Case-based agents within the OMAHA project

Pascal Reuss[1 2], Alexander Hundt[1], Klaus-Dieter Althoff[1 2], 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. For aircraft fault diagnosis much knowledge is required. This knowledge is distributed over various knowledge sources. In this paper we present our approach for a decision support system for diagnosis and maintenance within the OMAHA research project. We compare our approach to other diagnostic approaches for technical diagnosis and describe the case-based agents within the decision support system in more detail.

1 Introduction

The aircraft industry may very well be considered to have one of the highest demands for their maintenance personnel in terms of safety requirements and complexity. The latter is not only influenced by the complexity of one single aircraft alone but the one of a long-term lifespan with product life-cycles up to 50 years, ongoing development and introduction of new aircraft models. To describe aircraft maintenance procedures in a most basic fashion one can imagine a set of detection sensors for every piece of equipment which, once triggered, result in electronic error messages. Once the aircraft has landed all error messages are enriched with possible root causes, aggregated into reports and finally distributed via maintenance plans towards an airlines mechanic. Although driven by aerospace regulations and thus being highly standardized, maintenance procedures are still subject to optimization efforts.

1.1 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 [4]. The high-level goal of the OMAHA project is to develop an **O**verall **M**anagement **A**rchitecture for **H**ealth **A**nalysis of civilian aircrafts. 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, the system 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 project started in 2014 and will go 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 cross-system integrated system health monitoring (ISHM) for aircraft systems. 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. In the following subsection our initial concept for a decision support system for diagnosis and maintenance is described in more detail.

In the next section we give an overview of our system approach (Section 2). In Section 3 we characterize our approach and compare it to other approaches to technical diagnosis. We then give a more detailed description of three case-based agents for knowledge provision, adaptation, and planning that play an important role for maintaining our distributed knowledge bases (Section 4). Finally a short summary and outlook are given.

2 Decision support system for diagnosis and maintenance

Following a structured and thus later on easier maintainable approach we leverage elements of the SEASALT architecture for our design. SEASALT describes a domain independent architecture for extracting, analyzing, sharing, and providing experiences (Sharing Experience using an Agent-based System Architecture LayouT). It is especially intended for systematic development of distributed knowledge-based systems with a specific focus on CBR. It provides different layers for individual CBR-related task groups and distributes CBR knowledge over different CBR systems/agents.[[3]]

SEASALT in its elementary architecture consists of the following layers, briefly described. **Knowledge sources**, which represent not only actual sources (e.g. databases or textual web contents) but also dedicated agents that extract and collect the knowledge. **Knowledge formalization**, which provides necessary intelligent transformation processes of the acquired knowledge and aims at formalizing this knowledge into independent structures, either by a human or an

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

artificial agent. **Knowledge representation**, which provides a unified underlying knowledge model and thus offering great interoperability between individual components and layers. Different kinds of knowledge can be represented like case, rules, terminology and similarity knowledge. **Individualized knowledge** represents the user interface layer. [[3]] For our herein presented approach we will now discuss the layer **knowledge provision** in more detail.

As we glimpsed at aircraft maintenance procedures at the beginning of this paper, the established maintenance process ranges from the first signal aboard the aircraft over an accumulated set of reports to clear maintance instructions for a mechanic once the plane has landed. Our approach is meant to aid decisions in this process, as it will not suffice to completely replace existing systems, due to the nature of Case-Based Reasoning (Fig. 1).

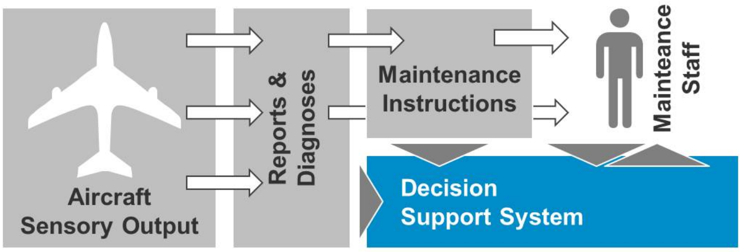

Fig. 1. Simplified scenario for aircraft maintenance

In order to enhance existing rule- or model-based diagnoses, which may lack of mechanisms to deal with exceptions, we leverage the SEASALT approach in our MAS to process valuable experience from human technicians in maintenance. Therefore Case-based software agents are introduced to diagnose occurred failures and suggest additional maintenance actions. In the existing diagnoses correlation rules are used to identify possible root causes for displayed failure message. Ultimately the rule-based approach will create a list of prioritized items which can be used by a human technician to plan maintenance procedures. In case of exceptions that are not covered by these rules, additional deviation lists with possible known exceptions are provided to the technician.

