Building a Narrative Conversational Agent using a Component-based Architecture

(Demonstration)

William Boisseleau
INSA de Rouen, France
william.boisseleau@insa-rouen.fr

Ovidiu Şerban
ISR Laboratory, University of Reading, United Kingdom
o.serban@reading.ac.uk

Alexandre Pauchet
LITIS, INSA de Rouen, France
alexandre.pauchet@insa-rouen.fr

ABSTRACT

This paper describes a component based-model for a Narrative Conversational Agent. This project has to interact with a child in a storytelling environment as natural as possible and replace the previous version of the platform, where the actions were triggered manually by a psychologist. The prototype was carried out as a student project, and had the goal to integrate existing components and libraries along with a new model for narrative dialogue interaction.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems Human**]: Human factors

Keywords

Virtual Agents, Child-Computer Interaction, Virtual Narrative Environments

1. INTRODUCTION

Designing models that handle natural human and machine interaction are particularly challenging. The main difficulty comes from the early aim of a natural virtual environment: to provide an easy way of interaction with a virtual agent, similar to the way humans interact with each other. Nevertheless, a realistic virtual conversational agent, also called Embodied Conversational Agent (ECA) [8], can increase the users' expectations up to the point that they are disappointed by the agent capabilities [7].

ECAs are autonomous and anthropomorphic user interfaces. As animated character, they tend to reproduce human behaviour in different levels of animation and communication skills, including natural language, facial expressions, gaze and gestures [1]. Recent research projects indicated that it was possible to improve the agent quality in increasing the interactiveness of ECAs and adding expressiveness and special graphic features [1, 9]. The project SEMAINE [10] or VHTookit [6] provide good examples of current capabilities of ECAs.

In this context, a study was performed on an interactive distributed platform [5], which uses also a pattern matching algorithm to detect the user's intervention. Our proposition consists in a dialogue model that integrates these interactive modules into a virtual narrative agents. The first version of the agent was first implemented as a Wizard of Oz platform, using a automata model for dialogue state transition. However, the links between steps were designed by a psychologist and triggered manually during the experiments. Our goal for this project is to rewrite several essential components of this platform and provide a semi-automatic version, in which the pilot does not have to interact much with the platform.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

2. GENERAL SCENARIO

The initial goal of this project is to create a narrative environment, in which the children could interact with a virtual agent. Our idea was to automatize as much as possible the actions done in the previous experiments by a psychologist, using AgentSlang components [5]. We choose the same story, "The Lost Ball", which is about a couple of children playing in the courtyard. Our platform has to be able to integrate narration, short interactive phrases (various questions asked by the child or the narrator) and to synchronize all of these with a couple of images, scanned from the original story. In order to make the agent more empathic and improve the understanding of the story, we integrated an Embodied Virtual Agent [4].

Different kind of input pattern were identified, namely those which permit to move on a next step of the story (contiguous step or not), those which do not permit to change step and those which are completely out of context. A "no action" from the user can also be taken as a valid pattern. When that is not the case, a counter fix the action time required. Furthermore, even though the user executes an action that is not listed in the pattern list from a specific step, the system reacts and mentions that he misunderstood, waiting for a new answer.

The system is also able to handle groups of patterns that are linked with group of answers, in order to prevent repetitive answers. The patterns are defined into the Syn!bad language [5], and they deal only with text data recovered by voice transcription or keyboard input.

3. OUR COMPONENT-BASED ARCHITECTURE

Our architecture consists in a series of AgentSlang [5] components, linked together by communication channels (the links between the components). Figure 1 describes this

