# Towards Continuous Knowledge Representations in Episodic and Collaborative Decision Making

Joachim Baumeister[1,3], Albrecht Striffler[1], Marc Brandt[2] and Michael Neumann[2]

[1] denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg, Germany
{firstname.lastname}@denkbares.com
[2] The Federal Environment Agency (Umweltbundesamt), Section IV 2.3 Chemicals
Wörlitzer Platz 1, 06844 Dessau-Roßlau, Germany
[3] University of Würzburg, Institute of Computer Science
Am Hubland, 97076 Würzburg, Germany

**Abstract.** With the success of knowledge-based approaches in decision support systems new requirements arise in practice. That way, users demand not only for the collaborative development of such systems, but also for the collaborative and episodic use in decision processes. Moreover, in complex decision domains multiple knowledge representations are available that need to be jointly processed. In this paper we introduce a novel approach and a system implementation that aims to meet these requirements.

## 1 Introduction

In the past, decision support systems based on knowledge bases emphasized the explicit representation of decision knowledge for its automated application in the target scenario. Typically, those systems are used monolithically by one user or automated by a machine. Examples are for instance the medical consultation system SonoConsult [12], the medical therapeutic system SmartCare [6], and TIGER [8] for the monitoring of gas turbines. With the success of those systems new requirements arise to adapt into new environments. Advanced requirements are as follows:

- *Collaborative use:* More than one person is working on the same decision process at the same time.
- *Episodic use:* The actual decision process is not a one-step question-answer interview, but needs (sometimes sporadically) input over time, i.e., a *decision episode*.
- *Mixed representation:* Systems are build from knowledge bases that do not use a single knowledge representation (e.g., rules) but a combination, for instance rules with models and ontologies.

The requirements stated above call for extensions of todays systems in the following manner:

- A systematic extension of systems that support the collaborative and the episodic decision making. Here, especially an approach of representing the provenance of decisions is required.

– A continuous knowledge representation to support heterogenous representations for decision making and its episodic application. Here, the already introduced *knowledge formalization continuum* [2] needs to be reconsidered in the light of its use in decision making.

In this paper, we try to shed more light into fulfilling the requirements mentioned above. The formalization and use of the knowledge formalization continuum is introduced in Section 2. In Section 3 we discuss a systematic approach for episodic decision making in collaborative use. A case study in Section 4 exemplifies the successful application of the described approach. The overall ideas are summarized and concluded in Section 5.

## 2 Continuous Knowledge Representation and Application

One main challenge in complex decision making is finding the appropriate scope of the knowledge base: Complex domains require a large number of aspects to be considered. Thus, a 'complete' knowledge base needs to include many aspects, to be later useful in practice. Most of the times however, not all aspects can be included in the knowledge base:

– *Uncertain domain knowledge:* Parts of the domain are not well-understood in a technical sense. Here, decisions in practice are often based more on past experience, evidence, and intuition than on strict domain laws and rules.
– *Bloated domain knowledge:* For some parts of the domain, the explicit representation of the knowledge would be too time-consuming and complex. For instance, much background knowledge needs to be included, that is required for proper decision making. Here, the expected cost-benefit ratio is low, e.g., because many parts will be rarely used in real-world decisions[1].
– *Restless domain knowledge:* Especially in technical domains, some parts of the domain knowledge are frequently changing due to technological changes. The explicit representation of these parts would require frequent maintenance. Here, also the cost-benefit of the maintenance vs. the utility of the knowledge needs to evaluated.

In this section we introduce an approach that allows for the combined representation and use of knowledge at a varying formalization granularity, i.e., the *knowledge formalization continuum*. The main idea of the knowledge formalization continuum is to use varying knowledge representations for one knowledge base and to select the best-fitting representation for each partition. Besides the representation of different knowledge representations, the approach also considers the mixed application of and reasoning with knowledge at different formalization levels.

