

Knowledge-driven systems for episodic decision support

Joachim Baumeister^{a,b,*}, Albrecht Striffler^b

^a University of Würzburg, Institute of Computer Science, Am Hubland, 97074 Würzburg, Germany

^b denkbare GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg, Germany



ARTICLE INFO

Article history:

Received 4 November 2014

Received in revised form 24 July 2015

Accepted 15 August 2015

Available online 21 August 2015

Keywords:

Decision support systems
Knowledge engineering
Ontologies

ABSTRACT

The paper describes a new approach of developing and maintaining state-of-the-art decision support systems. Such systems are able to capture the collaborative work on decision problems over time. Due to the complexity of large problem spaces a multi-modal knowledge representation is proposed. For the realization of a multi-modal knowledge base we integrate semantic technologies as a fundamental layer by combining the W3C ontologies PROV-O and SKOS. The approach is demonstrated by an implementation report of an industrially deployed decision support system.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In the past, decision support systems have been established in numerous domains. The term *decision support system* gathers a variety of system types helping humans to make appropriate decisions for a corresponding problem. Power [1], for instance, distinguishes the following types of decision support systems (DSS):

- **Data-driven DSS:** Provide decision support based on the analysis of large amounts of data. Business intelligence systems are a typical system class of data-driven DSS.
- **Model-driven DSS:** Provide decision support by using accounting, financial, representational, or optimization models. These systems provide access and manipulation of the model.
- **Document-driven DSS:** Support is provided by collecting, retrieving, and (automatically) classifying large amounts of (unstructured) information. Typical system classes are document management systems and information management systems.
- **Communication-driven DSS:** This system type adds capabilities to support the communication and collaboration between people working on the same task. Often this type is combined with the other types.

- **Knowledge-driven DSS:** Use problem-solving capabilities to derive appropriate actions for stated problems. Expert systems and recommender systems follow the knowledge-driven approach.

In this work we focus on knowledge-driven DSS as an implementation of knowledge-based systems, where solutions (decisions) are derived from a given input of facts. The typical process of such a system is depicted in Fig. 1 in BPM notation [2,3]: Initially a data entry activity is required to start the decision making activity, where the findings are processed by a problem solver. The derived solutions are returned in a decision output event.

Fig. 2 shows an extension of the classic decision process by allowing repeated data entry. Here, updated data is processed by the problem solver yielding a possibly updated set of decisions by incorporating non-monotonic reasoning [4].

Examples of such systems can be found in almost all domains, ranging from second opinion systems in medicine to fault diagnosis systems in the technical domain. In the past, those systems were built as monolithic applications, where single agents enter findings to derive one or more suitable decisions. Examples are found in medical consultation [5–7] and technical diagnosis [8–10].

In the context of the web and in the context of collaboration new requirements arise that motivate a rethinking of some principles of classic decision support systems.

Collaborative Use A complex decision problem is often not solved by a single user, but it is solved by the collaborative contributions of different participants. Collaboration may have diverse faces: Users participate in the decision problem by providing or

* Corresponding author at: University of Würzburg, Institute of Computer Science, Am Hubland, 97074 Würzburg, Germany.

E-mail addresses: joachim.baumeister@denkbare.com (J. Baumeister), albrecht.striffler@denkbare.com (A. Striffler).

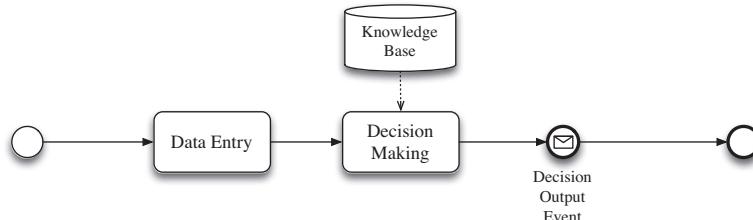


Fig. 1. Typical decision making process of knowledge-driven decision support systems.

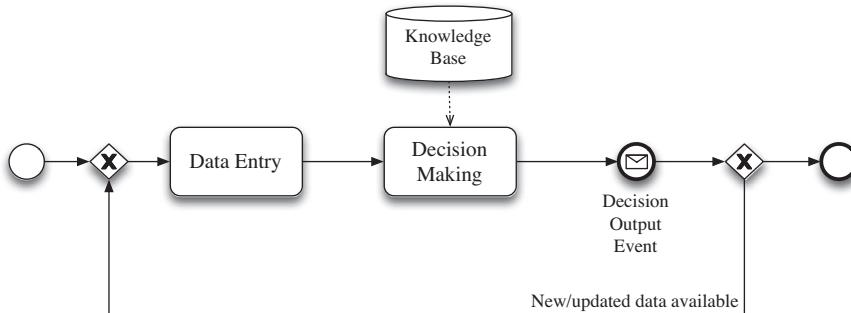


Fig. 2. Decision making process extended by repeated data entry thus yielding a non-monotonic decision process.

overriding important facts during a joint session. Alternatively, users solve disjoint sub-problems that in sum help to solve the overall decision problem.

Episodic Use Complex decisions are often not taken during a single session, but the actual decision process is partitioned over time into different *episodes*. We align to the semantics of the term *episodic* introduced by Russell and Norvig [11], where subsequent episodes do not depend on what actions occurred in previous episodes. In our setting, each episode may cover a different aspect of the decision problem. Commonly, the order of the handling of the different aspects has implications of the final reasoning process.

Mixing Knowledge Representations In traditional decision support systems a single knowledge representation is used to build the entire knowledge base. Successful knowledge representations are rules, decision trees, and Bayesian networks [12]. Complex and larger systems benefit from the use of hybrid approaches, integrating different representations into one knowledge base. Typically, a large knowledge base is partitioned into smaller *knowledge spaces*, where each knowledge space covers an aspect and uses a specific knowledge representation for its implementation. Here, for a single decision or fact, different knowledge representations can be *continuously* interwoven into a *multi-modal* knowledge representation.

In summary, advanced decision support systems need to deal with

- the collaborative use of the systems by a decision community,
- the episodic decision making of a problem,
- and multiple knowledge representations during the decision process.

Fig. 3 depicts an updated version of the decision process. The different knowledge spaces are represented by the different *sub-decision making* processes. These processes can be handled in parallel by different contributors of the decision making process.

A new episode is initiated by iterating the (same or different) sub-processes in the next episode. After every parallel execution of the sub-decision making processes the (final) decisions are aggregated in a subsequent process.

In the following we describe a novel approach of decision support systems by integrating different types of knowledge within reasoning for implementing the decision support. The rest of the paper is organized as follows: In Section 2 we motivate the use of a multi-modal knowledge base for building decision support systems, i.e., the knowledge formalization continuum. We also show how the use of multi-modal knowledge representations is implemented by an ontology layer. Section 3 describes the reasoning and explanation in multi-modal knowledge bases. The experiences with an industrial implementation of the presented concept is introduced in Section 4. The design decisions and experiences with building and running a collaborative decision support system in the domain of chemical safety are reported. Section 5 concludes the paper with a summary and a discussion of related work.

2. Engineering the knowledge formalization continuum

As we motivated in the introduction, complex decision support systems benefit from combining different representations instead of sticking to a single knowledge formalization. When the system needs to cover a complex domain, then it usually considers many aspects of the domain. However, for a number of practical reasons not all aspects can be included in a single knowledge base:

- **Uncertain domain knowledge:** Some aspects of the domain are not well-understood in a technical sense. In practice, decisions are often based more on past experience, evidence, and intuition than on strictly following domain laws and rules.
- **Bloated domain knowledge:** For some aspects, the explicit formalization 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* [13, p. 56] is low, e.g., because many parts will be rarely used in real-world decisions.

