Intelligent Elicitation of Military Lessons

Rosina Weber
College of Information Science and Technology
Drexel University
Philadelphia, PA 19104 USA
+1 215 895 1911
Rosina.Weber@cis.drexel.edu

David W. Aha
Navy Center for Applied Research in AI
Naval Research Laboratory (Code 5515)
Washington, DC 20375 USA
+1 202 404 4940
aha@aic.nrl.navy.mil

Abstract
We introduce LET (Lesson Elicitation Tool), which uses domain and linguistic knowledge to guide users during their submission of lessons learned. LET can detect a user's need for instructions and disambiguates expressions while collecting taxonomic domain knowledge.

Keywords: Lessons learned systems, intelligent information and knowledge management, intelligent interfaces, active help, verb classes, human-computer interaction, templates

INTRODUCTION

Lessons learned systems (LLS) are knowledge management initiatives for supporting the capture, validation, storage, distribution, and reuse of organizational lessons. LLS are ubiquitous in military organizations, where lessons have been captured and stored for more than ten years [6]. Nonetheless, these repository systems are not yet promoting knowledge reuse. Recent work [1][5] associated reuse impediments with lesson distribution, format, and collection. Using an intelligent user interface strategy, LET overcomes the reuse obstacles imposed by the collection methods currently being used in military organizations.

MOTIVATION, GOALS, AND CONTRIBUTION

The design of LET was motivated by the definition of lessons learned, a set of requirements associated with knowledge processes like knowledge creation, and some directions inferred from the identification of technological obstacles to the success of LLS [5].

LET's primary goal is to elicit a useful and disambiguated lesson from its users. To support this goal, LET must provide sufficient guidance to lesson submitters by communicating and enforcing lesson content. Also, to reduce the user's burden, LET must use an intuitive format for lesson authoring. Finally, LET must produce lessons that are amenable to computational treatment and shall provide instruments for disambiguation.

IUI'02, January 13-16, 2002, San Francisco, California, USA.

LET's contributions are as follows. First, it highlights the importance of knowledge elicitation (KE) for repository-based KM systems and demonstrates a KE approach. Elicitation from users can guarantee the creation of useful knowledge artifacts because authors who enter lessons play the same roles as their prospective users. Second, LET's use of verb classes to identify a user's knowledge needs represents a novel method for proactive knowledge dissemination, building on relevant previous work in the area (e.g., [4]). Finally, LET implements a method for word disambiguation that results in a natural and efficient way of collecting domain specific expressions. This is particularly important for military (as well in medical) domains, in which the frequent use of acronyms with multiple possible meanings can prevent information systems from being used across sub-domains (e.g., Navy and Marine Corps).

In the remainder of this paper, we describe LET's architecture and methods.

LESSON ELICITATION TOOL (LET)

LET was designed to implement the lesson collection process. Like some information extraction [3] methods, it focuses on obtaining the values for a template. However, instead of extracting values and expressions from text, LET coaxes information from users. LET asks questions, starts sentences, offers drop-down lists, imposes pre-defined structures for some sentences, and displays examples for the information being extracted. The template is domain-specific because it contains domain-specific elements (as lists of military roles of lessons' prospective users) and because it was conceived for the military audience through the identification of patterns in textual lessons (we examined a corpus of 35,000 lessons).

Architecture

LET's architecture, presented in Figure 1, consists of an interface, supporting modules, and a lesson database. A lesson template underlies the interface; it combines acquisition methods for eliciting lessons and expressions. Modules performing different functions support the interface operation; they range from simple lists to knowledge bases. The final component is the lesson database, which stores elicited lessons.

