Using Knowledge Wikis to Support Scientific Communities

Joachim Baumeister[1,*], Jochen Reutelshoefer[1]
Karin Nadrowsk[2], and Axel Misok[2]

1) Institute of Computer Science, University of Würzburg, Germany

2) Institute of Animal Ecology, University of Giessen, Germany

With the success of numerous applications of the Web 2.0 the interest in a web–based support of scientific communities has also gained significant relevance. The concept of Wikis, one building block of the Web 2.0, has shown to be a reasonable infrastructure for sharing and refining any kind of knowledge. The most prominent example is the encyclopedia Wikipedia, but many smaller wiki applications have proved to be beneficial, e.g., the availability of experience management and documentation projects in open–source communities and large enterprises.

In this paper, we introduce *knowledge wikis* that extend the features of a regular wiki by the representation and use of explicit problem–solving knowledge. We motivate the general ideas and benefits of knowledge wikis, and we describe a concrete knowledge wiki implementation. The actual use of a knowledge wiki is demonstrated by two case studies: the first case study reports on the share and reuse of ecological domain knowledge in the context of the BIOLOG project; the second case study describes a knowledge formalization pattern repository taken as a demonstrator from the knowledge engineering domain.

1 Introduction

Community–based applications have shown significant relevance in the era of the Web 2.0. From the scientific point of view the use of wikis [7] for sharing experience and knowledge is the most interesting application of this new kind of systems. Besides

*Corresponding author: Joachim Baumeister (joba@uni-wuerzburg.de)

the prominent example *Wikipedia* many other smaller projects for documentation and knowledge sharing have used wikis as their foundational infrastructure. Here, knowledge is created and maintained in a evolutionary way through a web–based interface by the community itself. In general, wikis are comparable with content–management systems by building, creating, and sharing knowledge (i.e., text, figure, and multimedia), but Wikis provide a much simpler and more open interface of the engineering process. The content is viewed and changed by a web browser using the edit pane of the particular wiki, offering a simplified syntax for the formatting process which is usually less complex than HTML.

An obvious drawback of classic wikis is their "informality": Since the content is represented by the informal arrangement and structure of text the included knowledge is not explicitly available for the automated use by problem–solving methods. With *knowledge wikis* we propose an extension of wikis that is capable to integrate explicit problem–solving knowledge with the classic wiki content like text and figures.

In this paper, we focus on the representation of classification knowledge that is mainly used in the context of recommendation problems, selection tasks, and diagnosis. Classification systems typically use the available knowledge to compute a range of possible solutions for a given problem description. For example, in the context of a classification system for the recommendation of an appropriate form of sport the system proposes a list of suitable sport forms (e.g., swimming, running) for an entered user profile (e.g., training goals, costs, interests). Besides the automated problem–solving the knowledge wiki also offers information for the specific solutions. Here, the wiki describes the particular forms of sport on each wiki page but also includes explicit problem–solving knowledge, i.e., findings of a user that would suggest the specific type of sport. In the described example, a sports community would extend the system by further forms of sport (together with its requirements to a user profile) or would modify critical points of the available knowledge. The creation and evolution of the knowledge is done within the normal edit pane of the wiki, i.e., developers insert and change problem–solving knowledge using a textual knowledge markup. In summary, we propose knowledge wikis to be a suitable infrastructure to support communities when collaboratively building and sharing knowledge–enabled systems.

The rest of the paper is organized as follows: we introduce the concept of knowledge wikis in Section 2 in more detail by describing an implementation of a knowledge wiki, i.e., the system KnowWE. In Section 3 we report on the progress of two case studies currently running and demonstrating the use and focus of the proposed approach. A summary and outlook for future work is given in Section 4.

2 Knowledge Wikis with KnowWE

We introduce the basic concepts of a knowledge wiki demonstrated by the implementation of the *KnowWE* system. Whereas, the system KnowME [1] was a rich client implementation of an environment for the development of (diagnostic) knowledge systems, the system KnowWE [2] uses the same knowledge engine, but approaches the collaborative development of knowledge–enabled wikis. KnowWE builds upon the open–source Wiki engine TWiki[1] and extends it in order to support the explicit knowledge representation of classification tasks.