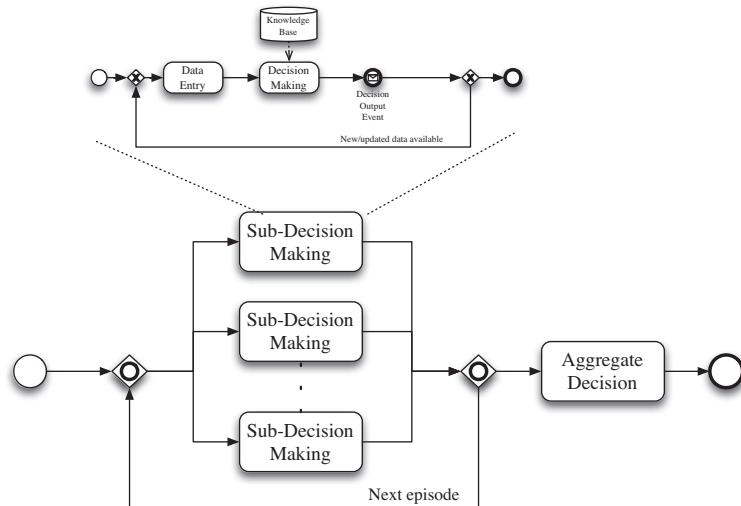


Fig. 3. Collaborative decision support process, where a complex decision problem is handled by multiple simultaneous decision makings.

In this context, the ratio is defined by developing and maintaining the knowledge base vs. the benefit/frequency of using the knowledge elements in practice.

- **Restless domain knowledge:** Especially in technical domains, some aspects are frequently changing due to engineering 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 be evaluated.

We introduce an approach that welcomes the combination of different knowledge representations into a single knowledge base. This approach makes use of the idea of the *knowledge formalization continuum* emphasizing the use of knowledge at varying formalization granularities (multi-modality). That way, the best-fitting representation for each aspect of the domain can be selected. Besides the interconnection of different knowledge representations the approach also considers the reasoning with knowledge at different formalization levels. This work extends the *knowledge formalization continuum* as presented in [14] by an implementation model of the knowledge space, which facilitates the reasoning with continuous and episodic knowledge. We are able to trace the reasoning process of multiple knowledge representations. This is important for decision support systems since we need a comprehensive derivation of decisions and corresponding explanation capabilities in the system, especially when multiple representations contribute to a particular decision.

2.1. The multi-modal knowledge space

For the engineering of decision support systems, the *knowledge formalization continuum* emphasizes the fact that a system can make use of more than one knowledge representation. Elements of the knowledge base can range from very informal representations (such as text and images) to very explicit representations (such as logic rules). The order of explicit knowledge vs. informal knowledge is defined by its capability of drawing automated inferences using the knowledge. For instance, rules are more capable to implement automated reasoning than plain images or textual tables. Fig. 4 shows a simplified depiction. Please note that the depiction is neither a precise nor a comprehensive scale but gives rough orders of formality of the knowledge representations. Approaches working with the knowledge formalization continuum do not need to commit to one single representation but use the

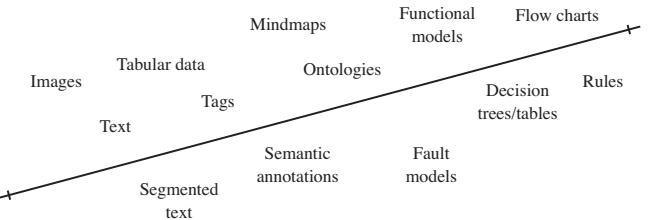


Fig. 4. An idealistic view of the knowledge formalization continuum.

best fitting representation for each module of the knowledge base. 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. rules. 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. The categorization of knowledge representations from informal to formal ones is also discussed by related research areas, especially in ontology engineering [15–17].

The flexible interpretation of knowledge representations needs strong tool-support to be useful for the development of decision support systems. That way, we need to define how to share and integrate decisions derived by the particular representations. Further, there needs to be the possibility to represent the decision making process of the diverse representations itself. We propose a stack of semantic technologies depicted in Fig. 5 for implementing these requirements.

Connecting the standard ontologies for provenance PROV [18] and the simple knowledge organization system SKOS [19], we are able to define the decision support ontology DSS. The reasoning process of proprietary knowledge representations is mapped to PROV, thus enabling the querying of the reasoning by the SPARQL language [20]. In the following, we discuss the approach in more detail.

2.2. Shared organization of decisions

In general, a decision support system derives appropriate decisions for a given input description. Consequently, each module of the system needs to know the set of possible decisions and it needs to have the capability to derive these decisions. By the set

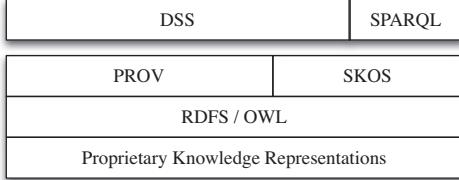


Fig. 5. The stack of used technologies.

of derivable decisions the *competence* of the module is defined. We propose the definition of decisions as ontology concepts, so that the decisions can be shared by each participating knowledge module.

The modeling of decisions by an ontology has a couple of advantages: First, a standardized interpretation of the decisions exists, since ontology languages such as RDF(S) [21] and OWL [22] define a clear semantics of the modeling primitives. Further, there already exist ontologies that support the knowledge acquisition task: The general vocabulary structure can be defined with the simple knowledge organization system SKOS [19]. For the generality of the approach we use SKOS for the organization of decisions, although tailored ontologies for decision support were proposed earlier [23,24]. In particular, SKOS defines the properties *skos:narrower* and *skos:broader* as inverse roles to build-up a hierarchy structure of concepts, i.e., the decisions.

For example, the decisions $D = \{d1, d2, d2.1, d2.2, d3, d3.1, d3.2\}$ are organized by *skos:narrower* relations as depicted in Fig. 6. The particular decisions are typed as instances of the class *dss:Decision* which is part of the general decision ontology. In the next section we explain this concept in more detail.

Having a unified hierarchy of decisions, every part of the knowledge base needs to align to this decision hierarchy to contribute to the general decision process. In Fig. 7 the previous decisions are linked to two different knowledge bases, where decisions from the branch *d2* are derived by a rule base and decisions from *d3* are derived by a workflow model.

Using the ontology structure we can jointly consider all decisions of the system and its parts correspondingly. However, the collaborative and episodic system also demands for a trace of the actual decision making process. We discuss an approach fulfilling this requirement in the following section.

2.3. Ontological representation of decisions making

The representation of decision making can be interpreted as the documentation of the reasons and facts for making a decision. Also all contributors and temporal data need to be represented. We call this documentation the *provenance* of the decision making process. Using multiple knowledge representations in a system, the representation of the decision making process also benefits from an ontological representation, since it provides an abstract but standardized language for this task.

We propose the use of the PROV ontology [18,25] to represent provenance information of the decision making process. As a W3C standard the PROV ontology already shows a wide range of applications in different domains. The PROV ontology explicitly represents the provenance of entities, i.e., in our case *decisions*, *entered data*, *users*, etc. are interpreted as PROV entities. For concepts defined in the PROV ontology the namespace *prov* is used. The three basic classes *prov:Agent*, *prov:Activity*, and *prov:Entity* are central for describing provenance information. Fig. 8 shows the general structure: An *prov:Agent* (a user/a problem-solving engine) is executing an *prov:Activity* (entering facts/deriving decisions) and produces an *prov:Entity* (input fact/derived decisions). Consequently, an *prov:Entity* can be attributed to an *prov:Agent* and the *prov:Entity* was generated

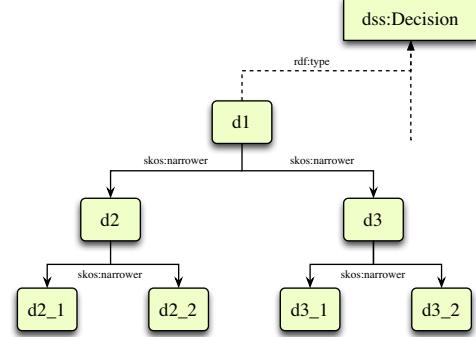


Fig. 6. An example hierarchy of decisions.

