Towards General-Purpose Annotation Tools – How far are we today?

Laila Dybkjær and Niels Ole Bernsen

Natural Interactive Systems Laboratory
University of Southern Denmark
Campusvej 55, 5230 Odense M, Denmark
laila@nis.sdu.dk, nob@nis.sdu.dk

Abstract

This paper discusses the notions of special-purpose natural interactivity and multimodality (NIMM) coding tools, limited-purpose tools, and general-purpose tools as defined in terms of a set of key requirements. In the light of these requirements the paper presents a detailed comparison of three special-purpose and six limited-purpose coding tools and discusses the challenges in building a first general-purpose NIMM annotation tool.

1. Introduction

Current research in natural interactive and multimodal systems is generating an unprecedented need for annotation tools which can accelerate the process of building detailed knowledge of how humans communicate when exchanging information with each other or with systems, using speech, gesture, facial expression, gaze, body posture, and object manipulation as part of the communication. Given the important limitations of established theory on human communication, attaining this knowledge has become fundamental to the development and evaluation of increasingly advanced system generations.

Progress requires new high-quality data resources, of course, and new, well-founded coding schemes, but, arguably, annotation tools are the key to the efficient exploration of data and coding schemes since natural interactivity is immensely complex, including multiple levels and an abundance of within-level and cross-level coordination. While good special-purpose tools already exist for, e.g., orthographic or phonetic transcription of speech, a general-purpose coding tool has still to be developed. Even if no-general-purpose tool is likely to supersede the special-purpose tool in its limited domain of application, the wealth of aspects of human communication which remain unexplored makes it unlikely that we shall have special-purpose tools for them all in the foreseeable future. This is why the goal of creating general-purpose natural interactivity and multimodality (NIMM) annotation tools has attracted a good deal of attention during the last few years. A general-purpose tool is still far preferable to no annotation support at all.

Today's most general NIMM coding tools are only limited-purpose. The goal of developing a general-purpose NIMM coding tool thus remains an obvious next-step challenge. Two key questions are: what are the main functional, design and workflow requirements of a general-purpose tool? And what are the main challenges facing its developers?

In the following, we first look back at how the need for powerful annotation support has evolved (Section 2). Section 3 presents a definition of special-purpose, limited-purpose and general-purpose coding tools and the requirements a general-purpose tool would have to meet. Section 4 presents nine existing coding tools which are compared in Section 5. Section 6 discusses main challenges ahead in building a general-purpose tool.

2. The need for powerful annotation support

The need for annotation and annotation support tools is far from new. However, the complexity of natural interactive and multimodal communication poses new tools requirements compared to the earlier focus on unimodal, such as speech-only, data. For instance, annotation has been important in the fields of written and spoken language processing for more than a decade. Tools were mostly developed in-house for specific purposes and projects and were mainly used for coding speech and language data at single coding levels rather than across levels. A few tools came to be used by several sites, see [Isard et al. 1998] for a review. The EU MATE (Multi-level Annotation Tools Engineering) project (1998-2000, http://mate.nis.sdu.dk) went a step further by demonstrating a tool which could support annotation at arbitrary levels of spoken dialogue, including cross-level annotation, such as annotation of prosodic cues to semantics. Illustrating the difficulties of developing limited-purpose tools, the MATE result was a proof of concept but not a tool which was broadly usable in practice.

Significantly, the annotation support needs addressed in MATE remain largely unfulfilled. Moreover, current interests in natural interactivity and new modality combinations have created a strong need for coding tools far more powerful than the MATE Workbench. A coding tool community is emerging world-wide, aiming to speed up progress in NIMM coding tools and advance their underlying theory.

3. Special-purpose, limited-purpose and general-purpose tools

We distinguish between three different kinds of tools, i.e. special-purpose, limited-purpose and general-purpose tools. A *special-purpose* NIMM coding tool caters for coding NIMM communication at (a) particular pre-defined level(s). The best special-purpose tools have capabilities unlikely to be surpassed by limited- or general-purpose tools. A *limited-purpose* NIMM tool is meant for coding at multiple, non-pre-defined levels and across levels, but it still has some limitations which makes it unsuitable for certain kinds of coding; it includes some or most of the functionality of a general-purpose tool and can be defined in terms of what it lacks to become a general-purpose tool. A *general-purpose* NIMM coding tool supports – within the set of modalities it aims to

address - coding of arbitrary levels and modalities, across levels and across modalities. It offers the functionality and flexibility needed to span the implied broad range of possible coding tasks and to sufficiently support the different users who will carry out those tasks.

