The NITE XML Toolkit Meets the ICSI Meeting Corpus: Import, Annotation, and Browsing*

Jean Carletta and Jonathan Kilgour

University of Edinburgh, HCRC Language Technology Group, 2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, UK jeanc,jonathan@inf.ed.ac.uk

Abstract. The NITE XML Toolkit (NXT) provides library support for working with multimodal language corpora. We describe work in progress to explore its potential for the AMI project by applying it to the ICSI Meeting Corpus. We discuss converting existing data into the NXT data format; using NXT's query facility to explore the corpus; hand-annotation and automatic indexing; and the integration of data obtained by applying NXT-external processes such as parsers. Finally, we describe use of NXT as a meeting browser itself, and how it can be used to integrate other browser components.

1 Introduction

The AMI project is developing meeting browsing technology. This requires multimodal data to be captured and annotated for a wide range of properties that are of use to the browser, reflecting not just low level properties like who is speaking, gesturing, or standing up when, but also properties that are less easy to read directly off signal, such as when the group has reached a decision and what the decision was. Here the intention is to hand-annotate the information required and then to use that to derive similar annotation automatically, either via statistical modelling, symbolic processing based on the understanding of the data that the hand-annotation affords, or a combination of the two. Finally, the automatically derived annotations must be made accessible to the browser. Data annotation and search are therefore central to the browsing technology.

In this paper, we describe work in progress to explore the possible uses of the NITE XML Toolkit [1, 2] within the AMI project. NXT, although it is relatively new, has been successfully deployed to annotate and search a range of data sets, and many of the uses relate to AMI concerns. Although the main concern for its users is to publish their core research, there are some reports about how NXT contributed to the work that are in the public domain [3, 4]. Taken together, the existing uses suggest that NXT could be helpful for AMI work not just in NXT's core areas of hand-annotation and data exploration, but for other purposes as well. During the initial stages of the project, several partners are prototyping

* This work was carried out under funding from the European Commission (AMI, FP6-506811).

their methods using the ICSI Meeting Corpus, which differs from the proposed AMI data in several ways (not least in being audio-only) but has the advantage of being available now. We discuss converting the existing data into the NXT data format; using NXT's query facility to explore the corpus; hand-annotation and automatic indexing; and the integration of data obtained by applying NXT-external processes such as parsers. Finally, we describe use of NXT as a meeting browser itself, and how it can be used to integrate other browser components.

2 The ICSI Meeting Corpus

The ICSI Meeting Corpus [5], available from the Linguistic Data Consortium (catalog number LDC2004S02) is a corpus of meetings recorded at the International Computer Science Institute in Berkeley. The data set consists of 75 natural meetings from ICSI's own research groups, recorded using both close-talking and far field microphones. Transcription is also available (LDC2004T04) and is given in XML files, one per meeting, where the meeting is divided into timestamped segments that contain the words as textual content. Segments change at utterance boundaries (that is, whenever there is a major pause for one speaker) and otherwise whenever the transcribers found it convenient to insert a timestamp. There is sometimes an intermediate tag between the words and their containing segments in order to indicate some quality of importance to speech recognition, such as emphasis. Such tags are also used to indicate places where the transcriber was uncertain of the correct transcription, where foreign language phrases are used and where pronounciation diverges significantly from normal or expected pronounciation variations. There are also the usual tags interspersed among the words, familiar from the Text Encoding Initiative [6], for indicating the placement of things like pauses and nonvocal and vocal sounds, and for transcriber comments. The textual transcription itself also contains punctuation, including left- and right-handed quotation marks. This includes the use of hyphens to punctuate the moment of interruption in disfluencies, and underscores occur after capitalized letters to indicate that the letter name has been pronounced. Otherwise mixed case and punctuation in the transcription are used at the transcriber's discretion and tend to mirror written English.

