Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), 1-4 October 2002 - Sigüenza (Spain)

TRELLIS: An Interactive Tool for Capturing Information Analysis and Decision Making

Yolanda Gil and Varun Ratnakar

USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
gil@isi.edu, varunr@isi.edu

Abstract. TRELLIS provides an interactive environment that allows users to add their observations, opinions, and conclusions as they analyze information by making semantic annotations about on-line documents. TRELLIS includes a vocabulary and markup language for semantic annotations of decisions and tradeoffs, and allows users to extend this vocabulary with domain specific terms or constructs that are useful to their particular task. To date, we have used TRELLIS with a variety of scenarios to annotate tradeoffs and decisions (e.g., military planning), organize materials (e.g., search results), analyze disagreements and controversies on a topic (e.g., intelligence analysis), and handle incomplete and conflicting information (e.g., genealogy research).

1 Introduction

In a world of overwhelming on-line information access and global communications, more and more people are asked to provide faster and more accurate answers based on up-to-date knowledge that is increasingly more disseminated in vast amounts of information sources. Research in text retrieval, extraction, and summarization, is aimed to sifting out relevant information to users [Croft 99, Cowie and Lehnert 96, Rader and McKeown 98]. Research in knowledge management and CSCW tools focuses on delivering information to interested parties in relevant formats [Smith and Farquhar 00, Ackerman & McDonald 96]. These tools can help users to manage all the information so they can make their decisions with reasonable accuracy and time bounds But users need support after they have made a decision, reached a conclusion, or made a recommendation. He or she will be often required to: 1) explain and justify their views to others, 2) update the decision in light of additional information or new data, 3) expose the intermediate products of the final recommendation to others that may be analyzing related information to make similar decisions. Our approach is to enable users to annotate the rationale for their decisions, hypotheses, and opinions as they analyze information from various sources. Once this rationale is recorded, it can be used to help users justify, update, and share the results of their analysis. This paper presents TRELLIS, an interactive tool that helps users create these annotations. TRELLIS includes a language for annotating information analysis, which can be

extended by users to suit their needs. Additional information about how TRELLIS represents and reasons about information sources can be found in [Gil and Ratnakar 2002].

The paper starts by describing our markup language to annotate information analysis, followed by an overview of the TRELLIS architecture and its functionality. We then show a use case scenario for intelligence analysis for feasibility of a special operations plan. We finalize with a discussion of contributions and plans for future work.

2 A Vocabulary to Help Users Annotate Information Analysis

The language that we propose uses the following basic components to describe this information. A *statement* is a piece of information or data relevant to an analysis, such as "The average water temperature in March is 63 degrees". A statement may have been extracted or summarized from a *document*, which is often a Web resource (text, imagery, or any other format) indicated by a URI or could also be a user-provided document such as an email message or a note relating a conversation. A statement can also be created by the user to introduce a hypothesis, conclusion, or observation that will be used in the analysis. Every document has a *source description*, describing its creator, publisher, date, format, etc. Each statement and its source can have a *source qualification* specified as a degree of *reliability* and *credibility*. Reliability is typically based on credentials and past performance. Credibility specifies the probable truth of a statement. Reliability and credibility are not the same, for example a completely reliable source may provide some information that may be judged to be not very credible given other known information.

A *compound statement or a unit* is composed of several statements related by a *construct*. Constructs reflect how individual statements are related in the analysis. For example, a causal construct is used to form the compound statement: "The average water temperature in March is 63 degrees" results in "unlikely use of divers". A *likelihood qualification* is a subjective informal indication of the analyst's reaction to a statement (or compound statement). This can indicate surprise, dismissal, saliency, accuracy, etc. A *reason* can be used to justify a compound statement, a source qualification, and a likelihood qualification.

These basic components are used to create *units*, such as the one shown in Figure 1. The basic structure of a unit is:

statement {**and** *statement*}* *construct* {**and** *statement*}*
is {not} *likelihood-qualifier* **because**
according to *source-description* **which is**
reliability-qualifier **because** *statement* **and**
credibility-qualifier **because** *statement*

The user may or may not provide all the components, only a statement is required to form a unit.

