Reflection Patterns for Interactive Knowledge Capture*

Jihie Kim
Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
jihie@isi.edu

Abstract

Current knowledge acquisition tools have limited understanding of how users enter knowledge and how acquired knowledge is used, and provide limited assistance in organizing various knowledge authoring tasks. In this paper, we present a novel extension to existing knowledge acquisition tools where the system 1) captures the episodes of knowledge acquisition and knowledge use through a set of declarative *reflection patterns* 2) performs assessment on how to improve the future knowledge acquisition and knowledge use based on captured episodes, and 3) provides assistance to the users by combining the assessment results.

1 Introduction

Knowledge acquisition (KA) remains a key challenge to knowledge-based AI applications. There have been increasing interests in supporting end users (i.e., ordinary users who do not have computer science background) directly enter complex problem solving knowledge on how to perform tasks. Although these techniques have been applied in building sizable knowledge bases in some cases, detailed analyses of the user interactions reveal that existing KA tools provide limited assistance in organizing and executing various knowledge authoring tasks [Kim and Gil 2000]. Systems do not reflect on how users perform KA tasks, how acquired knowledge (called *k-items*) is tested and used, and how future k-items should be built based on past experience. Users themselves have to keep track of the past mistakes, current status, potential new problems, and progresses made over time in order to decide the best options among possible courses of actions. The challenges in supporting these reflective capabilities include:

- The system should be *self-aware*, accessing and reasoning on interesting aspects of knowledge acquisition and knowledge use in assisting users.
- The system should relate the current situation to similar situations in the past and assess the levels of confidence in pursuing alternative options based on the relations.
- The system should recognize dynamic changes in the problem solving and decide how to guide users in modifying and using relevant k-items.

2 ECHO: Reflection Patterns for Interactive Knowledge Acquisition

In this paper, we present a novel extension to existing KA tools where the system adds an additional layer to existing tools and explicitly keeps track of knowledge acquisition and knowledge use episodes through a set of declarative *reflection patterns*. Reflection patterns allow the system to *be aware of* interesting knowledge acquisition and knowledge use episodes (called *meta-events*) that the system can make use of in assisting users. Each meta-event is a sequence of basic knowledge acquisition and knowledge use events such as the user ignored a suggestion and then the problem solving failed. The system *assesses the levels of confidence* in providing a suggestion based on its supporting and opposing meta-events captured in the reflection patterns. Any changes that are noticed (e.g. confident k-item became inconsistent with problem solving results) and associated k-item modifications are explicitly captured in the reflection patterns and are used in guiding the user.

We have built a system called Echo (mEta-Cognitive History analysis and Organization) that provides these capabilities. Echo dynamically captures a set of meta-events by matching the reflection patterns against the episodes of how different types of k-items are acquired, modified and used over time. The captured meta-events affect how the suggestions to the user are provided (i.e. how confident it is on the suggestion) as well as the content of the suggestions.

A prototype system has been developed for a domain of interactive scheduling where the user incrementally builds scheduling constraints and the user entered constraints assist users during scheduling. Since scheduling constraints can change over time, the system should be able to assist users in making associated modifications. The details of the system are described in (Kim 2005).

In developing reflection patterns and determining which meta-events the system needs to keep track of we have analyzed typical ways in which failures can occur in interactive knowledge capture and use. The reflection patterns are built help users avoid those failures. Figure 1 shows the paths

* The support of IJCAI, Inc. is acknowledged.

that Echo exploit currently. Depending on a particular need of a given KA system and its undesirable paths, we can include other meta-events in the reflection patterns and adjust the assistance provided by the system.

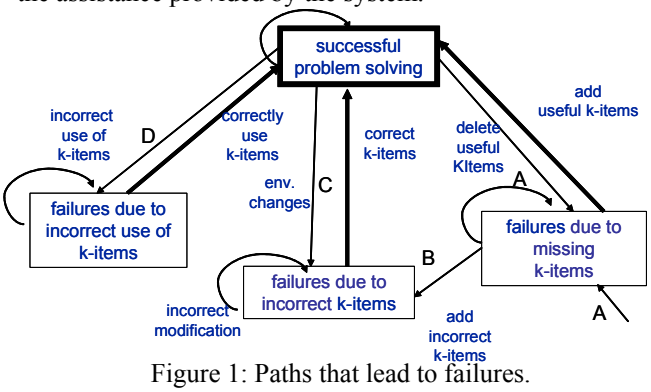

Figure 1: Paths that lead to failures.