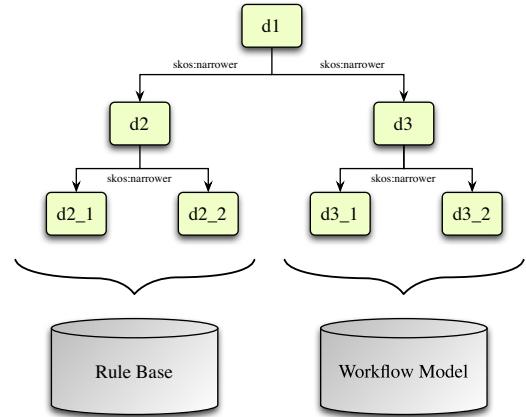


Fig. 7. Knowledge bases used for deriving decisions.

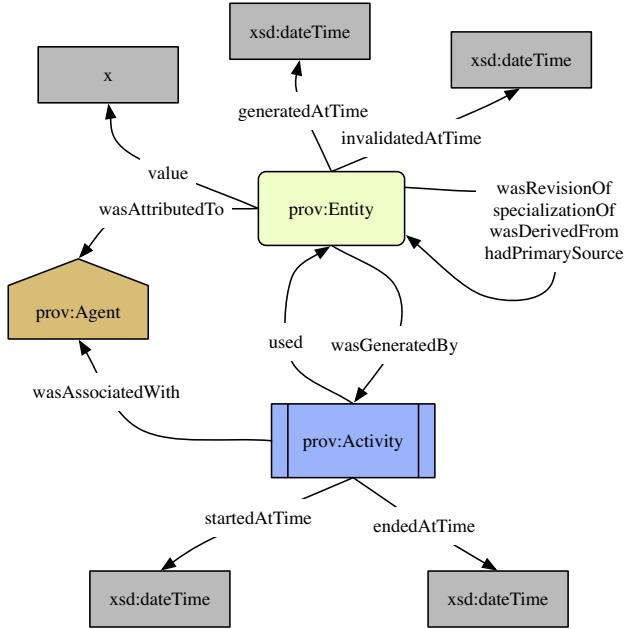


Fig. 8. The basic elements of the PROV ontology.

by a specific *prov:Activity*. An *prov:Activity* is also associated with an *prov:Agent*. In some processes an *prov:Activity* uses an *prov:Entity* for the creation of another *prov:Entity*, e.g., an entered fact is used for deriving a decision. An *prov:Activity* has a start and an end time; this is related to the generation time

of an `prov:Entity`. When the `prov:Entity` is superseded by a revision (`prov:wasRevisionOf`), then the `prov:Entity` is invalidated at a specified time. The following properties are also worthwhile to introduce: The property `prov:wasDerivedFrom` states that an instance of `prov:Entity` was transformed into another instance.

In its basic setting we see that the PROV ontology is suitable to generally represent the provenance of decision making. Concrete decisions are represented as `prov:Entity` instances. Participants of the derived decisions are represented as `prov:Entity`, `prov:Activity`, and `prov:Agent` instances.

For a more detailed representation we sub-class the concept `prov:Entity` into a decision support ontology (using the `dss` namespace). In Fig. 9 we see that decisions are instances of the `dss:Decision` concept, whereas entered data can depend on the knowledge representation used by the system. In the figure, we see examples of a formal rule base (`dss:RuleFact`) and an informal decision memo (`dss:DecisionMemo`). We further see that the state of a decision is represented by a `dss:DecisionAccount`. That way, the knowledge base stores the derivation state by using an instance of a `dss:Weight` instance. In principle, multiple representations can store different (and thus conflicting) `dss:Weight` instances. Then, the application needs to implement a conflict resolution strategy to avoid inconsistent decision states. We will discuss this topic together with other reasoning topics in the next section.

3. Episodic decision making with continuous knowledge representations

In this section, we *do not* discuss the different knowledge representations and their corresponding reasoning algorithms. There exists a broad range of approaches that all have strengths and weaknesses with respect to reasoning accuracy, knowledge acquisition costs, and maintenance processes. For a thorough introduction into knowledge representation and reasoning we refer to [4,26].

Here, we discuss the representation of the reasoning process within an episodic decision support system using multiple knowledge representations. In the previous section we proposed the use of the PROV ontology for this task. We show how this ontology is populated by instances generated during the reasoning process. Furthermore, we tackle the problem of explaining the reasoning process. In an episodic and collaborative application the explanation of made decisions is very important for users, since not all (sub-)decisions of a specific decision task are made by the same group of persons. Furthermore, some decisions may have been taken in the far past. Then, the 3W explanation of taken decisions (what?, why?, when?) will support the trustworthiness of the entire system. We will detail this in the following.

3.1. Broadcasting of decision making

In the introduction we depicted the general decision making process (Fig. 2), where a data entry initializes the reasoning and decision making. After the derivation of decisions the results are reported, for instance by simply presenting them to the user. In Fig. 10 we extend this process in the light of the requirements of collaborative and episodic decision making using mixed knowledge representations. Here, the solutions are not simply reported to the user anymore, but are broadcasted to a decision making ontology. Possible conflicts between decision making modules are also resolved in the ontology and the users are notified with

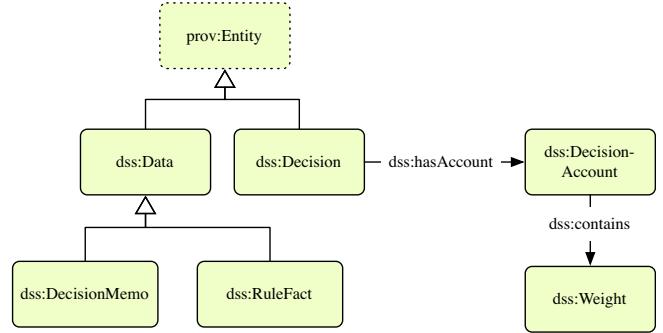


Fig. 9. `prov:Entity` concepts used for the decision making.

the final decisions. In many cases, the concrete conflict resolution strategy is implemented as a simple “last-come-first-serve” algorithm, i.e., in case of a conflict the latest decision will be established and will be presented to the user. In cases with strong user-interaction, this strategy may be extended by the “human-is-always-right” rule, i.e., decisions manually set by humans will always overrule decisions automatically derived by decision modules. The following example shows a decision process in ontological representation, where data is subsequently entered and decisions are derived.