An analysis is composed of many such units. They can be *linked as subunits* of one another. Units or statements can be left with no links to the overall analysis, and in that case they can be specified as *attachments* to the analysis. This is useful to

indicate that they have been considered by the user but do not appear in the derivation of the final analysis (for lack of time, or because the analyst found better options to justify their conclusions. An analysis can be done with an overarching *purpose*, which is often a question or request that the information analyst starts with.

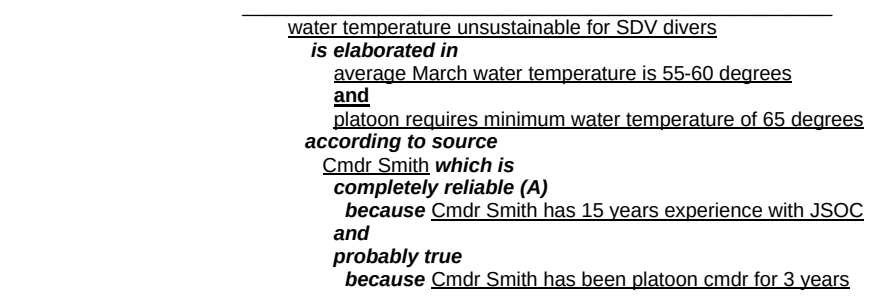

Fig. 1. An Example of a Basic Unit that Captures a Portion of the Analysis Regarding Water Temperature

We provide a default set of constructs. This default set can be extended by the user to incorporate new constructs useful in the particular topic of the analysis. Our default set of constructs is drawn from argumentation and discourse relations [Mann and Thompson 85, Pollock 94], logic connectives (drawing mostly from natural deduction, sequent calculus and tautologies), action representations (including temporal and causal relations) [Myers and Wilkins 96, Allen 84], and object representations (parts and roles) [Gruber 91]. In developing the default set of constructs, our concern was not completeness (since the user can extend the default set), nor precise semantics (since users would not necessarily be able or willing to follow the pre-specified meanings), nor computability (since, at least initially, we were not intending to automate or verify of the derivation of the analysis). Instead, our aim was to select a set of constructs that were understandable to end users and had the potential of being useful in a variety of analysis and situations. For example, to specify disjunction we did not include "or" as a construct and give it semantics, as would be done with a logic system. Instead, we included two constructs that indicate whether the disjunction is intended to be an exclusive or. The default set of constructs, grouped into three practical categories, include:

- *Discourse relations*: provides background for, depends on, stands though contradicted by, conceding, can be interpreted through, evaluated by, restates, summarizes, in contrast with, is solved by, shows how to do, is elaborated in (set and members, abstract and instances, whole and parts, process and steps, object and attributes, generalization and specialization), is motivation for, depends on, otherwise, causes, causes choice of, resulted in, choosing S1 results in S2, happened and resulted in, is purpose of.
- *Logic connectives*: not S1, S1 and S2, S1 or S2 but not both, S1 or S2 or both, S1 therefore S2, if S1 therefore S2 then not S2 therefore not S1, if S1 therefore S2 then S2, if not S1 and S1 or S2 but not both then S2, if not S2 and S1 therefore S2 then not S1.

- *Temporal relations*: is before, is after, meets, is met by, overlaps with, is overlapped by, starts, is started by, is during, contains, ends, is ended by, equals.

Users can also indicate partial knowledge by choosing from a small set of general constructs that include: related to, temporally related to, unrelated to.

We followed a similar path to design our initial set of likelihood qualifiers, drawing from modal logic [Lemmon & Scott 77]. Our initial set includes definitely true/false, probably true/false, maybe true/false, likely, impossible, surprising, shocking, reassuring, believable, absurd, accurate, dismissable, and salient.

3 TRELLIS: Capturing information analysis and decision making

Figure 2 shows the components of TRELLIS.

A user typically starts searching the Web for a certain document using the Search Tool, or indicating a pointer to a specific Web resource that contains useful information.

The Statement Editor is used to add statements about these documents. A Statement is normally a short statement, backed up by a document or user text, and by information on its source. All metadata that is allowed for the source comes from the Source Description Schema. See [Gil and Ratnakar 2002] for details.

