A Textual Case-Based Reasoning Framework for Knowledge Management Applications

Rosina Weber,[1,2] David W. Aha,[2] Nabil Sandhu,[3] & Hector Munoz-Avila[2,4]

[1]Department of Computer Science, University of Wyoming, Laramie, WY 82071-3682
[2]Navy Center for Applied Research in Artificial Intelligence,
Naval Research Laboratory, Washington, DC 20375
[3]ITT Industries AES Division, Alexandria, VA 22303
[4]Department of Computer Science, University of Maryland, College Park, MD 20742-3255
lastname@aic.nrl.navy.mil

Abstract. Knowledge management (KM) systems manipulate organizational knowledge by storing and redistributing corporate memories that are acquired from the organization's members. In this paper, we introduce a textual case-based reasoning (TCBR) framework for KM systems that manipulates organizational knowledge embedded in *artifacts* (e.g., best practices, alerts, lessons learned). The TCBR approach acquires knowledge from human users (via knowledge elicitation) and from text documents (via knowledge extraction) using template-based information extraction methods, a subset of natural language, and a domain ontology. Organizational knowledge is stored in a case base and is distributed in the context of targeted processes (i.e., within external distribution systems). The knowledge artifacts in the case base have to be translated into the format of the external distribution systems. A domain ontology supports knowledge elicitation and extraction, storage of knowledge artifacts in a case base, and artifact translation.

1 Introduction

Knowledge management (KM) concerns the gathering, organization, refinement, and distribution of knowledge [5]. Case-based reasoning (CBR) is a methodology that supports many of these activities, and has been frequently proposed as a methodology for KM applications [3,2].

One challenge to using CBR for KM is that knowledge is often stored in text format. Textual CBR (TCBR) [4] is a research area within CBR that addresses the manipulation of cases expressed in text form. Information extraction (IE) techniques, which transform a document collection into more structured representations [7], have been used in TCBR systems (e.g., [13]). For example, Weber et al. [17] used a variant of IE techniques called *template mining* [12] to extract information directly from text when there is an automatically recognizable pattern.

We introduce a KM framework to manipulate *knowledge artifacts* (i.e., structures that embed underlying concepts along with their conditions of applicability and rationale) that are well understood and accepted throughout organizations [18]. Example knowledge artifacts are best practices, lessons learned, incident reports, and alerts. In this framework, we use text extraction and knowledge elicitation techniques to acquire cases. We hypothesize that applying IE rules in a restricted natural language dialogue will: help to maintain and update a domain ontology; allow the content, context, and format of the elicited knowledge to be mapped into cases; yield clear and disambiguated cases that enable efficient case retrieval; and support conversion of that model into different formats of distribution systems. Our goal is to represent the knowledge so that it can be converted into the format of an external distribution system for subsequent proactive dissemination.

We are developing a multi-level TCBR framework and implementing it in a KM application for lessons learned systems [18]. In Section 2, we describe the motivations and benefits of our approach for lessons learned systems. We introduce our TCBR framework in Section 3 and describe an example application of it in Section 4. Related work is described in Section 5.

2 Lessons Learned Systems

Lessons learned (LL) *systems* have been deployed since the 1980's in private and government organizations to support lessons learned processes [18]. Existing LL systems are usually standalone retrieval tools that support variants of hierarchical browsing and keyword search in a repository of lessons. *Lessons learned* are the typical knowledge artifact stored in a LL repository. A lesson encodes validated knowledge from a work experience that can be reused to improve a targeted organizational process by suggesting a relevant contribution to a work practice.

2.1 Motivation

We are implementing the proposed TCBR framework (Section 3) to support KM tasks for the Navy Lessons Learned System (NLLS), which includes a lesson collection tool (NIIP) and a standalone lesson distribution tool that supports hierarchical browsing and keyword search for a repository of text-formatted lessons. NIIP's instructions do not enforce specific criteria on what to communicate when submitting a lesson. Consequently, text-formatted lessons are difficult to retrieve and interpret. Ergo, standalone distribution is viewed as inefficient.