Example. A given user (`user1`) enters new data in a data entry activity. The entry consists of the two facts `f1` and `f2`. Based on this entry and the facts, respectively, the knowledge base `k1` creates a new rule fact (`ruleFact 1`), that in turn derives the decision `d1`. The resulting ontology triples are given in Turtle [27] syntax:

```

1 ex:entry1 a dss:DataEntry ;
2   prov:wasAssociatedWith ex:user1 ;
3   prov:startedAtTime "2014-05-12+10:00"^^xsd:date ;
4   prov:endedAtTime "2014-05-12+11:00"^^xsd:date .
5
6 ex:f1 a dss:InputData ;
7   prov:hadPrimarySource ex:entry1 .
8
9 ex:f2 a dss:InputData ;
10  prov:hadPrimarySource ex:entry1 .
11
12 ex:ruleFact 1 a dss:RuleFact ;
13  prov:wasDerivedFrom ex:f1, ex:f2 ;
14  prov:wasAttributedTo ex:k1 .
15
16 ex:d1 a dss:Decision ;
17  prov:wasDerivedFrom ex:ruleFact 1 ;
18  prov:wasAttributedTo ex:k1 ;
19  prov:wasGeneratedBy ex:dataEntryActivity ;
20  prov:generatedAtTime "2014-05-12+10:00"^^xsd:date ;
21  prov:value "Established" .
  
```

Later, the user enters an informal memo (`entry2`) and connects this memo via the `memoFact 1` to the derivation of decision `d2` included in the knowledge base `k2`. The resulting triples added to the decision making ontology are:

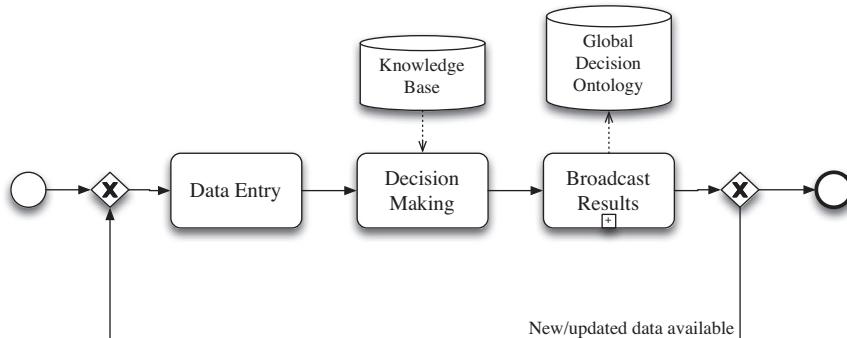


Fig. 10. Refined decision making process also broadcasting the results to the decision making ontology.

```

1 ex:entry2 a dss:MemoEntry ;
2 prov:wasAssociatedWith ex:user1 ;
3 prov:startedAtTime "2014-05-13+10:01"^^xsd:date ;
4 prov:endedAtTime "2014-05-13+11:01"^^xsd:date .
5
6 ex:f3 a dss:MemoFact ;
7 prov:hadPrimarySource ex:entry2 .
8
9 ex:memoFact1 a dss:MemoFact ;
10 prov:wasDerivedFrom ex:f3 .
11
12 ex:d2 a dss:Decision ;
13 prov:wasDerivedFrom ex:memoFact1 ;
14 prov:wasAttributedTo ex:k2 ;
15 prov:wasGeneratedBy ex:memoEntryActivity ;
16 prov:generatedAtTime "2014-05-13+10:00"^^xsd:date ;
17 prov:value "Established" .
  
```

Since we represent all activities as instances of the decision ontology, we are able to formulate the explanation queries as SPARQL statements [20]. The following SPARQL query gives an example answer to the 3Ws: It lists all derived decisions together with the primary acting agent (e.g., user responsible for deriving this decision) and the derivating sources of the decision. Also, the time of the decision derivation is shown. Due to the FILTER NOT EXISTS constraint all invalidated decisions are excluded from the result.

```

1 SELECT ?entry ?agent ?source ?generatedTime
2 WHERE {
3   ?entry rdf:type dss:Decision ;
4   prov:wasDerivedFrom+ ?source ;
5   prov:wasGeneratedBy ?activity ;
6   prov:generatedAtTime ?generatedTime .
7   ?activity prov:wasAssociatedWith ?agent .
8   FILTER NOT EXISTS {
9     ?entry prov:invalidatedAtTime ?invalidatedTime .
10  }
11 }
  
```

Typically, the query is specialized to a particular decision. Please note, that ad hoc explanations can easily be constructed by defining variants of the shown query.

4. Case study: collaborative decision support for chemical safety

The previous sections introduced a general approach for the implementation of collaborative decision support systems. There are many different possibilities for the realization of an episodic decision support system using mixed knowledge representations. In this paper, we describe the implementation of a decision support system that was deployed in 2012 in its first version and since then was extended and improved continuously until now. We describe the goals, development decisions, and experiences made with the collaborative decision support system KnowSEC (“Managing Knowledge of Substances of Ecological Concern”). The system is mainly in use by the unit *chemical safety* of the Federal Environment Agency of Germany (Umweltbundesamt).¹ The Federal Environment Agency of Germany is a scientific agency with about 1500 employees, and it observes the environment from different scientific disciplines and assesses the status of the nature. It advises politics and works on regulations in cooperation with other European environmental agencies. The main goal of KnowSEC is

¹ <http://www.umweltbundesamt.de>.

3.2. Explaining decisions and the decision process

When complex decision support systems are deployed into a collaborative and episodic environment, the system essentially needs to provide an explanation of derived decisions. Different people contribute to one decision over a longer period of time and a decision is derived by potentially heterogeneous knowledge representations. Then, transparent explanations improve the general acceptance of the system, but can also be used for tutorial and legal purposes by showing the reasons for a particularly derived decision [28].

An explanation approach for the described decision support system should give answers to the 3Ws (what?, why?, when?) and at least needs to meet the following requirements:

- **What:** Which decisions are derived?
- **Why:** Which knowledge and which users contributed to the derivation of a particular decision? The handling of different knowledge representations is an important requirement.
- **When:** What is the temporal development of a particular decision, i.e., the values within the past episodes?

Answering these questions requires the availability of the representation and versioning of each activity made in the decision support system. Every change needs to be traceable as it was described for instance by [29,30]. Also, the decision process itself needs to be represented as we introduced in the previous paragraphs.

the semi-automated support of expert users by ranking, assessing, and regulating new chemicals. As the foundation of the system, the semantic wiki KnowWE [31] is used for the development of the knowledge base but also for the application of the decision support system.

4.1. Knowledge base development

The development of the knowledge base started in 2011 with a small group of three domain specialists. Until now, eleven domain specialists have contributed to the knowledge base. Generally, the experts are organized in specialist teams, whereas these teams work on specific aspects of the knowledge base. For instance, there exists a team for the assessment of the persistence of chemicals, a team focusing on the bio-accumulation of chemicals, and a team for the exposure of chemicals. See Fig. 11 for different aspects that need to be considered for the assessment of a chemical. Each aspect corresponds to a single knowledge base, also called *module*. Initially, the modules were developed independently in order to facilitate faster development cycles, since no coordination was required between the teams. After the finalization of the first two modules we established knowledge formalization patterns [32] for the organization of the modules. That way, the modules are structured as similar as possible, using the same naming conventions, similar structures of decisions, and use similar inference strategies for decisions. In consequence, the maintainability was improved for existing modules, and the development of new modules was simplified significantly. When a module reached a steady state it was linked to the master knowledge base thus interweaving the linked modules. In the past and still now the teams are supported by two internal knowledge engineers that are mainly responsible for the organization and refactoring of the master knowledge base. Since, different (mainly independent) aspects are covered by the knowledge base, the development runs in parallel. At the moment of writing the knowledge base comprises eight decision modules, where the particular modules are implemented using different knowledge representations, see Fig. 12. The appropriate knowledge representation was selected according to the reasoning practice of the contributing expert team. For instance, some modules aggregate rather independent facts for a decision, whereas other modules show a strong dependency between particular facts. Simple decision modules with categoric logic are implemented by decision tables [32]. Complex decision modules elaborate sophisticated dependencies between different substance criteria. Those modules were implemented by scoring rules and diagnostic flowcharts [33]. Fig. 13 shows a non-monotonic scoring rule (simplified) included in the module assessing the *persistence* of a chemical.

