TOWARDS A MINIMAL STANDARD FOR DIALOGUE TRANSCRIPTS: A NEW SGML ARCHITECTURE FOR THE HCRC MAP TASK CORPUS

Amy Isard, David McKelvie, & Henry S. Thompson

Human Communication Research Centre,
University of Edinburgh, Scotland
email: `[amyi,dmck,ht]@cogsci.ed.ac.uk`

ABSTRACT

The rapid growth in availability of high-quality recordings of natural spoken dialogue (and natural spoken material more generally) has encouraged us to to improve the interchange of transcripts of such material, in order that these resources be easy to exploit by the scientific community as a whole.

In this paper, we describe a new SGML architecture which we have recently adopted for the HCRC Map Task corpus (a corpus of spontaneous task-oriented dialogues) with precisely these issues in view. This architecture is oriented towards ease of processing and update.

1. INTRODUCTION

The three main aspects of the transcription task which have motivated our design are:

- The data consists of multiple streams with a common timeline: e.g. multiple speakers (often overlapping), different modalities: speech, gaze direction, gestures, external noises.
- There are multiple hierarchies of annotation which may overlap. For example, syntactic, discourse and gaze annotations all may overlap within a single speaker's talk.
- All transcriptions, all annotation are approximations. Changes are thus inevitable, but changes to one type or level of annotation should have the minimum possible impact on any other, even those which depend on it.

We would like to store our corpus in an SGML format, since this is a good format for publishing and allows us to use generic software for processing. However, SGML was not expressly designed to cope with multiple aligned streams of data with overlapping annotation hierarchies.

Our solution is to keep each data stream and each level of annotation as separate XML[2] coded files, and to align them using hyperlinks. This approach minimises the duplication of data; it allows new levels of annotation to be added easily; and because all annotation elements are labelled and all links are expressed in terms of these labels, editing of annotations has minimal knock-on effects on other annotations.

The lowest level of structure is that of the timed transcription unit. In the case of the HCRC Map Task corpus, this is roughly a word, but either higher or lower levels could be chosen. There is a separate base-level transcript file for each talker (and if appropriate for other sound or data sources), and each such file is a simple sequence of timed transcription units and silences. We make a top-level distinction between words and other timed units, e.g. breaths, laughter, lip-smacks, etc. This level is produced semi-automatically from XWaves (TM) xlabel files.

All annotation, including in our case even that of orthographic words, is held in separate files in which each annotation unit 'points to' a span of other annotation units or base-level units, as appropriate. 'Pointing' is done using the draft XML-Link standard syntax for extended hyperlinks [6], [8].

This distributed coding requires software which supports the expansion of hyperlinks when required, the asking of questions which cut across the different annotation layers and the display of the annotated corpus in a human friendly fashion, all of which we have implemented (software freely available for research purposes) [3].

In summary, we will argue that using a syntax for dialogue transcription and annotation based on international standards (SGML and XML-Link) yields major benefits in terms of robustness, flexibility and exploitation, independent of the descriptive vocabulary used for annotation: tools for retrieval, intersection and tabulation of annotations depend only on the syntax of the markup and its reference to a single base level.

2. THE STRUCTURE OF THE CORPUS ANNOTATION

The structure of the annotations for a single speaker in a dialogue in our corpus [1] is shown in figure 1, where each box represents a separate file and arrows show the hyperlinks between files. There is a separate parallel set of files for the other speaker(s), which are only related at the move annotation level (and by sharing a common time line). The particular choice of annotation layers is corpus specific, and either more or different annotation levels could be chosen without affecting the general architecture.

