Evaluation of an Integrated Authoring Tool for Building Advanced Question-Answering Characters

Sudeep Gandhe\(^1\), Michael Rushforth\(^2\), Priti Aggarwal\(^1\), David Traum\(^1\)

\(^1\)USC Institute for Creative Technologies, 12015 Waterfront Drive, Playa Vista, CA 90094, USA
\(^2\)Department of Modern Languages and Literatures, University of Texas at San Antonio, USA

\{gandhe,aggarwal,traum\}@ict.usc.edu, michael.rushforth@utsa.edu

Abstract

We present the evaluation of an integrated authoring tool for rapid prototyping of dialogue systems. These dialogue systems are designed to support virtual humans engaging in advanced question-answering dialogues, such as for training tactical questioning skills. The tool was designed to help non-experts, who may have little or no knowledge of linguistics or computer science, build virtual characters that can play the role of an interviewee. The tool has been successfully used by several different non-experts to create a number of virtual characters used successfully for both training and human subjects testing. We report on experiences with seven such characters, whose development time was as little as two weeks including concept development and a round of user testing.

Index Terms: Virtual Human, Dialogue system, Authoring tool evaluation.

1. Introduction

Building a dialogue system can be a time-consuming and costly process. It requires system developers to have expertise not only in the domain of interaction but also in fields like computer science and linguistics. A variety of architectures have been proposed for building dialogue systems (e.g., [1, 2, 3, 4, 5, 6]) and the choice of the architecture is influenced by the specific goals for the dialogue system and its evaluation criteria. This chosen architecture, in turn, determines what specific types of resources are required to build the dialogue system. Once the dialogue system architecture and the required resources have been determined, the cost of developing a dialogue system can be lowered by reducing the cost of building those specific resources. Our goal in developing the DomainEditor authoring environment described in [7] is to reduce this cost and enable rapid prototyping of dialogue systems, by allowing non-experts to build such resources with the help of an integrated authoring tool. By non-experts we mean, designers with little or no experience in building dialogue systems and little or no background in computational linguistics. These non-experts might be experts in the specific domain of interaction (e.g., Tactical Questioning).

The DomainEditor authoring tool is especially suited to developing characters for complex question-answering dialogues. Unlike the architecture in [5, 8], DomainEditor is well suited to cases in which the character reasons about topics and decides under which conditions to provide an accurate answer or other alternatives such as lying or bargaining for release of the information. Interviewers can employ several strategies, such as building rapport, offering to perform certain favorable actions or pointing out the effects of non-cooperation, in order to persuade the character to cooperate and answer truthfully.

Building Tactical Questioning characters has been an ongoing project at Institute for Creative Technologies. The project has evolved through many different architectures for dialogue systems [9]. Gandhe et al. [4] provide the description of the latest architecture for the tactical questioning dialogue system and Gandhe et al. [7] provide a detailed description of the integrated authoring tool, DomainEditor, that was designed to be used by non-experts.

In this paper we report on the use of DomainEditor over the past two years and our experience in getting non-experts to build seven virtual human characters. In the next section we give a brief overview of the authoring process followed by the overview of dialogue manager functionalities available within the architecture. In section 3, we present the evaluation of the authoring process, the simple dialogue act (DA) scheme designed for tactical questioning and the resulting virtual human dialogue systems.

2. Architecture and Authoring Process

We use a standard pipeline architecture for our virtual human dialogue system (see [4] for more details). It has modules for speech recognition (ASR), natural language understanding and generation (NLU, NLG), Dialogue management (DM), non-verbal behavior generation [10] and speech (TTS) and behavior [11] synthesis.