As another example, the assessment knowledge of bio-accumulation was implemented by diagnostic flowcharts. A simple excerpt of the flowchart covering a part of the REACH regulation is depicted in Fig. 14: On the left, questions are depicted that are asked to the user (e.g., BCF value and source), whereas on the right a number of (sub-)decisions are inferred.

The modules are connected by an ontology of inputs and decisions as described in Section 2.2. Each decision module defines (sub-)decisions and corresponding input values; some inputs are shared with other decision modules. In total the ontology formulates 916 hierarchically structured decisions and 393 hierarchically structured input questions. Questions can provide multiple one-choice values, numeric inputs, and text values. The integrating concepts, however, are formulated in the decision hierarchy, where each module adds its own sub-tree of decisions. A part of the combined decision hierarchy is depicted in Fig. 15; here decisions of the module *bio-accumulation* are shown.

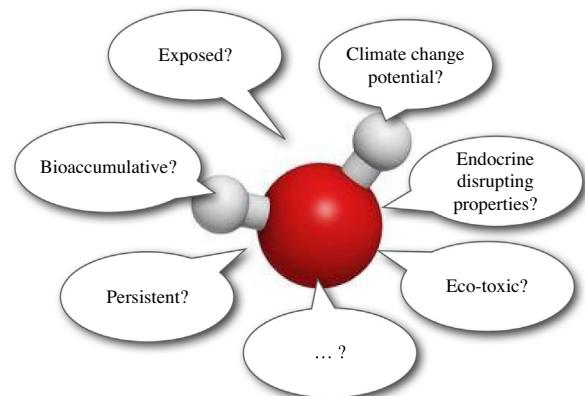


Fig. 11. Different target criteria during the decision process of a chemical substance.

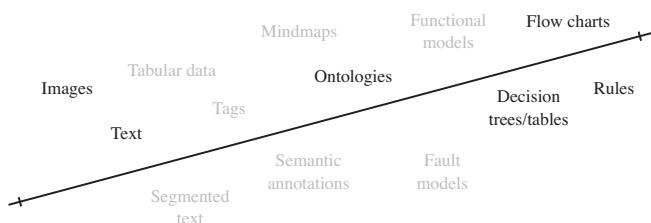


Fig. 12. Used and connected knowledge representations in the KnowSEC system.

```

IF "10-d window fulfilled" = Yes
AND ("ThCO2 (% degradation in 28 days)" >= 60
      OR "ThOD (% degradation in 28 days)" >= 60
      OR "DOC (% degradation in 28 days)" >= 70)
EXCEPT "Not readily biodegradable" = established
THEN "Readily biodegradable" = P6
  
```

Fig. 13. Assessment of the persistence of a chemical using non-monotonic rules.

4.2. The decision process

The initial goal of the system was the support of complex and time-consuming decision making on the chemical substances. A large number of specific chemical properties and their consequences need to be taken into account. Only substances under REACH² are managed. Typically, many different working groups (> 5 groups) are involved in the decision process that could run for months or years. For this reason, decisions need to be documented and traceable over time. The decision process is a collaborative task, where different sub-groups of the unit work on different aspects of the substance, for instance concerning safety (toxicity, persistence, bio-accumulation) or regulatory aspects, see Fig. 11 for some criteria. Every aspect is considered as an episode in the decision process, as we introduced in Section 1, since decisions taken for the particular criteria usually do not interact with each other. These criteria are tested in parallel or (more typically) subsequently, i.e., in true episodes.

Since the system is designed to not only support the decision process of a substance but also for its documentation, all actions and decisions about the substance are stored in the system. For that, each covered substance is represented by a distinct page in the system. That page gives an overview of the facts of the

² REACH stands for the European Community Regulation on chemicals and their safe use (EC 1907/2006).

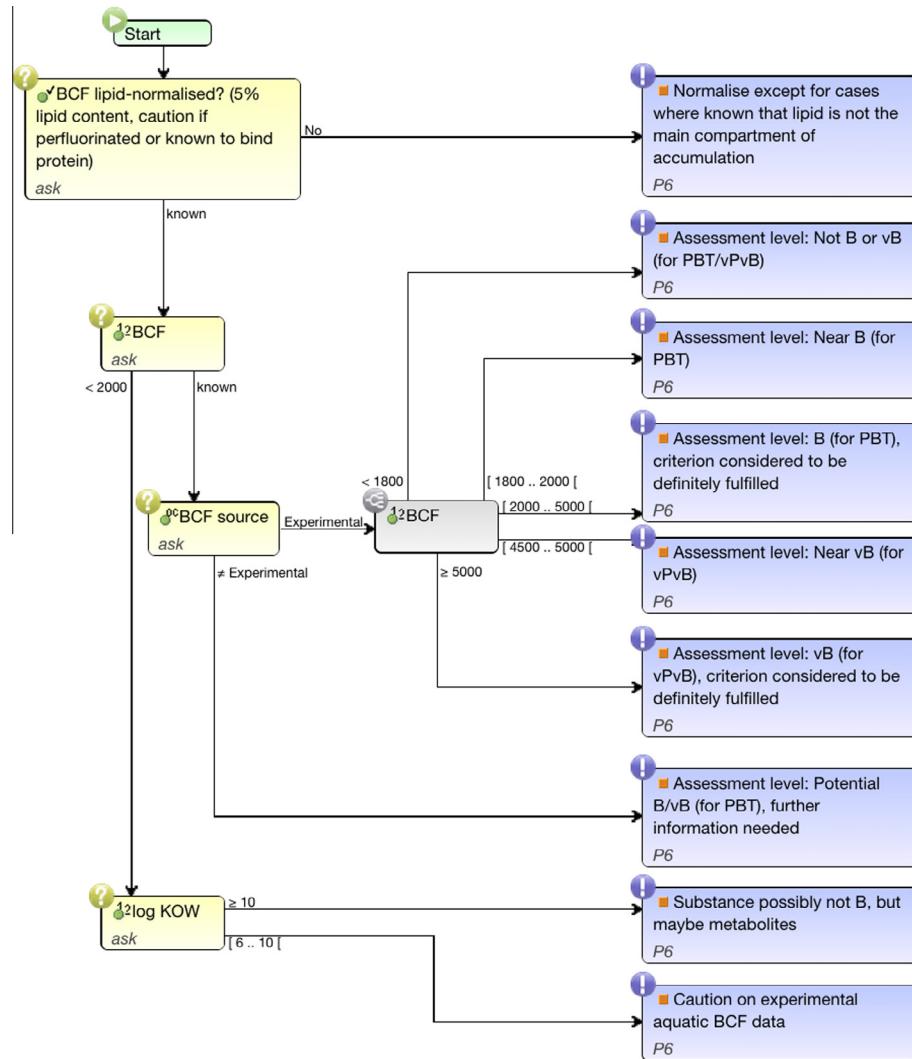


Fig. 14. Assessment of the bio-accumulation of a chemical using diagnostic flowcharts.

substance, summarizes all past decision actions, and offers the entry of new data and decisions. Provenance data is attached to each information displayed on the substance page, so that the involved users and the time of a particular decision is transparent to the visiting users. In Fig. 16 the page of the toy substance *Kryptonite* is shown (this substance is used for demonstration purposes, since no real substance data can be shown). On the right of the page the *Identifier* pane prints all known names and identifiers of the current substance. This pane is very important for the orientation of the users, since one substance can have many identifiers with inconsistent naming in the worst case. At the top of the center pane the made decisions are listed in the *Decisions* pane; explanation for selected decisions are generated when requested by the user. Below the overview pane, all informal decisions and comments are listed historically in the *Memos* pane. These memos can be filtered and sorted in various manners. At the left of the page a navigation bar offers quick links to frequently used services of the system. One important service links to (dynamically generated) *Substance Lists*, where users can define SPARQL queries to list substances with specific properties (please note that all actions are represented in the ontology and are thus SPARQLable). In the pane *Decision Making* the user can select different decision modules to be started for the current substance. When clicking on one of the module names, an interview form is opened asking for relevant

input values. Fig. 17 shows the interview form for the aspect *Persistence* and its sub-aspect *Biotic degradation*, respectively. For each given answer the decision module tries to derive new (sub-) decisions that are communicated to the user immediately.