The relations among our motivating issues are shown in Figure 1. Because currently deployed LL systems are viewed as inefficient, we focus on the primary goal of sharing knowledge. In particular, systems that manipulate knowledge artifacts will share knowledge by reusing these knowledge artifacts. There are two main reasons why these are not currently being reused [18]. First, because these artifacts are captured in unstructured text format, they are difficult to interpret (e.g., a lesson's conditions of applicability must be stated clearly). Knowledge artifacts must be unambiguous, making them easy to understand (e.g., the process(es) where they can be reused, their required conditions for reuse), and should employ a representation that highlights how they can be reused. Second, they are not being disseminated in the context of a distribution system (i.e., in the lesson's targeted organizational process(es)), when and where they are needed. To do this requires formatting the lessons according to the specification of the distribution system. Thus, it is necessary to model lessons from its original sources (i.e., human users and text documents) into an artifact representation that supports these two objectives.

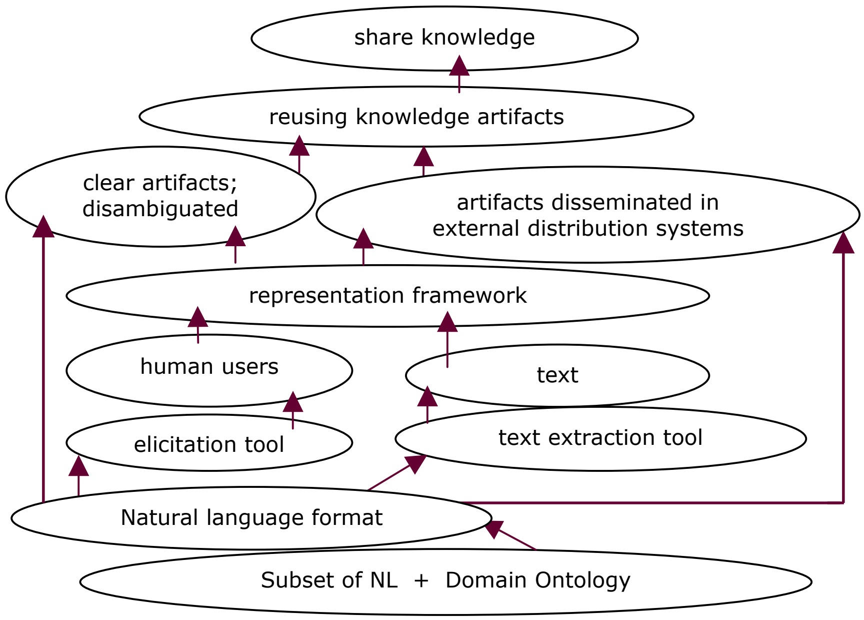

Fig. 1. Motivation diagram

An extraction tool converts documents into artifacts and an elicitation tool converts knowledge from users into artifacts, using the chosen case representation. These tools and the requirement that stored knowledge must be converted into the format of an external distribution system require addressing natural language (NL) understanding issues. To address these, we use an ontology and constrained NL techniques to identify known concepts, and to facilitate elicitation, extraction, and conversion. In addition, knowledge acquisition tools can help to populate an ontology. These motivations lead us to develop the TCBR framework described in Section 3.

3 Multi-level framework for Knowledge Management

3.1 Use for lessons learned

The basic components and dependency relations (illustrated by arrows) in the TCBR framework for KM are illustrated in Figure 2. Knowledge sources (i.e., human users, text documents) are the starting points for knowledge acquisition; users interactively provide knowledge to the elicitation tool when they access the system to submit a knowledge artifact, while an extraction tool is used to obtain knowledge artifacts from text documents.

Template mining [12,17] is used for extraction, combined with domain knowledge and knowledge of each artifact's structure. This approach is also implicitly employed to elicit knowledge artifacts from users; the elicitation tool guides and restricts a user's input by incrementally prompting menu selections to complete a previously defined template. This approach reduces the amount of text that has to be typed by avoiding the input of unnecessary details.

The ontology, which stores an abstraction and the mapping of all domain-specific concepts and meanings, can be used to verify the existence of all acquired concepts in the knowledge artifacts. By fully disambiguating the acquired knowledge, the system can then translate the artifacts into the format required by the distribution systems.