An issue with web resources is that they are not persistent. We have found it useful to have a Caching Module to cache any online resource that is added to the system.

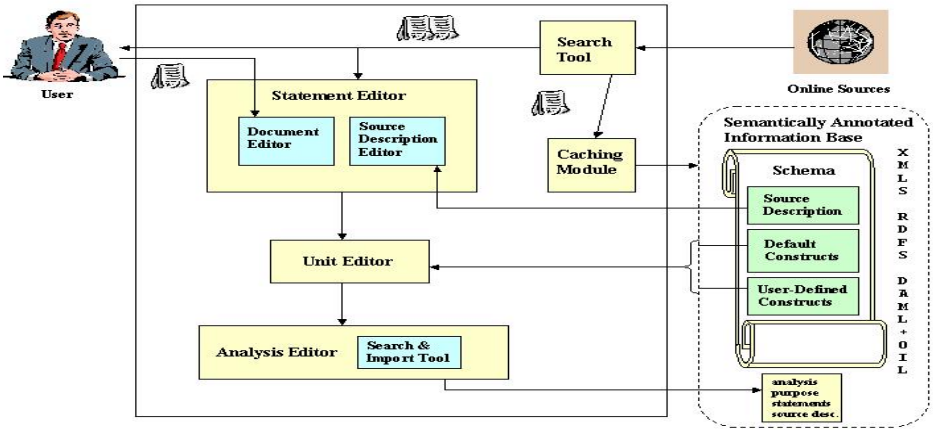

Fig. 2. The architecture of Trellis

The Unit Editor is used to add or edit units of knowledge in the system. It helps users compose statements and constructs into compound statements, and to add likelihood, reliability and credibility. These constructs are defined in the Schema. Users extend the schema when they define their own constructs specific to the domain being worked upon.

The Analysis Editor is used to organize the various units in a tree structure, which represents the reasoning pattern in reaching a conclusion for a given purpose. There is also a Search/Import utility for the Analysis Editor, which can be used to search analyses of other Trellis users for certain keywords either in the purpose or individual units. Any part of the other user's analysis can be imported into the current analysis.

TRELLIS is available on-line at trellis.semanticweb.org. In order to demonstrate the versatility and coverage of TRELLIS to annotate information analysis, we show here analyses created in a wide range of situations. Portions of the analyses discussed in this section are summarized in Figure 4. All these examples can be browsed on-line from the above Web site logging in as guest.

A genealogy example helps illustrate how TRELLIS helps annotate analysis of contradictory information. It shows how to capture an analysis of the date in which a user's family event occurred, in this case when an ancestor left Europe for the US. Another example concerns military planning and decision making. Here, a commander is trying to decide on the feasibility of using an SDV platform (Seal Delivery Vehicle) by analyzing weather conditions. Our last example captures the analysis of a user searching the Web in order to find a hotel for a trip to San Diego.

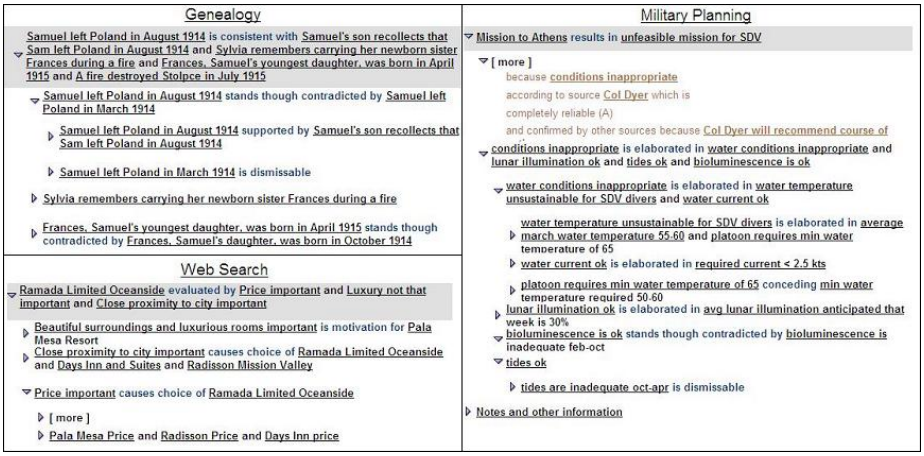

Fig. 3. Examples of Annotations in TRELLIS

5 Discussion

SEAS [Lawrence et al 01] shows an alternative approach to support intelligence analysis. Users define argument templates that contain questions to support the argument and an inference structure to derive the conclusion from the answers to the questions. The system contains a sizeable amount of patterns about early crisis warning for national security analysis. The approach emphasizes the use of shared patterns as well as support for automated inference on demand. TRELLIS has more generality but does not provide as much support for sharing and automation.