Each wiki page not only contains textual content, e.g., describing a particular solution or artifact, but also formalized knowledge, that can be used for interactive problem–solving. In general, we distinguish two basic elements of classification systems: *user inputs* represented by questions to the user and *solutions* derived by the systems for a given set of inputs. We support different types of user inputs, i.e., one–choice, multiple–choice and numeric inputs. The domain of an input i is denoted with $dom(i)$. For example, the domain $dom(TG)$ of a one–choice input TG (for Training Goals) contains all choice alternatives defined for the input, e.g., *endurance*, *rehabilitation*, and *staying fit*. In addition to these basic classes we provide different types of problem–solving knowledge for deriving suitable solutions for user inputs. The current implementation introduced in the following section is able to process (heuristic scoring) rules, set–covering knowledge, (heuristic) decision tables, and (heuristic) decision tree knowledge.

In this section we consider the knowledge acquisition issue using KnowWE, and then we show how this knowledge can be used within the knowledge wiki. Throughout the section we use an exemplary sport recommendation system that tries to select appropriate forms of sports as solutions for given user inputs (i.e., explicitly stated preferences).

2.1 Knowledge Acquisition with KnowWE

A knowledge wiki is organized in single wiki pages, whereas often each wiki page describes a particular solution by informal content and explicit knowledge. Revising the sports advisor wiki example we structure the wiki by representing every form of sport by a single wiki page. Then, the textual description of a sports form is enriched with explicit knowledge describing the typical preferences of a user that would derive the sports form as a suitable solution.

In normal wikis the acquisition of content, i.e., text, is typically accomplished via the edit pane in a web browser. In knowledge wikis the formulation of explicit knowledge is also managed using the edit pane. In consequence, we have to provide textual *knowledge markups* for the particular knowledge types. Informal content and explicit knowledge is created and modified together using the same edit pane.

2.1.1 Capture of Explicit Problem–Solving Knowledge

Figure 1 shows a screenshot of the edit pane of the wiki page "Swimming" in the exemplary knowledge wiki of the sports advisor system. Explicit knowledge is added to the normal wiki text surrounded by the special tag *Kopic* (for *knowledge topic*). In the example, we see the textual definition of a set–covering model [11, 4] of the solution "Swimming (occasional)", i.e., the listing of typical preferences/findings that would derive this solution as an appropriate recommendation. When saving the edit page

