A Common Ground for Virtual Humans:

Using an Ontology in a Natural Language Oriented Virtual Human Architecture

Arno Hartholt, Thomas Russ, David Traum, Eduard Hovy, Susan Robinson

University of Southern California's Institute for Creative Technologies & Information Sciences Institute
Marina del Rey, CA 90292
United States of America
{hartholt, traum, robinson}@ict.usc.edu & {tar, hovy}@isi.edu

Abstract

When dealing with large, distributed systems that use state-of-the-art components, there invariably is a mismatch in how these components internally represent concepts and the communications of these representations to other components. Without due care, distributed resources' representations can get out of synch, contain localized errors, or become manageable only by a small group of experts for each module. In this paper, we describe the use of an ontology as part of a distributed virtual human architecture in order to enable better communication between modules while improving the overall flexibility needed to change or extend the system. We focus on the natural language understanding capabilities of this architecture and the relationship between language and concepts within the entire system in general and the ontology in particular.

Introduction

Designers of large heterogeneous systems (such as task-oriented communicating agents) have an uncomfortable choice to make regarding their knowledge representations: should they choose a uniform representation for all modules that enforces common understanding and re-use, or should they allow each module to use its own representation, tailored specifically for that module. In either case, there are a number of difficult and perhaps insoluble problems. In the former case, it may be very difficult to decide which representation to use, given the different demands of such diverse processes as planning, perception in a real or virtual world, and natural language dialogue. Should one choose an impoverished language for which one can guarantee fast algorithmic complexity, or an overly rich language that has expressive capacity closer to that of natural language? On the other hand, if each module is free to choose its own language, how does one convert the necessary elements from one representation to another? How does one insure that the overlap in capacities is sufficient and faithful translation to the degree required is even possible?

In this paper we suggest that a middle-ground is indeed possible, in which a multi-phase project lifecycle can achieve the advantages of each approach while minimizing the disadvantages. In the early stages of the project, the best strategy is to allow each module designer to choose the representation language best suited for the state of the art in that area, while developing inter-process communication languages to bridge the gap. As understanding of the relationships and requirements are better understood, one can bring the languages closer together. Finally, one needs appropriate tools both within each module and across modules to make modification and creation of new domains easier and possible without specific work by the designers of each module.

We illustrate these points through our experiences with the Virtual Human Project at the University of Southern California (USC), which has built virtual agents for the Mission Rehearsal Exercise (MRE) (Rickel at al., 2001) and Stability And Support Operations – Simulation and Training (SASO-ST) (Swartout et al., 2006).

The Virtual Human Project

The Virtual Humans Project, at USC's Institute for Creative Technologies (ICT) and Information Sciences Institute (ISI), has the main goal of designing autonomous agents that support face-to-face interaction with people in many roles and in a variety of tasks. The agents must be embedded in the virtual world and perceive events in that world, as well as actions of human participants. They must represent aspects of the dynamic situation in sufficient depth to plan contingencies, develop beliefs and intentions, and form appropriate emotional reactions. They must communicate with each other and with human participants using multi-modal natural language communication.

Our latest scenario, an extension of SASO-ST, includes two virtual humans: a Spanish doctor and an Iraqi village elder. Set in a small Iraqi town plagued by violence, the human trainee takes on the role of an US Army captain with orders to move the doctor's clinic to a safer location (figure 1).

Figure 1: SASO-EN Scenario

Virtual Human Architecture

The Virtual Human Architecture includes a set of distributed modules that communicate through messages. These modules include:

- A task reasoner that can plan how to achieve goals and reason about alternatives and utilities of various actions (Gratch & Rickel)
- An emotion module that appraises the state of the world in relation to beliefs and goals resulting in emotion and specific coping strategies (Gratch & Marsella, 2003)
- A perception module, which updates internal state based on visual processing
- Natural language (NL) processing modules that can relate natural language to internal reasoning including:
    - Speech recognition (ASR), converting vocalizations into words, prosody, and emotional expression
    - Natural language understanding (NLU), converting unconstrained natural language expressions to internal representations
    - Dialogue management (DM), which relates the NLU output to the context of previous conversation and other internal state, updates the internal state, and plans new communications
    - Natural language generation (NLG), which converts internal communication goals to output text
- A body controller module, consisting of the following parts:
    - A non-verbal behavior generator, which decides which body movements should be performed in order to convey appropriate meaning of NLG output, emotions, perception and conversational regulation. (Lee & Marsella, 2006)
    - A behavior blending system, SmartBody, which takes directives for motions and allocates resources (Thiebaux 2007)
    - A speech synthesizer (TTS)
- The virtual and real environment.

An overview of the Virtual Human Architecture can be seen in figure 2. The cognitive agent is based on SOAR. Other modules are in Java and C++. For a more in-depth discussion of the general architecture and some of its application, see (Kenny et al, 2007).

Representation Languages and Knowledge Resources

There are a number of different types of resources needed and used in different ways by the modules. Before the introduction of the ontology, all these resources were managed by hand.

