a given threshold, a new case base is created and the new case is added. Otherwise the case is added to the existing case base. Each following case is processed in the same way. The similarity to all cases in the case bases is computed and the new case is added into the case base that contains the case with the highest similarity. If the similarity is below the threshold, a new case base is generated. This step is repeated until all generated cases are added to a case base. While the order of the cases has an impact on the clustering, the dimension of the impact has to be cleared.

Task eight uses sensitivity analysis to determine the weights of the problem description attributes, depending on the content of the cases. This sensitivity analysis is processed for every case base created in the task before. As a result initial relevance matrices are created with the diagnoses as rows and the problem description aka symptoms as columns. These relevance matrices will be used to compute the global similarity during a retrieval.

Task nine contains a consistency check of the vocabulary, similarity measures, and cases by a knowledge engineer to confirm or revise the changes made during the workflow. The feedback from the knowledge engineer is used in task ten to improve the individual tasks and steps within the workflow. The task nine and ten should be processed in periodic intervals and during each workflow execution.

This workflow is designed to be executed beside the CBR cycle as a maintenance workflow. Therefore the before mentioned Case Factory is responsible for the changes to the knowledge containers of a CBR system. This way the workflow is distributed to the knowledge formalization component and the knowledge provision component of the SEASALT architecture. One or more agents in the knowledge formalization component are responsible for the analysis tasks and steps and agents in the Case Factory performing the maintenance actions based on the analysis. But the workflow cannot only be used for maintenance beside the CBR cycle, but also within the CBR cycle. During the retrieval step, a query, especially a natural language query, could be analyzed in the same way as a new case. Therefore a "‘lighter"’ version of the workflow could be used, only containing tasks one to six and tasks nine and ten.

3.3 Current implementation

This section describes the current implementation of our workflow for semi-automated knowledge extraction. We implemented the workflow in Java, because

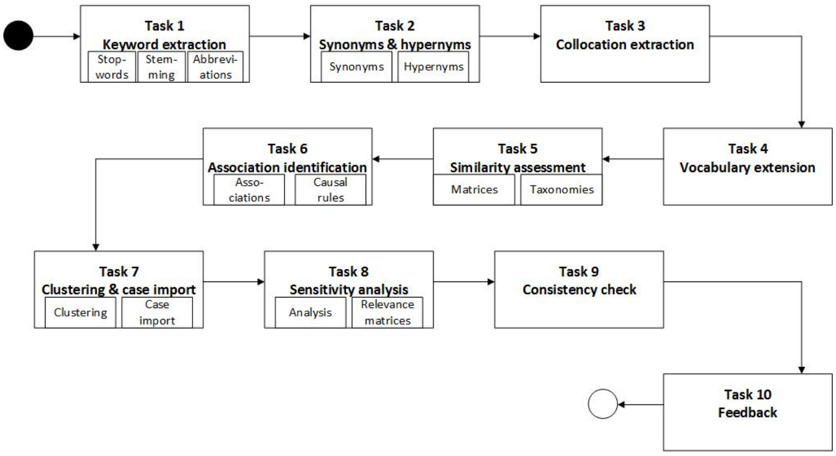

Fig. 3. Workflow for semi-automated knowledge extraction

the used CBR tool and the agent framework are Java based, too. Different import mechanisms are implemented to process data from CSV files, text files, and result sets from a database. Because of the different content and data structures of the documents, the data is processed differently for each document type. CSV files and result sets are processed row-wise, while text documents are processed in the whole. The mapping file is written in XML format and contains the information which column in a CSV file or result set should be mapped to which attribute in the case structure. The following code is an excerpt from the mapping file:

```
<mapping>
<part>problem</part>
<column>AC Type</column>
<attribute>ac_Type</attribute>
</mapping>
```

The keyword extraction is implemented using Apache Lucene and a part-of-speech tagger from the Stanford NLP group. Lucene provides several functions for text analysis, like stopword elimination and stemming and is combined with the Maxent part-of-speech tagger. At first a given input string is tagged with the Maxent tagger and then stopwords are eliminated based on a extended list of English stopwords. This extended list contains all stopwords from the common list of Lucene and some additional words from Airbus' simplified english document. After the elimination of the stopwords, for the remaining words stemming is performed. The result of this step is a list of stemmed keywords. This list is searched for abbreviations based on the Airbus document of used abbreviations in the aircraft domain. All found abbreviations are replaced with the appropriate long word. At last duplicate keywords are removed from the list.

The second task of the workflow is implemented using Wordnet, which provides a large database of synonyms and hypernyms for the English language.