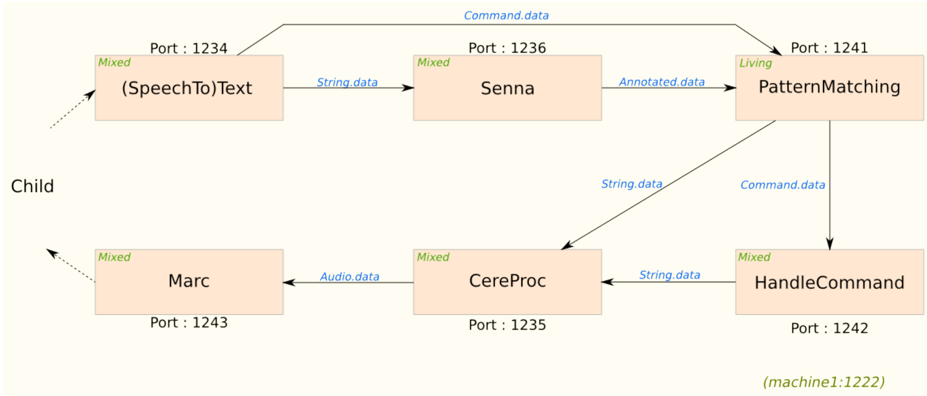

Figure 1: Narrative conversational agent architecture proposed

model, where each component is identified by a unique ID, equivalent to a TCP port for this configuration. Since AgentSlang is a generic platform for building Conversational Agents models, there are several several specific components that had to be build in the context of our project:

1. The child performs an action, which will result into some textual input. This is recovered either from a keyboard input, managed by the *Text Component*, either from a transcription component: *SpeechToText Component*.
2. The *Text Component* publishes a string which is collected by the *Senna Component* [3], which annotates it with Part-of-Speech tags.
3. The annotated data is processed by the *PatternMatching Component* that first checks if the annotation is recognized from a list of patterns. The component uses an automata model including Syn!Bad patterns [5] when the recognition process is launched. Two pattern lists previously initiated with Syn!Bad are checked iteratively, on the one hand those that are local and step specific, and on the other hand the ones that are global and fixing the out of context answers. In addition, a pattern group is linked with an answer group, a specific command, and the next step to play (that could be the current). Depending on cases, a command can be published, which will accomplish graphical or interactive tasks. But in any case, the user is boost by a string published.
4. *CereProc Component* [2] transforms the text into an audio file, which is finally read by the *MARC Component* [4]. Marc is an Embodied Conversational Agent, which used to animate certain facial and vocal expressions.

Data of each steps are stored and structured into XML files and can be maintained easily, including steps content, out of context content and commands content, so that the tool can evolve incrementally. Another script tool was also proposed in order to create scenario and generate rapidly these data.

4. CONCLUSION AND FUTURE WORK

In this paper, we presented our component-based architecture based on new and existing AgentSlang modules. The main result of our work is having a first working prototype of the platform and the development of a transition-based model for the narrative interactive agent.

There are several limitations of the current prototype. First of all, Syn!bad and the AgentSlang platform now supports multi-language patterns, but at the time of the development only English was supported. Second, the current scenario and the interactive states are very limited that could be gradually enlarged. Last, our agent does not have any memory model, so we are studying the possibility of adding it.

Acknowledgments

This demonstration is a part of the NARECA project, funded by the ANR (ANR-13-CORD-0015).

5. REFERENCES

[1] Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsson, H., and Yan, H. Embodied conversational agents. MIT Press, 2000, ch. Human conversation as a system framework: designing embodied conversational agents, pp. 29–63.

[2] CereProc. Cerevoice sdk. `http://www.cereproc.com/`.

[3] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research 12* (2011), 2493–2537.

[4] Courgeon, M., Martin, J., and Jacquemin, C. Marc: a multimodal affective and reactive character. In *Proceeding of Workshop on Affective Interaction on Natural Environment* (2008).

[5] Şerban, O., and Pauchet, A. Agentslang: A fast and reliable platform for distributed interactive systems. In *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on* (2013), IEEE, pp. 35–42.

[6] Hartholt, A., Traum, D., Marsella, S. C., Shapiro, A., Stratou, G., Leuski, A., Morency, L.-P., and Gratch, J. All together now: Introducing the virtual human toolkit. In *International Conference on Intelligent Virtual Humans* (Edinburgh, UK, Aug. 2013).

[7] Mori, M. The uncanny valley. *Energy 7*, 4 (1970), 33–35.

[8] Ogan, A., Finkelstein, S., Mayfield, E., D'Adamo, C., Matsuda, N., and Cassell, J. Oh dear stacy!: social interaction, elaboration, and learning with teachable agents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 39–48.

[9] Pelachaud, C. Modelling multimodal expression of emotion in a virtual agent. *Philosophical Trans. of the Royal Society B: Biological Sciences 364*, 1535 (2009).

[10] Schröder, M. The SEMAINE API: towards a standards-based framework for building emotion-oriented systems. *Advances in HCI 2010* (2010), 2–2.