Part of the base level transcription for one of the speakers is given in figure 2. This transcription is a sequence of TU (timed units) of

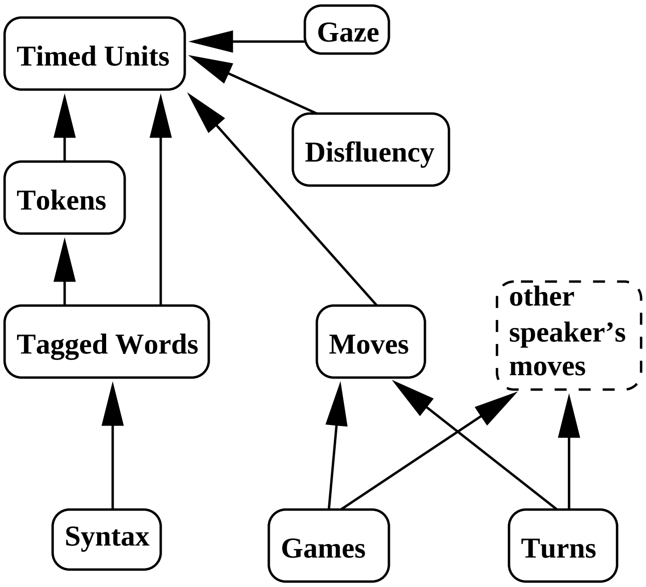

Figure 1: The hyperlink structure of a dialogue.

speech, silences and noises. Timed units are normally single words, but also include some word sequences such as word plus clitic (e.g. “there’s”), or words run together (e.g. “do you have a”), for which it is not sensible to provide individual word timings. This file is the only place where times are stored, higher levels of annotation refer to this base by means of symbolic identifiers (the ’id’ attribute).

```
<?xml version="1.0"?>
<!DOCTYPE timed_unit_stream SYSTEM "timed-
units.dtd">
<timed_unit_stream id="tu.q1ec1.g">
<tu id="tu.1" start="0.00" end="0.32">okay</tu>
<noi id="tu.2" start="0.32" end="0.44"
     type="inbreath"/>
<tu id="tu.4" start="0.44" end="0.84">starting</tu>
<tu id="tu.5" start="0.84" end="1.37">off</tu>
<sil id="tu.6" start="1.37" end="1.57"/>
<tu id="tu.7" start="1.57" end="1.84">we</tu>
<tu id="tu.8" start="1.84" end="2.22">are</tu>
<sil id="tu.9" start="2.22" end="2.35"/>
<tu id="tu.10" start="2.35" end="2.87">above</tu>
<sil id="tu.11" start="2.87" end="2.96"/>
<tu id="tu.12" start="2.96" end="3.03">a</tu>
<tu id="tu.13" start="3.03" end="3.52">caravan</tu>
<tu id="tu.14" start="3.52" end="3.93">park</tu>
...
```

Figure 2: Part of one speaker’s base level timed unit transcription.

The dialogue move annotation [4] is constructed out of the base level transcriptions. Part of the move coding is shown in figure 3. For example, move ’m2’ consists of the timed units from timed unit 4 to timed unit 14 inclusive. This inclusion relation is denoted by the ’href’ attributes of the moves.

The dialogue game annotation is constructed out of the move level coding for both speakers, part of this coding is shown in figure 4. It is only at this level that we introduce a logical ordering of the two speakers speech. Due to the hyperlinking, this logical ordering can be changed without changing the base transcripts.

```
<!DOCTYPE move_stream SYSTEM  "moves.dtd" [
<!ENTITY g "q1ec1.g.timed-units.xml">
]>
<move_stream id="move.q1ec1.g">
<move id="m1" label="ready" href="&g;#id(tu.1)"/>
<move id="m2" label="instruct"
      href="&g;#id(tu.4)..id(tu.14)"/>
...
```

Figure 3: Part of the giver’s move structure, built on top of the timed unit transcription.

```
<?xml version="1.0"?>
<!DOCTYPE game_stream SYSTEM "game.dtd" [
<!ENTITY F "q1ec1.f.moves.sgm">
<!ENTITY G "q1ec1.g.moves.sgm">
]>
<game_stream id=’q1ec1’>
<game id=’g1’ initiator=’giver’ type=’instruct’>
<moveseq href=’&G;#id(m1)..id(m2)’/>
<moveseq href=’&F;#id(m3)’/>
</game>
...
```

Figure 4: Part of the game structure, built on the move annotation for both speakers.

The part of speech tagging is also stored in a separate file, hyperlinked to the timed-unit file. Some of the part of speech annotation is shown in figure 5. Although not shown in this example, the part of speech annotation can also link to the ’token’ level. This is an intermediate level which allows us to apply part of speech tags to words which are not complete timed units, e.g. clitics.