For each keyword from the result list of Task 1 the synonyms are determined via Wordnet database and the found synonyms are stored. After searching for synonyms for the given keywords, an additional search is performed based on the found synonyms. This additional search is repeated until the returned synonyms from the Wordnet database contain only already known synonyms. Based on this list of keywords and synonyms, the Wordnet database is requested for hypernyms and for single worded hypernyms a synonym search is performed. The result of this implemented task is a list of keywords with their synonyms and hypernyms in form of a multiple linked list.

In the third task, collocations are identified based on the raw data with the help of the Dragon toolkit. This toolkit provides a phrase extractor based on the frequent occurrence of collocations and a given set of technical terms provided by Airbus. Before using the extractor the abbreviations in the input string are replaced to match the technical terms. The found collocations are stored in a list.

The next task is implemented using the open source tool myCBR. This tool is used to model the case structure, vocabulary, and similarity measures of our CBR systems. It also provides an API to interact with our workflow. This API is used to add all keywords, synonyms, hypernyms, and collocations to the vocabulary of our CBR systems. The mapping information is used to distribute the added words and phrases to the appropriate attributes in the case structure.

The fifth task is only implemented partially at this time. For the added keywords and their synonyms initial similarity values are set in a symmetric similarity matrix. Each keyword has a similarity value of 0.8 to each synonym. This relationship is bidirectional. Additional content-based similarity values have to be assigned manually. The taxonomy creation is not implemented yet.

After extending the vocabulary and setting the similarity values, cases are generated based on the rows of CSV files or database result sets. For each case a retrieval is performed with the problem description of the case as query using the API of myCBR. If the computed similarity is below 80 percent, a new case base is created and the case is added, otherwise the case is added to the case base with the case that has the highest similarity to the query. This process is repeated until all generated cases are added to a case base. If more than one case base has to be considered for adding a case, the case base with the first found case is enlarged.

3.4 Evaluation setup and results

This section describes the evaluation setup of the current implementation of our workflow and the diagnosis retrieval. The workflow was used to analyze and process 670 data sets with SIL context and 120 data sets with ISR context. From each data set a case was generated. During the first and third task 872 keywords and 76 collocations were extracted. The second task produced 2862 synonyms and 213 hypernyms. In the first evaluation scenario the raw data and the extracted keywords, synonyms, and hypernyms are compared by maintenance experts from Airbus and Lufthansa. In the second evaluation scenario 50 queries are performed on the system with ten cases as retrieval result. These retrieval results are checked by the maintenance experts from Airbus and Lufthansa Systems for appropriate diagnoses to the given queries.

As a result from the first evaluation scenario the experts rated 628 keywords as correct (ca. 72 percent). From the remaining 244 keywords, 98 keywords are wrongly extracted because of false abbreviation replacement or stemming problems, while 146 keywords are false because of an inappropriate word sense. This means there is an overhead of 27 percent from word sense problems. 62 collocation are rated correctly (82 percent), while 14 collocations are wrong, because of false abbreviation replacement. The synonyms and hypernyms have a similar success rate. 2260 synonyms were rated correct and useful, while 602 synonyms were wrong because of inappropriate word sense. Only 124 hypernyms were rated correct, while the remaining 89 hypernyms are wrong as a consequence of the inappropriate synonym word sense.

The result of the second evaluation scenario is that an average of 78 percent of the retrieved cases have an appropriate diagnosis. For each query this number differs slightly. For some queries all retrieved cases were appropriate, for other queries only a few cases were appropriate. Not only the cases itself were checked, but also the ranking of the cases. An average of 18 percent of the retrieved cases were ranked wrong from an expert point of view.

The evaluation shows that the initial version of our workflow produces good result, but there is still potential for improvement. The results from the workflow are good enough to perform a meaningful retrieval, while the number of correct diagnoses has to be improved. The main problem in both scenarios is the word sense of keywords and synonyms that is in many cases not compatible with the aircraft domain. This problem has to be addressed to identify the useful word senses. Another problem is the missing similarity measures for attribute values, which are not synonyms.

4 Related Work

There is a lot of related work on CBR and information extraction, association rule mining, processing textual data in CBR and text mining. This section contains a selection of related work from these topics. Bach et al. describe in their paper an approach for extraction knowledge from vehicle in-service reports. This approach is also based on the SEASALT architecture like our approach, but uses only automated keyword extraction to process the reports. As an additional step the extracted keywords are classified. Then the extracted keywords are reviewed by experts and inserted manually into the vocabulary [5]. Our approach still has the review process of an expert or knowledge engineer, but aims on a more detailed text processing workflow with phrases, synonyms and hypernyms. We try to create a more automated workflow to populate the vocabulary and initial similarity measures.