We use an information-state [6] based dialogue manager that uses dialogue acts (DAs) for reasoning about communicative effects and communication with NLU and NLG modules. This information-state is in part based on conversational game theory [12]. The main responsibilities of the dialogue manager are to update the information state of the dialogue based on the incoming DA from NLU and to select the response DA for NLG. The information state update rules describe grammars for conversational game structure and are written as state charts. We are using State Chart XML\(^1\) (SCXML), a W3C working draft, for describing the state charts. These state charts model various subdialogues like question-answering, offer, threat, greetings, closings, etc. The dialogue manager also implements advanced features like topic-tracking and grounding [13]. See Figure 1 for a sample dialogue.

The resources required for this architecture are the domain knowledge for the virtual character, a set of relevant dialogue acts and examples of NL text for all the DAs. These examples can be used as training data for ASR, NLU and NLG. DomainEditor (GUI shown in Figure 2) supports both top-down and bottom-up authoring approaches.

\(^1\)http://www.w3.org/TR/scxml/. Specifically we use the commons SCXML implementation. [http://commons.apache.org/scxml]
The soldier’s name is Ryan Benton.

My name is PFC Sean Avery.

He was carrying a puppy.

The suspicious package contained a puppy.

If you don’t tell me your privileges will be revoked.

I don’t want to be on restriction.

Who was the soldier?

Tell me more about the soldier.

The PFC’s name is Ryan Benton.

Tell me what you saw.

The soldier got out of the car and took a package from the back seat.

Do you know who the soldier was?

I have no clue what the soldier’s name is.

If you don’t tell me your privileges will be revoked.

I don’t want to be on restriction.

Who was the soldier?

Tell me more about the soldier.

The PFC is in my platoon and one of my closest friends.

What was inside the package?

He was carrying a puppy.

What was inside the package?

As I said earlier, I don’t want to be on restriction.

The suspicious package contained a puppy.

Whose idea was it to smuggle the puppy on the base.

Will you give the soldier a less severe punishment if I talk?

That’s comforting, I wouldn’t want him to get thrown off base or anything.

I think the soldier told his girlfriend it wouldn’t be a big deal to sneak a puppy on base.

Figure 1: A sample dialogue with Avery character. Player refers to the human trainee.

Figure 2: DomainEditor: An Integrated Authoring tool for designing the conversational domain, and specifying the utterances that map to various dialogue acts.

Working in a top-down fashion, the authoring process begins with specifying a domain of knowledge for the character. The basic unit of domain knowledge is an \(<object, attribute, value>\) triple. DomainEditor automatically generates all relevant DAs following a dialogue act schema. The default schema was authored by expert dialogue system designers specifically for tactical questioning dialogue systems. It can be easily tailored to add different types of DAs for other kinds of virtual humans. Each DA has a detailed XML representation and a pseudo-natural language gloss that is generated using templates. E.g. A template like “Attribute of Object is Value” is used for the event dialogue act type. Scenario authors then generate examples of natural language texts whose meaning is represented by these DAs.

This top-down approach can be augmented with a bottom-up approach. Once the dialogue system has been built, the designers can collect a dialogue corpus by having human subjects interview the virtual human character. The collected corpus can then be annotated with the most appropriate DAs. For some utterances this may require expanding the character’s domain of knowledge. DomainEditor provides these necessary utilities while ensuring consistency and completeness. Here consistency refers to generating only the valid DAs that can be correctly handled by the dialogue manager and completeness refers to generating all the DAs that are relevant with respect to the domain knowledge of the character.

3. Evaluation

3.1. Non-experts can author Dialogue Systems

DomainEditor has been used for creating several tactical questioning characters (viz. Hassan, Amani, Ali Sadat, Sean Avery) as well as other non-tactical question-answering characters (viz. Victor, Amber, Bradley). Figure 3 shows a list of these characters along with the information about their authors, corresponding scenarios and the amount of dialogue system resources that were collected.