[1] Costs for developing/maintaining the knowledge vs. the benefit/ frequency of using the single parts in practice

### 2.1 The Knowledge Formalization Continuum

In general, the *knowledge formalization continuum* is a conceptual metaphor extending the knowledge engineering model for a domain specialist. The metaphor emphasizes that entities of a knowledge base can have different facets ranging from very informal representations (such as text and images) to very explicit representations (such as logic formulae), see Figure 1. Here, it is not necessary to commit to a specific knowledge representation at the beginning of a development project. Rather, it supports concentrating on the actual knowledge by providing a flexible understanding of the knowledge formalization process. Specific domain knowledge can be represented in different ways, where adjacent representations are similar to each other, e.g., tabular data and cases. More extreme representations are much more distinct, e.g., text vs. logic rules. It is important to note that the knowledge formalization continuum is neither a physical model nor a methodology for developing knowledge bases. Rather, the concept should help domain specialists to see even plain data, such as text and multimedia, as first-class knowledge that can be transformed by gradual transitions to more formal representations when required. On the one hand, data given by textual documents denote one of the lowest instances of formalization. On the other hand, functional models store knowledge at a very formal level.

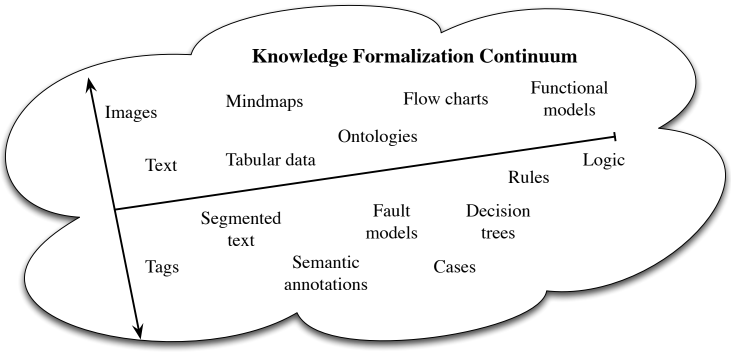


**Fig. 1.** An idealistic view of the knowledge formalization continuum.

When working with different representations of knowledge one has to keep in mind, that every granularity of formalization has its advantages and disadvantages. On the informal side, textual knowledge can be easily acquired and it is often already available. No prior knowledge with respect to tools or knowledge representation is necessary. However, (automated) reasoning using textual knowledge is hardly possible. The knowledge can only be used/retrieved through string-based searching methods. The formal side proposes rules or models as knowledge representation; here automated reasoning is effective but the acquisition of such knowledge is typically complex and time-consuming. Further, the knowledge engineer needs to "model" the knowledge in a much more precise manner.

The knowledge formalization continuum embraces the fact that knowledge is usually represented at varying levels of formality. A system supporting the knowledge formalization continuum should be able to store and work with different representations, and it should support transitions between the representations where its cost-benefit ratio is (in the best case) optimal.

In typical projects, prior knowledge of the domain is already at hand, often in the form of text documents, spreadsheets, flow charts, and databases. These documents build the foundational reference of the classic knowledge engineering process, where a knowledge engineer models domain knowledge based on these documents. The actual utility and applicability of knowledge usually depends on a particular instance. The knowledge formalization continuum does not postulate the transformation of the entire collection into a knowledge base at a specific degree but the performance of transitions on *parts* of the collection when it is possible *and* appropriate. This takes into account the fact that sometimes not all parts of a domain can be formalized at a specific level or that the formalization of the whole domain knowledge would be too complex, considering costs and risks.

## 2.2 Reasoning in the Knowlegde Formalization Continuum

When using different types of knowledge representations the most important question is how to connect these elements when used during the reasoning process.

**Pragmatic Reasoning** As a pragmatic approach to be used in decision support systems, we propose to define a taxonomy of decisions and connect entities of knowledge (*knowledge elements*) with decisions of the decision taxonomy. See Figure 2.2 for an exampled depiction. Here, the knowledge base contains rules, workflow models, and

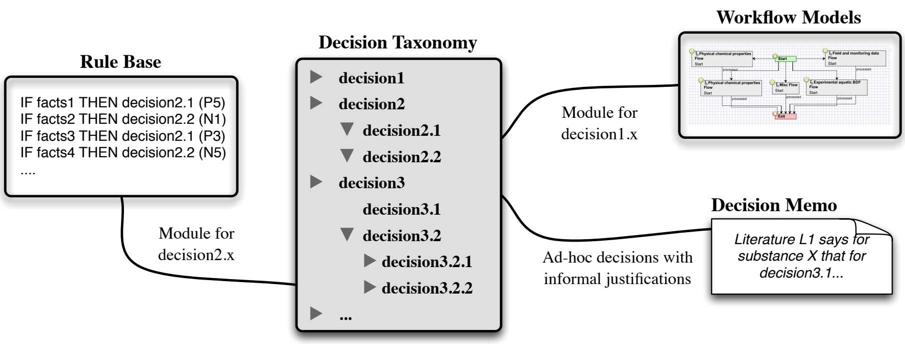


**Fig. 2.** An example for connecting different knowledge elements with a decision taxonomy.

textual decision memos. All elements reference the same collection of decisions and thus can jointly infer decisions. When a knowledge element is activated during decision making–the knowledge element fires–then the corresponding decision element is established and presented as derived decision.