The existing system suffers from several drawbacks which are addressed by our proposed MAS enhancement, especially focusing on challenges with incorporating exceptions in the maintenance process. We aim at capturing the experience gained by mechanics and providing this knowledge within the reasoning process for creating maintenance plans. Additionally we seek to reduce efforts necessary to integrate new knowledge, which are typically rather high when dealing with complex rule-systems. The following augmentation approach describes the intention of a future decision support system.

In brief the MAS will be present a supplementary solution to the mechanic in addition to the regular solution from the established system. The MAS will either confirm the same diagnosis and maintenance action of the existing system or present a different solution and maintenance action, in case the stored experience leads to a different solution.

We now further elaborate on the process intended by us to incorporate case-based agents into the established and highly sensitive maintenance diagnosis. The MAS uses several input sources throughout an aircraft's typical work cycle, ranging from electronic signals logged during flight to textual notes from logbook entries. Information is accumulated in post-flight reports which consist of items, each of those to be handled by a technician in compulsory series of tasks. Each item consists of a diagnosis and a maintenance proposal. The items, broadly speaking, are each the result of a rule-based reasoning (RBR) system. Now in addition to these strict procedures each diagnosis and suggestion will be retrieved by the MAS and processed as supplementary CBR query. As a result we think of three different outcomes in this system.

1. The maintenance solution from RBR system is identical to the displayed diagnosis and maintenance proposal from the CBR system.
2. The CBR system recognizes a different diagnosis which would lead to a different maintenance proposal.
3. The CBR system has the same diagnosis as the the RBR system but differs in the resulting maintenance proposal.

We will illustrate these three decision support outcomes along our aformentioned maintenance scenario. Figure 2 demonstrates the most relevant information flows where "A" represents the RBR systems output, "B" represents the CBR systems output, "C" represents relevant maintenance instructions for either system's output and "D" represents the feedback that enables our CBR system to learn and improve. Output is in both systems' cases defined as diagnoses and a proposed maintenance approach. Maintenance proposals must follow explicit maintenance instructions which are in any case provided seperately.

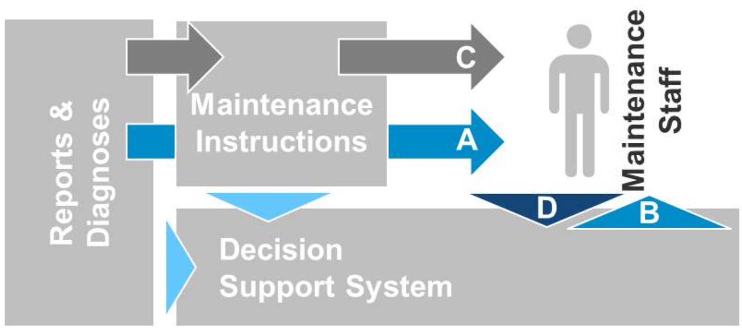

Fig. 2. Simplified workflow for cross-system diagnosis

In the first outcome the received diagnoses and maintenance plans from A and B are identical, thus the mechanic proceeds as usual but returns an additional feedback to the DSS afterwards.

In the second outcome A and B result in different diagnoses. Both systems use the same initial data input yet approach their reasoning differently. While the RBR system follows its established model the CBR system incorporates knowledge about known exceptions which are subject of different reports that are not part of the RBR system. We think of this second event whenever there is not enough evidence by the RBR system to come to a fully satisfied conclusion about the root cause. In this case the mechanic will need to decide on which diagnosis he will focus. The CBR system is intended to support this decision by offering additional information about known exceptions for this diagnosis as well as the frequency of correct or false retrievals of the displayed solution. If the inherited maintenance procedures of the diagnosis the technician decided on did not lead to a fault repair, he follows the next relevant diagnosis in line.