The syntactic analysis [7] is constructed from the part of speech annotation, an example of this is shown in figure 6. Note here that (a) the syntactic analysis has multiply nested markup, and (b) that we use a <twseq> element to hyperlink to sequences of tagged words. This means that we use ’replace’ rather than ’embed’ hyperlink semantics (see next section).

3. EXPANDING THE STRUCTURE

Given that the corpus is now distributed over several files, there is a need to combine the different parts together. For example, when one is parsing the corpus, one may want to access the actual words as well as the sequence of part of speech tags. This means that we want to expand the <tw> elements to include the <tu> and <token> elements. This is done by a program called ’knit’ (part of the LTXML toolkit [3]) which expands hyperlinks. Two kinds of expansion are currently supported: (a) inclusion, where the hyperlinked elements are included inside the containing element; and (b) replacement, where the containing element is replaced by the sequence of hyperlinked elements. For a particular set of XML elements, the choice of which type of expansion to perform can be made by giving command options to ’knit’ or by assigning certain attribute values to the elements in the DTD. An example of this ex-

```
<?xml version="1.0"?>
<!DOCTYPE part_of_speech_stream SYSTEM "part-of-
speech.dtd" [
<!ENTITY tok "q1ec1.g.tokens.xml">
<!ENTITY tu "q1ec1.g.timed-units.xml">
]>
<part_of_speech_stream id="pos.q1ec1.g">
<tw id="pos.1" tag="sent" href="&tu;#id(tu.1)"/>
<tw id="pos.2" tag="vbg" href="&tu;#id(tu.4)"/>
<tw id="pos.3" tag="rp" href="&tu;#id(tu.5)"/>
<tw id="pos.4" tag="pau" href="&tu;#id(tu.6)"/>
<tw id="pos.5" tag="ppss" href="&tu;#id(tu.7)"/>
<tw id="pos.6" tag="ber" href="&tu;#id(tu.8)"/>
<tw id="pos.7" tag="pau" href="&tu;#id(tu.9)"/>
<tw id="pos.8" tag="in" href="&tu;#id(tu.10)"/>
<tw id="pos.9" tag="pau" href="&tu;#id(tu.11)"/>
<tw id="pos.10" tag="at" href="&tu;#id(tu.12)"/>
...
```

Figure 5: Part of the giver’s part of speech annotation.

```
<?xml version="1.0"?>
<!DOCTYPE parsedchannel SYSTEM "syntax.dtd" [
<!ENTITY tw ’q1ec1.g.pos.xml’>
]>
<parsedchannel id=’syn.q1ec1.g’>
<text>
<para id=’q1ec1.g.1’ n=’1’ type=’ok’>
<vp vform=’ing’ pers=’2’>
  <twseq href=’&tw;#id(pos.2)’/>
  <pp prep=’off’>
    <twseq href=’&tw;#id(pos.3)..id(pos.4)’/>
  </pp>
</vp>
<s vform=’fin’>
  <np num=’plur’ case=’subj’ pers=’1’ type=’pro’>
    <twseq href=’&tw;#id(pos.5)’/>
  </np>
  <vp vform=’fin’ num=’plur’ pers=’1’>
    <twseq href=’&tw;#id(pos.6)..id(pos.7)’/>
    <pp prep=’above’>
      <twseq href=’&tw;#id(pos.8)..id(pos.9)’/>
      <np>
        <twseq href=’&tw;#id(pos.10)..id(pos.12)’/>
      </np>
   </pp>
  </vp>
</s>
...
```

Figure 6: Part of the giver’s syntactic annotation.

pansion is shown in figure 7.

4. QUERYING

One major advantage of SGML coding is that it makes the structure of the corpus explicit. This structure can then be used to help the process of querying the corpus for particular forms. Various SGML query language processors are available, including LTXML [3] and SgmlQL [5].

The combination of ’knitting’ and SGML querying is successful when one is processing a single annotation hierarchy. When, however, one wants to compare more than one hierarchy, e.g. asking for pronouns spoken by one speaker when the other speaker is talking, life is more difficult. A complete solution to this problem probably involves a complex database and inverted indices on the base transcription. However, we have implemented a simpler partial solution in a program called ’intersect’ which reads a number of annotation files, performs a number of queries on each one, and returns the intersections of these by time, using the common time line.