[1] http://www.twiki.org

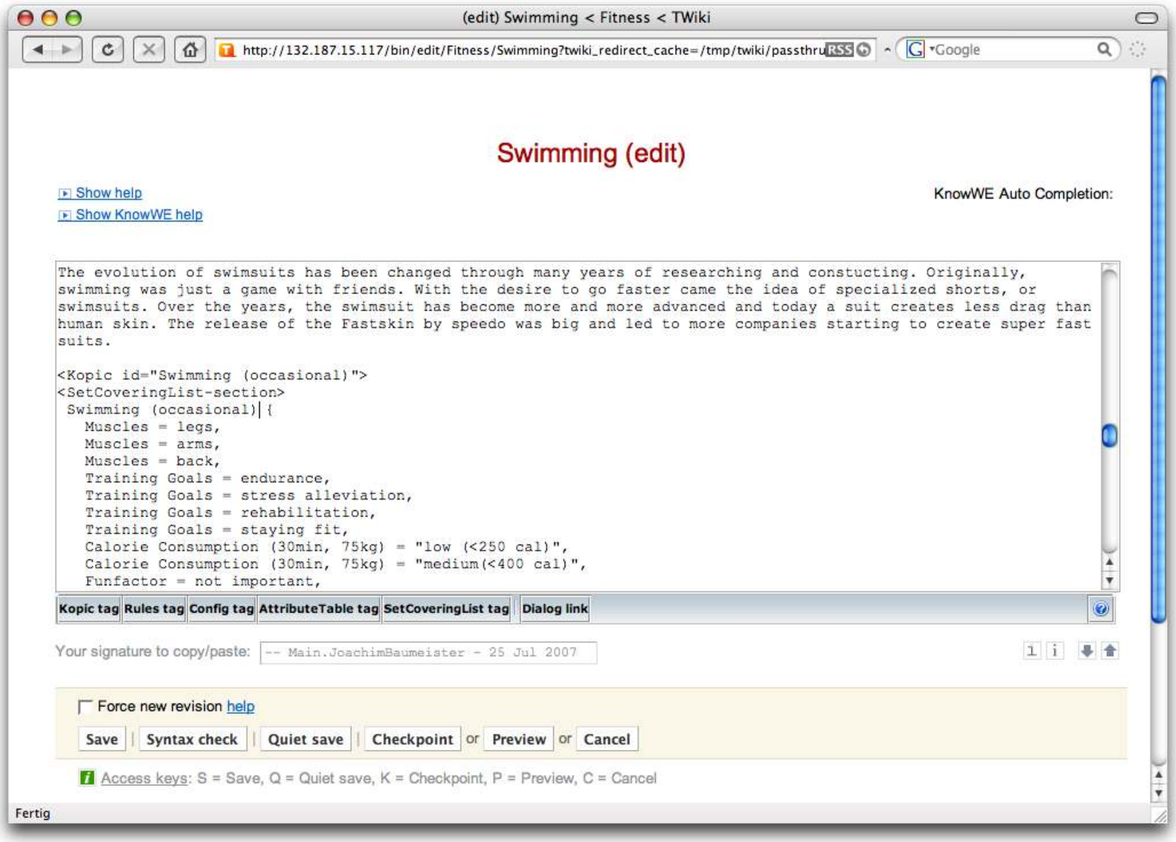

Figure 1: The creation of a page for the sports form "Swimming" and the corresponding problem–solving knowledge.

the knowledge wiki parser interprets the listed findings as new user inputs and creates set–covering relations for the specified inputs and the defined solution "Swimming (occasional)". Besides set–covering models the knowledge wiki is also able to interpret further knowledge markups for the formalization of decision trees, decision tables, and (heuristic scoring) rules [3].

In order to simplify the definition and maintenance of the entered knowledge the wiki interface provides a help page describing the syntax of the particular knowledge markups. Furthermore, an auto completion feature suggests inputs and solutions, that have been already defined in the current or other wiki pages, and thus encourages the reuse of already known ontological concepts. A syntax check gives verbose feedback in the case of the incorrect use of markup definitions.

The entered knowledge is stored together with the remaining wiki text by simply pressing the "Save" button. With the storage of the wiki text in the regular repository the contained knowledge markup is extracted and compiled into an executable knowledge base. This knowledge base is then stored in a *knowledge repository* together with knowledge bases of other wiki pages. In consequence, this process induces a large collection of knowledge bases for the entire knowledge wiki. These knowledge bases are loosely coupled by an alignment process that automatically aligns user inputs with the same name and type. The alignment process can be simplified by enforcing a *pre–defined terminology* for the entire wiki. When the knowledge wiki administrator constrains the possible inputs by setting a global terminology of inputs, new knowledge bases are bound to use this fixed set of possible user inputs. In this case, the inputs of the particular knowledge bases are trivially aligned to the corresponding inputs pre-defined in the global ontology. Reusing already defined names of user inputs in new knowledge bases yields a dense connection of the knowledge bases by the automatic alignment process, which is utilized during the distributed problem–solving process described in the following section.

2.1.2 Semantic Annotation of Input Concepts

The previous paragraph described the capture of explicit problem–solving knowledge by an example using set–covering models. Ontological concepts (inputs, solutions) and the corresponding derivation knowledge is easily defined in textual manner together with the text of the corresponding wiki page.

Knowledge wikis additionally provide a semantic annotation approach for tightly integrating the informal text phrases with explicit knowledge. Similar to annotation techniques known from *semantic wikis*, e.g. [6], in knowledge wikis the explicit semantics of text phrases can be annotated by concepts of the input terminology. In contrast to semantic wikis the annotation in knowledge wikis is mainly used for problem–solving, i.e., by creating interactive menus to enable data entries of the user within a wiki page.

In a knowledge wiki text phrases are annotated with defined input concepts; thus the following markup

```
%QUESTION{ns="myKnowledge" Name="myInput" Text="myText"}
```

annotates the wiki text `myText` displayed in the view mode with an input concept `myInput` known from the knowledge base with the namespace `myKnowledge`.