The idea of a general-purpose NIMM coding tool probably is only about 5 years old. The type of general-purpose tool we consider here aims at supporting markup of modalities captured by audio and video. General requirements to such a tool would include the ability to:

- enable good and precise handling of raw data (audio, video), millisecond/single frame control, different data formats;
- support entry and revision of coding schemes;
- support efficient exploratory and mature coding, including time-stamped and structure coding;
- support efficient querying of coded data;
- support data analysis, be it via some basic statistics functionality, e.g. inter-coder agreement computation, or via export/linking to existing statistics packages;
- support import/export of coding files and query results in standard formats, e.g. XML;
- provide good and customisable visualisation of everything, including symbolic and analogue views;
- support meta-data descriptions.

As opposed to the (fully) symbolic view, the "analogue" data view allows labelled segment time length visualisation along a visible timeline. Structure coding enables cross-level linking of coordinated phenomena, temporally or otherwise, coded at different levels using different coding schemes. The listed requirements probably reflect broad consensus but the general requirements can be met in extremely different ways aiming at different user groups. The realisations of today's limited-purpose tools span from a do-it-yourself programmer's kit for satisfying – in principle - any particular coding and visualisation purpose to an easy-to-use non-programmers' tool for doing certain kinds of natural interactivity coding, sometimes visualising process and results in different customisable ways.

4. Existing tools for annotation of natural interactive and multimodal data

Special-purpose as well as limited-purpose tools are available for coding of NIMM data. Twelve of the most promising tools and tools projects in existence in year 2000 were reviewed in [Dybkjær et al. 2001]. Since then, some of those tools have been further developed and new tools have emerged. In the tables below, we compare nine existing coding tools. The three tools in the first table are special-purpose tools while the six other tools are limited-purpose. The parameters included reflect the requirements listed in Section 3 and provide, in addition, some general information on each tool.

Tool / **Parameter**	**Transcriber 1.4.2**	**WaveSurfer 1.6.2**	**PRAAT 4.1.27**
Functionality	Segmentation, labelling, orthographic transcription of speech signals. Developed for broadcast news	Sound visualisation and manipulation, phonetic transcription, orthographic transcription	Phonetic transcription, visualisation, analysis and manipulation of speech, orthographic transcription
Overall purpose	Special	Special	Special
Providers	Centre Technique d'Arcueil, France,http://www.etca.fr/CTA/gip/Projets/Transcriber/	KTH, Sweden, http://www.speech.kth.se/wavesurfer	University of Amsterdam, The Netherlands, http://www.praat.org
Platforms	Unix, Linux, Windows	Unix, Linux, Windows, Macintosh,...	Unix, Linux, Windows, Macintosh
Implementation	Tcl/Tk, C extensions, Snack sound extension	Tcl/Tk, Snack	C/C++
Internal data rep.	XML	Text	Text
License issues	Open source, GNU General Public License	Open source, BSD style license	Open source, GNU General Public License
Supported formats	Most common audio files, e.g. wav, mp3	Sound: wav, au, aiff, mp3, CSL, SD, Ogg/Vorbis, NIST/Sphere Transcription: HTK (and MLF), TIMIT, ESPS/Waves+, Phondat	aiff, aifc, wav, au, nist
Media control	Millisecond control via bar and buttons	Millisecond control via buttons and bar	Millisecond control via bar
Coding schemes support	Tags can be added/changed/deleted in the included scheme. Only orthographic transcription	Not really	None
Coding palette	Via keys	Almost none	None
Interface	No programming skills required	No programming skills required	No programming skills required
Types of coding	Time-stamped	Time-stamped	Time-stamped
Info extraction	Simple search	Via analysis tools	Search and via analysis tools
Analysis	None	Waveform, spectrogram, pitch, spectrum	E.g. spectrogram, pitch, formant, intensity, statistics (multidimensional scaling, principal component analysis, discriminant analysis)
Import/export	Export to: STM, Childes, LDC.typ, MATE; more can be added. Import from: .typ transcription files,	Used as widget in custom applications	Save to different sound formats

	xwaves, OGI segmentation files		
Customisable visualisation	Fonts, colours	E.g. colours, sample rate, channels, panes	Sizing menus, fonts, views, sound devices, buttons
Coding file view	Analogue, symbolic	Analogue	Analogue
Meta-data support	Limited, e.g. transcriber's name, date, language	None	None