In addition to this base data, several sites have annotated it for various types of information, with more types of annotation planned. So far, we have worked with two of the existing annotations. The first of these is dialogue act annotation using the MRDA tagset [7]. The basic dialogue act annotation gives start and end times, speaker, and the words for each tag. The SRI speech recognizer has been used to produce a forced alignment of the words in each act, and so the dialogue act annotation also contains the start and end times for each word. Meanwhile, hot spot annotation [8] simply gives start and end times for sections of the meeting during which the participants were unusually animated. Both the dialogue act and hot spot annotation are distributed separately from the main corpus, the former in a comma-separated value format and the latter in a tab-delimited format.

3 The NITE XML Toolkit

The core of NXT consists of two types of functionality: routines that load, access, manipulate, and save data according to a particular data model; and an engine that evaluates queries expressed in NXT's Query Language (NQL). Several groups plan library support with similar functionality, of which the Atlas project is perhaps the closest in style [9]. The furthest developed of these is the Annotation Graph Toolkit (AGTK) [10]. NXT differs from AGTK in two ways. First, its data model and query language are oriented towards those users who build descriptive analyses of heavily cross-annotated multimodal corpora in preparation for defining appropriate statistical models, and therefore it allows easy access to an expressive range of data relationships, at the expense of processing speed. Second, it supplements the data and query facilities with library routines for building displays and interfaces based on Java Swing. The libraries include a default top level interface that allows one to choose an observation (in this case, a meeting) and a tool to run on it from those registered in the corpus metadata; audio and video players; a search interface for running queries and displaying the results; basic display layouts such as text areas and trees that synchronize with the data and with the media and search facilities; and standard utilities for things like opening an observation and saving some annotation. These libraries are intended to make it possible to build tailored end user tools at low cost. There is also a default data display that is never as good as a tailored one but at least allows any corpus in the correct format to be viewed and searched without further programming.

4 Conversion to NXT Data Format

The first step in testing NXT on the ICSI Meeting Corpus was to transform the data into NXT's stand-off XML storage format. NXT represents the data for one meeting as a related set of XML files, with a "metadata" file that expresses information about the structure and location of the files. For corpora such as this one that contain timing information, NXT stores data for each participant separately. This is because the data in one NXT file must follow a strict temporal order, but speech from different speakers can overlap. Relationships among files are represented using stand-off links that reference individual elements using a filename and id; in one of the possible link syntaxes ranges can be expressed, with the meaning that the relationship holds from the first to the last element in the range. Figure 1 gives an extract across the files for words, dialogue acts, and speech quality tags for one agent in one meeting in order to show that both the links and the individual files are quite simple in structure.

4.1 Orthography

Since the orthography was already in an XML format, up-translation was relatively easy. We first divided the orthographically transcribed segments from the

