

Towards Using Case-Based Explanations as a Knowledge Foundation

Jakob Michael Schoenborn¹
schoenb@uni-hildesheim.de

Klaus-Dieter Althoff¹
klaus-dieter.althoff@dfki.uni-kl.de

Institute of Computer Science, Intelligent Information Systems Lab
University of Hildesheim, Samelsonplatz 1, 31141 Hildesheim, Germany
<https://www.uni-hildesheim.de>

Abstract. *Due to the GDPR, the need of explanation-aware systems is rising. To include a component which can explain the decisions made by a given system is often not feasible or requires at least a lot of effort. On top, the user acceptance of decisions made by artificial intelligence agents is more sceptical than welcoming. Therefore, plausible explanations have to be generated for each decision made so that the user can develop trust in the decision making process. This is important for knowledge management as well, since knowledge needs also to be trusted - otherwise the knowledge would not be reused and is therefore without value. This should be prevented by building an explanation-aware system. To guarantee the improvement of value, the incoming input from a user needs to be sanitized before stored in the case-base. The process of how knowledge can be extracted and then furthermore be used and trusted will be further investigated. The future aim is to build up a distributed case-based reasoning system which explains its own building process so that a given knowledge engineer can guide the way in which the system is building up and adjust it to his needs.*

Keywords: Knowledge Management and Maintenance · Case-Based Reasoning · Trusted Knowledge · Explanations

1 Introduction and motivation

Whenever a person encounters a new situation, this person tries to reuse knowledge from past situations that encountered before. This is done by trying to remember every case, which inherits a high similarity and therefore could be applicable. It might be required that a few adaptations will have to be made to reuse the existing knowledge. The person then evaluates if the past solution can also be applied to this new problem [1].

Everyone is going through this process multiple times per day, mostly unconsciously. Judges try to remember similar cases to judge the current, new case. Teachers try to express similar explanations to a current problem of a student to make their teaching material more clear. Administrators of a company try to remember the application procedure of two candidates to compare them. The memory is managing experiences and infers knowledge out of the stored experiences. During daily decisions a person is taking, erroneous remembering can occur but does not lead to vast disasters. But “*the imperfect nature of memory*” leads to multiple false remembering, as highlighted by Gonsalves [7] among many others. But when a decision is made only by knowledge used from recently memorized cases and has an greater influence on other people, this is where explanations should support the given memory, i. e. by referring to evidence. Binns et al researched the perception of justice in algorithmic decisions while the “...aim was to find explanation styles which could plausibly meet or exceed the regulatory requirements regarding transparency of automated decisions, in particular the requirement that organisations provide ‘meaningful information about the logic involved’ in an automated decision“ [5, p. 4.] which is an important step towards the GDPR 15.1.(h).

Explanations build up on knowledge. One might argue that explanations would suffer from the same problematic as described above. The proposed approach is to build up a system from scratch to prevent arguing on basis of an erroneous knowledge foundation. Otherwise, given there is a case-based reasoning system which shall be modified to express an explanation: It has to be determined, in which way the explanation will be provided. There are at least three different types of explanations: textual (templates, reports, ...), semantic relations (cause-effect, is-part-of, ...), and graphical representations (plots, graphs, ...). The transformation from the given state to an actual explanation has to be revised by the knowledge engineer inhibiting the technical knowledge and an expert inhibiting the domain knowledge. When revising the existing knowledge, the view should not be only limited on cases (as evaluated by Leake and Wilson [9] and suggested by Smyth [13]). Instead, additional components which also take effect in generating an explanation (i. e. adaptation rules and the used similarity measure) should be manipulated and then the different output of explanations has to be observed. How does the explanation change and is it still a valid explanation given the current situation? While building up the case base, it is important to keep in mind where the cases (and their inherited knowledge) were coming from. The long-term goal of the idea aims to be valid for any given domain, since there are multiple new domains showing up, trying to incorporate the usage of case-based reasoning (i. e. phishing detection [2], real-time strategy games [14], and mood detection [3] to name a few). This seems to be a promising connection point to build up on explanations parallel to establish the CBR system.

2 Application domain: aviation

In the following, the aviation domain is the domain to be considered and the system is supposed to support an intern knowledge engineer with the knowledge management when creating a maintenance routine for a new type of air plane. This domain is chosen, because it is a very technical domain with a lot of structured information in form of attribute-value pairs, taxonomies and ontologies. The complexity of the aviation domain comes with “*the hundreds of components* (e.g. Ventilation Control), *which consists of dozens of systems* (e.g. Air Conditioning), *which contains dozen of individual parts*(e.g. Cabin Air Filter)” [10] called Line Replaceable Units. Using appropriate vocabulary and similarity measures, these information can be stored as cases and thus be used by a CBR system. To generate an explanation there needs to be at least *some* knowledge. Rule-based and model-based knowledge can be retrieved from the given LRUs as a baseline for an explanation-aware system. Using this way, physical impossible combinations of components can be excluded and a proper explanation generated on why they are not possible. Having a first set of core functionality, the more challenging part has to be considered: When should which knowledge be added to the system and especially: Why? The motivation in general to split up the development of the explanation-aware CBR system can be viewed similar to the motivation of software product line engineering: Reduction of development costs, enhancement of quality, and customers get products adapted to their needs and wishes [8]. A product line can here be the introduction of a new air plane type to the air plane fleet. Since there cannot be any practical experience when building a new air plane type, it is crucial for a cost-effective introduction to exclude as many failure risks as possible. This is the entry point for an explanation-aware CBR system.