Please note, that more formal approaches like RIF [16] do use a comparable connection, i.e., decisions are formalized as concepts/instances and rules are defined to derive the existence of the concept/instance.

**Decision Making using Scoring Weights** With the simple approach sketched above, decisions can be only taken categorically. For a more leveled approach, we propose to introduce scores as weights for decisions. Scores are a well-understood weighting scheme in knowledge engineering [11, 7] and has a simple reasoning semantics: Each decision has an account which stores the scoring weights given to the decision by knowledge elements during the reasoning process. When a knowledge element "fires", then the corresponding score is added to the account of the particular decision. Scoring weights included in the account are aggregated in a predefined manner. A decision element is established and shown as derived decision, when the aggregated scoring weight exceeds a given threshold.

*Example:* Often a reduced set of score weights S = {N3, N2, N1, 0, P1, P2, P3} is sufficient for developing large knowledge bases. Given the weight categories a developer can select from seven weights N1 (weakly negative) to N3 (excluded) for negative scoring and seven weights P1 (weakly positive) to P3 (clearly established) for positive scoring. The weight 0 represents an unclear state. The score weights of a decision account are aggregated as follows: The sum of two equal weights results in the next higher category, e.g., P2 + P2 = P3. Positive and negative weights are aggregated, so that two equal score weights nullify each other, e.g., P2 + N2 = 0. A decision is established (confirmed), if the aggregation of the collected scoring weights exceeds the category P3.

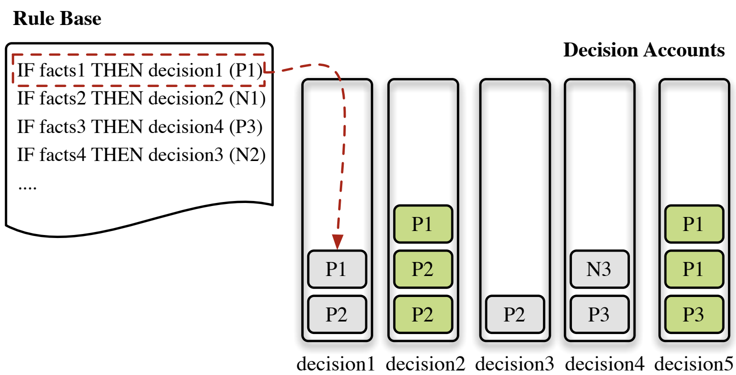


**Fig. 3.** Exemplary score accounts for five decisions.

In Figure 3 the accounts of five decisions and an excerpt of a rule base are shown. One rule fires and adds the weight P1 to the account of `decision1`. We see that `decision2` and `decision5` are established, since the aggregation of their collected scoring weights exceeds the weight P3. In contrast, `decision4` is not established because of the negative weight N3.

## 3 Episodic and Collaborative Decision Making

Complex decisions often are not made by taking one step, but are typically divided into a number of sub-decisions. Each of them may need further research and collaborative interaction for clarifying details. Collaboration is necessary when a complex decision can only be made by joining experts from different domains into the decision process. These requirements can be fulfilled by specific extensions of a decision support system:

1. Contemporary access to the data and decisions.
2. Episodic collaboration during decision making.
3. Provenance of data and decisions.

### 3.1 Contemporary Access

Authorized persons need to be able to access the system at the same time. They should be able to work with the system in order to make decisions or to retrieve already taken decisions. Contemporary access can be provided by a web-based implementation of the system, as for example implemented by semantic wiki systems [13]. Further examples are collaborative ontology development environments such as WebProtégé [9, 14].

In such a distributed setting we need to consider concepts like rights management for access control, revision management of old versions of the knowledge, and conflict management of simultaneous edits.

