The Spoken Dialogue Challenge XML format

Blaise Thomson and Helen Hastie (October 2010)

1. Introduction

This document proposes a standardised XML format for storing logs of human-machine spoken dialogues. Current evaluation of different spoken dialogue systems is often hampered by the wide variation in logging formats used by different systems. This document describes a proposed logging structure to be used in the Spoken Dialogue Challenge (SDC), so that the effects of different formats can be minimized. It builds on XML logging formats previously developed for the EU projects: CLASSiC (Lemon et al., 2009) and TALK (http://www.talk-project.org).

2. The directory structure

Each dialogue is stored in a separate directory. The contents of this directory should include the following files:

- session.xml: The xml file describing the dialogue (format described below).
- {separated}.wav: Segmented audio recordings of the user's speech. Each of these files records a section of speech as segmented by the system's speech detection software. Filenames are arbitrary.
- {full}.wav: Audio recording of the complete, unsegmented, audio of the user's speech. Filenames are arbitrary.
- transcription.xml: Extended xml file with any transcriptions / evaluation tags added.

Additional files may also be included in the directory. In particular, many systems may wish to include segmented / unsegmented recordings of the system's speech.

3. The data format

All log files are stored in a simple xml format, see example below.

The attributes will be of four different types:

Type	Denoted by
String	s
Float	f
Date	d
Integer	i

All times should be given in seconds from the start of the dialogue. Below is the log outline.

```
<dialog>
        <header>
                <!-- Some header info -->
        </header>

        <turn speaker=”s” turn_number=”i”>
                <!-- Info about what was said in a given turn -->
        </turn>  <!-- One of these turn tags for each utterance of system or user -->

        <dialogue_rec speaker=”s” fname=”s”/>
        <!-- The filename of the recording of the completed dialogue, multiple tags
             are allowed, e.g. for speaker=”user”, speaker=”system” -->
</dialog>
```

4. Basic Tags

This section describes the basic set of tags that will be minimally required by SDC, other tags may be included as needed by the system designers, suggestions for these are given in the Section 5. Each table provides the required sub-tags for each of the main tags given in the above example. The header must store:

<header>	
Host name (the computer that ran the dialog)	<host>
Date and time the dialogue started (with time zone information)	<date>
System name	<system>
Version	<version>

The contents of the <turn> tag, will depend on whether the speaker is the system or a user. Each system turn must support the following sub-tags, note that system specific dialogue act format may be in either text or xml form. Turn numbers will indicate number of turns for the user/system separately, i.e. user turns will increment after each user turn.

<turn speaker=”system” turn_number=”x”>	
The dialog act	<dialogue_act> System specific format </dialogue_act>
The system prompt, as output to the user	<text> Text of prompt </text>

Each user turn must contain the sub-tags in the following table. Providing the n-best list of ASR hypotheses is important, particularly for statistical dialogue management (Thomson and Young, 2010). User hang-up is stated here, other errors may also need to be logged (for example VXML errors).

<turn speaker=”user” turn_number=”x”>	
The filename of the user utterance recording, with start and end times	<rec fname=”s” starttime="f" endtime="f">
The ASR as a list of ASR hypotheses with confidence scores. In the case of a system which output only 1-best, the xml must still be represented as a list of 1.	<asr> <hypothesis p=”f”> text </hypothesis> </asr>
The semantic input as a list of semantic hypotheses	<slu> <interpretation p=”f”> dialogue act </interpretation> </slu>
User hang-ups	<hangup/>

The transcribed data is exactly the same as the logging files except that some additional tags are added in the user turns:

<slu_transcription>, <asr_transcription>	
Any number of speech transcriptions along with the person who transcribed it. Date must include time zone	<asr_transcription transcriber=”s” date=”d”> text </asr_transcription>
Any number of dialog act transcriptions along with the person who transcribed it. Date must include time zone.	<slu_transcription transcriber=”s” date=”d”> dialogue act </slu_transcription>

5. Optional Tags

In the header, the developer may wish to log evaluation metrics for easy searching and categorisation of dialogues into, for example, good/bad and long/short dialogues. Many of these evaluation tags would typically be added in the transcription.xml file.

<header>	
Evaluation metrics, e.g. number of turns, task success, user satisfaction, reward score	<evaluation num_turns=”i” task_success=”s” user_sat=”s” score=”f” />
Source channel (e.g. voip / headset)	<input_source type=”s”/>

Barge-in may be included, options for time stamping include the time the TTS stopped and/or the time the asr started.