As stated above, the core functionality and knowledge containers need to be expanded so that valid and trustworthy explanations can be offered. Additional sources of knowledge are free texts of aircraft incidents and reports written by maintenance technicians or other staff members. To retrieve the knowledge out of free texts, the framework FEATURE-TAK has been developed - a **F**ramework for **E**xtraction, **A**nalysis, and **T**ransformation of **U**nstructured - **T**extual **A**ircraft **K**nowledge which combines several methods from natural language processing and CBR [10]. This framework consists of five layers to store domain specific informations like abbreviations and technical phrases which can be accessed by other components (i. e. software agents). The workflow is processed by multiple, distributed agents and coordinated by a central supervisor agent. To support the knowledge engineer, eight tasks are completed automatically ranging from phrase and keyword extraction, identifying synonyms and hypernyms to a similarity assessment and sensitivity analysis¹. The knowledge engineer will then be offered a suggestion to add the retrieved knowledge, but without an explanation why the framework has come to this decision. Either way, the knowledge engineer has to do a consistency check and stored feedback on the process instance. This

¹ for a detailed description of the tasks, refer to [10]

could be supported by a process on evaluating the current state of knowledge and whether this retrieved piece of knowledge has actually a positive effect on the system if stored in the case base.

3 Building an explanation-aware CBR system

When building up the knowledge foundation from scratch and the goal is to build up an case-based explanation system which can also explain its own building process, the system needs to gather initial knowledge to express at least a basic explanation. One possibility to determine a domain-accurate case structure is using reports of experts (here: maintenance engineers free text reports) with statements on their domain and comparing these to extracted text results - filtering out keywords to use them as case attributes. Another possibility is to retrieve basic knowledge by crawling through networking/communities, FAQs or Wikis. Even if there are networking communities where it might be possible to deploy an automated web-crawler and build up a knowledge base, legal concerns has to be respected as the GDPR restricts the usage of data gained by mentioned web-crawlers without the creators permission in Europe [6]. Since automatically generated information is not guaranteed to be valid, a knowledge engineer has to add the suggested knowledge to the case base carefully. Whenever new knowledge will be suggested to the knowledge engineer, the system should try to generate an explanation on why this piece of knowledge should be added and, especially, on basis of which evidence this piece of knowledge has been retrieved. Then the evidence should be tied to the saved case as attribute of the case. Because the evidence might become outdated, maintenance processes should be called on a regular basis. Depending on the applied domain, evidence might become faster outdated than in other domains (e. g. laws usually do not change rapidly, while a person learning to play chess has a rapid changing state of knowledge). The idea on outdated knowledge and actual mechanisms to counter this issue was originally presented by T. Roth-Berghofer. His SIAM methodology extends the CBR cycle by adding two more steps, review and restore, which are triggered after the retain step [12] (here: after the knowledge engineer includes and accepted case with its explanation). He distinguishes between an application phase (retrieve, reuse, revise) and a maintenance phase (retain, review, restore). This is important for the maintenance, because in the original CBR cycle was no way to maintain the knowledge when the environment changes [4, 12]. This is especially important for explanations when they are building up on the current knowledge and it is crucial to be able to review the current state of knowledge (as the added review-step does).

Using explanations to improve the communication between the system and the maintenance engineer, certain use cases on when an explanation is actually needed have to be determined. Usually, in the maintenance area, explanations are further informations on why a given fault has occurred - they are error-based.

The first part of the explanation is actually given by the error message itself. The addition of using CBR would be an explanation on what this error actually “means” in a semantic way and additionally to provide a possible solution. Therefore, an error-based explanation could be structured in the following way:

$$\text{error-based explanation} = \begin{cases} \text{error cause} \\ \text{semantical consequence} \\ \text{possible solution} \end{cases}$$

Nevertheless, to find the root cause for an occurred fault is difficult to reproduce since it can be caused by a single part, the interaction between parts, or even the communication infrastructure between these parts [11]. Thus, the knowledge engineer has to manually replace the LRU which is accused to be the most probable reason for a given fault. Currently, this is based on the best-guess of the technician who might need guidance through using error-based explanation.

4 Conclusion

This work presented a brief overview on the establishment of explanations as a support to knowledge engineers in the aviation domain. Due to the high risk nature of the aviation domain where a false-positive has a critical impact, the bottom-up approach has been chosen. Thus, existing knowledge has to be reviewed and will only be added to knowledge containers, if it has been validated and has been connected with evidence. The approach was to build up error-based explanations for a faster identification of fault-evoking LRUs and thus to reduce the overall maintenance time of an air plane. Since there are very few similar approaches, this preliminary approach concludes with multiple questions unanswered, e.g. how exactly a proper case structure should be structured to incorporate explanations in the maintenance process, when an explanation is actually desired, and how to offer unquestionable explanations.