Tool **Parameter**	**AGTK** **(Annotation Graph Toolkit)**	**Anvil 4.0.7**	**Tasx**
Functionality	Software components for building annotation tools for audio and video data annotation	Annotation of video and audio data. Developed for gesture research	Annotation of video and audio data. Time Aligned Signal data eXchange format
Overall purpose	General in principle (do it yourself)	Limited	Limited
Providers	Linguistic Data Consortium http://agtk.sourceforge.net/	DFKI, Saarbrücken, Germany http://www.dfki.de/~kipp/anvil	University of Bielefeld, Germany http://tasxforce.lili.uni-bielefeld.de/
Platforms	MacOS X, Windows, POSIX	Unix, Linux, Windows, Macintosh	Windows, Unix, Linux, Macintosh
Implementation	C++, Java, Python, Tcl	Java	Java
Internal data rep.	Based on annotation graphs	XML	XML, relational database
License issues	Open source, OSI-approved Common Public License	Free for research, not open source	GNU General Public License
Supported formats	xlabel, TIMIT, Penn Treebank, Switchboard, BAS Partitur, CSV, LDC Callhome, aif	Formats supported by JMF 2.1.1, including QuickTime and avi	Formats supported by JMF
Media control	Examples show seconds for sound, control via buttons and bar	Milliseconds, frame, control via buttons and bar	Seconds (sound), frame, control via buttons and bar
Coding schemes support	Supporting new coding schemes requires programming skills	Entering new coding schemes requires XML skills	No coding scheme can be indicated
Coding palette	Yes, if the programmer made it	Yes, for the selected coding scheme	None
Interface	Only for programmers	XML skills needed to add new coding scheme	No programming skills required
Types of coding	Examples show time-stamped	Time-stamped, structure	Time-stamped
Info extraction	None	Search	Search
Analysis	None	Not much (Sonogram plug-in)	Via built-in link to PRAAT spec-trogram and pitch (in principle)
Import/export	Interfaces to WaveSurfer	Import from PRAAT, Xwaves, export to ASCII format for SPSS	Import from/export to, e.g., annotation graphs, Exmaralda, PRAAT; import from, e.g., Anvil, SyncWriter, Transcriber
Customisable visualisation	As much as the programmer prepares for	Colours, video size, speed of video, collapse/open data groups	Font size and type
Coding file view	Depends on programmer; can be both analogue and symbolic	Analogue	Analogue, symbolic (only one layer at a time)
Meta-data support	Can be programmed	Little support (coder and coding scheme)	Via menu entries

Tool **Parameter**	**NXT** **(NITE XML Toolkit)**	**NWB 3** **(NITE Workbench for Windows)**	**The Observer**
Functionality	Software components for building annotation tools for audio and video data annotation	Annotation of video and audio data	Annotation of video data. Developed for behavioural studies
Overall purpose	General in principle (do it yourself)	Limited	Limited
Providers	HCRC, Edinburgh, UK http://www.ltg.ed.ac.uk/NITE/	NISLab, University of Southern Denmark http://nite.nis.sdu.dk	Noldus, Wageningen, The Netherlands http://www.noldus.com
Platforms	Unix, Linux, Windows	Windows	Windows
Implementation	Java	C++	C++
Internal data rep.	XML	Relational database	Relational database
License issues	Open source, GNU General Public License	Free for research	Commercial
Supported formats	E.g. avi, mpeg, au	wav*, au, aiff, midi, mp3, wma, asf, cda, avi*, mpeg, wmv, ivf, vob	E.g. mpeg, avi, QuickTime, Digital Video
Media control	Examples show video frame control via buttons	Millisecond, frame, control via buttons	Frame, control via buttons, seconds shown
Coding schemes	Entering new coding schemes	Interface for entering new coding	Interface for entering new coding

support	requires XML and stylesheet skills	schemes	schemes (configuration)
Coding palette	Yes, if the programmer made it	Yes, for the selected coding scheme	Yes, for the selected coding scheme
Interface	Only for programmers	No programming skills required	No programming skills required
Types of coding	Time-stamped, structure	Time-stamped	Time-stamped
Info extraction	Query via own query language	Query via SQL interface, search	Search
Analysis	None	None	Time events, reliability, elementary statistics, lag sequential analysis
Import/export	Can be programmed	Export to XML	Export to formats for further statistical processing, import of graph (bitmap) and Ethovision data
Customisable visualisation	As much as the programmer prepares for	Colours, zoom, timing (seconds, frames)	E.g. speed, timing, toolbar, auditory feedback
Coding file view	Depends on programmer	Symbolic	Symbolic
Meta-data support	Can be programmed	Free form text meta-data can be entered in meta-data table	Some (fixed) parameters can be entered via configuration