Collaboration is often supported through annotations. Web annotation and document annotation tools enable users to add commentary to documents [Koivunen & Swick 2001; Nagao et al 2001]. The emphasis of these approaches is more on collaboration, while our work has a more specific focus on information analysis for decision making. Other annotation tools provide a structured vocabulary such as the one used in TRELLIS to annotate debates and arguments [Lawrence et al 2001; Shum et al 2000]. These tools provide more ontologies and templates that the users must follow in order to enforce sharing and understanding by a specific user community.

TRELLIS provides an interactive environment that allows users to add their observations, opinions, and conclusions as they analyze information by making semantic annotations to documents and other on-line resources. This is in essence a knowledge acquisition problem, where the user is adding new knowledge to the system based on their expertise as they analyze information.

References

1. Ackerman, M.S. and McDonald, D.W. 1996. Answer Garden 2: Merging Organizational Memory with Collaborative Help. In Proceedings of CSCW'-96.
2. Allen, J. F. 1984. A General Model of Action and Time. Artificial Intelligence 23 (2).
3. Cowie, J. and Lehnert, W. 1996. Information Extraction. Communications of the ACM, 39(1): 80-91, Jan 1996.
4. Croft, W.B. 1999. Combining Approaches to Information Retrieval," in Advances in Information Retrieval: Recent Research from the CIIR, Kluwer Academic Publishers.
5. Gil, Y. and Ratnakar, V. 2002. Trusting information sources one citizen at a time, Proceedings of the First International Semantic Web Conference, Sardinia, Italy.
6. Gruber, T.R. 1991. The Role of Common ontology in achieving sharable, reusable knowledge bases. In Proceedings of the Second International Conference, Principles of Knowledge Representation and Reasoning.
7. Koivunen, M.R. and Swick, R. 2001. Metadata Based Annotation Infrastructure offers Flexibility and Extensibility for Collaborative Applications and Beyond. In: Proceedings of the K-CAP 2001 Workshop on Knowledge Markup and Semantic Annotation, BC.
8. Lawrence, J. D.; Harrison, I.W.; and Rodriguez, A. C. 2001. Capturing Analytic Thought. In Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001). Victoria, British Columbia, October 2001.
9. Lemmon, E. and Scott, D. 1977. An Introduction to Modal Logic. Oxford: Blackwell.
10. Mann, W. C. and Thompson, S. A. 1988. Rethorical Structure Theory: Toward a Functional Theory of Text Organization. Text 8(3).
11. Nagao, K., Shirai, Y., Squire, K. 2001. Semantic Annotation and Transcoding: Making Web Content More Accessible. IEEE MultiMedia 8(2): 69-81.
12. Pollock, John L. 1994. Justification and Defeat, Artificial Intelligence, 67 p. 377 – 407.
13. Radev, D. and McKeown, K. 1998. Generating natural language summaries from multiple online sources. Computational Linguistics, 1998.
14. Shum, S.B., Motta, E., Domingue, J. 2000. ScholOnto: An Ontology-Based Digital Library Server for Research Documents and Discourse. Journal on Digital Libraries, 3 (3).
15. Shum, S.B. 1996. Design Argumentation as Design Rationale. Encyclop- edia of Computer Science and Technology (M.Dekker Inc: NY).
16. Smith, R. and Farquhar, A. 2000. AI Magazine, Fall 2000.
17. Wilkins, D.E., Myers, K.L. 1995. A Common Knowledge Representation for Plan Generation and Reactive Execution, Journal of Logic and Computation, 5 p. 731-761.