References

1. Aamodt, A. and Plaza, E.: *Case-based reasoning; Foundational issues, methodological variations, and system approaches*. AI Communications. 1994.
2. Abutair, H., Belghith, A.: *Using Case-Based Reasoning for Phishing Detection*. Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia, Oct 2017.
3. Adebayo, J., Adekoya, A., Ekwonna, C.: *Temperament and Mood Detection Using Case-Based Reasoning*. International Journal of Intelligent Systems and Applications. 6. DOI: 10.5815/ijisa.2014.03.05. 2014.
4. Althoff, K.-D., Wess, S., Traphöner, R.: *INRECA - A Seamless Integration of Induction and Case-Based Reasoning for Decision Support Tasks*, 1995.
5. Binns, R. et al.: *'It's Reducing a Human Being to a Percentage': Perceptions of Justice in Algorithmic Decisions*. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 377, 14 pages. DOI: <https://doi.org/10.1145/3173574.3173951>. 2018.

6. Goodman, B., Flaxman, S.: *EU regulations on algorithmic decision-making and a "right to explanation"*. AI Magazine. 38. 10.1609/aimag.v38i3.2741. 2016.
7. Gonsalves, B., Reber, P., Gittelman, D.: *Neural Evidence That Vivid Imagining Can Lead to False Remembering*. Sage Journals. 2004.
8. Pohl, K., Böckle, G., and van der Linden, F.: *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag, Berlin, Heidelberg. 2005.
9. Leake, D., Wilson, M.: *How Many Cases Do You Need? Assessing and Predicting Case-Based Coverage*. In: Ram A., Wiratunga N. (eds) Case-Based Reasoning Research and Development. ICCBR 2011. Lecture Notes in Computer Science, vol 6880. Springer, Berlin, Heidelberg, 2011.
10. Reuss, P., et al.: *FEATURE-TAK - Framework for Extraction, Analysis, and Transformation of Unstructured Textual Aircraft Knowledge*. In: Case-Based Reasoning Research and Development, Springer International Publishing, Pages 327–341, 2016.
11. Reuss, P., et al.: *Knowledge Engineering for Decision Support on Diagnosis and Maintenance in the Aircraft Domain*. In: Advances in Intelligent Systems and Computing, p. 173-196, 2017.
12. Roth-Berghofer, T.: *Knowledge maintenance of case-based reasoning systems: the SIAM methodology*. IOS Press. 2003.
13. Smyth, B.: *Case-base maintenance*. In: Pasqual del Pobil A., Mira J., Ali M. (eds) Tasks and Methods in Applied Artificial Intelligence. IEA/AIE 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1416. Springer, Berlin, Heidelberg. 1998.
14. Stalnaker, R.: *Knowledge, Belief and Counterfactual Reasoning in Games*. In: Arló-Costa H., Hendricks V., van Benthem J. (eds) Readings in Formal Epistemology. Springer Graduate Texts in Philosophy, vol 1. Springer, Cham, 2016.

AGREEMENT TO GRANT A NON-EXCLUSIVE PUBLICATION PERMISSION (V 20170316)
FOR THE

Original work with title: Towards Using Case-Based
Explanations as a Knowledge Foundation

Authors: Jakob Michael Schönborn,
Prof. Dr. Klaus-Dieter Althoff

Prepared for the Workshop/Conference/Book: _____
LWDA 2018 / FGWM

I/we (the Authors) hereby grant a non-exclusive, irrevocable and non-time limited publication permission over the above-named original work (the Material) to the following persons (hereinafter the editors/publishers):

Pascal Reuss, Joachim Baumeister, and
any editor/publisher for LWDA 2018.

In order to include in any and all forms, compile, publish, print, distribute and spread such a work with the unique aim of making it available as open-access for everybody.

The authors of the Material preserve all their copyrights and publication rights.

In the event that any elements used in the Material contain the work of third-party individuals, I/we understand that it is my responsibility to secure any necessary permissions and/or licences and will provide it in writing to the editor/publishers. If the copyright holder requires a citation to a copyrighted work, I/we have obtained the correct wording and have included it in the designated space in the text.

I/we hereby release and discharge the editors/publishers and other publication sponsors and organizers from any all liability arising out of the inclusion of my Material in the publication, or in connection with the performance of any of the activities described in this document as permitted herein. This includes, but is not limited to, my right of privacy or publicity, copyright, patent rights, trade secret rights, moral rights or trademark rights.

I have used third-party material: Yes () No

If the previous answer is yes, I/we have the necessary permission to use the third-party material: Yes () No () Not applicable

J. Schönborn
(Signature of one author of the Material representing all its co-authors)

Print Name: Jakob Michael Schönborn

Location/Date: Hildesheim, 29.7.2018

Please return this form to:
<Editor Name, Address details>