Demonstration: Using GIFT to Support Students' Understanding of the UrbanSim Counter Insurgency Simulation

James. R. Segedy, John S. Kinnebrew, & Gautam Biswas

Institute of Software Integrated Systems, Department of Electrical Engineering and Computer Science, Vanderbilt University, 1025 16th Avenue South, Nashville, TN, 37212, U.S.A.
{james.segedy,john.s.kinnebrew,gautam.biswas}@vanderbilt.edu

Abstract. This paper presents our recent work with the Generalized Intelligent Framework for Tutoring (GIFT) for authoring tutors and training systems in concert with already developed external applications that provide a wide variety of educational experiences. In this paper, we describe our efforts to extend the GIFT system to develop metacognitive tutoring support for UrbanSim, a turn-based simulation environment for learning about counterinsurgency operations. We discuss specific extensions to GIFT as well as the links we have developed between GIFT and UrbanSim to track player activities. Additionally, we discuss a conversational approach that we are designing to interpret players' strategies and provide feedback when they adopt suboptimal approaches for their counter-insurgency operations.

Keywords: GIFT, UrbanSim, Scaffolding, Adaptive Support

1 Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) provides a software platform and authoring system for designing, developing, and implementing online and in-class educational programs [1-2]. An important aspect of GIFT that makes it different from a number of conventional tutoring systems is its emphasis on interoperability across a variety of existing *training applications* (TAs). The overall goals are to reduce the high design and development costs of building computer-based tutors and to increase the reusability of educational applications while also creating engaging and adaptive learning spaces that students can access as needed.

While this is a significant advantage of GIFT, it introduces challenges in the number of use cases that must be considered in order to fully leverage and develop a general framework that is compatible with different forms of available educational resources. In this paper, we present our work in exploiting the GIFT platform to develop a metacognitive tutoring environment for the UrbanSim TA [3], a counter-insurgency (COIN) command simulation developed by the Institute for Creative Technologies at the University of Southern California. We describe the steps involved in developing

adfa, p. 1, 2011.

generalized connectors that are currently tailored to support communication from UrbanSim to GIFT. Our work illustrates the flexibility of the GIFT platform to accommodate dynamic tracking of student activities in the UrbanSim COIN environment. Our overall goals are to simultaneously model student problem solving performance, behavior, and strategies, so that the developed GIFT tutor will provide dynamic support when students are involved in training episodes. Our experiences in developing GIFT to support cognitive and metacognitive tutoring lead to a set of design recommendations for further increasing the capabilities, adaptability, and flexibility of developing a variety of tutor-supported TAs with GIFT. We hope that our experiences and development efforts will help future GIFT developers working with other TAs.

2 UrbanSim

UrbanSim [3] (Figure 1) is a turn-based simulation environment in which users assume command of a COIN operation in a fictional Middle-Eastern country. Users have access to a wealth of information about the area of operation they have been assigned to. This includes: intelligence reports on key individuals, groups, and structures; information about the stability of each district and region in the area of operation; economic, military, and political ties between local groups in the region; the commanding team's current level of population support; and the team's progress in achieving six primary lines of effort. The actions that users take are scenario-specific, but they generally involve increasing the area's stability by making progress along the different lines of effort: (1) improving civil security; (2) improving governance; (3) improving economic stability; (4) strengthening the host nation's security forces; (5) developing and protecting essential services and infrastructure; and (6) gaining the trust and cooperation of the area's population.

Students conduct their operations by assigning orders to available units under their command (*e.g.*, *E CO b* and *G CO a* in Figure 1). To commit their orders, they press the *COMMIT FRAGOS* (FRAGmentary OrderS) button to complete one turn in the simulation environment. The simulation then executes the user's orders; simultaneously, it has access to a sociocultural model and complementary narrative engine that determine the actions of non-player characters in the game, which also affects the simulation results. For example, a friendly police officer may accidentally be killed during a patrol through a dangerous area. These *significant activities* and *situational reports* are communicated to the user, and the results of all activities may result in net changes to the user's population support and line of effort scores (see bottom right of Figure 1).