5. Comparison of tools

The three special-purpose tools are all specialised to address particular well-defined codings of audio files such as orthographic transcription. They offer good control of the sound signal. The two tools supporting phonetic transcription (WaveSurfer and PRAAT) also offer a number of speech signal analysis tools. But since the tools are not meant for other types of coding than they were designed for, there is close to no support for coding scheme changes, the type of coding is time-stamped only, and the offered customisations are limited to those known from many other programs, such as colours and fonts.

The six limited-purpose tools all have in common that they are not meant to handle pre-defined coding levels. On the other hand, none of them provide sufficient support for arbitrary coding of audio/video data to be called general-purpose tools. Two of them (AGTK and NXT) are do-it-yourself tools. They come with examples but otherwise leave it to the user to build the tool needed based on the offered components. Do-it-yourself tools may be useful if nothing better is around, if a programmer is available, and if one needs functionality which comes fairly close to the included examples. If the latter is not the case, it may be faster and better to tailor a tool to one's needs just using an ordinary programming language. AGTK examples show audio control but no video control. NXT examples show video control but no audio control.

The limited-purpose tools often reveal, via their strong and weak sides, what they originally were developed for. For example, the Observer has quite limited support for handling audio data and spoken dialogue, and Anvil reveals in its visualisation that focus has not been on spoken dialogue annotation. The analogue coding file view is often sub-optimal for spoken dialogue coding.

One issue which really categorises all these tools as limited-purpose rather than general-purpose is the lack of an appropriate interface for entering new coding schemes and the possibility to enter any coding scheme which one may find relevant for the kind at raw data that can be handled by the tool. NWB and the Observer have interfaces for coding scheme entry. However, none of them are easy to comprehend and, in both cases, there are limits to which kinds of coding schemes can be entered.

A second issue is the customisation and visualisation options. A general-purpose tool would need very considerable flexibility in these respects since it would have to accommodate many different needs and preferences. As an example, most of the limited-purpose tools only offer either a symbolic or an analogue coding file view but not both nor at the same time. Also, most tools only offer time-stamped coding. This makes a tool unsuitable for the coding of cross-level and cross-modality relationships. Moreover, the kinds of customisation offered are typically quite basic, such as fonts, colours, size, zoom and speed.

Sophisticated information extraction is frequently missing apart from some kind of – often not very advanced - search. Exceptions are NXT which includes a query module with a home-grown query language and NWB which offers information extraction via an SQL interface. None of these interfaces are for novice users, however. Analysis tools are typically absent apart from what has been made obtainable via plug-ins and links (Anvil, Tasx). Only the Observer includes some simple analysis tools and – importantly – supports export to existing statistics packages.

Meta-data support has received varying attention in the reviewed tools from close to no support to some support.

6. Challenges ahead

Comparing NIMM coding tools is clearly a multi-dimensional exercise. No tool is just simply better or poorer than another. But we have still not many tools to choose among, and a general-purpose tool is still a challenge for the future.

As we see them, the three main challenges in building such a tool are: how to allow for easy entry of coding schemes of one's own choice or design, how to enable unlimited cross-level and cross-modality coding, and how to provide sufficient flexibility in visualisation to optimise presentation of data coded by using arbitrary (sets of) coding schemes.

References

Dybkjær, L., Berman, S., Kipp, M., Olsen, M. W., Pirrelli, V., Reithinger, N. and Soria, C.: Survey of Existing Tools, Standards and User Needs for Annotation of Natural Interaction and Multimodal Data. ISLE Deliverable D11.1, 2001.

Isard, A., McKelvie, D., Cappelli, B., Dybkjær, L., Evert, S., Fitschen, A., Heid, U., Kipp, M., Klein, M., Mengel, A., Møller, M.B. and Reithinger, N.: Specification of workbench architecture. MATE Deliverable D3.1, August 1998.