### 3.2 Episodic Collaboration

Authorized persons should be able to enter data for making a particular decision. The data entry needs not to be made at one time but can be partitioned over multiple sessions, i.e., decision episodes. Also, different users can enter data used for the same decision.

### 3.3 Provenance of Data and Decisions

When more than one person contributes to a complex decision making process and when the process is partitioned into episodes, then the process and reasoning should be traceable and understandable by the users. This implies the documentation of the decisions including their history but also the provenance of the data used for making the decision (see below). Therefore, the system needs to provide versioning of the decisions made including a documentation by the respective users. When representing the history and documentation of decisions by an ontology, then known approaches can be applied, for instance [10, 4].

Provenance of data and decisions is needed in collaborative and episodic environments. Here, the following questions need to be clearly answered:

- At which time was a particular data element entered?
- Who entered the data?
- Which knowledge elements are responsible for a particular decision?
- What is the history of a particular data and decision?
- Which persons contributed to the process of a particular decision?

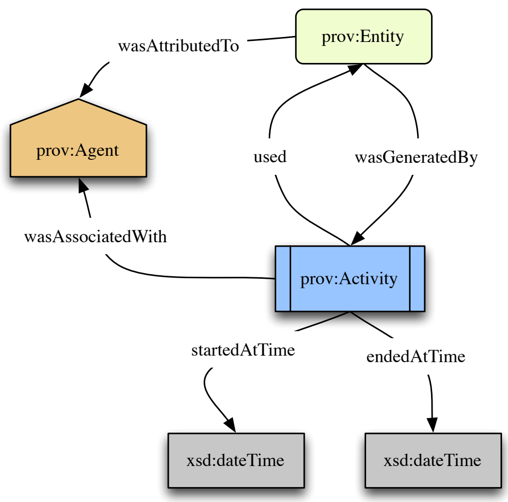


**Fig. 4.** Simple version of the PROV ontology.

We propose the application of the PROV ontology [15] to knowledge elements and the entities of the decision process. That way, an extensible and standardized ontology is used to represent the use and origin of decisions. In Figure 4 the three Starting Point classes and properties of the PROV ontology are depicted. Here, an `prov:Agent` is responsible for taking an `prov:Activity`. The `prov:Activity` generates an `prov:Entity`, but instances of `prov:Entity` can be also used in (other) instances of `prov:Activity`. An `prov:Entity` is a general representation for a thing, being physical, digital, conceptual, or any other kind of interpretation. We can see that answers to the questions stated above can easily be represented using the simple version of the PROV ontology, when people involved in the decision making process are represented as `prov:Agent` instances, entered data and the decisions themselves are represented as `prov:Entity` instances, and the data entry and decision episodes are represented as `prov:Activities`.

# 4 Case Study: KnowSEC – A System for Managing Chemical Substances of Ecological Concern

In this section we describe the KnowSEC project and its corresponding tool. KnowSEC stands for "Managing Knowledge of Substances of Ecological Concern" and it is used to support substance-related work and workflows within a unit of the Federal Environment Agency (Umweltbundesamt). More precisely, the tool supports decisions by a number of knowledge-based modules. In the context of the KnowSEC project only substances under REACH[2] are considered. For the implementation of the KnowSEC project the semantic wiki KnowWE [3] was extended by KnowSEC-specific plugins. KnowWE is a full-featured tool environment for the development of diagnostic knowledge bases and RDF(S) ontologies. It provides plugins for automatically testing and debugging knowledge bases including continuous integration. For a recent overview we refer to [1].

Many of the requirements stated in the introduction of this paper apply to the KnowSEC project: The group is divided into sub-groups; each sub-group collaboratively works on a number of substances. For each substance under consideration a number of complex decisions need to be made concerning the safety and regulation of the substance. Decision making on a substance sometimes can take a couple of months or even years, therefore support for episodic decision making is required.

## 4.1 Substances as Wiki Instances

Since the single substances are the primary target of decision making, every substance under consideration is represented by a distinct (semantic) wiki article. The article stores relevant information of the substance such as chemical end-points, relevant literature, and comments of group members. The information is entered by group members using (user-friendly) editors. In the background the information is silently translated into an ontology representation for automated reuse and processing. That way, any information (e.g., alternative identifiers, end-points, paragraphs, comments) is represented as an RDF triple. Consequently, the visualization of the latest changes and specific overviews are easily defined by SPARQL queries [17]. The article of the imaginary substance "Kryptonite" is depicted in Figure 5. The article is maintained for demonstration purposes and reflects by no means any real work of the agency.