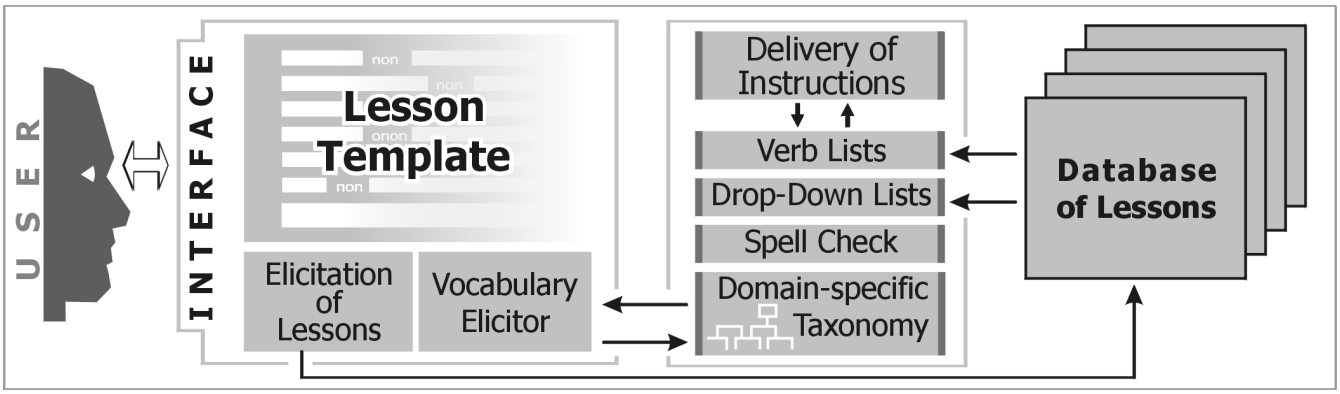

Figure 1. Architecture of LET

In addition to the lesson template, LET has two intelligent components that aid the collection process. The first supports proactive delivery of guidance, while the second continuously acquires domain-specific words (e.g., acronyms).

Proactive delivery of guidance

This component attempts to identify when users make mistakes and, thus, are in need of guidance. Our initial implementation targeted misuse of verbs with Levin's [2] verb classes. For example, the field APPLICABLE ACTION is composed of a verb phrase composed by a command verb and a noun phrase. The command verb is an activity verb. Activity verbs are part of a limited verb class, and are also always transitive verbs (because they always describe an accomplishment). Therefore, we can detect that a user did not understand the idea underlying the field if the user types an intransitive verb. In this case the user will be prompted with a short explanation and directed to the definition of the field. If the user types a transitive verb that is not in the set of activity verbs, then LET prompts a short explanation suggesting the user revise his verb choice. Unexpected prepositions in verb complements are also indicative of the need for guidance.

Acquisition of domain-specific vocabulary

During lesson submission, words are spell checked against a dictionary that embeds general and domain-specific words. When an unknown word is identified, LET's interface prompts the user, asking whether the unknown word is a typo or if it refers to a domain-specific word (e.g., nickname) that has not yet been identified. When confirming a word, the user will trigger the Vocabulary Elicitor. This module prompts categories in a simple choice list and asks the user to type a short description. As a result, a new word is added to the domain-specific taxonomy, or the originally typed word is corrected. LET's Vocabulary Elicitor can also be used to detect and clarify possible ambiguities in the elicited text, avoiding their permanent storage. As the taxonomy's growth levels off, automatic methods for conflict resolution tend to become necessary.

CONTINUING EFFORTS

LET's design is complete with respect to what was described here although we plan to extend its capabilities to promote effective support during the collection process. Most of what was described has been implemented; currently we are implementing the various dialogues that follow identification of inappropriate verbs from verb lists.

We are currently preparing an experiment with human subjects to assess the effectiveness of LET in producing useful lessons in comparison to currently deployed collection tools. We will evaluate LET with respect to its intended goals and report the results in future publications.

REFERENCES

[1] Aha, D.W., Weber, R., Muñoz, H., Breslow, L.A. and Gupta, K. Bridging the Lesson Distribution Gap, in *Proceedings of IJCAI'01* (Seattle, WA, Aug 2001), Morgan Kaufmann Publishers, Inc., 987-992.

[2] Levin, B. *English Verb Classes And Alternations: A Preliminary Investigation* The University of Chicago Press, Chicago IL, 1993.

[3] Gaizauskas, R. and Wilks, Y. Information Extraction: Beyond Document Retrieval. *Computational Linguistics and Chinese Language Processing* 3, 2 (Aug 1998), 17-60.

[4] Johnson, C., Birnbaum, L., Bareiss, R., and Hinrichs, T. War Stories: Harnessing Organizational Memories to Support Task Performance. *Intelligence: New Visions of AI in Practice* 11, 1(2000), 17-31.

[5] Weber, R., Breslow, L., and Sandhu, N. On the Technological, Human, and Managerial Issues in Sharing Organizational Lessons, in *Proc. of FLAIRS* (Key West FL, May 2001), AAAI Press, 334-338.

[6] Weber, R., Aha, D.W., and Becerra-Fernandez, I. Intelligent lessons learned systems. *Expert Systems with Applications 20*, 1 (January 2001), 17-34