In their article about knowledge extraction from web communities, Sauer and Roth-Berghofer describe the KEWo Workbench and the mechanisms provided by this workbench to extract knowledge from semi-structured texts. The KEWo workbench is able to create taxonomies from extracted keywords and phrases based on the relative frequency of the occurrence [11]. In our approach we will generate the taxonomies not from the relative frequency, but from found hypernyms and synonyms from the Wordnet database and useful technical terms from the aircraft domain vocabulary.

Many systems with textual knowledge use the textual CBR approach, like [12], [10] and [7]. The data sources available for our project are mainly structured data, therefore we choose a structural CBR approach. But the most important information about an occurred fault can be found in fault descriptions and logbook entries, which are free text. We decided to use a hybrid approach with the combination of structural CBR and textual CBR techniques, to integrated all available information.

[8] describes an approach for enriching the retrieval using associations. They use the Apriori algorithm to extract relevant cases for correlation between cases. We will use algorithm like Apriori or FP-Growth to extract associations between attribute values in a case. This aims on generating completion rules to enrich a query by setting attribute values automatically based on the completion rules.

5 Summary and Outlook

In this paper we described the idea of a semi-automatic knowledge extraction workflow for a decision support system within the aircraft domain. We give an overview over the decision support system and the tasks and substeps of the workflow. In addition, we show our current implementation of the workflow and the evaluation results, based on the current implementation.

As the evaluation shows there is potential for improvement of the individual tasks of the workflow as well as for the complete workflow. The main problem of the inappropriate word sense, that causes the overhead of the vocabulary and the similarity measures, will be addressed by the extend use of an aicraft domain vocabulary provide by Airbus and Lufthansa Systems. Another idea for solving this problem is to restrict the adding of keywords, based on the relative occurrence frequency. In addition to the enhancement of implemented tasks, the next steps will be the implementation of the tasks for taxonomy creation, the sensitivity analysis and association extraction.

References

1. Althoff, K.D.: Collaborative multi-expert-systems. In: Proceedings of the 16th UK Workshop on Case-Based Reasoning (UKCBR-2012), located at SGAI International Conference on Artificial Intelligence, December 13, Cambride, United Kingdom. pp. 1–1 (2012)
2. Althoff, K.D., Bach, K., Deutsch, J.O., Hanft, A., Mänz, J., Müller, T., Newo, R., Reichle, M., Schaaf, M., Weis, K.H.: Collaborative multi-expert-systems – realizing knowledge-product-lines with case factories and distributed learning systems. In: Baumeister, J., Seipel, D. (eds.) KESE @ KI 2007. Osnabrück (Sep 2007)

3. Althoff, K.D., Reichle, M., Bach, K., Hanft, A., Newo, R.: Agent based maintenance for modularised case bases in collaborative mulit-expert systems. In: Proceedings of the AI2007, 12th UK Workshop on Case-Based Reasoning (2007)
4. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), dr. Hut Verlag Mnchen
5. Bach, K., Althoff, K.D., Newo, R., Stahl, A.: A case-based reasoning approach for providing machine diagnosis from service reports. In: Case-Based Reasoning Research and Development. International Conference on Case-Based Reasoning (ICCBR 2011). pp. 363–377 (2011)
6. BMWI: Luftfahrtforschungsprogramms v (2013), `http://www.bmwi.de/BMWi/Redaktion/PDF/B/bekanntmachung-luftfahrtforschungsprogramm-5,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf`
7. Ceausu, V., Despres, S.: A semantic case-based reasoning framework for text categorization. In: The Semantic Web, Lecture Notes in Computer Science. pp. 736–749 (2007)
8. Mote, A., Ingle, M.: Enriching retrieval process for case based reasoning by using certical association knowledge with correlation. International Journal on Recent and Innovation trends in Computing and Communication 2, 4114–4117 (2015)
9. Richter, M., Wess, S.: Similarity, uncertainty and case-based reasoning. Automated Reasoning - Essays in Honor of Woody Bledsoe 1, 249–265 (1991)
10. Rodrigues, L., Antunes, B., Gomes, P., Santos, A., Carvalho, R.: Using textual cbr for e-learning content categorization and retrieval. In: Proceedings of International Conference on Case-Based Reasoning (2007)
11. Sauer, C.S., Roth-Berghofer, T.: Extracting knowledge from web communities and linked data for case-based reasoning systems. Expert Systems, Special issue in Innovative Techniques and Applications of Artificial Intelligence 31, 448–456 (2013)
12. Weber, R., Aha, D., Sandhu, N., Munoz-Avila, H.: A textual case-based reasoning framework for knowledge management applications. In: Proceedings of the ninth german Workshop on Case-Based Reasoning. pp. 244–253 (2001)