The task reasoner reasons primarily about states, actions and the relationships between them, but not about their internal structure. On the other hand, NL processing modules must represent a finer grain of detail to be able to capture the meanings of referring expressions and partial matches (e.g. questions). We adapt an attribute value representation for both kinds of knowledge, with designated “slots” linking objects to values of attributes. There are different ways in which this kind of knowledge is integrated in the system, including:

- SOAR productions that directly create objects and links as part of SOAR’s working memory
- TCL macros, that take in arbitrary argument structures and create a set of SOAR productions
- Task model, a set of actions with states as preconditions and effects
- NL “frames”, containing an action or state with added linguistic information for both the NLU and NLG.

Creating a consistent specification for NLU frames has been the most challenging level of concept matching between components. The specification must fit well with the NLU training method and is constrained by the formalisms and conceptual model of the Dialogue Manager and task model. Yet it should be rich enough to capture the intended meaning of the user’s utterances.

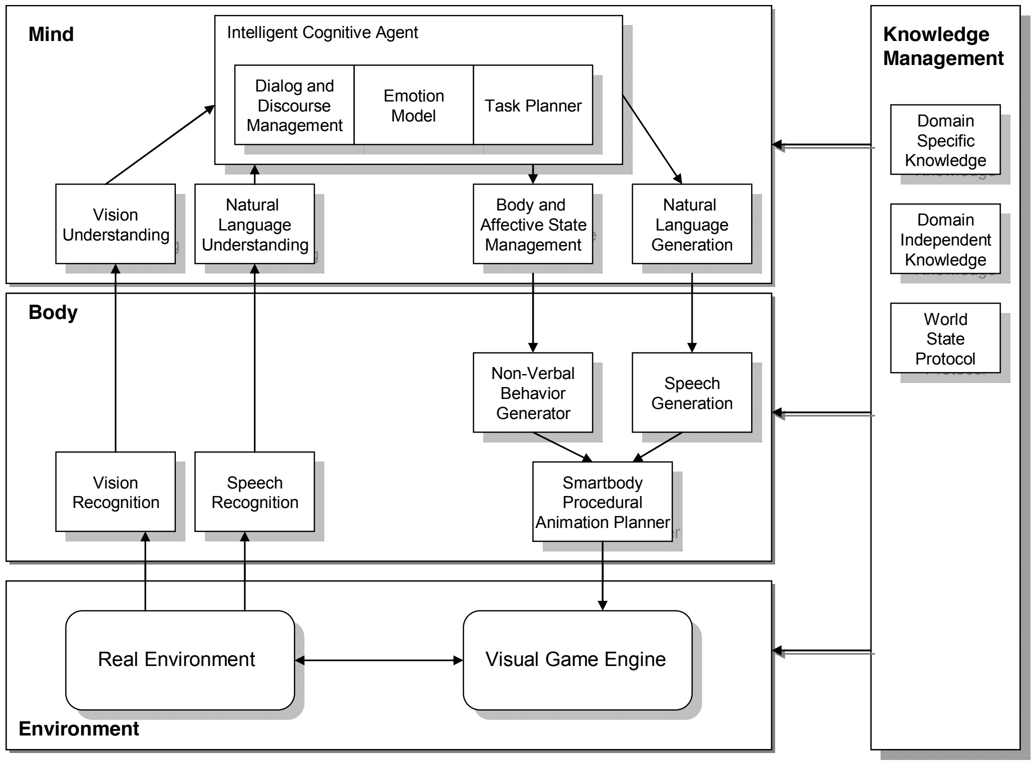

Figure 2: Virtual Human Architecture

Ontology

General

The main modules that now use the ontology are the task model and the NLU. We have two iterations of our ontology and use Stanford's Protégé (Knublauch 2004) to manage both. For our first iteration we choose to use Protégé Frames, as this lies conceptually close to the existing data sources and does not have the overhead that OWL brings in. Our philosophy was to create an ontology that did not require many modifications to the existing system. This version gave us the benefit of integrated data sources and created the necessary experience needed to leverage all the benefits an ontology can give.

The goal for the second version of our ontology was the re-use of knowledge and the introduction of a more principled ontology design. We switched the representation language to OWL in order to enable us to use inference procedures for automatically classifying instances and most crucially because it allowed us to institute a hierarchical structure of domain independent and domain dependent concepts. Instituting a principled design of the ontology meant making changes to existing representations of the system.

Structure

The world ontology is structured to provide a domain-independent set of concepts that can be specialized and instantiated at the domain-specific level. The domain independent part of the ontology defines for instance entities such as military officers; specific actors like our captain are then defined at the domain-specific level.

Instantiated actions and states exist at the domain-specific level in a basic form. These instances are used by both the task model and the NLU frames who add module specific information to them, like relations and linguistic information. This ensures consistency between modules and enables re-use of knowledge

In the OWL ontology, we introduced the notion of *generic actions* that include templated descriptions of the preconditions and effects. The templates use values from case roles on a particular action and states of the fillers to realize the constraints needed by the agent's planning code.