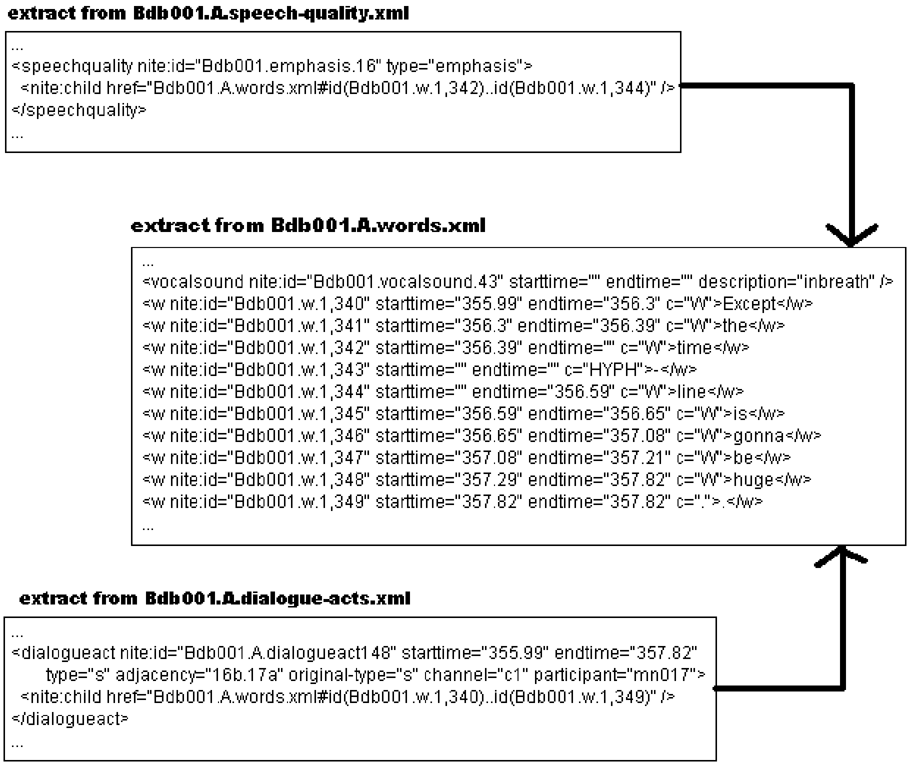

Fig. 1. Example file and link structure.

original by speaker. Although we wished to preserve the original segmentation, it was felt that most new annotations should point either directly to words or to new, more theoretically motivated segments. For this reason, the old segments were pulled out into a separate file that applications could load or fail to load at will. Words are represented in their own flat file along with other transcription elements such as silences and anchors that represent the placement of disfluencies. In the original, speech quality tags (such as those for emphasis) came between segments and transcription elements; this is inconvenient for many kinds of processing because these tags only occur sporadically. For this reason, we pulled these tags out into a separate file that points to words independently of the original containing segments.

NXT's references between files are based on XML element IDs. This means that in order to have the segments and speech quality tags reference words, we needed to tokenize the transcription. In order to do this we altered a preexisting tokenization algorithm [11] to take into account some of the conventions of the ICSI transcription. Our tokenization attempts to encode the information given by syntactic conventions used in the ICSI data in a systematic way. For example we aim to provide a word token containing the string "ICSI" whether or not the letters are pronounced separately or they are pronounced together as "Icksy". Information about the pronounciation is encoded in an attribute value. Our tokenization is designed to split words into the minimal units that might be required should we wish to apply natural language processing the data, so for example hyphenated words are normally split into their constituent parts with a "HYPH" token between them.

4.2 Dialogue Acts and Hot Spots

Although the file format for the dialogue act annotation is simple, up-translation was nontrivial because we needed to transfer the dialogue acts onto the top of the newly tokenized orthography, preserving both the timestamping from the former and the non-word content from the latter. This was made more complex by the fact that the dialogue acts did not cover all of the spoken words, in particular where the transcription was not taken from close-talking microphones, and in some relatively rare cases where the forced alignment between transcript and audio signals failed. There is one part of the ICSI meeting data that is routinely transcribed but deliberately does not form a part of the dialogue acts: the "Digits Task" where meeting participants read out strings of digits. For an individual speaker, words in the orthography and in the acts came in the same order, but this was the only property the input data guaranteed. The algorithm that we devised traversed the two inputs in parallel, running through the orthography until it found a match for the words to the next act. It then considered the left and right context up to the immediately preceding and following words in order to decide whether or not to assimilate non- word content into the move. In the left context, we assimilated any preceding left-handed quotation mark. In the right context, content was assimilated up until the last punctuation (excluding left-handed quotation marks) before the following word.

We have one side comment about the process of developing a strategy for translating the dialogue act annotation: it was easier to know what to do once we understood what the data contained, and NXT was itself useful for this purpose. Our first step in the translation process was to transform the dialogue act annotation into a rival segmentation and transcription in NXT format and load both it and the original into a simple NXT display. This allowed us compare the versions visually; where the alignment was not obvious we were able to use the search highlighting to find which segments overlapped which dialogue acts. We could also search, e.g., for segments that did not overlap with any dialogue act and for words from the orthography that did not match any words with compatible timings from the dialogue acts.