At the right of the article all decision work on the substance "Kryptonite" is displayed giving a summary of the currently taken (sub-)decisions, the comments by group members, and a fact sheet showing the identifiers of the substance.

At the time of writing, KnowSEC stores more than 11,000 substances as separate wiki articles including a number of critical chemical characteristics. A small part of these substances are currently under decision making.

[2] REACH stands for the European Community Regulation on chemicals and their safe use (EC 1907/2006). The regulation handles the registration, evaluation, authorization, and restriction of chemical substances.

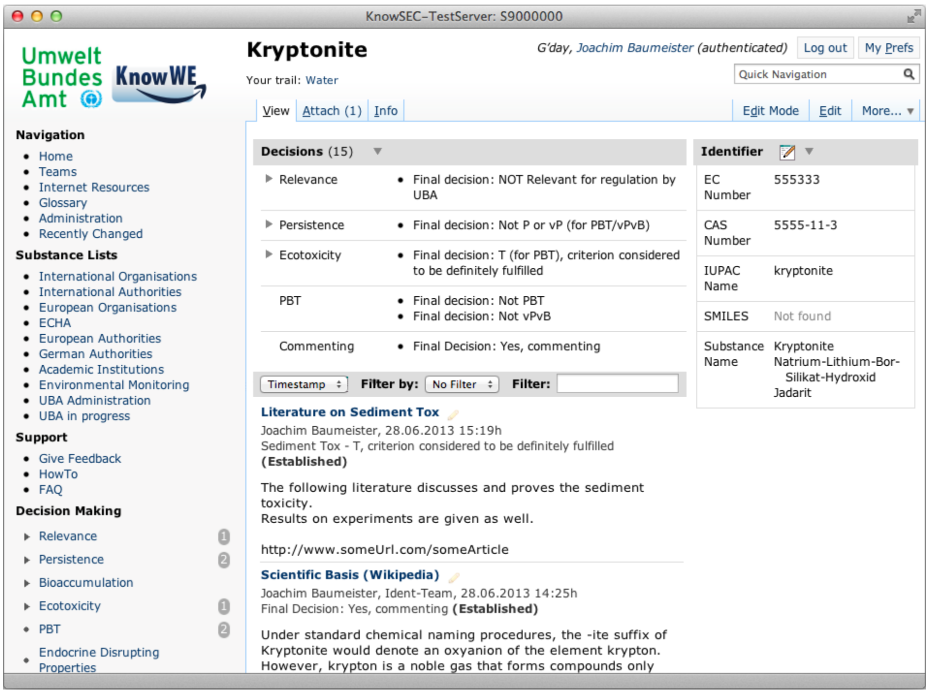


**Fig. 5.** The article describing the imaginary substance "Kryptonite".

## 4.2 Continuous and Collaborative Decision Making

When displaying a substance article in KnowSEC, the left menu of the wiki is extended by a decision making bar; see Figure 5. Here, all decision aspects are listed that are relevant for the working group. When clicking on a decision aspect, the sub-aspects of the selected aspect are shown. By selecting one of these (sub-)aspects the user can initiate a decision form, where specific questions of the aspect are asked and decisions are proposed automatically. These interactive decision forms are generated by explicit knowledge represented by scoring rules or DiaFlux models [5]. Any data entry and taken decision is recorded by KnowSEC including the time and user. An explanation component shows the justifications of taken decisions by visualizing the supporting data and the acting users of the data. For the explanation the PROV ontology—as described in Section 3.3—is applied. Users, i.e., team members, are instances of `prov:Agent` and entered data and (decision) memos are instances of `prov:Entity`. The creation or edit of a (decision) memo and an interactive decision form are represented as `prov:Activity` instances including the corresponding edit times. A simplified depiction of this application is shown in Figure 6; the prefix `dss` (decision support system) stands for the KnowSEC ontology namespace.