For example, the domain independent "Move" generic action defines effects such as adding "the *theme* is at the *destination*." that are later instantiated for our scenario. This type of reasoning goes beyond the standard OWL inference capabilities and required the construction of our own template interpretation code.

Procedures, Interfaces & Tools

We have created specialized interface elements for the Protégé editor (tabs) to support the generation of our output code that is directly used by the system modules. Meta-concepts are used to enable easier generation of code.

Related Work

Mikrokosmos (Mikrokosmos) and FrameNet (Ruppenhofer, et al. 2006) are large, broad-coverage ontologies based on linguistic principles. Our ontology, in contrast, is more focused on the particular domain, and is anchored to a semantic representation that the agents understand.

Our use of templated generic actions is similar to the Parameterized Action Representation (Badler et al., 1998; Bindiganavale et al. 2000). Our underlying representation is tied to a different agent control system, and the contents of the templates are filled in by instantiation from the ontology rather than as being used as a means of communication between users and the agents in question.

The Smartkom Project (Porzel et al) is inspirational in its use of an ontology to solve a number of natural language processing issues for a system including a virtual character and several simple command tasks.

Future Work

We plan to integrate more components with the ontology, most notably the NLG and SmartBody. This involves extending the knowledge base with concepts from the virtual environment and the development of a rich lexicon. The ultimate goal is to tie together all the information that different modules use about a single concept.

In addition, we are extending the tools both in terms of interfacing with the system as well as for creating and tweaking scenarios. These will ultimately result in a suite of authoring tools as an additional layer on top of Protégé. We are also investigating whether we can leverage the use of external recourses, like WordNet and existing ontologies.

So far we have mostly discussed the use of an ontology to facilitate cross-module information sharing and internal representational consistency. But a centralized point of entry to the representation and editor also bring about the benefits of greatly improving the speed of adding new knowledge and other extensions. This potentially enables non-experts to change and extend the systems themselves. Including an ontology into an existing system can thus be seen as a step on our Virtual Human's maturation process from research pilot system to prototype to eventually, possibly, a distributable system. In this task we face the challenge of determining the optimal tradeoff point between system simplification and complexity. In USC's Virtual Human Project, the ontology and associated framework provide a rich context for investigating this challenge.

References

Gratch, J., Rickel, J., André, E., Badler, N., Cassell, J., Petajan, E., 2002. Creating Interactive Virtual Humans: Some Assembly Required, *IEEE Intelligent Systems,* July/August, 54-63

Gratch, J. and S. Marsella (2004). "A domain independent framework for modeling emotion." *Journal of Cognitive Systems Research 5(4): 269-306*.

Patrick Kenny, Arno Hartholt, Jonathan Gratch, William Swartout, David Traum, Stacy Marsella, Diane Piepol, Building Interactive Virtual Humans for Training Environments, 2007, *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*

Knublauch, H., Fergerson, R. W., Noy, N. F., Musen, M. A. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. Third International Semantic Web Conference, Hiroshima, Japan, 2004.

Lee, J., Marsella, S: Nonverbal Behavior Generator for Embodied Conversational Agents. *6th International Conference on Intelligent Virtual Agents*, pp 243-255, Marina del Rey, CA. (2006).

Rickel, J., Gratch, J., Hill, R., Marsella, S., Swartout, W., 2001. Steve Goes to Bosnia: Towards a New Generation of Virtual Humans for Interactive Experiences. *In AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Stanford* University, CA

Swartout, W., Gratch, J., Hill, R., Hovy, E., Marsella, S., Rickel J., Traum, D., 2006. Toward Virtual Humans, *AI Magazine,* v.27(1)

Thiebaux, M., Marshall, A., Marsella, S., Fast, E., Hill, A., Kallmann, M., Kenny, P., Lee, J., SmartBody: Behavior Realization for Embodied Conversational Agents, *Submitted to IVA07*, (2007), Paris, France

William Swartout, 2006, Virtual Humans, *Twenty-First National Conference on Artificial Intelligence (AAAI-06). (Senior paper)* Boston, MA

N. Badler and R. Bindiganavale and J. Bourne and M. Palmer and J. Shi and W. Schuler, "A Parameterized Action Representation for Virtual Human Agents", *Workshop on Embodied Conversational Characters*, Lake Tahoe, CA, Oct. 1998.

N. Badler and R. Bindiganavale and J. Bourne and M. Palmer and J. Shi and W. Schuler, "A Parameterized Action Representation for Virtual Human Agents", *Workshop on Embodied Conversational Characters*, Lake Tahoe, CA, Oct. 1998.

Bindiganavale, R., Schuler, W., Allbeck, J., Badler, N., Joshi, A., and Palmer, M., Dynamically Altering Agent Behaviors using Natural Language Instructions. In *Proceedings of Autonomous Agents 2000*, pp. 293–300., August, 2000

Ruppenhofer, J., Ellsworth, M., Petruck, M. R. L., Johnson, C. R., and Scheffszyk, J., *FrameNet II: Extended Theory and Practice*, version 1.3, (Berkeley FrameNet Project, University of California, Berkeley) 2006.