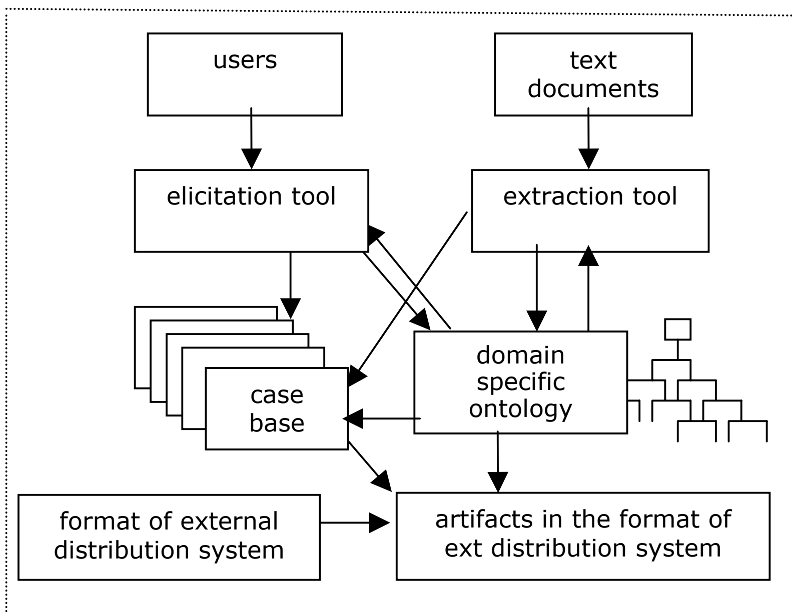

Fig 2. A textual case-based reasoning framework for knowledge management tasks

The novelty of our proposed approach is that we implicitly employ an adaptation of template mining to interactively elicit knowledge from users. In particular, we identify where to search, what to search, and, when requirements are met, what to extract. In addition to filling in templates by assigning values to attributes, case retrieval requires that these values be weighed to support knowledge intensive similarity assessment. For lesson acquisition, we define weights on each condition of a lesson by verifying its sufficiency and necessity. For example, when a user verifies that a condition was present when the lesson was learned and agrees that the same lesson can be reused even when this condition does not hold, then it is assigned a (relatively) lower weight.

The goal of both elicitation and extraction tools is to model the acquired knowledge as cases to support case retrieval. Besides the elements and components acquired by the elicitation or extraction tool, additional attributes are added to case descriptions as a result of inferences from the domain ontology (see example in 4.2).

Finally, the knowledge acquired has to be converted into the format used by an external distribution system to support active distribution in the context of the targeted organizational processes whenever an artifact is applicable.

3.2 Extending the methodology to other knowledge artifacts

Template mining is feasible when the text and its content embed some recognizable patterns, thus decreasing system requirements (e.g., no parser is required) and increasing its chances of success. Previous knowledge about the content and structure of these documents is required to apply template mining [17].

In our survey on lessons learned systems [18] we concluded that a KM application should focus on one knowledge artifact at a time, and thus a different implementation of the framework should be used for different knowledge artifacts. However, in the same domain, the specific ontology for one artifact is likely to be reusable for another, which can significantly reduce knowledge engineering efforts. Each knowledge artifact's representation can be extended as needed by the users. For example, all alerts have a feature for part, problem, and processes but the description of these processes will differ among industries. As this is a KM context, indexing is organization- rather than task-specific; each case will contribute the artifact's perspective of the knowledge for multiple tasks. For example, when the knowledge artifact is a lesson, an originating action in a military organization can be represented by a simple action, whereas in a chemical plant the originating action might be represented by a causal model involving many actions. Lessons will also record an applicable tasks and applicability conditions. In a best practices repository, artifacts will describe a process and teach how to best implement this process. And an incident reports repository will explain an incident's cause and ways to prevent it.

A different ontology must be generated for each domain because the meaning of a concept can differ across domains and user groups. For example, in the military, a *helicopter* is a resource that supports movement services. Yet in a medical domain, a *helicopter* is a transportation mode for trauma patients. In the aircraft industry, a *helicopter* is a product. Common to all of them is that a helicopter is a flying craft built to land and take off vertically.

This TCBR framework covers acquisition, storage, and dissemination and can be tailored to other knowledge artifacts other than lessons. There are some technical restrictions, like the requirements to employ text extraction and defining the domain ontology. In addition, concept disambiguation is an ambitious goal.

4 Application for lessons learned systems

Implementing the proposed framework has different challenges in each application domain depending on the quality of the available knowledge sources and the requirements of the representation into which the knowledge is to be converted for distribution. Implementation is simplified if the formats required for storing and distributing knowledge are similar (e.g., in language, level of detail, the audience, time, media).

Both users and a repository of text-formatted lessons are available for acquiring lessons for the NLLS,. The type of knowledge artifact (lessons) and the (military) audience affect the artifact representation, whose components are:

Applicable task: When (in which task) the lesson should be reused.
Suggestion: What to reuse (repeat or avoid) that changes a work practice.
Conditions for reuse: The circumstances defining when to reuse this lesson.
Originating action: The reason for recording this lesson.
Result: This identifies whether the originating action was a success or a failure.
Lesson contribution: The change/contribution to the applicable task that was learned.