<turn>	
User barge-in, resulting in halted system speech.	<barge-in tts_time=”f” asr_time=”f”/>

The system turn may also support the optional recording of system prompts and optional TTS costs (Boiden et al. 2009).

<turn speaker=”system”>	
The filename of the recording, with start and end times	<rec fname=”s” starttime="f" endtime="f"/>
TTS cost attribute to the <text> tag	<text cost=”f”/>

The optional <user_model> tag would supply a probability for each of the SLU hypotheses (n=”x”). Depending on the exact user model, it may require access to the system action in the preceding turn and the state hypotheses at the preceding turn, as well as the current SLU hypotheses (Lemon et al., 2009).

<user_model>	
User	<interpretation_probability n="i" p="f"/>

The NLG component may require logging of detailed information (Rieser et al., 2010) such as the communicative goal of the utterance (e.g. offering a bus time) and content planning including the dialogue act type to be generated with certain values such as bus time, departure and the destination locations. Finally, it would log the alternate text realizations.

<nlg communicative_goal=”x”>	
Content plan contains dialogue act type with key values which can be more than one.	<content_plan> <dialogue_act type=”s”/> <key_value key="s"/>.. </dialogue_act> </content_plan>
Realizer output	<realizer_outputs> <output> text1 </output> <output>text2</output> </realizer_outputs>

The set of items in the database matching the user’s current constraints, and their attributes/slot-values can be logged. For example, it would list all the bus numbers that match the user’s constraints.

<database_hits hits=”x”>	
Database item	<item> <key_value key=”s” value=”s”/> </item>

The <user_preference_model> tag lists dispositional preferences where order of the <preference> tags is important. An example of this is that the user would prefer a bus that is coming sooner from a location nearby rather than one later at their exact location.

<user_preference_model>	
Preference	<preference name="s"/>

6. Example Transcribed Dialogue with Basic Tags

```
<dialogue>
        <header>
                <host> bermuda.eng.cam.ac.uk </host>
                <date>  25/05/10 10:51:94 BST </date>
                <system>BUDS</system>
                <version>2.1</version>
        </header>

        <turn speaker=”system” turn_number=”1”>
                <dialogue_act> hello() </dialogue_act>
                <text> Welcome to the Lets Go bus system. How may I help you? </text>
        </turn>

        <turn speaker=”user” turn_number=”1”>
            <rec fname=”001.wav” starttime=”2.21” endtime=”4.79”/>
            <asr>
                <hypothesis p=0.8> I want to go to Forbes and Murray </hypothesis>
                <hypothesis p=0.1> I want to go  Forbes and Murray </hypothesis>
            </asr>
            <slu>
                <interpretation p=0.9> inform(tostop=”Forbes,Murray”) </interpretation>
            </slu>
            <asr_transcription transcriber=”Jamie” date=”30/05/10 10:53:20 BST”>
                I want to go to Forbes and Murray
            </asr_transcription>
            <slu_transcription transcriber=”Kate” date=”02/06/10 08:45:52 BST”>
                inform(tostop=”Forbes,Murray”)
            </slu_transcription>
        </turn>

        <dialogue_rec speaker=”user” fname=”dialogue_user.wav”/>
</dialogue>
```

References

Cedric Boidin and Verena Rieser and Lonneke van der Plas and Oliver Lemon and Jonathan Chevelu. “Predicting how it sounds: Re-ranking dialogue prompts based on TTS quality for adaptive Spoken Dialogue Systems”, in *Proceedings of the Interspeech Special Session: Machine Learning for Adaptivity in Spoken Dialogue*, 2009.

Oliver Lemon, Olivier Pietquin, Herve´, Frezza-Buet Verena Rieser, Xingkun Liu, Philippe Bretier, Steve Young and James Henderson. "Shared Context Model (XML Schema)", *CLASSiC project Deliverable* D3.1, http://www.classic-project.org/deliverables, 2009.

Verena Rieser, Oliver Lemon and Xingkun Liu. "Optimising Information Presentation for Spoken Dialogue Systems", in *Proceedings of ACL,* 2010.

Blaise Thomson and Steve Young. "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems", in *Computer Speech and Language*, 24(4), p.562-588, 2010.