ERIC: an agent framework for embodied real-time intelligent commentary

Martin Strauss

May 19, 2007

1 GALA 2007 submission document

Title ERIC: an agent framework for embodied real-time intelligent commentary

Track Student

Category RaceReporter

Authors Martin Strauss
Masters Student
Informatik, Universität des Saarlandes, Germany

2 Contact data

Name Martin Strauss

e-mail martin.strauss@dfki.de

URL `http://www.martinstrauss.id.au`

Phone +49 163 146 2550

Postal address DFKI
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Germany

3 URL

`http://www.dfki.de/~mstrauss/`

4 Movie file submitted

`martin_strauss.avi`

5 Reference teacher

Name Michael Kipp

Affiliation DFKI

e-mail kipp@dfki.de

6 Process of development

The agent framework was developed between February and May 2007, as part of the author's Masters thesis.

7 Resources used

- Nuance RealSpeak Solo for TTS speech generation, `http://www.nuance.com/realspeak/solo`
- Charamel CharaVirld2 character environment, and CharaSpy debugger, `http://www.charamel.de`
- Jess: the Java Expert System Shell, `http://herzberg.ca.sandia.gov/jess`
- ALMA: A Layered Model of Affect, `http://www.dfki.de/~gebhard/alma.html`
- Eclipse development environment for Java, `http://www.eclipse.org`

8 Resources required

The agent framework was developed and tested on two PCs, both with a Pentium 4 CPU (3.2 GHz), 2 GB of RAM, and a NVidia GeForce 6800 Ultra graphics card with 256 Mb onboard memory. One of the PCS runs Windows XP Professional SP2, and the other runs SuSE Linux 9.2. In addition, the Charamel software requires a parallel port hardware lock.

Title ERIC: an agent framework for embodied real-time intelligent commentary

9 Application and Context of the work

ERIC is an intelligent embodied agent that can provide running commentary on a continuous event in real-time. In this agent, we have focused on knowledge reasoning with a world model, generating and using emotions and affect, and generating coherent natural language, synchronised with nonverbal modalities. The graphical and TTS output of the agent is provided by commercial systems.

ERIC is currently implemented to commentate a horse race provided by the RaceSim system; with minimal modification the system is configurable to provide commentary in any continuous dynamically changing environment; for example, it could commentate sports matches and computer games, or play the role of "tourist guide" during a self-guided tour of a city.

10 Novelty

ERIC is implemented as a modular framework of Java-based components, and observes a strict separation of domain-specific and domain-independent knowledge. It uses an expert system to generate a rich world model of declarative knowledge in real-time. The agent features a template-based natural language generation system capable of generating anaphora and coherent discourse, using the Centering Theory of [2]. It also uses a layered model of emotions, mood and personality to guide its output generation; these affects are generated from dynamic appraisal of events, actions and objects against goals and desires.

ERIC receives one signal from the RaceSim simulator every second containing the current location and speed of each of the four horses in the simulated race. An elaborate world model (containing facts such as "a horse is about to overtake another") is deduced from this input by an expert system implemented as rules in Jess. In order to maintain consistency over the course of a continuous event, the facts in the world model created by the knowledge inference module are all tagged with a timestamp.

In order to generate emotional responses, the agent has an affective appraisal module which uses a set of causal and belief relations to assign appraisal tags to facts in the world model based on a set of goals and desires. These tags are used by ALMA [1] to generate an affective state (an overall personality, medium-term moods, and emotion events) according to the OCC cognitive model of emotions. The resulting module is much smaller and more easily maintainable than a hash table mapping events/actions/objects to appraisal tags.

The template-based NLG in ERIC is similar to [3]; it uses semantic centers (inspired by [2]) for macro-level discourse structure, and affective state for lexical selection. A template consists of a priority (for comparison with other templates), some conditions that must be true for the template to be active, and a set of information conveyed by the template, as well as a single backward-looking and a number of forward-looking semantic centers for discourse coherence.

11 Architecture

ERIC has a highly modular architecture based on parallel rule-based reasoning engines, implemented in Java and Jess, a rule-based system itself implemented in Java. The key modules in ERIC are those concerned with knowledge reasoning (the domain model), natural language generation, and affective appraisal. As well as these, the agent has a fusion module for combining generated language, facial expressions and gestures into a form that can be sent to the Charamel character.

Throughout the architecture of ERIC, domain-specific knowledge is kept separate from domain-independent reasoning: for example, the NLG templates are kept outside the generation reasoning rules, and the goals/desires and cause/effect relations are separate from the affective appraisal rules. This results in an architecture that is highly reusable across domains.

12 Performance

In the current configuration, the CharaVirld software depicting the "Paul" agent runs on one PC, while all other components (including the RaceSim simulator) run on another, and communicate with CharaVirld via network sockets: neither PC is running at full CPU or memory capacity. No user evaluation has been performed yet.

References

[1] Gebhard, P. Alma – a layered model of affect. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)* (Utrecht, 2005), pp. 29–36.

[2] Grosz, B. J., Joshi, A. K., and Weinstein, S. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics 21*, 2 (1995), 203 –225.

[3] Theune, M., Klabbers, E., Odijk, J., de Pijper, J.-R., and Krahmer, E. From data to speech: A general approach. *Natural Language Engineering 7*, 1 (2001), 47–86.