Knowledge acquisition is guided by the artifact representation. Although our framework supports knowledge acquisition from both users and text documents, we focus here only on the elicitation tool.

4.1 Eliciting lessons

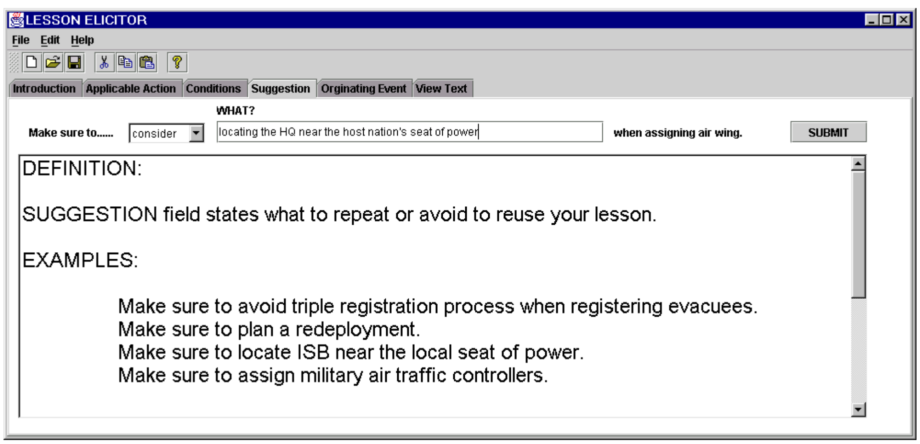

Fig. 3. Examples of suggestions in military lessons

The elicitation tool is implemented in a conversational fashion with a restricted subset of natural language, where we implicitly employ template mining rules of the type used in IE systems to locate patterns and then identify what to extract. This method prompts the user with intuitive styles of speech based on typical expressions that are identified by examining, in this example, the NLLS lesson repository. In the context of eliciting lessons, one of the attributes to extract is the suggestion (i.e., the recommendation that results from a lesson). Lesson authors tend to use expressions like *make sure*, or *ensure* before they communicate a suggestion, as shown in the examples displayed in Figure 3 The system imposes this format when it displays these examples, prompting the user with a template that outlines the information to be extracted. An example of this is illustrated in Figure 3, where a dialog box is prompted to the user with the expression *make sure* preceding a pull-down list of selectable verbs. After selecting a verb, the user can insert the remaining suggestion. The *applicable action* field is the first value extracted in the lesson elicitation process.

Knowledge elicitation is implemented in a conversational fashion where questions guide users to communicate desired concepts and to think logically about their input. For example, after a user enters an *applicable action*, an optional dialog box asks. “Are you sure this is the action where you can reuse your lesson?” When a lesson has more than one condition, the elicitation tool will try to identify their relevance ordering to support an appropriate weighing in the similarity functions. This is done by querying the user as to the necessity of each condition. Under user guidance, it determines the importance that a condition be present in lesson reuse opportunities.

4.2 Using the domain specific ontology in the elicitation process

The ontology is used to verify if the user’s inputs are valid domain concepts. Whenever the user’s input is in the ontology, the procedure is as follows. Suppose the user enters “it was a disaster relief operation” (Figure 4) for a military lesson. This condition triggers a method that assigns *disaster relief* to the attribute *operational objective*, which triggers another method that contains the knowledge that disaster relief operations have a *permissive hostility level*. Operational objective and hostility level are additional attributes that complement the artifact representation.