In the third outcome the displayed diagnoses of A and B are identical, however differ in the proposed maintenance plan. Again the human technician will decide which maintenance action will be executed by him first. In case all regular maintenance procedures for the diagnosis fail, the mechanic needs to find solution on his own, thus creating a new exceptional case. Thesis is, that this event will occur whenever a new diagnoses is not covered by a rule-based maintenance plan yet. The fault repair relies on the personal experience of the mechanic which it is the intention of our approach to make these experiences available for other technicians in a comparable situation.

The intended workflow incorporates human feedback (D) in each of the aforementioned outcomes, thus improving the solution retrieval of the MAS with each cycle and especially learning new cases from human experience without the need to derive new rules for the RBR.

To maintain the CBR systems within the MAS so-called Case Factories are used. The basic idea was developed by Althoff et al.[1] and improved by Reuss and Althoff[9] to support the maintenance of distributed CBR systems. The Case Factory uses several agents to evaluate and maintain a single CBR system, four agents to evaluate the CBR system, one for each knowledge container[10], and four agents to maintain the knowledge containers. Based on evaluation results and user feedback, maintenance actions are derived. To coordinate multiple Case Factories with dependencies between the knowledge, a so-called Case Factory Organization (CFO) is used. This CFO contains several software agents for logging, planning and explanation tasks. The maintenance actions from the single Case Factories and additional actions based on the dependencies between the knowledge are combined to a maintenance plan. This plan is enhanced with explanations to support the understanding of a knowledge engineer. The knowledge engineer confirms the plan or parts of the plan and the confirmed maintenance actions are executed.

3 Related Work

DAME [6] was a British research project that started about twelve years ago and lasted several years. The subject of DAME was the design and the implementation of a fault diagnosis and a prognostic system based on grid computing and the related development of grid services. Its special focus was on the development of an improved computer aided fault diagnosis and prognosis ability and the integration of these features to a preventive maintenance program. The context of the developed demonstrator was aircraft engine diagnosis. To facilitate engine fleet management, engine sensor data was routinely analyzed using the COMPASS health monitoring application, developed by Rolls-Royce, and prognostic applications, employed by Data Systems and Solutions. The On-Wing-Monitoring-System QUOTE made it possible to detect the basic reasons of unknown anomalies and tried to line up suitable measures of remedy. DAMEs data mining service consisted of the AURA pattern match system that allowed the required fast search among flight data archives by means of a special pattern matching method. Within the DAME demonstrator context the AURA system supplied vibration data that suited the varying conditions of the engines best. CBR was used for the Flightline Maintenance Advisor that was tested by Singapore Airlines. Cases were developed based on the knowledge of engineers and mechanics about development, maintenance, and lifetime of the engines. DAME achieved some experience in integrating different services; however according to our understanding, integration on the conceptual / knowledge level was not provided. It is also unclear whether the developed case structure can be reasonably reused within our approach. The approach from Saxena[11] is based on a specific variant of CBR the authors call Dynamic Case-Based Reasoning (DCBR). "Dynamic" here means that DCBR is not only dynamic and by this able to learn - with respect to its case base through adding cases but also through statistic vectors that contain abstract knowledge condensed from groups of similar cases. The authors applied their approach for technical diagnosis and argue that it is applicable to fleet vehicles including vehicles in the aircraft domain. So-called analytical and descriptive knowledge is used as knowledge sources. This includes also knowledge described in an informal technical language, which is processed using techniques from natural language processing. The approach using the mentioned statistic vectors maybe interesting for specific diagnostic subtasks. Ferret and Glasgow[5] describe a hybrid approach that combines model-based reasoning (MBR) and CBR. It is based on a hierarchical decomposition of mechanical devices. Using a MBR only approach encounters some difficulties arising from imperfection of the model and the experts designing the model. These imperfections lead to incomplete or incorrect models and this leads to false diagnoses. The combination of MBR and CBR tries to counter the mentioned imperfections. Therefore the CBR part is used only after the MBR part to evaluate and criticize the results of the model-based diagnosis process and to help to decide which diagnoses should be selected. The CBR components allow the system to improve and overrule design and implementation mistakes of the model by evaluating potential diagnoses and finding additional diagnoses. The CBR component is a generic component that does not depend on specific devices or type of model used by the model-based diagnosis. This approach has been applied for technical diagnosis outside the aircraft domain. It uses explicit knowledge about structure and behavior of the technical systems and fault diagnosis experience in the form of cases. This approach provides a deep integration between MBR and CBR.