**Explicit Knowledge and the Taxonomy of Decisions** From the technical point of view, the explicit part of the knowledge base is partitioned into modules, that are con-

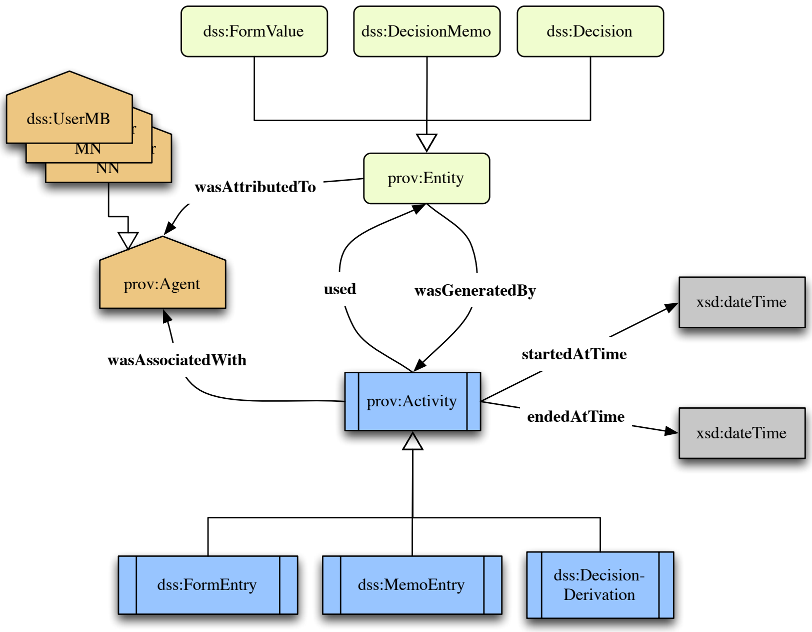


**Fig. 6.** Simplified version of the applied PROV interpretation for tracing the provenance of decisions.

nected by a taxonomy of decision instances. Since the taxonomy is represented as an RDF ontology, it is strongly connected with the ontology of the article information (see paragraph above). The formal versions of the aspects are implemented by knowledge base modules and connected by the taxonomy of decision instances. Some modules are using decision trees, other modules use scoring rules, also RDF ontologies are used.

**Decision Memos** For some aspects and decisions, respectively, no explicit knowledge base is available. When the user wants to document a decision without using an explicit knowledge base, he/she is able to create a *decision memo*. A decision memo is, entered by an authorized user and consists of free text, some meta-data (e.g., tags, time, etc.), and an explicit decision with a scoring weight. The decision memos are attached to the article of the corresponding substance. The included decision is used in the overall reasoning process. A decision memo is an implementation of an implicit reasoning element of the knowledge formalization continuum. An example of a decision memo can be the note of a group member that a particular aspect was proven by a specific experiment giving the reason for deriving a specific (sub-)decision. For instance, see the decision memos about the persistence of the substance "Kryptonite" being created in Figure 7. Decision memos are automatically attached to the articles of the corresponding substances.


**Fig. 7.** A decision memo created for the exemplary substance Kryptonite.

**Size and Current Status** Currently, KnowSEC provides explicit decision modules for supporting the assessment of the relevance, the persistence in the environment, the bioaccumulation potential, and the toxicity of a given substance. The taxonomy of decisions however, contains 15 different main decisions on substances having a larger number of sub-decisions.

The static part of the knowledge base currently consists of 282 questions (user inputs to characterize the investigated substance) grouped by 92 questionnaires, 558 decisions (assessments of the investigated substance), and about 1,000 rules to derive the decisions. The rules are automatically generated from entered decision tables that allow for an intuitive and maintainable knowledge development process. Two knowledge engineers are supporting a team of domain specialists, that partly define the knowledge base themselves, partly giving domain knowledge to the knowledge engineers.

At the beginning of the project a couple of internal data bases were integrated into KnowSEC as (decision) memos. Currently, the system contains more than 27,000 (decision) memos for the 11,000 substances. In the form dialog more than 51,000 questions were answered; partially automatically by imports of internal data bases. Both, decision memos and the explicit rule base derived more than 42,000 module decisions.

## 5 Conclusions

Advanced decision support systems allow for the distributed and episodic handling of complex decision problems. They implement large knowledge spaces by mixing different knowledge representations with informal decision justifications. In this paper, we introduced a novel approach for building decision making systems, that support collaborative and episodic decision making. Furthermore, we motivated how the application of the knowledge formalization continuum helps to create knowledge in complex domains. The practical applicability and relevance of the presented approach was demonstrated by the discussion of an installed decision support system for the assessment of chemical substances. When decisions are derived in a collaborative and episodic setting, the transparency of found decisions is of prime importance. Thus, we are currently working on an elaborated explanation approach based on the provenance ontology PROV, that is capable to provide intuitive and effective ad-hoc explanations even for end users.

