Multi-Agent Case-Based Diagnosis in the Aircraft Domain

Pascal Reuss[1,2], Klaus-Dieter Althoff[1,2], Alexander Hundt[1], Wolfram Henkel[3], and Matthias Pfeiffer[3]

[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

Abstract. Aircraft diagnosis is a highly complex topic. Many knowledge sources are required and have to be integrated into a diagnosis system. This paper describes the instantiation of a multi-agent system for case-based aircraft diagnosis based on the SEASALT architecture. This system will extend a existing rule-based diagnosis system, to make use of the experience of occurred faults and their solutions. We describe the agents within our diagnosis system, the planned diagnosis workflow and the current status of the implementation. For the case-based agents, we give an overview of the initial case structures and similarity measures. In addition, we describe some challenges we have during the development of the multi-agent system, especially during the knowledge modeling.

1 Introduction

An aircraft is a complex mechanism, consisting of many subsystems. Occurring faults cannot be easily tracked to their root cause. A fault can be caused by one system, by the interaction of several systems or by the communication line between systems. Finding the root cause can be very time and resource consuming. Therefore the use of experience from successfully found and solved root causes can be very helpful for aircraft diagnosis. This paper describes the instantiation of a multi-agent system (MAS) based on the SEASALT architecture. The MAS contains several Case-Based Reasoning (CBR) systems to store the experience and provide this knowledge for diagnosis. In the next section, we give an overview of the OMAHA project and the SEASALT architecture. Section 2 contains related work with comparing our approach to other diagnosis and multi-agent approaches. In Section 3 we describe the instantiation of the SEASALT components for our MAS and describe the case-bases agents with the case structure and similarity measures of the underlying CBR systems in more detail. Section 4 gives a short summary of the paper and an outlook on future work.

1.1 OMAHA project

The multi-agent system for case-based aircraft diagnosis is under development in the context of the OMAHA research project. This project is supported by the German Federal Ministry of Economy and Energy and tries to develop an **O**verall **M**anagement **A**rchitecture for **H**ealth **A**nalysis of civilian aircraft. Several topics are addressed within the project 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 integrate not only the aircraft and its subsystems, but also systems and processes in the ground segment like manufacturers, maintenance facilities, and service partners into the maintenance process. Several enterprises and academic and industrial research institutes take part in the OMAHA project: the aircraft manufacturer Airbus, the system manufacturers Diehl Aerospace and Nord-Micro, the aviation software solutions provider Linova and IT service provider Lufthansa Industrial Solutions 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 project started in 2014 and will last until the end of March, 2017. [1]

The OMAHA project has several different sub-projects. Our work focuses on a sub-project to develop a so-called integrated system health monitoring (ISHM) for aircraft systems. The main goal is to improve the existing diagnostic approach to identify faults with root cause in more than a single subsystem (cross-system faults). Therefore, a multi-agent system (MAS) with several case-based agents will be implemented to integrate experience into the diagnostic process and provide more precise diagnoses for given faults.

1.2 SEASALT architecture

The SEASALT (**S**hared **E**xperience using an **A**gent-based **S**ystem **A**rchitecture **L**ayou**T**) 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 interface. The knowledge modularization allows the compilation of comprehensive solutions and offers the ability of reusing partial case information in form of snippets.

The SEASALT architecture consists of five components: knowledge sources, knowledge formalization, knowledge provision, knowledge representation, and 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.

The knowledge formalization component is responsible for formalizing the extracted knowledge into a modular, structural representation. This formalization

[1] `www.dlr.de/lk/desktopdefault.aspx/tabid-4447/7274_read-39606`

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.

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 Agents. 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],[9].

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.

1.3 Application domain: aircraft diagnosis

An aircraft is a highly complex machine consisting of a large number of subsystems that interact with each other, like hydraulic, cabin, ventilation, and landing gear. Each subsystem has a large number of components. The smallest component that can be replaced during maintenance is called Line Replacement Unit (LRU). The challenge is to find the root cause of a fault, because there could be more than one LRU causing the fault or a fault chain. In a fault chain, the first fault causes additional faults, which could also cause additional faults again. Faults are not limited to have their root cause in the subsystems that stated the fault, but the root cause can be found in a different subsystem. Therefore, a cross-system diagnosis is required to improve the precision of the diagnosis process.

In the next section we give an overview of some related work. In Section 3 we describe the multi-agent system concept and the instantiation of the SEASALT architecture. Section 3.3 describes the current status of our implementation. Finally a summary and outlook on future work is given.