4 Case-based agents within the decision support system

This section describes the case-based agents in our decision support system in more detail. First the case-based tasks within the knowledge provision are presented. Furthermore our ideas for case-based adaptation and planning are described.

4.1 Case-based knowledge provision

The knowledge provision within SEASALT builds upon the Knowledge Line idea which modularizes knowledge analogously to how software is modularized in the Product Line approach within software engineering [[7]]. As opposed to focusing on software core components and variability, the modularization in SEASALT is designed towards individual topics that are represented within the respective knowledge domain.

For our MAS we need to elaborate more on the agents represented here. According to the SEASALT architecture each of these topics is governed by a respective Topic Agents, which can be any kind of information system or service including CBR systems, databases, web services, or other kinds of machine accessible knowledge stores. Additionally, the Topic Agents' CBR systems are extended with case factories, which take care of the individual agents' case maintenance.

For further structuring, all Topic Agents are administrated by a central Coordination Agent. From the individualized knowledge the Coordination Agent receives semi-structured natural language queries and analyses them, using a rule-based question handler and subsequently queries the respective Topic Agents. A query uses incremental reasoning that is using one agent's output as the next agent's input. In doing so the Coordination Agent's course of queries resembles the approach of a human amateur trying to answer a complex question. This reasoning process is formalized by using a graph-like structure called Knowledge Map. This map encodes formal representations of all Topic Agents and possible output/input connections, thus providing the comprehensive and general knowledge that is needed for carrying out the incremental reasoning process. Finally the Coordination Agent uses the query results and prefabricated templates to compose the information to be given to the user[[2]].

For the decision support system, the Knowledge Map is not yet realized. Figure 3 shows an exerpt of the Knowledge Map from an already implemented multi-agent system named docQuery.

```
<rdf:Description rdf:about="docquery-disease">
        <conf:agentspercoord>1</conf:agentspercoord>
        <conf:abstract>false</conf:abstract>
        <conf:topic>Disease</conf:topic>
        <conf:table>disease</conf:table>
        <conf:threshold>0.2</conf:threshold>
        <conf:information quality="100" costs="100" speed="100"  limit="100" />
        <conf:link use="prophylaxis">docquery-medicament</conf:link>
        <conf:link use="prophylaxis">docquery-associatedcondition</conf:link>
</rdf:Description>
```

Fig. 3. Exerpt from the docQuery Knowledge Map

The Knowledge Map contains information like the topic of the agent and the table of a database, from which the case base will be imported. In addition, a threshold for the minimum similarity for the retrieval can be defined and links to other case bases that represents the dependencies between them [[8]].

4.2 Case-based adaptation

Realizing adaptation in our decision support system faces several challenges. Not only the application domain is very complex, but also the knowledge formats of diagnoses and maintenance suggestions are very diverse and distributed over various knowledge sources. Diagnosis and maintenance knowledge is stored in excel documents, free text, or even in non digital formats like paper logbooks. While the diagnosis knowledge could be transformed into attribute value pairs with moderate effort, the transformation of maintenance knowledge requires very high effort or could be not possible. Maintenance procedures are described in free text format and a first analysis of the knowledge shows that it would be very difficult to use information extraction techniques to transform the free text into attribute value pairs with symbolic values.

Using rules for adaptation of diagnoses and maintenance suggestions would cause high effort, because of the complexity of aircraft systems, the dependencies between these systems, and the possibility of failure chaining. A detected fault might have several root causes in different systems and the adaptation of a diagnosis has to consider forward and backward chaining of faults. The required adaptation rule system has to be built from scratch and has to cover common adaptation situation as well as exceptional adaptation situations based on a high number of parameters like operational values, engineering data, and soft- and hardware configurations. In addition, the maintenance of the rule system causes high effort, too. Every time when adding, changing, or deleting a rule, all dependencies and affected rule chains have to be evaluated and adapted. Another problem is the adaptation of knowledge in free text format. Finding the affected text passage and change or partially or fully replace it, would be very difficult. For these reasons a rule-based adaptation is not adequate for our decision support system.