UrbanSim provides documentation and tutorials that should help students gain an appreciation for the challenges inherent in managing COIN operations. For example, they should learn the importance of maintaining situational awareness, managing trade-offs, and anticipating 2nd- and 3rd-order effects of their actions, especially as the game evolves [3]. They should also understand that their actions themselves produce intelligence (through their consequences as observed in the simulation environment), and, therefore, the need to continually "*learn and adapt*" in such complex domains where the available information is often overwhelming, but at the same time may be incomplete. In other words, students should realize that their decisions produce intelligence that may be critical for decision making and planning during the next set of turns. Students can learn about the effects of their actions by viewing causal graphs provided by their security officer (S2). Users who adopt strategies to better understand the area of operation and its culture by viewing and interpreting the effects of their actions using these causal graphs should progressively make better decisions in the simulation environment as the COIN scenario evolves.

Fig. 1. UrbanSim

3 Developing an Application to Connecting UrbanSim to GIFT

Connecting a TA to the GIFT environment involves creating an *interoperability interface*. This interface is responsible for reporting the actions performed in the TA (and the resulting TA state) to GIFT while also handling control messages sent by GIFT to the TA to keep the two systems in alignment. The various components and their interactions necessary for connecting UrbanSim and GIFT are shown in Figure 2. UrbanSim produces log files that include information on the actions taken by actors in UrbanSim (and the effects of those actions). To report this information to GIFT, we have authored a Java application that monitors the log files and transmits the data to the interoperability interface, which passes the information to GIFT in a predefined structured format. GIFT can then use this data to tutor the student through a web-based interface.

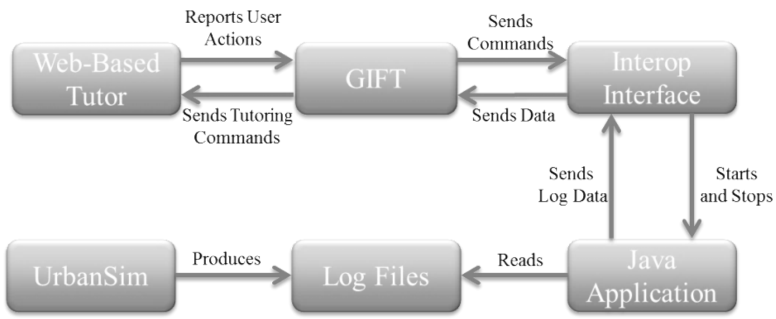

Fig. 2. Communication between GIFT and UrbanSim

The first step in developing this infrastructure required us to create the log parsing application. This involved completing the following steps:

1. Representing the complex set of data models used by UrbanSim.
2. Representing the actions taken by users and the contexts in which they occurred.
3. Monitoring the UrbanSim log directory and translating the log data into the representations created during steps 1 and 2.
4. Implementing code to establish a socket connection with the interop interface and publish the information obtained from the UrbanSim log files.

To represent the data models used by UrbanSim, we reverse engineered the plain-text save files generated by the program, extracted the data objects, their properties, and relationships to other objects and then created 22 Java classes to represent these data models. We then analyzed UrbanSim to extract the set of 38 measurable *actions* available to students in the program. Finally, we analyzed the set of 19 measurable *contexts* in which actions could occur. In this instance, a context can be considered to be equivalent to an interface configuration. For example, the configuration shown in Figure 1 shows a map of the area of operation. By tracking the actions and contexts logged by UrbanSim, we were able to create a detailed understanding of students' behaviors in the program. Once these objects had been defined, we focused on developing the algorithm for detecting changes in a log file, extracting the new information, processing it effectively, and then communicating it to the GIFT environment.