Revisiting our running example the following markup in the article *Swimming*, the input "Training goals" defined in the wiki page "Swimming" and its included knowledge base "Swim" is annotated twice: first, with the text "reducing stress" and second with the text "endurance".

```
Swimming is good for %QUESTION{ns="Swimming..Swim"
                       Name="Training goals" Text="reducing weight"}%
successfully or to train %QUESTION{ns="Swimming..Swim"
                       Name= "Training goals" Text="endurance"}%.
```

Using the semantic annotation we are able to combine the textual contents of a wiki page with the formally defined knowledge. The view mode of such an annotated wiki article uses the annotations to generate pop–up menus within the text, in order to provide a problem–solving interview seamlessly integrated with browsing the text, i.e., *in–place answers* (see Figure 2). We describe this technique in more detail in the following section.

2.2 Knowledge Consumption with KnowWE

We distinguish two ways of knowledge consumption in a knowledge wiki: 1) classic browsing of the wiki pages in order to manually retrieve the information required for solving the current problem (*implicit problem–solving*), and 2) an interactive interview with the user to solve the problem in a knowledge–based manner (*knowledge–based problem–solving*). With the presented knowledge wiki we further introduce a third approach, that tries to combine the manual browsing of web sites with a knowledge–based flavor, i.e., *in–place answers*. In the following, we discuss knowledge–based problem–solving with structured data acquisition and the in–place answers technique of KnowWE in more detail.

2.2.1 Reasoning using Structured Data Acquisition

For a knowledge wiki using a pre–defined terminology of inputs (cf. Section 2.1.1) a standard questionnaire is generated automatically, and a link to this questionnaire is then provided on every wiki page. A click on this link starts a structured interview with the user, where the user has to answer the presented inputs as questions. When not using a pre–defined terminology such a questionnaire can be easily defined within a knowledge wiki page.

Internally, the system collects all acquired inputs in a blackboard and propagates the values and the corresponding alignments to all knowledge bases in the knowledge repository of the knowledge wiki. In consequence, an answered question is provided to all registered knowledge bases that share the particular input. Subsequently, the distributed reasoning results of every registered knowledge base are again returned to the blackboard for further broadcasting and interpretation. Furthermore, solutions already derived by the knowledge bases are collected and presented to the user within the wiki interface.

Figure 2 shows the sports advisor example with already derived solutions given in the right pane. A link for starting the questionnaire ("Questionnaire for sports advisor") is also automatically provided in the center frame of the wiki. The structured questionnaire is opened in a new window when clicking this link, as shown in Figure 3. Here, the user is asked to give some input with respect to his expectations to a proposed sports form. As explained before, the entered inputs are propagated to all registered knowledge bases and solutions derived by the particular bases are presented. Thus, not only the current sports form is clarified, but also all other forms of sport for which explicit knowledge is represented in the knowledge wiki.

2.2.2 In–Place Answers

The structured interview described above is often perceived to be uncomfortable in typical "problem–solving sessions". In contrast, users are more used to browse web pages and to read the presented information in order to solve the problem by themselves, i.e., implicit problem–solving. Therefore, the presented knowledge wiki also provides interactive elements when reading the wiki text. We use the semantic annotation described