Case-based adaptation has several advantages compared to rule-based adaptation. Successful adaptations could be stored as cases and applied to adaptable situations using similarity measures. This way one adaptation case could be used for several situations. Furthermore abstract cases could be generated from sufficient similar cases to cover a greater spectrum of situations. In addition, adaptation cases could be linked together to support the user with more than on adaptation suggestion. The maintenance of the adaptation CBR system could be done by another Case Factory and be integrated into the Case Factory Organization to consider the dependencies between several adaptation CBR systems and the diagnosis CBR systems. This way changing the knowledge in a diagnosis CBR system will affect the adaptation knowledge in the dedicated adaptation CBR system to avoid inconsistencies and false adaptations. Another advantage is that feedback on an adaptation or new experiences could be easier integrated by creating a new case. Challenges of the case-based adaptation approach are the uncertainty based on the similarity assessments to find an adaptation case and the modeling of an adequate case structure.

Based on the considerations above, we decided to integrate agents with underlying CBR system for case-based adaption in our decision support system.

4.3 Case-based planning

The Case Factories generate several maintenance suggestions based on the evaluation of the knowledge containers and feedback of the users. In addition, the Case Factory Organization generates maintenance suggestions based on the dependencies between CBR systems. These suggestions are combined to a maintenance plan which is displayed to a knowledge engineer for confirmation. Using CBR to support the maintenance planning will reduce the effort of the plan generation. Depending on the maintenance strategies that are defined for the knowledge containers, several different maintenance actions are available. These maintenance actions recur in different combinations when a maintenance plan is generated. Maintenance plans, which are confirmed and successfully executed, could be stored partially or complete as cases. These cases could be used to reduce the effort of the plan generation, by using the retrieved plan as s basis for the new one. Using CBR for maintenance planning allows the usage of abstract plans to cover more planning situations. The generation of the additional maintenance actions will be done with the help of the Maintenance Map that contains dependencies between maintenance actions as well as constraints and preferences of these actions.

Considering explanations and the confirmation of a maintenance plan by a knowledge engineer, a plan state based approach for the case-based planning is intended. This way the single generation steps can be explained to the knowledge engineer to support the understanding of the maintenance plan by providing the operations and intermediate plans.

5 Summary and Outlook

This paper described the idea of a decision support system for diagnosis and maintenance in the domain of aircraft fault diagnosis. The system will be realized as a multi-agent system and has several case-based agents for knowledge provision, adaptation, and planning. We gave an overview over our approach and compared it to other approaches for diagnosis systems for technical diagnosis. We describe the types of case-based agents within our decision support system at a level of detail as possible till now.

Currently, the requirements for the decision support system are being defined and a detailed concept will be developed based on these requirements. In parallel, a first prototype of a multi-agent system will be implemented to test ideas of the concept and prepare the implementation of a demonstrator for the decision support system.

References

1. 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)
2. 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)
3. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), dr. Hut Verlag Mnchen
4. BMWI: Luftfahrtforschungsprogramms v (2013), `http://www.bmwi.de/BMWi/Redaktion/PDF/B/bekanntmachung-luftfahrtforschungsprogramm-5,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf`
5. Feret, M., Glasgow, J.: Combining case-based and model-based reasoning for the diagnosis of complex devices. Applied Intelligence 7, 57–78 (1997)
6. Jackson, T., Austin, J., Fletcher, M., Jessop, M.: Delivering a grid enabled distributed aircraft maintenance environment (dame). Tech. rep., University of York (2003)
7. van der Linden, F., Schmid, K., Rommes, E.: Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering. Springer Verlag Berlin (2007)
8. Reuss, P.: Concept and implementation of a Knowledge Line - retrieval strategies for modularized, homogeneous topic agents within a multi-agent-system (in German). Master's thesis, University of Hildesheim (2012)
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. Richter, M.M.: Handbuch der knstlichen Intelligenz, chap. Fallbasiertes Schlieen, pp. 407–430. Oldenbourg Wissenschaftsverlag (2003)
11. Saxena, A., Wu, B., Vachtsevanos, G.: Integrated diagnosis and prognosis architecture for fleet vehicles using dynamic case-based reasoning. In: Autotestcon 2005 (2005)