Once our log parser application had been written and tested, we turned our attention to writing the GIFT interoperability interface that would connect to the log parser, receive data, and report it to GIFT. To test this functionality, we configured a GIFT performance assessment *condition*. A condition receives data from the interoperability interface and uses it to assess a student's current level of performance with respect to a *concept*. In GIFT, a learner model is defined as a set of named *concepts* that are assessed continually while students are interacting with designated course materials. At any time, each concept may be assessed as being below, at, or above expectation. The data representation is similar to the sampling of a stream: GIFT monitors the student's task performance over time and updates the concept assessments based on the student's most recent performance. Thus, a student may perform above expectation on one concept at some point in the simulated scenario, but fall below expectation on the next turn because they missed a critical piece of information (situational awareness). A history of these assessments is maintained for feedback purposes during a particular learning session and also across multiple sessions. In the tutor we are developing for UrbanSim, the condition we created detects when a student commits their orders and then presents them with a survey through GIFT's tutor user interface, as shown in Figure 3. We expect that the data collected through this survey will provide valuable insight into how students analyze situations in UrbanSim and learn from them as the simulation progresses.

4 Design Recommendations

Our goal in the work is to develop a tutor for UrbanSim using the GIFT framework that can analyze users' understanding of the current COIN scenario, and determine what strategies the user is adopting (if any) in determining their next moves. As we have moved toward this goal, our experiences in coupling UrbanSim and GIFT by authoring a log parsing tool and implementing an interoperability plugin resulted in the following design recommendations to facilitate tutor development:

1. **Expand Instructional Triggers**: GIFT is designed such that all tutoring decisions are bound to changes in a student's concept assessments (below, at, or above expectation). This makes it difficult to author instructional interventions based on non-performance factors. For example, to configure GIFT to show the survey in Figure 3, we had to create a performance assessment condition that detected when the student committed orders and assessed the *committed orders* concept as above expectation (instead of at expectation). The survey was then triggered by a change in the assessment of the committed orders concept. It may be desirable to expand these triggers such that instructional decisions may be directly bound to elapsed time or the occurrence of an event of interest. This could lead to more straightforward authoring of such instruction.
2. **Allow for Contextualized Conversational Instruction and Assessment**: GIFT allows a course author to develop mid-lesson surveys and uses the AutoTutor Lite [4] conversations to administer instructional interventions in appropriate situations. However, the content of these surveys and conversations must be determined ahead of time and may not be parameterized by variables derived from student performance and the state of the system. For example, question 1 in Figure 3 cannot be modified to ask the student about a specific FRAGO that they just committed. Ad-

ditionally, GIFT does not allow many of these student responses on surveys and in conversations to serve as on-line assessments of their understanding (the exception is that specific answers to multiple choice questions may be linked to assessments of specific concepts). Thus, a student may, in their interactions with surveys and conversations, reveal information about their understanding that is not utilized in future GIFT interactions. Contextualized conversational feedback has been shown to positively affect learner behavior [5], and so we recommend that such feedback capabilities be incorporated into future versions of GIFT.