## References


1. Baumeister, J., Reutelshoefer, J., Belli, V., Striffler, A., Hatko, R., Friedrich, M.: KnowWE - a wiki for knowledge base development. In: The 8th Workshop on Knowledge Engineering and Software Engineering (KESE2012). http://ceur-ws.org/Vol-949/kese8-05_04.pdf (2012)
2. Baumeister, J., Reutelshoefer, J., Puppe, F.: Engineering intelligent systems on the knowledge formalization continuum. International Journal of Applied Mathematics and Computer Science (AMCS) 21(1) (2011), http://ki.informatik.uni-wuerzburg.de/papers/baumeister/2011/2011-Baumeister-KFC-AMCS.pdf
3. Baumeister, J., Reutelshoefer, J., Puppe, F.: KnowWE: A semantic wiki for knowledge engineering. Applied Intelligence 35(3), 323–344 (2011), http://dx.doi.org/10.1007/s10489-010-0224-5
4. Franconi, E., Meyer, T., Varzinczak, I.: Semantic diff as the basis for knowledge base versioning. In: 13th International Workshop on Non-Monotonic Reasoning (NMR). pp. 7–14 (2010)
5. Hatko, R., Baumeister, J., Belli, V., Puppe, F.: Diaflux: A graphical language for computer-interpretable guidelines. In: Riaño, D., ten Teije, A., Miksch, S. (eds.) Knowledge Representation for Health-Care, Lecture Notes in Computer Science, vol. 6924, pp. 94–107. Springer, Berlin / Heidelberg (2012)
6. Mersmann, S., Dojat, M.: SmartCare$^{tm}$ - automated clinical guidelines in critical care. In: ECAI'04/PAIS'04: Proceedings of the 16th European Conference on Artificial Intelligence, including Prestigious Applications of Intelligent Systems. pp. 745–749. IOS Press, Valencia, Spain (2004)
7. Miller, R.A., Pople, H.E., Myers, J.: INTERNIST-1, an Experimental Computer-Based Diagnostic Consultant for General Internal Medicine. New England Journal of Medicine 307, 468–476 (1982)
8. Milne, R., Nicol, C.: TIGER: Continuous diagnosis of gas turbines. In: ECAI'00: Proceedings of the 14th European Conference on Artificial Intelligence. Berlin, Germany (2000)
9. Noy, N.F., Chugh, A., Liu, W., Musen, M.A.: A framework for ontology evolution in collaborative environments. In: ISWC'06: Proceedings of the 5th International Semantic Web Conference, LNAI 4273. pp. 544–558 (2006), http://dx.doi.org/10.1007/11926078_39
10. Noy, N.F., Musen, M.A.: PromptDiff: a fixed-point algorithm for comparing ontology versions. In: In 18th National Conference On Artificial Intelligence (AAAI-2002. pp. 744–750 (2002)
11. Puppe, F.: Knowledge Reuse among Diagnostic Problem-Solving Methods in the Shell-Kit D3. International Journal of Human-Computer Studies 49, 627–649 (1998)
12. Puppe, F., Atzmueller, M., Buscher, G., Hüttig, M., Luehrs, H., Buscher, H.P.: Application and evaluation of a medical knowledge system in sonography (SONOCONSULT). In: ECAI'08/PAIS'08: Proceedings of the 18th European Conference on Artificial Intelligence, including Prestigious Applications of Intelligent Systems. pp. 683–687. IOS Press, Amsterdam, The Netherlands, The Netherlands (2008)
13. Schaffert, S., Bry, F., Baumeister, J., Kiesel, M.: Semantic wikis. IEEE Software 25(4), 8–11 (2008)
14. Tudorache, T., Nyulas, C., Noy, N.F., Musen, M.A.: WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web. Semantic Web (2012)
15. W3C: PROV-O: The PROV Ontology: http://www.w3.org/tr/prov-o/ (April 2013)
16. W3C: RIF-Core Recommendation: http://www.w3.org/tr/rif-core/ (February 2013)
17. W3C: SPARQL 1.1 recommendation: http://www.w3.org/tr/sparql11-query/ (March 2013)