2 Related Work

Decision support for diagnosis (and maintenance) in the aircraft domain means that a lot of engineering knowledge is available to support this process. In the past various diagnostic approaches tried to improve diagnosis and maintenance in this domain: among others case-based reasoning, rule-based reasoning, model-based reasoning, Bayesian belief networks, Fuzzy inference, neural networks,

fault trees, trend analysis, and a lot of combinations. For OMAHA, that is OMAHA work package 230, the exploitation of available experiences as supplementation to other already used knowledge sources is of high priority. See also the work from Reuss et al.[10] for relating our approach with a selection of related other experience reusing diagnostic approaches: the British research project DAME [7] dealing with fault diagnosis and prognosis based on grid computing , Dynamic Case-Based Reasoning [13] learning also through statistic vectors containing abstract knowledge condensed from groups of similar cases, and the hybrid approach of Ferret and Glasgow [6] combining model-based and case-based reasoning.

For optimizing the relation between cost and benefit we decided to use the various available textual knowledge sources (cf. also Section 3). A recent overview of using textual sources for CBR is given in the textbook of Richter and Weber [12]. The paper of Reuss et al. [11] also gives an overview of some related approaches in this direction.

In addition to other specific characteristics of our approach one property differentiating it from many other (CBR) approaches is the fact that we develop a multi-agent system that applies a lot of CBR agents (among other) ones. The following approaches have in common that they also combine a multi-agent system approach with CBR. Researchers also dealing with CBR from different perspectives and trying to combine the specific insights to an improved overall approach are [16]. Of course, what makes our approach different here is that we are concerned with the development of concrete framework with existing applications. Corchado et al.[5] present in their work an architecture for integrating multi-agent systems, distributed services, and application for constructing Ambient Intelligence environments. Besides addressing a different domain and task this approach appears to be more open concerning the potential tasks agents can take over, while our approach is more focused in applying software engineering strategies for decomposing problems into sub-problems resulting in a distributed knowledge-based system. Zouhaire and his colleagues[17] developed a multi-agent system using dynamic case-based reasoning that learns from traces and is applied for (intelligent) tutoring. Our approach does not learn from traces but instead has to deal with a lot of technical knowledge and in addition has to solve critical problems. Srinivasan, Singh and Kumar[14] share with our approach that they develop a conceptual framework for decision support systems based on multi-agent systems and CBR systems. Our approach appears to be more on the side of integrating software engineering and artificial intelligence methods implementing concrete application systems, while the authors discuss how their framework influences decision support system in general. Khelassi[8] developed the IK-DCBRC system basing on a multi-agent architecture using a CBR approach with fuzzy-enhanced similarity assessment and being able to explain the results for different users. Our approach is not explanation-aware with respect to its current implementation status, however there is a conceptional extension of the SEASALT architecture (together with Thomas Roth-Berghofer and his research team) defined that includes explanation awareness. In addition,

there are two PhD research projects ongoing focusing on explanation awareness. What also makes us different from Khelassis work is that our approach is embedded in an overall methodology resulting in a systematic process of how to develop an instance of our architecture with applications in travel medicine, technical diagnosis, and architectural design.

3 Multi-agent case-based diagnosis in the aircraft domain

In this section we describe the current version of our multi-agent system for case-based diagnosis. Based on the SEASALT architecture we describe the instantiation of the single components in context of our multi-agent system and the diagnosis workflow. In addition, we give an overview over the case structures and similarity measures of our case-based agents.