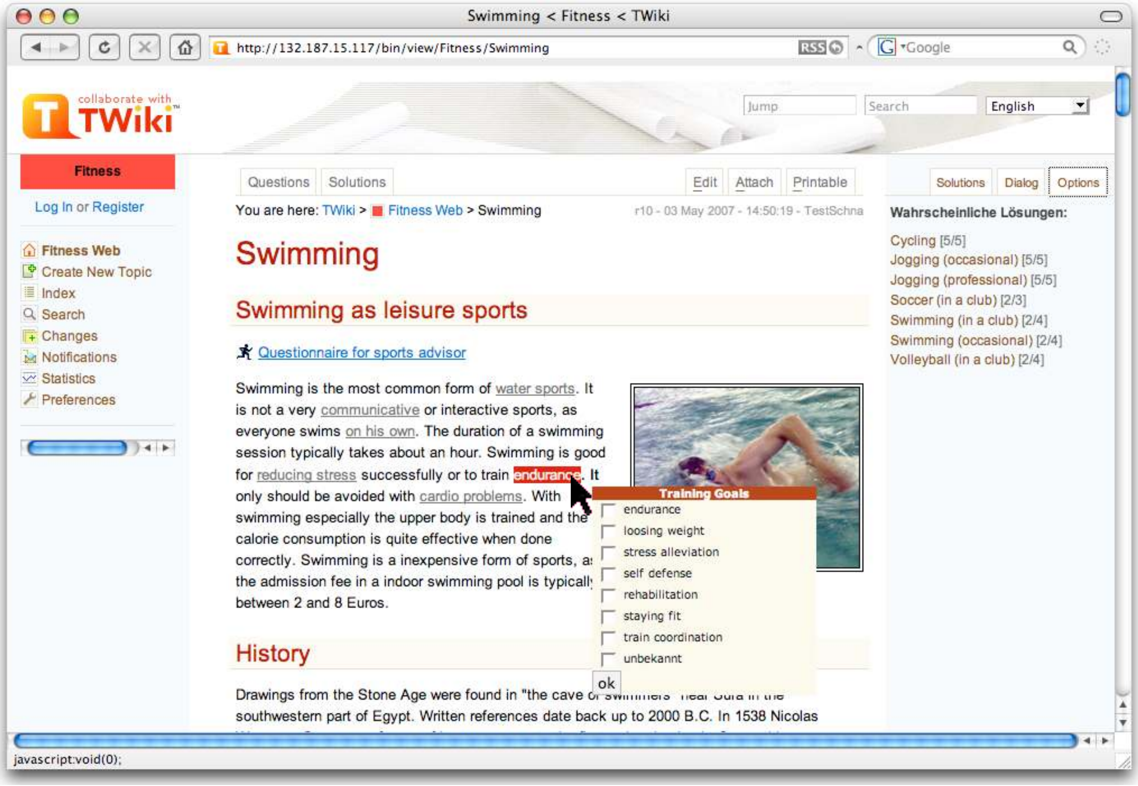

Figure 2: The view interface of the knowledge wiki KnowWE showing the sports advisor example.

in Section 2.1.2 to generate pop–up menus on tagged text phrases. We call this feature *in–place answers* since users are now able to actually provide explicit user input while reading the text, thus allowing for a seamless integration of informal and formal problem–solving.

For example, in the previously shown Figure 2 the click on the text phrase "endurance" activates a pop–up asking for the "Training Goals" of the user, where "endurance" is a possible answer. As the structured data acquisition approach described above the findings, acquired by in–place answers, are also propagated to all knowledge bases in order to derive suitable solutions. Consequently, the inferred solutions are presented in the solutions pane.

In this section, we introduced the concept *knowledge wikis* and described its current implementation KnowWE. A more detailed introduction to knowledge wikis and especially the possible knowledge markups is described in [2, 3]. In the next section we report on two applications that are currently developed using the presented KnowWE system.

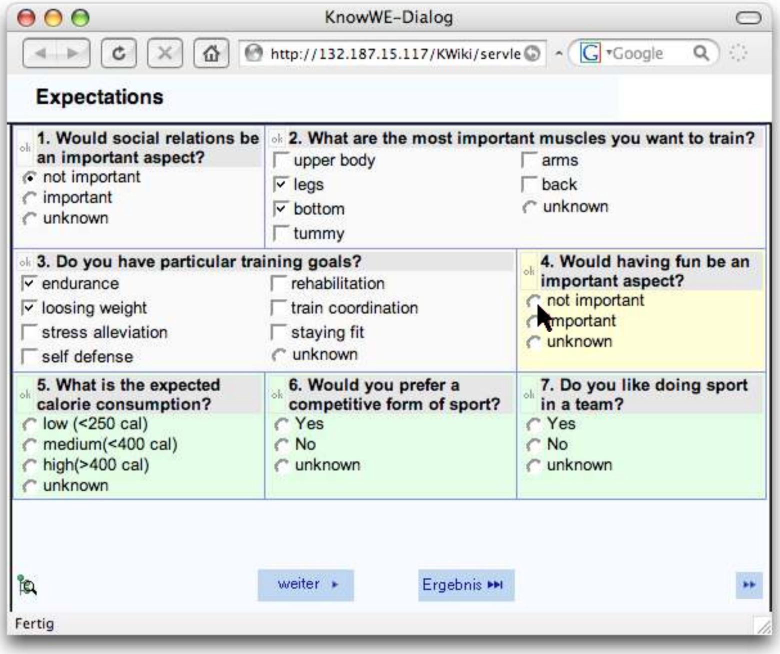

Figure 3: A questionnaire for the structured acquisition of user inputs.

