The BE&E Tutorial Learning Environment (BEETLE)*

Claus Zinn, Johanna D. Moore, Mark G. Core,
Sebastian Varges, and Kaśka Porayska-Pomsta
School of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, UK
[zinn|jmoore|markc|varges|kaska]@cogsci.ed.ac.uk

Abstract

We describe the architecture and the components of BEETLE, a dialogue-enhanced Basic Electricity and Electronics Tutorial Learning Environment.

BEETLE

There is mounting evidence from cognitive science and intelligent tutoring systems research that the most effective tutors are those that prompt students to construct knowledge for themselves. Natural language dialogue offers an ideal medium for eliciting knowledge construction, via techniques such as co-construction of explanations and directed lines of reasoning. These strategies unfold over multiple turns and require a dialogue system to be flexible enough to deal with an unexpected response, interruption, or failure of tactics. To provide these capabilities, we have developed a dialogue management framework, inspired by the three-level architectures used in robotics.

Fig. 1 displays BEETLE's underlying generic and modular architecture for the management of tutorial dialogue. It is divided into four major parts (from left to right): external knowledge sources, the information state, the update engine, and the three-layered planning and execution architecture. BEETLE's architecture emphasises the importance of clearly separating the knowledge sources involved in tutorial dialogue, and thus minimises the re-representation of knowledge in different parts of the tutor system. Moreover, this design emphasises and encourages the reusability of components. Our system is composed of the following modules, which interact using the Open Agent Architecture (Martin et al., 1999):

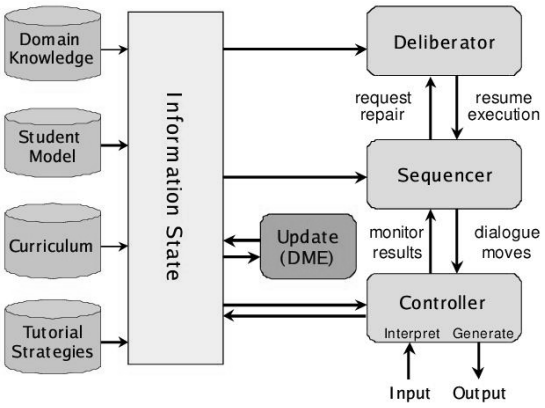

Figure 1: BEETLE's 3-level Architecture.

O-Plan, a deliberative planning and execution monitoring module implemented using the Open Planning Architecture (Currie and Tate, 1991). O-Plan is capable of performing on-the-fly plan repair when tutorial situations occur that have not been anticipated during the planning of the dialogue.

NUBEE, an NLU module which translates the user's typed English input to a logical form. This is built using the CARMEL NLU toolkit (Rosé, 2000), which includes a robust parser and a wide coverage grammar. We augmented CARMEL by

*This research is supported by Grant #N0001402IP20005 from the Office of Naval Research, Cognitive and Neural Sciences Division.

adding a reference resolution module, and an interface to WordNet (Miller, 1990) that can look for known synonyms of unknown words.

BEETLEGEN, a generation module which synthesises English text from logical forms for each system turn. BEETLEGEN is a hybrid generator combining a template-based approach with grammar-based processing within a pipeline of XML document transformations. BEETLEGEN's built-in intelligence includes reasoning about implicit speech-acts, NP and VP pronominalisation, and lexical choice. BEETLEGEN uses standard XSLT stylesheet processors (Clark, 1999).

TIS, a dialogue manager developed using the TRINDIKIT dialogue system shell (Larsson and Traum, 2000). This module maintains the system's blackboard, including the dialogue history, and encodes rules of conversation (*e.g.*, listeners are obliged to address questions). Note that knowledge of how to converse is kept separate from the planner's knowledge of how to tutor, and knowledge of the domain being tutored.

BEER, a Basic Electricity and Electronics Reasoner which encodes all of the system's knowledge about the subject matter to be tutored. BEER is able to compute answers to student questions and to simulate the execution of student lab actions. The domain reasoner is written in LOOM, a description-logic-based knowledge representation and reasoning engine (MacGregor and Bates, 1987).

CURBEE, a curriculum agent which encodes BEETLE's representation of the teaching material. The material is annotated for *learning goals, level of difficulty,* and *allocated time*.

BEESM, a situational modeller which infers relevant information about the immediate tutorial situation based on the student's interaction with the system (time elapsed, amount of material covered, student's answer correctness, *etc.*). BEESM derives values for the *autonomy* and *approval* to be given to the student, which inform (i) the tutorial planning component that selects the next appropriate tutorial strategy, and (ii) the text generator BEETLEGEN that generates the appropriate linguistic realisations.

BEEGLE, a Tcl/Tk-based GUI displaying hypertext lessons, multiple choice questions, a chat interface, and the circuit simulation environment. A screenshot of BEETLE is depicted in Fig. 2.

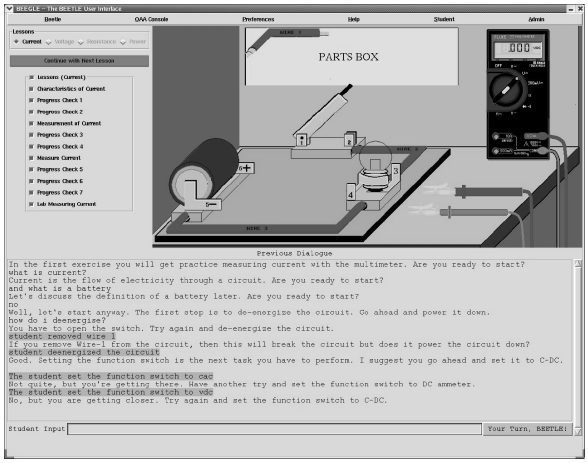

Figure 2: A Screenshot of BEETLE.

Currently, BEETLE has a series of hypertext lessons and multiple choice questions covering the topics of current, voltage, resistance, and power. The practical exercise *measuring current* is supported with natural language tutorial dialogue.

References

J. Clark. 1999. XSL Transformations (XSLT), Version 1.0, W3C Recommendation. http://www.w3.org/TR/xslt.

K. Currie and A. Tate. 1991. O-Plan: the open planning architecture. *Artificial Intelligence*, 52:49–86.

S. Larsson and D. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6(3–4):323–340.

R. MacGregor and R. Bates. 1987. The Loom knowledge representation language. Technical report, USC Information Sciences Institute, Marina del Rey.

D. Martin, A. Cheyer, and D. Moran. 1999. The Open Agent Architecture: a framework for building distributed software systems. *Applied Artificial Intelligence: An International Journal*, 13(1-2).

George Miller. 1990. WordNet: An on-line lexical database. *Intl. Journal of Lexicography*, 3(4).

Carolyn P. Rosé. 2000. A framework for robust semantic interpretation. In *Proc. of the 1st Annual Meeting of the North American Chapter of the ACL (NAACL-00), Seatle*.