4.3. Technical aspects

The open-source system KnowWE [31,34] was extended for the implementation of KnowSEC. KnowWE is a full-featured semantic wiki for the development of diagnostic knowledge bases and OWL/RDF(S) ontologies. A broad range of intelligent applications were already developed with KnowWE, for instance, clinical guidelines [35,36], the configuration of HCI devices [37], and the ontological formalization of ancient history [38]. KnowWE provides plugins for the automated testing and debugging [39] of knowledge bases including a continuous integration framework. In general, the use of a semantic wiki [40] offers a number of helpful features:

- Distributed and collaborative use due to a web-based interface.
- Different types of knowledge are already mixable such as text and images (informal) and ontologies (formal).
- A number of technical engineering features are commonly included like the versioning of content, access control of users, etc.

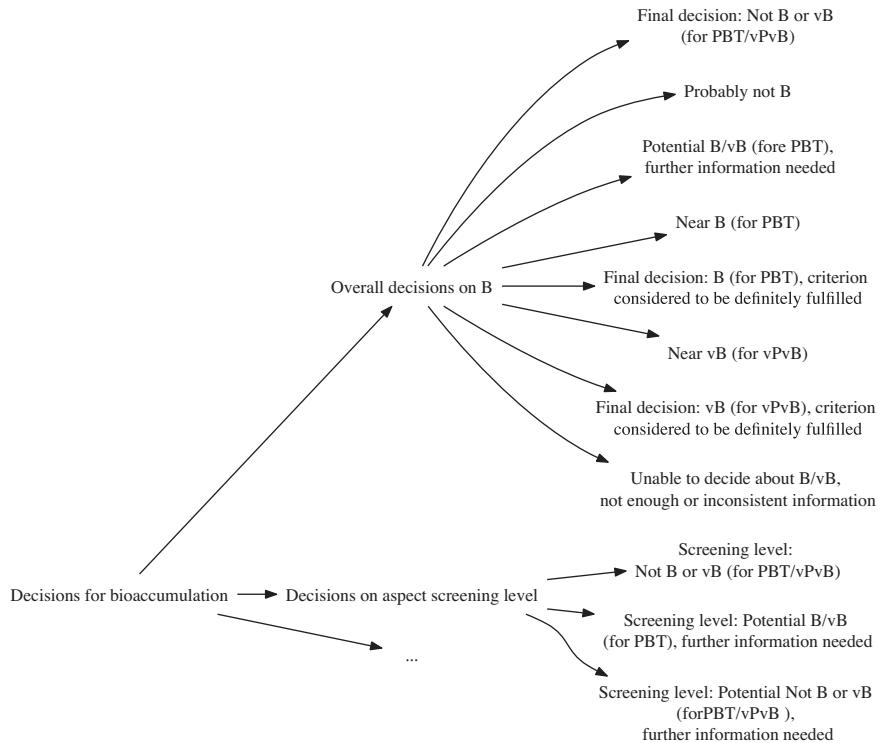


Fig. 15. An excerpt of the decision hierarchy; here a part of the decisions for evaluating the bio-accumulation of a substance is shown.

Kryptonite

G'day, Joachim Baumeister (authenticated)
[Log out](#)
[My Prefs](#)

[Quick Navigation](#)

Your trail:

[View](#) | [Attach \(1\)](#) | [Info](#)
[Edit](#)
[More...](#)

Decisions (13)	History	Identifier	
▶ Relevance	current	EC Number	
▶ Ecotoxicity	Final decision: T (for PBT), criterion considered to be definitely fulfilled	123-456-3	
▶ Raw Water Protection	There was already a full CCH on the substance	IUPAC Name	
CCH	There was already a full CCH on the substance	Not found	
TPE	There was a TPE on the substance	SMILES	Not found
Analysis of Risk Management Options	Substance Evaluation is no RMO	Substance Name	Kryptonite
▶ Substance Lists		Cryptonite	
Memos (3)	Sort by Timestamp Filter		

Initial Memo [🔗](#)
Joachim Baumeister, Abbau-Team, 31/10/2014

This is a example page!
Kryptonite is a fictional substance known from the Superman world. As a radioactive element originating from Superman's home planet of Krypton.

Toxic [🔗](#)
Joachim Baumeister, Trinkwasser-Team, 16/09/2014
T-Criterion fulfilled (Raw water) **(Established)**

Superman: The Man of Steel Sourcebook (1992). while non-canonical, concurs, referring to Kryptonite as "the common ore of the super-actinide Kryptonium, an unusually stable transuranic element, whose atomic number is believed to be 126". Kryptonium is given a radioactive half-life of 250,000 years. (from Wikipedia)

Fig. 16. The substance page of the toy substance Kryptonite.

Fig. 17. Data entry concerning the biotic degradation properties of the toy substance Kryptonite. Explanation texts for specific questions pop up on demand.

During the project the system was extended by KnowSEC-specific plugins, that support the episodic decision process and the optimized view of chemical properties.

4.4. Summary

The described system KnowSEC is in active use, providing the decision support and documentation for about 15,000 substances. The significance of the system is mirrored in the high number of about 76,000 (sub)-decisions documented so far; many of them were automatically inferred. The main benefit of the system is its combined use of decision support and documentation on varying levels of formality. The flexibility of the system is highly appreciated. Since the decisions, independent of their original knowledge representation, are structured into a combined decision hierarchy, the results can easily be retrieved using the standard query language SPARQL. Such queries can be defined by the users themselves and can be embedded into arbitrary pages of the system. Albeit not expected in the beginning, these flexible query definitions evolved as one of the most-appreciated features. Experienced users define their own queries to select and filter substances of their current interest. Furthermore, the ontological representation of almost all actions in the system (not only the decisions) allows for the simple use of explanation capabilities.

At the moment, the system is enhanced by decision modules covering less frequently used decision criteria. Also the linkage to other information systems is implemented making KnowSEC a SPARQL endpoint and thus a full member of the (internal) substance *linked data cloud*.

KnowSEC was already installed in other environmental agencies for conducting complex studies on related domains.

5. Conclusions

We conclude the paper with a summary of the contributions, discuss related work, and give a brief outlook to future work.

5.1. Summary

The collaborative and episodic use of information systems demands for advanced decision support systems. Such systems are able to handle knowledge and decisions at different levels of formality in a unified approach. They can be used collaboratively by working groups with different intentions and they can flexibly be tailored to project requests.

This work presented a novel design approach for such systems. We introduced the knowledge formalization continuum as a metaphor for handling knowledge and decisions at mixed levels, and we discussed a concept for representing the decision process and its trace in an ontological manner.