3 Case Studies

In the last months we have gained a couple of experiences when using the knowledge wiki as a recommendation system. In different applications, we evaluated the usability and efficiency of the implementation. In a first "proof–of–concept" system a sports advisor wiki was developed by three computer science students, demonstrating the full range of features of the KnowWE system, including 56 knowledge bases for 31 of the most common forms of sports (some sports are redundantly defined reflecting different opinions of the students). A second case study considered the creation of knowledge wikis with a larger group of initially untrained persons. In total, 45 students started to implement 11 knowledge wikis with varying sizes. The experiences made in this case study motivated some improvements, mostly simplifications with respect to the knowledge markups.

In this paper, we introduce two running projects that are currently under development. Both projects are motivated by real world scenarios.

3.1 LaDy – Landscape Biodiversity

Landscape ecologists have gained a great amount of knowledge on how landscape structures and management affects the diversity of life [8]. The current effort to form inter- and transdisciplinary research projects with economists have resulted in socioeconomic knowledge on how biodiversity can be fostered in managed agroecosystems. At the same time, this knowledge is often inaccessible to decision makers in private or governmental agencies, since it is processed by and for an scientific audience. The BIOLOG Europe project[2] aims at integrating socioeconomic and landscape ecological research to study the effects of environmental change on managed ecosystems. To make the results of their research accessible both to domain specialists as well as diverse people from other backgrounds they decided to build a knowledge system (decision support system). The system is named “LaDy”, as an acronym for landscape biodiversity.

Although LaDy is open to any topics of interest to participants, as for example exchanging methodological advice or pictures, it’s core task is to structure the provision of textual knowledge and explicit relations on biodiversity, ecosystem services and management decisions.

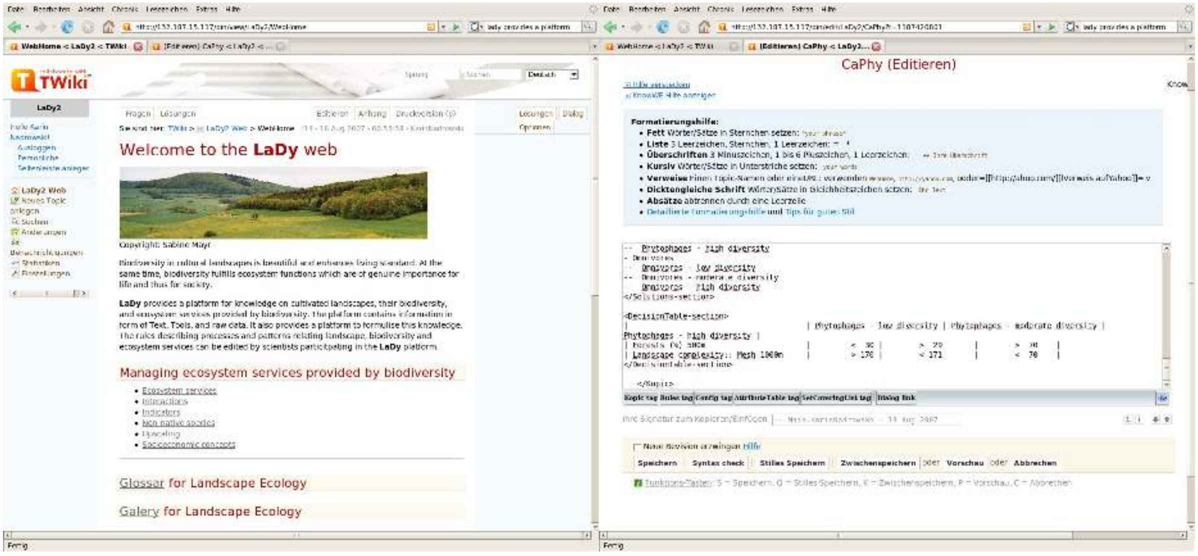

Figure 4: On the left: the welcome page of LaDy, a platform aiming at integrating knowledge on biodiversity, ecosystem services and environmental change in managed landscapes. On the right: explicit knowledge for predicting biodiversity of an animal group for given landscape structure; here, a decision table us used.