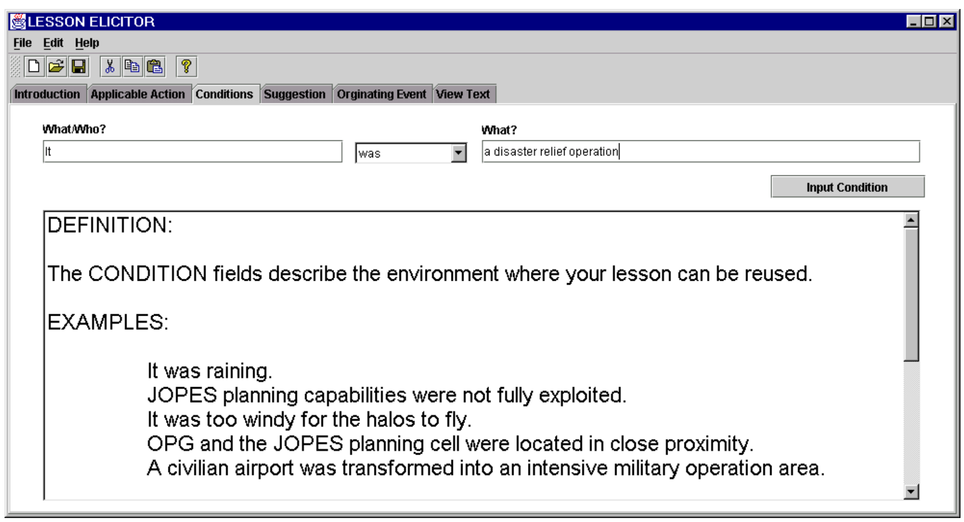

Fig. 4. Dialog for inserting a *condition*

4.3 Status of implementation

In our project, we are currently examining approximately 40,000 unclassified NLLS lessons for *corpus analysis* as the main knowledge source for automatic ontology construction [9]. Our next steps include integrating the elicitation tool with ontology management tools and using it to aid in populating the ontology. We intend to implement the extraction tool after evaluating the utility of the elicitation tool. We have tested lesson dissemination in the context of HICAP [15], which plays the role of an external distribution system. The conversion of lessons into HICAP's format is simple. We are examining other distribution systems that will challenge, test, and improve our translation method.

5 Related work

A KM approach should use and minimally modify existing work processes, or it may not be accepted by an organization's users. Thus, we suggest using well-known knowledge artifacts (e.g., lessons learned, best practices, incident reports, alerts) as a focus for KM activities. Landes et al. [11] previously proposed that identifying types of reusable experiences facilitates organizational learning because, once organized and represented, they can be easily retrieved and applied during subsequent problem solving. Many industries (e.g., aerospace, software engineering) use these knowledge artifacts because they represent experiences with an embedded purpose and clarifying structure, thus facilitating representation and retrieval for future reuse. Similarly, expectations are employed in elicitation tools. For example, Davis [8] uses rule models to derive expectations, while Kim and Gil [10] derive them from inter-dependencies among components of a knowledge base. Analogously, we derive expectations based on the structure underlying a knowledge artifact and its users.

CBR has been proposed as a foundation for several KM approaches (e.g., [2,3]) for several reasons. First, it can be used to collect, store, reuse, and adapt knowledge artifacts. Second, it supports partial matching, which is essential for this framework because many artifacts (e.g., organizational lessons) have multiple conditions of applicability, and a lesson's applicability depends on the degree to which its conditions match with known facts. Our approach share the use of cases and the acquisition from texts with the framework proposed in Schmalhofer et al. [16] for knowledge-based systems.

In addition to eliciting knowledge artifacts, knowledge acquisition techniques are also needed to acquire the specific ontology of the chosen domain. Our proposed framework's strategy is to elicit the knowledge and disambiguate it before storage. This is analogous to machine translation approaches [14], which fully disambiguate the source language before converting it into the target language.

Lenz's [13] work shares some commonalities with ours in that both propose using TCBR in a KM application but differs in the audience, in the targeted knowledge, and in the use of IE. Lenz's framework targets semi-structured documents such as FAQ's, which embed <question,answer> pairs but do not have a structure that indicates a purpose or utility to the targeted application.

Althoff et al. [3] describe a system that shares many commonalities with our approach but targets software engineering. Commonalities include the use of CBR to retrieve experiences, an organizational focus, a requirement for validating recently acquired knowledge, and an ability to modify the reuse potential of an experience (e.g., we use a domain-specific ontology to do this in our approach). While our framework manipulates knowledge embedded in knowledge artifacts (e.g., organizational lessons, best practices, alerts) in any domain, theirs manipulates knowledge typically from software engineering projects.

Our approach focuses on supporting distribution of knowledge in context [6;1;18]. The main distinction and contribution of our approach is that the elicited knowledge can be converted and then disseminated in context.

6 Concluding Remarks

We proposed a framework for case retrieval in KM applications. We adopt textual case-based reasoning techniques to extract text and to elicit knowledge from users. We claim that to be efficiently acquired, stored, and distributed, organizational knowledge has to be in a format that is well understood and accepted within these organizations. Knowledge artifacts such as lessons learned, best practices, and incident reports have been widely used. For this reason, it is reasonable to make them the central part in KM applications.