Existing KA systems make use of these implicitly. That is, consideration of the above failures not explicit in the design and they are reflected in the implementation of the tool. Example-based validation approaches refine k-items collectively, and cannot tell 'when' and 'how' k-items can be improved and 'how' they should be used.

The above steps can be enhanced by exploiting related meta-events. In particular, we make use of predictive assessment of similar situations and assessment on dynamic changes including how k-items can be improved.

Algorithm

A. assess likeliness of potential failures due to missing k- items by collecting and relating meta-events of
- steps failed without k-item assistance
- how similar steps failed without k-item assistance
- how similar steps were assisted by k-items

and produce suggestions to create new k-items based on
- assessments on how to create k-items (see **E**)

B. assess likeliness of potential failures due to incorrect k- item definitions by collecting and relating meta-events of
- inconsistencies with the current problem solving result
- the degree of inconsistencies with valid past results
- inconsistencies with confident k-items

and produce suggestions to modify k-items based on
- assessments on how to modify k-items (see **E**)

C. assess likeliness of potential failures due to changes by collecting and relating the meta-events of
- how similar k-items were modified in similar situations
- modification in similar situations led to successes
- the user agreed to modify k-items in similar situations
- the user specified when to modify k-items

and produce suggestions to modify k-items based on
- assessments on how to modify k-items (see **E**)

D. assess likeliness of potential failures when k-items are ignored by collecting and relating the meta-events of
- consistencies between k-items and current result
- consistencies between k-items with past valid results
- inconsistent decisions resulting in failures
- consistent decisions resulting in successes

and produce suggestions to use k-items based on the assessment

E. assess how to create or modify k-items by collecting and relating the meta-events of
- incorrect definitions or modifications of similar k-items resulting in further failures
- k-item definitions and modifications resulting in successes

The above algorithm is represented by a set of declarative reflection patterns described in [Kim 2005].

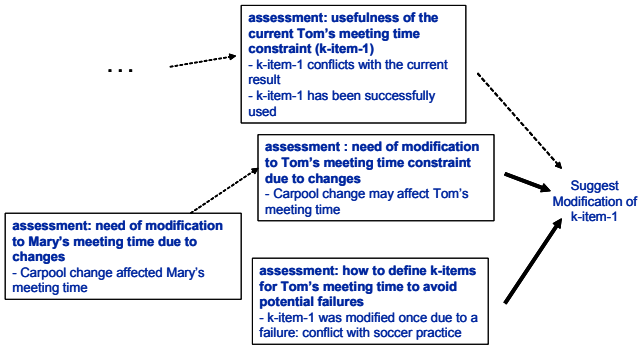

Figure 2: relations between assessments

Figure 2 shows how different assessments assist a k-item modification. In this case, Echo predicts that k-item-1 needs to be modified because it is less confident about the usefulness of the k-item (due to the current conflict) and more confident on potential changes needed The suggestions include how the system predicts the k-item can be modified based on relevant past modifications (i.e. consideration of soccer practice time).

3. Preliminary Results

We performed a preliminary evaluation of Echo with a set of synthetic scenarios with varying user interactions and mistakes. We compare two KA systems where both of them use the same episodes of problem solving and problem changes but one of them was enhanced with Echo's reflection patterns. The results show that with Echo, the KA system can reduce the number of incorrect suggestions and the number of problem solving failures.

Average per scenario	With reflection patterns	Without reflection patterns
Avg # decisions assisted with k-items(# total decisions)	24.0 (84.0)	22.03 (82.03)
Avg # wrong user decisions	8	8
Avg # additional user mistakes	2.73	5.97
Avg # k-items creation or modification	4.87	4.34
Avg # sugg (Avg # confident sugg)	13.83 (10.27)	13.0 (0)
Avg # wrong sugg	0.90	3.07
Avg # failures	4.10	6.73

Table 1. With and without reflection patterns

References

[Kim 2005] Reflection Patterns for Interactive Knowledge Capture,http://www.isi.edu/~jihie/papers/echo-RP.pdf.

[Kim & Gil 2000] Acquiring Problem-Solving Knowledge from End Users: Putting Interdependency Models to the Test. *Proceedings of AAAI-2000.*