The left window in Figure 4 shows the welcome page of LaDy, with captions on ecosystem services, interactions, indicators, non-native species, upscaling and socioeconomic concepts. The window on the right shows the edit pane of a topic which implements a decision table. The decision table allocates three classes of biodiversity in phytophagous carabids (beetles) to variables describing landscape structure, in this case the percent cover of forest and the landscape complexity within a specified distance.

Providing a framework for both, exchanging textual knowledge and implementing explicit rules on ecosystem behaviour, LaDy aims at serving as a platform to condense and to communicate knowledge needed for an efficient management of ecosystem services provided by biodiversity.

[2]www.biolog-europe.org

3.2 Knowledge Formalization Repository

When building knowledge-based systems, the actual formalization of the required knowledge is the most complex task. Often, domain specialists find it very difficult to formalize the available knowledge in an appropriate way, thus describing the knowledge acquisition bottleneck. *Knowledge formalization patterns* [9] can help knowledge engineers and domain specialists by providing suitable templates for different scenarios of application requirements and features of the knowledge, i.e., *best practices*. Depending on the type of application and the special characteristics of the knowledge a particular knowledge formalization pattern can be selected that best matches the requirements of the projects. For each knowledge formalization pattern specific guidelines are given on how to formalize the domain knowledge, and consequences are discussed that are implied by the usage of the selected pattern. In this way, knowledge formalization patterns have a similar intuition for knowledge engineering as Design Patterns [5] have for software engineering.

The *knowledge formalization pattern repository* is available to all researchers and practitioners of knowledge systems, in order to share their experiences with using the described patterns or by introducing new patterns that have been successfully implemented. Knowledge formalization patterns are mostly described by the following elements: A meaningful name, a motivation for using the pattern, it's general applicability, an implementation structure, a discussion of the consequences when using the pattern, known uses of the pattern, related patterns and an example. Not all of the elements sketched above are used for the description of every pattern, but only when applicable.

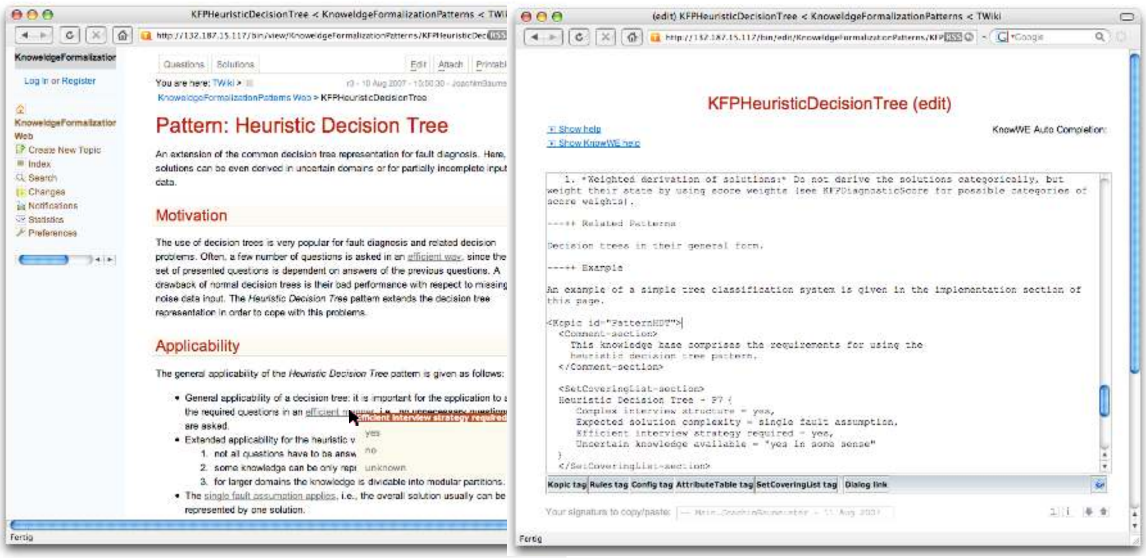

Figure 5: On the left: an extract of the knowledge formalization pattern repository describing the heuristic decision tree pattern. On the right: explicit knowledge for deriving the heuristic decision tree is defined in–text in the edit pane of the corresponding wiki page; here, set–covering knowledge is used.