The practical applicability and relevance of this concept was shown in a case study, where the design and use of a collaborative decision support system for chemical safety is described. The presented case study is only one example of an industrial application, which is applicable to a broad range of domains. In the past years, the approach of episodic and continuous knowledge systems was implemented successfully in a number of further projects: In the medical domain, the implementation supported the collaborative development of mechanical ventilation guidelines [41]. Here, informal specifications were iteratively refined to explicit diagnostic flowcharts. At the time of writing there are about 300 ontological main concepts diagnostically connected by about 400 relations. Another medical application considered the interactive decision support system for cataract surgeries [42,43]. During the development, an expert textbook was incrementally transformed into explicit knowledge; the system also supports the episodic and step-wise-decision making by a group of users. Finally, the system comprised more than 320 explicit concepts connected with corresponding medical expert texts and 200 distinguished cross-connections served for decision support rules. In the machine building industry the approach was successfully implemented by developing a decision support system for selecting appropriate fault descriptions for given real world claims. Partly integrating resources automatically, the system includes about 50,000 concepts including components, failures, and conditions. Here, the claim formulation is a collaborative effort and the system supports the asynchronous improvement and use of the system.

5.2. Related work

Collaborative decision support systems are surveyed by Shim et al. [44]. Concerning their categorization, the presented approach is interpretable as a system for supporting group processes, since the decision process itself is also represented. A related design approach is described by the web-based system HERMES [45]. It augments classical decision making by supporting argumentative discourse among decision makers. The argumentation in the system is comparable to the decision steps *decision memos* and *decision dialogs* of the presented approach. However, HERMES provides an elaborated approach for the discussion and argumentation before making the actual decision. Palomares et al. [46] also approach *group decision making*, i.e., the problem of managing

situations when multiple experts need to come to a unified decision. They propose the visual interface MENTOR to depict the multiple opinions of the decision's participants and the evolution of their opinions. The approach presented in this paper tries to provide explanation capabilities by declarative SPARQL queries. A graphical visualization, however, is not available by now.

From a technical viewpoint, the paper proposes the combination of the standard ontologies SKOS and PROV, i.e., the integration of SKOS as a frame for representing the structure of a decision support system in combination with using PROV for representing the inference provenance of decisions. To the knowledge of the authors, this special combination of PROV and SKOS is novel, but a number of related approaches already combined instances of the SKOS and PROV ontologies. As part of the Getty vocabularies the Art and Architecture Thesaurus (AAT) was integrated into the linked open data cloud (LOD). The AAT vocabulary [47] uses a number of concepts of the ontologies SKOS and PROV. The AAT is a multilingual vocabulary for general concept describing art, architecture, and further cultural heritage. Here, SKOS is used for the taxonomic structure of the concepts and especially the labeling system of SKOS is applied, e.g., `skos:prefLabel`. The PROV ontology is used to represent the provenance of the actual ontology concepts, i.e., the creation and modification activities. In contrast to the work of this paper, no provenance of reasoning actions is represented.

The PAV ontology (Provenance, Authoring and Versioning) is defined for describing the provenance information of scientific digital content [48]. The ontology extends the concepts of the PROV ontology and introduces SKOS-based mappings for aligning the ontology concepts with terms from the LOD cloud, such as DublinCore entries. The PAV ontology is applied in the Open PHACTS project [49–51], which is a data hub linking pharmacological data bases by semantic technologies. The alignment of instances in Open PHACTS is implemented by corresponding SKOS relations, such as `skos:relatedMatch`, `skos:narrowerMatch`, and `skos:exactMatch`. The ontology mainly represents a data store for linked data bases but includes no reasoning trace as described in this paper.

5.3. Future work

In the future it will be interesting to extend the presented concept of a collaborative and episodic decision support system to a *linked data application*. Here, not only the integration into the linked open data cloud is reasonable but also the conceptual extension, where the system is integrated into closed environments as coined by the term *Corporate Semantic Web* [52]. Then, transparent user rights and tailored data interfaces become important topics. With the linkage of the system to external data resources the consideration of data quality becomes even more important, as for instance described in [53,54].

References

- [1] D.J. Power, *Decision Support Systems: Concepts and Resources for Managers*, Greenwood Publishing Group, 2002.
- [2] Object Management Group (OMG), 2014. <<http://www.omg.org/spec/bpmn/2.0/>>.
- [3] M. Dumas, M.L. Rosa, J. Mendling, H.A. Reijers, *Fundamentals of Business Process Management*, Springer, 2013.
- [4] F. Puppe, *Systematic Introduction to Expert Systems*, Springer, Berlin, 1993.
- [5] F. Puppe, M. Atzmueller, G. Buscher, M. Hüttig, H. Luehrs, H.-P. Buscher, 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, IOS Press, Amsterdam, The Netherlands, The Netherlands, 2008, pp. 683–687.
- [6] S. Mersmann, M. Dojat, SmartCare™ – 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, IOS Press, Valencia, Spain, 2004, pp. 745–749.
- [7] M. Kimura, M. Sakamoto, T. Adachi, H. Sagara, Diagnosis of febrile illnesses in returned travelers using the PC software GIDEON, *Travel Med. Infect. Dis.* 3 (2005) 157–160.
- [8] R. Milne, C. Nicol, TIGER: continuous diagnosis of gas turbines, in: ECAI'00: Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, 2000, pp. 711–715.
- [9] R. Helfman, J. Dumer, T. Hanratty, TED-turbine engine diagnostics: an expert diagnostic system for the M1 Abrams AGT1500 turbine engine, in: 11th Conference on Artificial Intelligence for Applications, IEEE Computer Society, Los Alamitos, CA, USA, 1995, pp. 1032–1038. <<http://doi.ieeecomputersociety.org/10.1109/CAIA.1995.378774>>.
- [10] J.C. da Silva, J.A. Matelli, E. Bazzo, Development of a knowledge-based system for cogeneration plant design: verification, validation and lessons learned, *Knowl.-Based Syst.* 67 (2014) 230–243.
- [11] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, second ed., Prentice-Hall, Englewood Cliffs, NJ, 2003.
- [12] M. Stefik, *Introduction to Knowledge Systems*, Morgan Kaufman Publisher, 1995.
- [13] W. Lidwell, K. Holden, J. Butler, *Universal Principles of Design*, Rockport Publishers, 2003.
- [14] J. Baumeister, J. Reutelshofer, F. Puppe, Engineering intelligent systems on the knowledge formalization continuum, *Int. J. Appl. Math. Comput. Sci. (AMCS)* 21 (2011) 27–39.
- [15] N. Guarino, D. Oberle, S. Staab, What is an ontology?, in: S. Staab, R. Studer (Eds.), *Handbook on Ontologies*, second ed., Springer, 2009, pp. 1–17.
- [16] M. Uschold, M. Gruninger, Ontologies and semantics for seamless connectivity, *SIGMOD Rec.* 33 (2004) 58–64.
- [17] D.E. Millard, N.M. Gibbins, D.T. Michaelides, M.J. Weal, Mind the semantic gap, in: HYPERTEXT'05: Proceedings of the 16th ACM Conference on Hypertext and Hypermedia, ACM, New York, NY, USA, 2005, pp. 54–62. <<http://doi.acm.org/10.1145/1083356.1083367>>.
- [18] L. Moreau, P. Groth, Provenance: An Introduction to PROV, *Synthesis Lectures on the Semantic Web: Theory and Technology*, Morgan and Claypool, 2013.
- [19] W3C, SKOS Simple Knowledge Organization System Reference, 2009. <<http://www.w3.org/TR/skos-reference>>.
- [20] W3C, SPARQL 1.1 Recommendation, 2013. <<http://www.w3.org/TR/sparql11-query>>.
- [21] W3C, RDF – Resource Description Framework Recommendation, 2004. <<http://www.w3.org/RDF>>.
- [22] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider, S. Rudolph (Eds.), *OWL 2 Web Ontology Language: Primer*, W3C Recommendation, 27 October 2009. <<http://www.w3.org/TR/owl2-primer>>.
- [23] C. Evangelou, N. Karacapilidis, O.A. Khaled, Interweaving knowledge management, argumentation and decision making in a collaborative setting: the KAD ontology model, *Int. J. Knowl. Learn.* 1 (2005) 130–145.
- [24] E. Kornyshova, R. Deneckère, Decision-making ontology for information system engineering, in: ER, Lecture Notes in Computer Science, vol. 6412, Springer, 2010, pp. 104–117. <<http://dblp.uni-trier.de/db/conf/er/er2010.html#KornyshovaD10>>.
- [25] W3C, PROV-O: The PROV Ontology, 2013. <<http://www.w3.org/TR/prov-o>>.
- [26] F. van Harmelen, V. Lifschitz, B. Porter (Eds.), *Handbook of Knowledge Representation*, Elsevier, 2008.
- [27] W3C, RDF 1.1 Turtle – W3C Recommendation, 2014. <<http://www.w3.org/TR/turtle>>.
- [28] T.R. Roth-Berghofer, M.M. Richter, On explanation, *Künstl. Intell.* 22 (2008) 5–7.
- [29] N.F. Noy, M.A. Musen, PromptDiff: a fixed-point algorithm for comparing ontology versions, in: 18th National Conference On Artificial Intelligence (AAAI-2002), 2002, pp. 744–750.
- [30] E. Franconi, T. Meyer, I. Varzinczak, Semantic diff as the basis for knowledge base versioning, in: 13th International Workshop on Non-Monotonic Reasoning (NMR), 2010, pp. 7–14.
- [31] J. Baumeister, J. Reutelshofer, V. Belli, A. Striffler, R. Hatko, M. Friedrich, KnowWE – a wiki for knowledge base development, in: The 8th Workshop on Knowledge Engineering and Software Engineering (KESE2012), 2012. <http://ceur-ws.org/Vol-949/kese8-05_04.pdf>.
- [32] F. Puppe, Knowledge formalization patterns, in: PKAW 2000: Proceedings of the Pacific Rim Knowledge Acquisition Workshop, Sydney, Australia, 2000.
- [33] R. Hatko, J. Baumeister, V. Belli, F. Puppe, DiaFlux: a graphical language for computer-interpretable guidelines, in: D. Riaño, A. ten Teije, S. Miksch (Eds.), *Knowledge Representation for Health-Care*, Lecture Notes in Computer Science, vol. 6924, Springer, Berlin/Heidelberg, 2012, pp. 94–107, http://dx.doi.org/10.1007/978-3-642-27697-2_7.
- [34] J. Baumeister, J. Reutelshofer, F. Puppe, KnowWE: a semantic wiki for knowledge engineering, *Appl. Intell.* 35 (2011) 323–344.
- [35] R. Hatko, J. Baumeister, V. Belli, F. Puppe, DiaFlux: a graphical language for computer-interpretable guidelines, in: KR4HC'11: Proceedings of the 3rd International Workshop on Knowledge Representation for Health Care, 2011, pp. 94–107.
- [36] R. Hatko, J. Reutelshofer, J. Baumeister, F. Puppe, Modeling of diagnostic guideline knowledge in semantic wikis, in: Proceedings of the Workshop on Open Knowledge Models (OKM-2010) at the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW), 2010.
- [37] S. Kreutzer, Ein Expertensystem zur Unterstützung körperbehinderter Menschen, Diplomica, 2012.
- [38] J. Reutelshofer, F. Lemmerich, J. Baumeister, J. Wintjes, L. Haas, Taking OWL to athens – semantic web technology takes ancient greek history to students, in:

- ESWC'10: Proceedings of the 7th Extended Semantic Web Conference, Springer, pp. 333–347.
- [39] S. Furth, J. Baumeister, An ontology debugger for the semantic wiki KnowWE, in: 10th Workshop on Knowledge Engineering and Software Engineering (KESE10), 2014.
- [40] S. Schaffert, F. Bry, J. Baumeister, M. Kiesel, Semantic wikis, *IEEE Software* 25 (2008) 8–11.
- [41] R. Hatko, D. Schädler, S. Mersmann, J. Baumeister, N. Weiler, F. Puppe, Implementing an automated ventilation guideline using the semantic wiki KnowWE, in: H. Stuckenschmidt, A. ten Teije, J. Voelker (Eds.), EKAW 2012: 18th International Conference on Knowledge Engineering and Knowledge Management, 2012, pp. 363–372.
- [42] K.-P. Scherer, C. Rieder, C. Henninger, J. Baumeister, J. Reutelshöfer, New concepts for knowledge based cataract surgery assistance, in: Proc. of the 8th International Joint Conference on Software Technologies, 2013.
- [43] K.P. Scherer, C. Rieder, C. Henninger, M. German, J. Baumeister, J. Reutelshöfer, Modeling and visualization of cataract ontologies, in: ADVCOMP 2014: 8th International Conference on Advanced Engineering Computing and Applications in Sciences, 2014.
- [44] J. Shim, M. Warkentin, J.F. Courtney, D.J. Power, R. Shardad, C. Carlsson, Past, present, and future of decision support technology, *Decis. Support Syst.* 33 (2002) 111–126.
- [45] N. Karacapilidis, D. Papadias, Computer supported argumentation and collaborative decision making: the hermes system, *Inform. Syst.* 26 (2001) 259–277.
- [46] I. Palomares, L. Martínez, F. Herrera, MENTOR: a graphical monitoring tool of preferences evolution in large-scale group decision making, *Knowl.-Based Syst.* 58 (2014) 66–74.
- [47] G.R. Institute, The Art & Architecture Thesaurus (AAT), Getty Vocabulary Program, 2015. <<http://www.getty.edu/research/tools/vocabularies/aat/index.html>>.
- [48] P. Ciccarese, S. Soiland-Reyes, K. Belhajjame, A.J. Gray, C. Goble, T. Clark, PAV ontology: provenance, authoring and versioning, *J. Biomed. Semantics* 4 (2013).
- [49] A.J. Gray, P. Groth, A. Loizou, S. Askjaer, C. Brenninkmeijer, K. Burger, C. Chichester, C.T. Evelo, C. Goble, L. Harland, S. Pettifer, M. Thompson, A. Waagmeester, A.J. Williams, Applying linked data approaches to pharmacology: architectural decisions and implementation, *Semantic Web* 5 (2014) 101–113.
- [50] A.J. Williams, L. Harland, P. Groth, S. Pettifer, C. Chichester, E.L. Willighagen, C.T. Evelo, N. Blomberg, G. Ecker, C. Goble, B. Mons, Open PHACTS: semantic interoperability for drug discovery, *Drug Discov. Today* 17 (2012) 1188–1198.
- [51] P.T. Groth, A. Loizou, A.J.G. Gray, C. Goble, L. Harland, S. Pettifer, API-centric linked data integration: the open PHACTS discovery platform case study, *Web Semantics: Science, Services and Agents on the World Wide Web* 29, 2014.
- [52] G. Coskun, O. Streibel, A. Paschke, R. Schäfermeier, R. Heese, M. Luczak-Röscher, R. Oldakowski, Towards a corporate semantic web, in: International Conference on Semantic Systems (I-SEMANTICS '09), Graz, Austria, 2009, pp. 602–610.
- [53] C. Fürber, M. Hepp, Using SPARQL and SPIN for data quality management on the semantic web, in: 13th International Conference on Business Information Systems, 2010, pp. 35–46.
- [54] T. Berners-Lee, 2009, Linked Data, <<http://www.w3.org/DesignIssues/LinkedData.html>>.