On Adding Question Answering to an Adaptive Programming Language Tutor

Kate Taylor and Simon Moore

University of Cambridge Computer Laboratory, Cambridge, United Kingdom
{kate.taylor, simon.moore}@cl.cam.ac.uk
http://www.cl.cam.ac.uk/intelligent-verilog/

Our aim is to produce an e-tutor that behaves like an experienced colleague looking over your shoulder as you learn a language. Our e-tutor, the Intelligent Verilog Compiler (IVC) uses an explainer with the compiler to check syntax and explain errors. A model checker is used to check the dynamic semantics of pre-defined exercises.

The motivation for the explainer is to spot and report typographical and other errors that get in the way of understanding Verilog, whilst encouraging deep learning by not providing the answer directly. This models the way that a human supervisor leads a student towards a correct solution, and implements the concept of *deep learning*

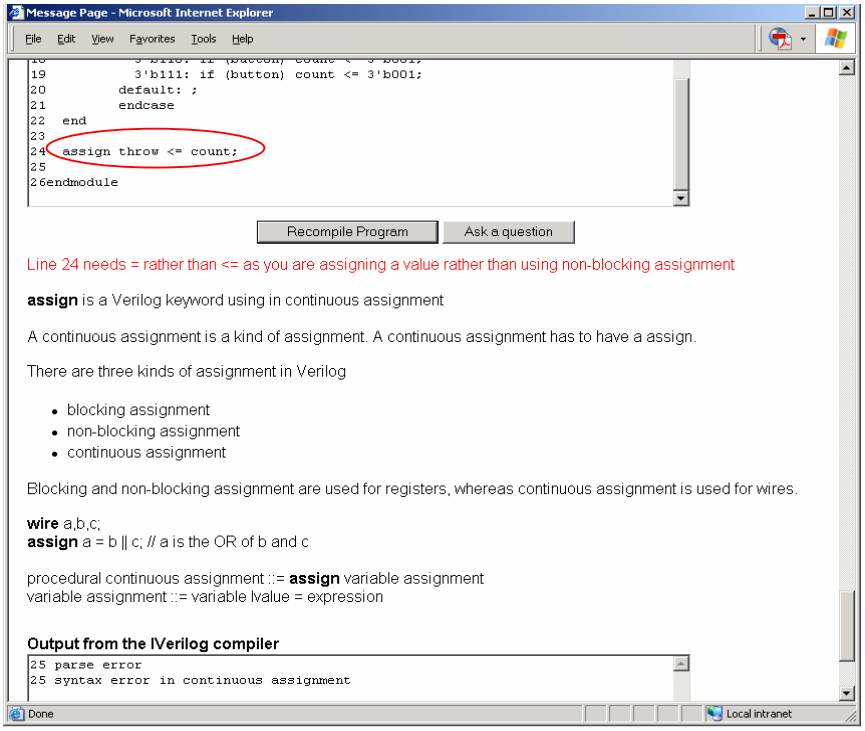

Fig. 1. The student has typed (<=) instead of (=) at line 24. The voogle component generates links back into the tutorial whilst the explainer (in red text) points out the error.

The student feedback from [1] indicates that they prefer to keep code, explanation and help in one window, resulting in a *pop-in* rather than a *pop-up* style of handling the user interface. Fig 1 shows the explanation between the code and the errors. The student is able to ask questions as well as being taught, allowing them to move the focus to what they want to ask to create a dialogue rather than a monologue. A Prolog program, **voogle** (**v**erilog g**oogle**), provides automatic generation of text from the tutorial web content from a variety of viewpoints from glossary entry to full explanation guided by an appreciation of the task that the student is trying to perform.

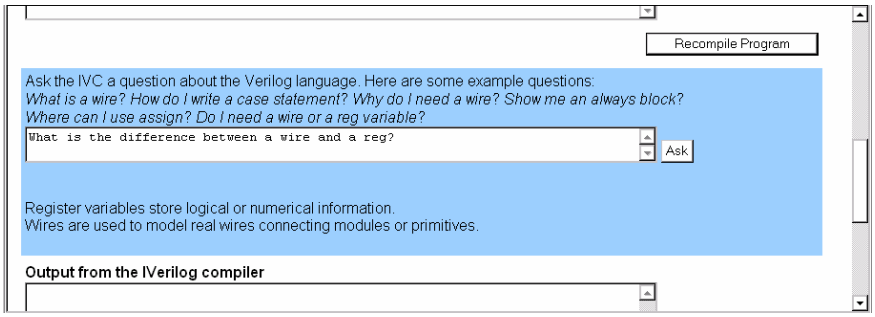

Fig. 2. The student asks a question to understand the error and explanation.

The *ask a question* button replaces the help text shown in Fig. 1 with a dialogue as shown in Fig. 2. English, Verilog keywords and symbol ontologies connected by synonyms create a richness of vocabulary to support the question answering in unrestricted English. Early results show that the question answering is being used to find an answer to the current bug, often as an alternative to reading the help provided: *"I don't want to read what you provide, I want to ask specific questions"*. Fig. 2 shows comparisons between alternatives which are mostly satisfied by glossary definitions. Questions about programming in general indicate that the ontology needs to link into other resources using the search engines Swoogle or Google. At the time of writing there is no other ontology advertised for Verilog: it is one of our aims to provide one.

IVC is a shallow system but we believe this is sufficient to help with the initial learning curve for a language as well as revision a few months later, and differing capabilities in different parts of the language to be learnt. This work was funded by the Cambridge MIT Institute (CMI) and the evaluation performed with the Centre for Applied Research in Educational Technology (CARET). We wish to thank the undergraduate and postgraduate students for their enthusiasm and commitment to this project.

References

1. Moore, S, Taylor K An Intelligent Interactive Online Tutor for Computer Languages, 25th Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence (SGAI) (2005).