The left window in Figure 5 shows the normal wiki interface of the knowledge formalization pattern repository describing the pattern *heuristic decision tree.* the right window of Figure 5 shows the corresponding edit pane. Here, set–covering knowledge is defined in–text formally describing the applicability of the heuristic decision tree pattern.

Interested users can browse the pattern wiki and learn to find the most appropriate pattern for their specific domain application. Using the in–place answers technique the specific user can instantly provide requirements and preferences of her application domain, and the knowledge wiki will offer suitable patterns (i.e., solutions) in the right pane of the wiki.

Currently, the repository is under development and in its initial version the pattern will contain a couple of patterns, that were initially documented in [9, 10]. With this initial version it is planned to invite related researchers to refine the existing patterns, and to share their experience on how to build knowledge bases using other knowledge representations like case–based reasoning and probabilistic networks.

4 Conclusions

We have presented the concept *knowledge wiki* as a novel infrastructure to support (scientific) communities of interest for sharing knowledge in a web–based environment. Knowledge wikis extend regular wikis by the representation of explicit problem–solving knowledge, that can be created, shared, discussed and modified in a wiki–like manner. The described approach showed applications of knowledge wikis for problem areas with classification tasks, but the general ideas are easily transferable to other knowledge tasks, like planning and simulation.

Since the work only discussed the initial aspects of a knowledge sharing method with wikis, we face many open issues for future work: from the technical point of view we are planning to refine the presented interfaces of the knowledge wiki. Since we envision experienced users to participate in a knowledge wiki and not thoroughly trained knowledge engineers, we have to simplify the access to the system and its technology as far as possible. One possible direction for future improvements is the simplification of the markups of semantic annotations and knowledge formalizations. Furthermore, similarly to some approaches of semantic wikis, e.g., IkeWiki [12], it may be beneficial to evaluate the usability of WYSIWYG editors for the formalization and maintenance of the included knowledge. From a social point of view, the presented approach has also space for further improvements. In the current implementation, the user is already able to attach comments to specific parts of the wiki text (including the formalized knowledge) in order to initiate a discussion on the specified area. In the future, more features like rating methods and user profiles may also improve the social aspects of the system.

References

[1] J. Baumeister. *Agile Development of Diagnostic Knowledge Systems*. IOS Press, AKA, DISKI 284, 2004.

[2] J. Baumeister, J. Reutelshoefer, and F. Puppe. KnowWE – Community-based Knowledge Capture with Knowledge Wikis. In *Proceedings of the 4rd International Conference on Knowledge Capture (K-Cap),to appear*, New York, NY, USA, 2007. ACM Press.

[3] J. Baumeister, J. Reutelshoefer, and F. Puppe. Markups for Knowledge Wikis. to appear, 2007.

[4] J. Baumeister, D. Seipel, and F. Puppe. Incremental Development of Diagnostic Set-Covering Models with Therapy Effects. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(Suppl.):25–50, November 2003.

[5] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[6] M. Krötzsch, D. Vrandecic, and M. Völkel. Semantic MediaWiki. In *Proceedings of the 5th International Semantic Web Conference (ISWC06), LNAI 4273*, pages 935–942, Berlin, 2006. Springer.

[7] B. Leuf and W. Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, New York, 2001.

[8] A. Otte, D. Simmering, and V. Wolters. Biodiversity at the Landscape Level: Recent Concepts and Perspectives for Multifunctional Use. *Landscape Ecology*, 22:639–642, 2007.

[9] F. Puppe. Knowledge Formalization Patterns. In *Proceedings of PKAW 2000*, Sydney, Australia, 2000.

[10] F. Puppe, S. Ziegler, U. Martin, and J. Hupp. *Wissensbasierte Diagnosesysteme im Service-Support [Knowledge-Based Systems for Service-Support]*. Springer, Berlin, 2001.

[11] J. A. Reggia, D. S. Nau, and P. Y. Wang. Diagnostic Expert Systems Based on a Set Covering Model. *Journal of Man-Machine Studies*, 19(5):437–460, 1983.

[12] S. Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In *1st International Workshop on Semantic Technologies in Collaborative Applications (STICA'06)*, 2006.