Before expansion:

```
<x/>
<a href="#id(b1)..id(b3)"/>
<y/>
```

After expansion (inclusion semantics):

```
<x/>
<a href="#id(b1)..id(b3)">
  <b id=’b1’/>
  <b id=’b2’/>
  <b id=’b3’/>
</a>
<y/>
```

After expansion (replacement semantics):

```
<x/>
<b id=’b1’/>
<b id=’b2’/>
<b id=’b3’/>
<y/>
```

Figure 7: Hyperlink expansion

In particular queries, there is sometimes a need to ask for the start times of higher level segments. However in the corpus examples above, only the base level units have times. Rather than explicitly storing start/end attributes on each higher level segment, which would lead to duplication and possible inconsistency, we prefer to calculate the start/end times for these segments on demand. This suggests that a general program for functionally defined attributes would be a useful addition to a suite of SGML tools.

5. DISPLAY

One advantage of using SGML annotation for our corpus is that it lets us use SGML/XML expertise/software to allow flexible options for displaying the corpus, by using the concept of stylesheets and generic software. An example of what is possible can be seen at our demo web page `http://www.ltg.ed.ac.uk/∼amyi/maptask`.

6. CONCLUSION

Despite the disadvantages of distributed SGML annotation coding, i.e. verbosity and difficulty of integrating the information, we believe that there are advantages in using generic techniques for representing annotated corpora. Continuing software development is necessary before SGML corpora are as useful as they could be, but we believe that present results show promise. An important point is that this approach encourages one to think of general (i.e. not DTD specific) solutions, which are more likely to continue to be useful when one's annotation and corpora change.

The work reported here has been funded by the EPSRC (NScope project), and ESRC UK research councils. Future development of these ideas is being undertaken in the European Union funded Mate project (see `http://mate.mip.ou.dk/`).

7. REFERENCES

1. A.H.Anderson, M.Bader, E.G.Bard, E.Boyle, G.Doherty, S.Garrod, S.Isard, J.Kowtko, J.McAllister, J.Miller, C.Sotillo, H.Thompson & R.Weinert, "*The HCRC Map Task Corpus*", Language and Speech, 34(4), pp 351-366, 1991.

2. T. Bray, J. Paoli and C. M. Sperberg-McQueen (eds), 1998, "*Extensible Markup Language (XML) version 1.0*", W3C Recommendation 10-February-1998, Available at `http://www.w3.org/TR/REC-xml`.

3. C. Brew, D. McKelvie, R. Tobin, H. Thompson, & A. Mikheev, 1998, "*The XML Library LT XML version 1.0: User documentation and reference guide*", Language Technology Group,University of Edinburgh, Available from `http://www.ltg.ed.ac.uk/software/xml`.

4. J. Carletta, A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon, and A. Anderson, 1996, "*HCRC Dialogue Coding Manual*", HCRC Technical Report HCRC/TR-82, June 1996, Available from `http://www.hcrc.ed.ac.uk/publications/`.

5. J. Le Maitre, E. Murisasco, and M. Rolbert, 1996, "SgmlQL, a language for querying SGML documents", In *Proceedings of the 4th European Conference on Information Systems (ECIS'96)*, Lisbon, 1996, pp. 75-89, Information available from `http://www.lpl.univ-aix.fr/projects/multext/MtSgmlQL/`.

6. E. Maler and S. DeRose, eds., 1998, "*XML Linking Language (XLink)*", World Wide Web Consortium Working Draft 3-March-1998, Available from `http://www.w3.org/TR/WD-xlink`.

7. D. McKelvie, "*SDP - Spoken Dialogue Parser*", HCRC Technical Report HCRC/RP-96, May 1998, Available from `http://www.hcrc.ed.ac.uk/publications/`.

8. H.S. Thompson & D. McKelvie, 1997, "Hyperlink semantics for standoff markup of read-only documents", In *Proceedings of SGML Europe'97*, Barcelona, May 1997.