Hassan was implemented in previous architectures and was ported to the new architecture as the authoring tool was being developed. The rest of the seven characters were authored by non-experts. Amani was initially developed by a non-expert as a tactical questioning character. We conducted a pilot user testing for this character as well as user testing at U.S. Military Academy (USMA) Westpoint where we had access to the target users for tactical questioning systems. A total of 34 cadets interviewed Amani twice and practiced their interviewing skills. This corpus allowed us to identify and fix the deficiencies in our initial dialogue act schema [16, 17]. Sean and Avery represent two scenarios involving the same character PFC Sean Avery. These scenarios were developed by same author over 3 months. The Avery scenario is more complex, including dialogue policies such as deciding on whether or not to lie about certain information based on what has happened in the dialogue. Ali Sadat was developed by a USMA cadet, who used his subject matter expertise to effectively circumscribe the character’s domain knowledge. One of the guidelines for tactical questioning is to fill out a SALUTE (Size, Activity, Location, Uniform, Time, Equipment) report [18]. Such structure for an interview helps define the domain of interaction rapidly.

Besides tactical questioning, our tool has been used by psychology researchers to build question-answering characters which can be used in their experimental methodologies. These virtual characters provide a consistent experience compared to human confederates and can be controlled precisely by the system designer. Victor and Amber are two such characters that were developed to teach how to use verbal cues for deception detection. They can answer questions truthfully or deceptively.
depending on the mode in which they are being operated. DomainEditor is well suited for creating such characters. A total of 35 participants interacted with these two characters [14]. For this study, the input interface was typed text with an optional multiple-choice between suggested similar questions while the virtual humans responded with speech performed by animated bodies. Bradley was another such character designed to study social influence of humor [15]. A total of 54 participants had conversations with either a humorous or non-humorous versions of Bradley using typed text interface for both input and output.

Figure 4 shows the authoring progress of four such characters which were developed by non-experts during summer 2010. This shows that non-experts can use the authoring tool to build virtual human dialogue systems in a small amount of time. The ease with which a domain of interaction can be defined affects development time. In fact, Ali Sadat was developed in only 2 weeks.

The authoring process for these characters has two phases. The first phase begins with a top-down process which includes defining the character’s domain knowledge first and then authoring the NL texts for all relevant dialogue acts. The growth in the number of dialogue acts represents the growth in character’s domain knowledge. As can be seen from figure 4, the domain reaches a stable level relatively early. Most of the domain authoring occurs during this phase. Scenario designers author one or two utterances for each of the character’s DAs for some variability. Substantially more examples are authored for player DAs in order to ensure robust NLU performance. The second phase is a bottom-up phase which involves collecting a dialogue corpus by having volunteers interview the virtual human character that has been built. The utterances from this corpus can then be annotated with the most appropriate dialogue act. It can be seen that this second phase is responsible for a rapid growth in player utterances. It can also lead to minor domain expansion and a small increase in character utterances.

### 3.2. Evaluating the Dialogue Act scheme

Since DomainEditor only allows utterances to be annotated with a DA that has been automatically generated, and dialogue act specification is an expert-task, the chosen dialogue act scheme could be a limiting factor in system development. To ver-
Figure 4: Amount of dialogue system resources collected across time for 4 characters that were authored by non-experts.

<table>
<thead>
<tr>
<th>Corpus</th>
<th># utts</th>
<th># player DAs</th>
<th>Reliability α</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot study</td>
<td>224</td>
<td>113</td>
<td>0.49</td>
<td>0.38</td>
</tr>
<tr>
<td>Westpoint (original)</td>
<td>768</td>
<td>143</td>
<td>0.49</td>
<td>0.33</td>
</tr>
<tr>
<td>Westpoint (expanded)</td>
<td>799</td>
<td>287</td>
<td>0.63</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 1: A summary of DA annotation reliability (Krippendorf’s α) and domain coverage at different developmental stages for Amani.


4. Acknowledgements

This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). The content does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred. We would like to thank all of the authors who used the DomainEditor to create characters, as well as the other members of the TACQ team who helped design the architecture.

5. References