The hot spot annotation was trivial to translate because it only consists of tags with start and end times that are independent of any other data. We transformed it into a separate XML file containing a flat list of tags.

5 Hand Annotation

One of the strengths of NXT's library-based approach to GUI support is that it allows new tools that support new kinds of hand-annotation to be built. Our current priorities for this data set are a kind of hierarchical topic segmentation and a variant of Bales' Interaction Process Analysis [12, 13]. A preliminary screenshot for the former is shown in figure 2. NXT has been used for a range of other annotations that may also be of interest on this data, including coreference, named entities, the linking of gestures to deictic language, and an n-way classification of utterances by the type of information they contain in preparation for argument-based summarization.

Although NXT's libraries have some facilities for editing annotation times, they are designed primarily for adding structural information, such as that required for linguistic annotation, over the top of some existing time- stamped segmentation of the data, such as orthographic or gestural transcription. Thus we expect time-aligned data to be constructed using some other tool as input to NXT. This could include transcription tools like The Transcriber [14], time-aligned coding tools such as The Observer [15], Anvil [16], or TASX [17], or automatic derivation from signal.

6 Automatic Annotation

NXT provides a facility for automatic indexing based on query language matches for queries expressed in NQL. This is primarily for caching intermediate results of complicated queries that will be needed again, although it is also useful for adding theoretically motivated constructs to the data. For instance, one project has employed this facility in order to identify "markables", or the sorts of entities which contribute to information structure and to coreferential relationships, based on prior syntactic annotation [3]. This facility may be of use within AMI, especially for positing higher level structures from annotations that are closer

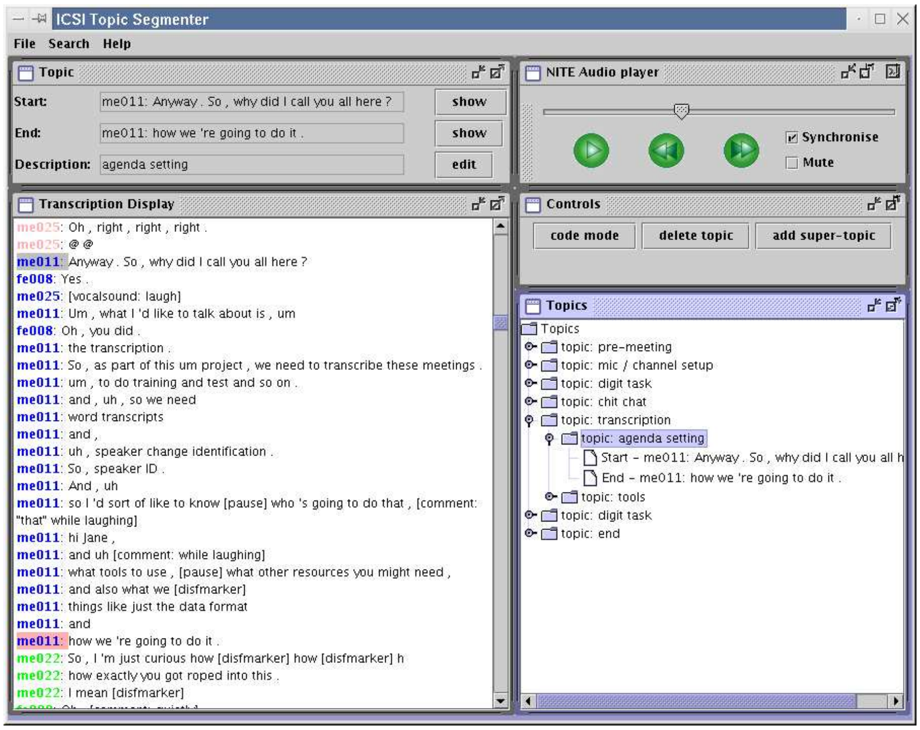

Fig. 2. A tool for hierarchical topic segmentation built using NXT.