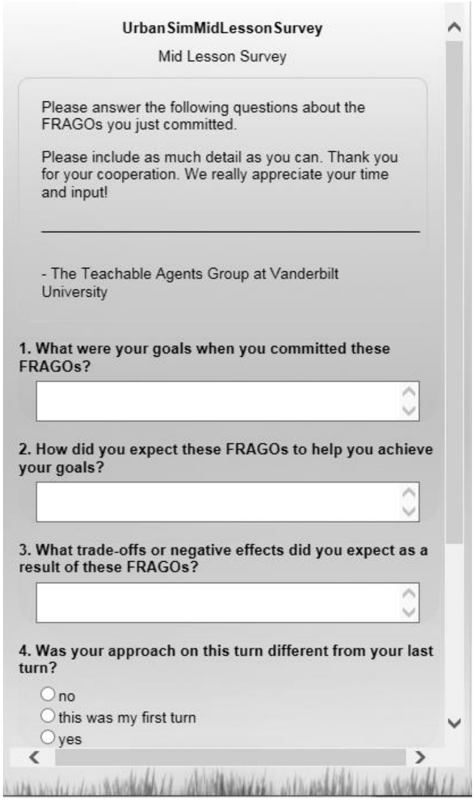

Fig. 3. UrbanSim survey presented through GIFT

3. **Expand Configurability of Dynamic Course Flow**: Currently, the primary structure of a GIFT course is fixed and specified in configuration files. Thus, even if concept assessments show that the student lacks pre-requisite skills, it is difficult to dynamically reconfigure the GIFT course to provide tutorial interventions that help the student develop that skill. In recent versions of GIFT, a system called eMAP [1] has been implemented which allows for dynamic assessment and instruction with regard to mastering a set of domain concepts. While this provides some dynamic capabilities in terms of course flow, we recommend that this system be expanded in the future. In particular, the potential of dynamic GIFT courses could be

greatly enhanced with the ability to configure additional aspects of a course or instructional intervention to adapt to the needs of learners. For example, a future version of GIFT could support dynamic flow between *multiple* training applications if a student's performance in one training application proves that they need training in pre-requisite skills before they are ready to succeed at their task.

5 Conclusions and Future Work

In this paper, we have presented our experiences in creating an application to synchronize the UrbanSim counter-insurgency command simulation with the Generalized Intelligent Framework for Tutoring (GIFT). We provided an overview of the process and potential in employing GIFT to augment a training application with new capabilities for learner modeling and support. The work presented here is part of a larger project aimed at developing metacognitive tutoring functionalities for GIFT to enhance students' future learning and problem-solving abilities. Our future work includes collecting data from students using UrbanSim, performing a systematic study of the strategies they employ and their sources of confusion, and using the insight obtained from this study to identify opportunities for providing feedback and scaffolding in our GIFT tutor for UrbanSim. A study of strategies at the cognitive and metacognitive levels may require us to build an extended task model of the COIN operations that are relevant to the UrbanSim scenario. We will also work toward implementing the design recommendations that we discussed in the previous section.

Acknowledgements. This work has been supported by an ARL research grant through funding from ONR grant #W911NF-14-2-0050. We wish to thank our ARL sponsors Drs. Robert Sottilare and Benjamin Goldberg, and Lt. Col. Kenric Smith with their help in understanding and interpreting the GIFT and UrbanSim COIN environments. We would also like to thank Nate Blomberg and Mike Hoffman from Dignitas Technologies, who provided significant technical assistance as we developed the technology necessary for connecting UrbanSim and GIFT.

References

1. Nye, B.D., Sottilare, R.A., Ragusa, C., Hoffman, M.: Defining Instructional Challenges, Strategies, and Tactics for Adaptive Intelligent Tutoring Systems. In Sottilare, R., Graesser, A., Hu, X., Goldberg, B. (eds.): Design Recommendations for Intelligent Tutoring Systems (Vol. 2). U.S. Army Research Laboratory, Orlando, FL (2014)
2. Sottilare, R., Graesser, A., Hu, X., Holden, H. (eds): Design Recommendations for Intelligent Tutoring Systems (Vol. 1). U.S. Army Research Laboratory, Orlando, FL (2013)
3. McAlinden, R., Durlach, P., Lane, H., Gordon, A., Hart, J.: UrbanSim: A game-based instructional package for conducting counterinsurgency operations. In: Proceedings of the 26th Army Science Conference, Orlando, FL (2008)
4. Hu, X., Cai, Z., Craig, S.D., Wang, T., Graesser, A.C.: AutoTutor Lite. In: Proceedings of the 14th International Conference of Artificial Intelligence in Education, pp. 802 (2009).

5. Segedy, J.R., Kinnebrew, J.S., Biswas, G.: The Effect of Contextualized Conversational Feedback in a Complex Open-Ended Learning Environment. Educational Technology Research and Development, 61, 71-89 (2013)