3.1 Multi-agent system for aircraft diagnosis

First we will describe the instantiation of our multi-agent system. The multi-agent system is an additional component of the diagnosis mechanism. It will not replace the existing rule-based diagnosis, but will extend the current diagnosis mechanism. The main component for our multi-agent aircraft diagnosis is the *knowledge provision* component. This component contains the Knowledge Line, which is responsible for providing a diagnosis for a given fault situation. The Knowledge Line consists of several topic agents with underlying CBR systems. The topic agents use the knowledge of their CBR systems to provide a part of the diagnosis. If only the knowledge of one topic agent is required, the topic agents delivers the complete diagnosis. There are several homogeneous teams of topic agents in the Knowledge Line, each responsible for diagnoses of an aircraft type (e.g., A320, A350, or A380). Each team has an additional agent, called solution agent to coordinate the topic agents and rank the individual solutions. Because each individual solution represents a possible diagnosis, a combination of solutions is not appropriate. The approach of separated agent teams for each aircraft type is based on the idea to split the knowledge into several smaller CBR systems. This way the number of cases for a retrieval and the maintenance effort for each system can be reduced. Nevertheless, for a diagnosis more than one agent team may be necessary. Therefore, a query can be distributed to several agent teams, either by default or if a diagnosis from the primary agent team for a query cannot provide a sufficient diagnosis. A coordination agent is responsible for coordinating the agent teams, distributing a query, and combining the team's solutions. The complete diagnosis process requires some more software agents that do not belong to the Knowledge Line itself: an interface agent, a composition agent, a knowledge map agent, and an output agent. The interface agent receives the query either from a web interface and/or a data warehouse. The main data source is a Post Flight Report (PFR) containing all the faults having occurred during the last flight of an aircraft. This PFR is based on the rule-based diagnosis in the aircraft. Each fault is represented as a so-called PFR

item. Additional data like aircraft configuration, operational parameters (e.g., weather conditions, temperature, etc.), and logbook entries can be received, too. The PFR data and the additional data have to be correlated to assign the additional data to the corresponding PFR item. This task is done by the correlation agent. The extended PFR items are sent to the coordination agent. For each PFR item, a request to one or more agent teams is performed. To determine which topic agents of a team should be requested, a so-called Knowledge Map is used. This Knowledge Map contains information about existing agents and their dependencies and underlying CBR systems. The task to determine a so-called retrieval path (the topic agents to be requested and the sequence of retrievals) is done by a knowledge map agent. This agent has access to the general Knowledge Map and a CBR system, which stores past successful retrieval paths for given fault situations. The knowledge map agent uses the CBR system to retrieve the most similar retrieval paths and adapt the path to the new situation if necessary. Based on the determined retrieval path, the topic agents are requested and a ranked list of diagnoses is generated. The list of diagnoses is sent to the output agent. The output agent forwards the list to the web interface and the data warehouse. One more agent is located in the *knowledge provision* component. The so-called query analyzer takes each extended PFR item and checks for new concepts, which are not yet part of the vocabulary of the CBR systems. If any new concepts are found, a maintenance request is sent to the so-called Case Factory [9]. The Case Factory checks the maintenance request, derives appropriate maintenance actions, and executes the actions after confirmation from a knowledge engineer. The query analyzer is not part of the diagnosis process itself, but provides some learning capabilities to the multi-agent system.

The user interface can be found in the *individualized knowledge* component. The user interface is a web interface, which provides the options to send a query to the multi-agent system and present the returned diagnoses. In addition, the user can enter new cases, edit existing cases, and browse a entire selected case base. In addition to the web interface, a connection to a data warehouse is part of this component. The data warehouse contains PFRs and the additional data and will be the main query source for the multi-agent systems. If additional information is required that is not provided by the data warehouse, it can be added via the web interface.

The *knowledge formalization* component transforms structured, semi structured, and unstructured data into a modular, structural knowledge representation used by all CBR systems. This way the knowledge is represented in the same way all over the multi-agent system. Because a structural approach for the CBR systems in the Knowledge Line was chosen, semi-structured and unstructured data have to be transformed into attribute value pairs. This transformation workflow is performed by a so-called case base input analyzer. The workflow consists of several steps: At first, information extraction methods are used to extract keywords and collocations and to find synonyms and hypernyms for the extracted keywords. Then the input data is analyzed to find associations within the allowed values of an attribute as well as across different attributes.

This way want to generate completion rules for query expansion. The keywords, synonyms, hypernyms, and collocations are added to the vocabulary and initial similarity values for keywords and their synonyms are set. The keywords and their hypernyms can be used to generate taxonomies for similarity measures. After the vocabulary extension, cases are generated from the input data and stored in the case bases. The last step is to perform a sensitivity analysis on the stored cases to determine the weighting for the problem description attributes. The workflow is presented in more detail in [11].

In the *knowledge sources* component a collector agent is responsible for finding new data in the data warehouse, via web services or in existing knowledge sources of Airbus. New data could be new configurations or operational parameters, new synonyms or hypernyms, or complete new cases.