to signal. Simple queries in NQL express variable bindings for n-tuples of objects, optionally constrained by type, and give a set of conditions on the n-tuples combined with boolean operators. The defined operators for the condition tests allow full access to the timing and structural relationships of the data. A complex query facility passes variable bindings from one query to another for filtering, returning a tree structure. To give the flavour of the query language, consider the following example.

```
($a w):(TEXT($a) ~ /th.*/)::
   ($s speechquality):($s ^ $a) && ($s@type="emphasis")
```

On this data set, this query finds instances of words starting with "th", and then for each finds instances of speech quality tags of type emphasis that dominate the word, discarding any words that are not dominated by at least one such tag in the process. The query returns a tree structured index into the data with words at the first level down from the root and the emphasis tags below them. Return values can be saved in NXT format, in which case they can be co-loaded with the data set itself. In general, queries provide a very flexible mechanism for data processing.

NQL has its limits, both in terms of what can be expressed in it and the speed at which queries are applied, and it will of course be convenient to add annotation by other means. Because NXT stores data in a standard format and because it divides an entire data set into several files that are in themselves quite simple, this is relatively easy to do. Some natural language processing tools, such as those available from [18] for part of speech tagging and chunking, work natively on XML data. Many useful annotations, such as word counts and durations for segments, can be added using a simple XSLT stylesheet. It is often useful to supplement this approach with fsgmatch, an XML transduction mechanism that provides good support for regular expressions, and xmlperl, a stylesheet language that has access to perl programming statements. Both of these are also available from [18]. Recourse to non-XML processes requires translation to and from the required format, although it is sometimes possible within xmlperl to set up communication with the external process so that the stylesheet transforms information into the right format, streams it externally, accepts the output back, and reintegrates it. Another useful technique is passing XML ids through the external process as an extra field so that output can be spliced into the original XML data. However an annotation has been added, the metadata must be modified to reflect the change, a process that in distributed data annotation requires careful management.

Because of the way in which NXT breaks up a data set into files, it is often possible to send single files for external processing. For when this is not the case, we are currently developing a number of utilities that create new XML documents with tree structures that draw elements from across a data set using the stand-off pointers and that explode a tree back into the correct component structures, complete with any modifications.

7 Browser Component Integration

NXT widgets have two useful properties built in: registration with a central clock that allows a set of signal and data displays to synchronize with each other, and the highlighting of query results on a data display. These two properties are exactly what is needed for a meeting browser, since together they allow users to navigate around the meeting by time or by properties of interest, albeit ones currently expressed in a rather formal language. Figure 3 shows the screenshot of a very simple NXT-based browser. When the user evaluates a query and selects part of the result, the corresponding part of the data display is highlighted in grey. As one plays the audio signal, the data corresponding to the current time is highlighted in green. It is possible to play extracts corresponding to particular annotations, although due to limitations in the libraries underlying the current NXT implementation, the start and end times of the extracts are approximate. Whatever kinds of annotations a data set contains, a browser can be built for them. An Edinburgh student project, for instance, has used NXT to compare a meeting transcription with extractive summaries derived from it using several different approaches.

Anything can be an NXT data display with time and search highlighting as long as it implements the interfaces that NXT widgets themselves use to provide these properties. This raises the interesting prospect that integrating a new component into a meeting browser could be a matter of implementing these interfaces – that is, that AMI meeting browsers could themselves be NXT applications.

One AMI project partner has gone partway to proving that the most important part of this approach, time synchronization, will work by demonstrating NXT with a radically different kind of display than the text-based ones that come in the interface library. The demonstration is of coding head orientation by manipulating a coffee cup instrumented with a ”flock-of-birds” motion sensor, where the cup's handle is a convenient stand-in for the nose. An on-screen display based on the Java 3D graphics libraries provides feedback whilst coding or for replay. As the video plays, the coder captures and writes new annotations for head orientation every frame. The partner reports that implementing the TimeHandler interface that makes the system work was simple to do [19].