What should organizations do with knowledge that is not formatted as a knowledge artifact? Any knowledge source should have a clearly distinguished purpose of applicability (i.e., for which task it is useful) and distribution intent (e.g., training knowledge, best practice). Any KM system that stores corporate memories that do not highlight their reuse has potentially nothing to add beyond information.

Acknowledgements

This research was supported by grants from the Office of Naval Research and the Naval Research Laboratory.

References

1. Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., & Sintek, M.: Towards a Technology for Organizational Memories. *Intelligent Systems*. *13 3* (1998) 40-48.
2. Aha, D.W., Becerra-Fernandez, I., Maurer, F., & Muñoz-Avila, H. (Eds.): *Exploring Synergies of Knowledge Management and Case-Based Reasoning: Papers from the AAAI 1999 Workshop* (Technical Report WS-99-10). Menlo Park, CA: AAAI Press (1999)
3. Althoff, K.-D., Bomarius, F., & Tautz, C. (2000). Using a Case-Based Reasoning Strategy to Build Learning Software Organizations. *Accepted for the IEEE Journal on Intelligent Systems, special issue on KM and Knowledge Distribution over the Internet.*
4. Ashley, K.D., & Lenz, M. (eds.): *Textual Case-Based Reasoning, Papers from the AAAI-98 Workshop* (Technical Report WS-98-12). Menlo Park, CA: AAAI Press (1998)
5. Benjamins, V.R., Fensel, D., & Gómez Pérez. A.: Knowledge Management Through Ontologies. In: U. Reimer (ed.): *Proc. of the 2nd Int'l Conference on*

Practical Aspects of Knowledge Management (PAKM98), October 1998. URL http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-13
6. Cole, K., Fischer, O., & Saltzman, P.: Just-in-time Knowledge Delivery. *Communications of the ACM*, *40* 7, (1997) 49-53
7. Cowie, J., & Lehnert, W.: Information Extraction. *Communications of the ACM, 3 9* 1, (1996) 80-91
8. Davis, R.: Interactive transfer of expertise: Acquisition of New Inference Rules. *Artificial Intelligence 12* (1979) 121-157.
9. Grishman, R., & R. Kittredge (eds.): *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Hillsdale, NJ: Lawrence Erlbaum. (1986)
10. Kim, J., & Gil, Y.: Deriving Expectations to Guide Knowledge Base Creation. In: *Proc. of the 16th National Conference on Artificial Intelligence*. Orlando, FL, AAAI Press (1999) 235-241
11. Landes, D., Schneider, K. and Houdek, F.: Organizational Learning and Experience Documentation in Industrial Software Projects. In: *Proc. of the 1st Workshop on Building, Maintaining, and Using Organizational Memories* (OM-98), Brighton, UK, August, 1998. URL http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/
12. Lawson, M., Kemp, N., Lynch, M., & Chowdhury, G.: Automatic Extraction of Citations from the Text of English-language patents: An example of template mining. *Journal of Information Science* 22 6 (1996) 423-436
13. Lenz, M.: Knowledge Sources for Textual CBR applications. In M. Lenz & K. Ashley (Eds.) *Textual Case-Based Reasoning: Papers from the AAAI-98 Workshop* (Technical Report WS-98-12). Menlo Park, CA, AAAI Press (1998) 24-29
14. Mitamura, T., E. H. Nyberg, and J. G. Carbonell: An Efficient Interlingua Translation System for Multi-Lingual Document Production. In: S. Nirenburg (ed.): *Progress in Machine Translation*. Amsterdam, NL, IOS Press (1993)
15. Muñoz-Avila, H., McFarlane, D., Aha, D.W., Ballas, J., Breslow, L.A., & Nau, D.: Using Guidelines to Constrain Interactive Case-Based HTN planning. *In Proceedings of the Third International Conference on Case-Based Reasoning*, Munich, Springer (1999) 288-302
16. Schmalhofer, F., Reinartz, T., & Tschaitschian, B.: Unified Approach to Learning for complex real-world domains. *Applied Artificial Intelligence 9 2* (1995) 127-156
17. Weber, R., Martins, A., & Barcia, R. On Legal Texts and Cases. In M. Lenz and K. Ashley (Eds.): *Textual Case-Based Reasoning: Papers from the AAAI-98 Workshop* (Technical Report WS-98-12). Menlo Park, CA, AAAI Press. (1998) 40-50
18. Weber, R., Aha, D.W., & Becerra-Fernandez, I. Intelligent Lessons Learned Systems. *International Journal of Expert Systems Research & Applications* 20 1 (2001)