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

3.2 Case-based agents

This section focuses on the case-based agents within our multi-agent diagnosis system. We will describe the agents' tasks and the underlying CBR systems with their case structure and similarity modeling. In addition to the PFR, we have to consider several different data structures like Service Information Letters (SIL), In-Service Reports (ISR), elogbooks and aircraft configuration documents. While a PFR contains only information about the problem description, SIL, ISR and eLogbooks contain problem descriptions and solutions. Configuration documents contain data about the latest system configuration of an aircraft with hard- and software versions. We performed an analysis on these data to identify relevant information for cases, relationships between these information and data anomalies. Based on the result of this analysis we derived two case structures with attribute-value pairs and their value ranges. One case structure is based on PFR and SIL (C_{SIL}) and the other case structure is based on PFR and ISR (C_{ISR}). The case structures overlap to some degree, because attributes derived from the PFRs are part of both structures, like ATA chapter, aircraft type, and fault description. The C_{SIL} structure contains 32 attributes, while the C_{ISR} structure consists of 28 attributes. The attributes are distributed among problem description, diagnosis, quality information, and pointer to other cases. The problem description contains attributes like ATA chapter, aircraft type (e.g., A380), aircraft model (e.g., 380-641), fault code, displayed message, fault description and affected Line Replacement Units (LRU). Attributes like recommendation, comments, maintenance reference, corrective LRUs and root cause are part of the solution. For quality assessments the number of positive (a retrieved diagnosis was helpful) and negative (a retrieved diagnosis was not helpful) retrievals are stored.

The configuration of an aircraft has great impact on the probability of fault occurrence. If a certain system is not built in, corresponding faults will not occur. The occurrence of faults depends also on the soft- and hardware version

of built in systems. Therefore, the configuration of an aircraft can exclude faults and root causes and have an impact on the similarity of cases. Because of the complexity of the configuration data for an entire aircraft, we decided to consider the configuration separate for each aircraft component. For each subsystem of a component the so-called modification status (mod-status) is stored. With the help of this mod-stati, cases could be excluded and similar configuration could be compared.

Most of the attributes have a symbolic data type and a taxonomy as similarity measure. The attributes ATA chapter, fault code and affected LRUs have a natural hierarchical structure, that can be mapped to a taxonomy. A great challenge is the similarity measure for the fault description attribute. The symbolic values of this attribute are extracted via a workflow in the knowledge formalization component as described in [11]. During the automatic vocabulary expansion, the values are added to a similarity table. Similarities between the automatically added values are only set between values and their synonyms. The other values have to be set manually. To reduce the effort, an automatic taxonomy generation from the extracted values and their synonyms and hypernyms is planned.

The multi-agent system will contain several topic agents with the same case structure to reduce the number of cases in one case base. Most faults can be assigned to a specific ATA chapter. Therefore, for each ATA chapter an own topic agent is generated. An agent team within our multi-agent system will consist of agents discriminated by ATA chapter and data source (SIL, ISR, etc.).

Another case-based agent is the so-called knowledge map agent. This agent is responsible for determining which topic agents have to be requested to find a solution for a given request. For each request, a retrieval on the underlying CBR system is performed. The cases will contain the characteristics of a request as the problem description and a successfully used retrieval path. This way we try to address the cross-system faults. Cross-system faults may have their root causes in LRUs of different ATA chapters. Requesting only the topic agent of a single ATA chapter may not give the correct root cause identification and diagnosis. Based on experience from solved faults, the cases for the knowledge map agent could contain information when the request of additional topic agents may be useful to find the correct diagnosis.

There are several challenges to be met while modeling the case structures and the similarity measures. One major challenge is based on the fact, that the ATA chapter differs for the same subsystem in different aircraft types. The cabin entertainment system is linked to two different ATA chapters in the A320 and the A380. Therefore, a mapping between the different ATA chapters is required to compare fault cases from different aircraft types. Another challenge is modeling the fault description in the case structure. The description of a fault is mainly given in free text provided by pilots or cabin personal. Unfortunately, there is no standard description language for faults. Therefore, every person describes a fault with slightly different words and technical terms. Extracting the key symptoms from this fault descriptions and comparing two fault descriptions requires the integration of natural language processing techniques in the modeling process

and the diagnosis process of the multi-agent system. In addition, the amount of knowledge that can be found in the fault descriptions is very high. Analyzing 3000 example fault descriptions, we found more than 21000 different keywords and phrases describing the occurred faults. Modeling all these keywords and phrases in one attribute is not practicable. While it is possible to add all keywords automatically, setting the similarity between these keywords within a similarity matrix or a taxonomy is not practicable. In addition, the maintenance effort for such an attribute would be very high and in no relation the gained benefit.