Performance will be an issue with this approach if NXT is used unadulterated, especially when the browser is used to look at a series of meetings. Streaming data in a radically non-tree-structured data model such as NXT's is still a research issue, and NXT's design is predicated on the assumption that graphical user interfaces will only be required for a limited amount of data at a time. We believe this difficulty can be overcome by loading the individual XML files that make up an NXT data set incrementally and building external indices into the data model for the properties of most importance to the browser. Integrating non-Java components is theoretically possible but has not yet been attempted, nor has the interface that enables search highlighting been implemented for anything apart from textual displays. Thus this approach to brower

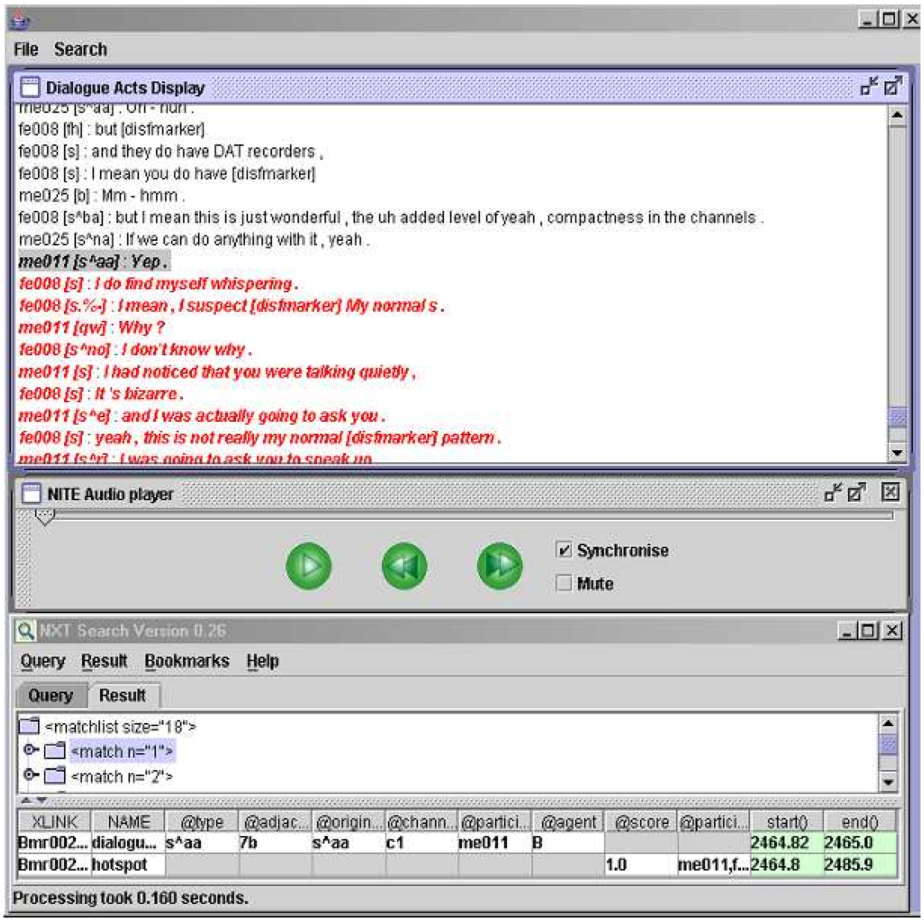

Fig. 3. A simple NXT-based meeting browser that places one dialogue act per line and shows hotspots italicized and in red, with a user search highlighting one dialogue act by speaker me011 during a hotspot.

component integration looks promising, but proving that it is useful requires future work.