The main challenge for the knowledge map agent is to identify the characteristics of a request and the according knowledge sources to solve the request.

3.3 Status of implementation

We implemented a prototype to test some functionalities of the desired multi-agent system. This application serves as a testing system for knowledge modeling and diagnosis process. The prototype consists of two CBR systems and a user interface to interact with the systems. We modeled the case structure, vocabulary and similarity using the open source tool myCBR [15]. One CBR system contains cases based on SIL documents, the other one on ISR documents. The SIL case base contains 670 cases and the ISR case base 220 cases. The user interface provides the functionalities to perform a retrieval, enter new cases, edit existing cases, and browse the case base based on filter criteria. In addition, the workflow of the knowledge extraction is partially implemented. The keyword extraction, collocation extraction, synonym and hypernym identification, and automatic vocabulary extension are implemented. For more detail on the implementation of the knowledge extraction workflow see [11].

4 Summary and Outlook

In this paper we describe the instantiation of our multi-agent system for case-based diagnosis. We give an overview of the individual components and describe the case structure and similarity of our case-based agents. As Section 3.3 shows, the multi-agent system is not fully implemented, yet. The next steps are the implementation of the additional agents (interface, coordination, output, knowledge map) and the refinement of the case structures and similarity measures. In addition, the learning mechanism based on the knowledge extraction workflow will be realized.

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., Hanft, A., Schaaf, M.: Case factory - maintaining experience to learn. Advances in Case-Based Reasoning Lecture Notes in Computer Science 4106/2006, 429–442 (2006)
4. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), dr. Hut Verlag Mnchen
5. Corchado, J.M., Tapia, D.I., Bajo, J.: A multi-agent architecture for distributed services and applications. International Journal of Innovate Computing 8, 2453–2476 (2012)
6. Feret, M., Glasgow, J.: Combining case-based and model-based reasoning for the diagnosis of complex devices. Applied Intelligence 7, 57–78 (1997)
7. Jackson, T., Austin, J., Fletcher, M., Jessop, M.: Delivering a grid enabled distributed aircraft maintenance environment (dame). Tech. rep., University of York (2003)
8. Khelassi, A.: Reasoning System for Computer Aided Daignosis with explanation aware computing for medical applications. Ph.D. thesis, Abou Bakre Belkaied University, Tlemcen, Algeria (2013)
9. Reuss, P., Althoff, K.D.: Explanation-aware maintenance of distributed case-based reasoning systems. In: LWA 2013. Learning, Knowledge, Adaptation. Workshop Proceedings. pp. 231–325 (2013)
10. Reuss, P., Althoff, K.D., Henkel, W., Pfeiffer, M.: Case-based agents within the omaha project. In: Case-based Agents. ICCBR Workshop on Case-based Agents (ICCBR-CBR-14) (2014)
11. Reuss, P., Althoff, K.D., Henkel, W., Pfeiffer, M., Hankel, O., Pick, R.: Semi-automatic knowledge extraction from semi-structured and unstructured data within the omaha project. In: Proceedings of the 23rd International Conference on Case-Based Reasoning (2015)
12. Richter, M.M., Weber, R.: Case-Based Reasoning - A Textbook. Springer-Verlag Berlin Heidelberg (2002)
13. Saxena, A., Wu, B., Vachtsevanos, G.: Integrated diagnosis and prognosis architecture for fleet vehicles using dynamic case-based reasoning. In: Autotestcon 2005 (2005)
14. Srinivasan, S., Singh, J., Kumar, V.: Multi-agent based decision support system using data mining and case based reasoning. International Journal of Computer Science Issues 8, 340–349 (2011)
15. Stahl, A., Roth-Berghofer, T.: Rapid prototyping of cbr applications with the open source tool mycbr. In: Advance in Case-Based Reasoning, Proceeding of the 9th European Conference on Case-Based Reasoning (2008)
16. Sun, Z., Han, J., Dong, D.: Five perspectives on case based reasoning. In: 4th International Conference on Intelligent Computing. pp. 410–419 (2008)
17. Zouhair, A., En-Naimi, E.M., Amami, B., Boukachour, H., Person, P., Bertelle, C.: Incremental dynamic case based reasoning and multi-agent systems (idcbr-mas) for intelligent touring system. International Journal of Advanced Research in Computer Science and Software Engineering 3, 48–56 (2013)