8 Conclusions

Translating the basic ICSI Meeting Corpus plus other existing annotations for it into NXT format has been immediately useful in allowing us to understand the data and add annotations to it by hand. It has also provoked an interesting thought exercise in how NXT could be used for some kinds of automatic annotation that are necessary for browsing meetings, as well as providing the mechanism by which different meeting browser components could be integrated.

References

1. Carletta, J., Kilgour, J., Evert, S., Heid, U., Chen, Y.: The NITE XML Toolkit: data handling and search. (submitted for publication)
2. Carletta, J., Evert, S., Heid, U., Kilgour, J., Robertson, J., Voormann, H.: The NITE XML Toolkit: flexible annotation for multi-modal language data. Behavior Research Methods, Instruments, and Computers **35** (2003) 353–363
3. Carletta, J., Dingare, S., Nissim, M., Nikitina, T.: Using the NITE XML Toolkit on the Switchboard Corpus to study syntactic choice: a case study. In: Fourth Language Resources and Evaluation Conference, Lisbon, Portugal (2004)
4. Heid, U., Voormann, H., Milde, J.T., Gut, U., Erk, K., Pad, S.: Querying both time-aligned and hierarchical corpora with NXT Search. In: Fourth Language Resources and Evaluation Conference, Lisbon, Portugal (2004)
5. Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., Wooters, C.: The ICSI Meeting Corpus. In: ICASSP, Hong Kong (2003)
6. TEI Consortium: TEI: The Text Encoding Initiative (n.d.) http://www.tei-c.org/; accessed 26 May, 2003.
7. Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., Carvey, H.: The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In: HLT-NAACL SIGDIAL Workshop, Boston (2004)
8. Wrede, B., Shriberg, E.: Spotting “hot spots” in meetings: Human judgements and prosodic cues. In: EUROSPEECH, Geneva (2003)
9. National Institute of Standards, Technology: ATLAS Project (2000) http://www.nist.gov/speech/atlas/; last update 6 Feb 2003; accessed 1 Mar 2004.
10. Linguistic Data Consortium: AGTK: Annotation Graph Toolkit (n.d.) http://agtk.sourceforge.net/; accessed 1 Mar 2004.
11. Grover, C., Matheson, C., Mikheev, A., Moens, M.: LT TTT - a flexible tokenisation tool. In: Second International Conference on Language Resources and Evaluation (LREC 2000). Volume 2. (2000) 1147–1154
12. Bales, R.F.: Social Interaction Systems: Theory and Measurement. Transaction Publishers (1999)
13. Bales, R.F.: Interaction Process Analysis: A method for the study of small groups. Addison-Wesley, Cambridge, MA (1951)

14. Barras, C., Geoffrois, E., Wu, Z., Liberman, M.: Transcriber: development and use of a tool for assisting speech corpora production. Speech Communication **33** (2000) special issue, Speech Annotation and Corpus Tools.
15. Noldus, L., Trienes, R., Hendriksen, A., Jansen, H., Jansen, R.: The Observer Video-Pro: new software for the collection, management, and presentation of time-structured data from videotapes and digital media files. Behavior Research Methods, Instruments & Computers **32** (2000) 197–206
16. Kipp, M.: Anvil - a generic annotation tool for multimodal dialogue. In: Seventh European Conference on Speech Communication and Technology (EUROSPEECH), Aalborg (2001) 1367–1370
17. Milde, J.T., Gut, U.: The TASX-environment: an XML-based corpus database for time aligned language data. In Bird, S., Buneman, P., Liberman, M., eds.: Proceedings of the IRCS Workshop on Linguistic Databases. University of Pennsylvania, Philadelphia (2001) 174–180 Anvil look-alike; open source.
18. University of Edinburgh Language Technology Group: LTG Software (n.d.) http://www.ltg.ed.ac.uk/software/; accessed 1 Mar 2004.
19. Reidsma, D. personal communication, (2004) 11 March.