THE NITE WORKBENCH
A Tool for Annotation of Natural Interactivity and Multimodal Data

Niels Ole Bernsen, Laila Dybkjær and Mykola Kolodnytsky

Natural Interactive Systems Laboratory
University of Southern Denmark
Science Park 10, 5230 Odense M, Denmark
{laila, nob, mykola}@nis.sdu.dk

Abstract

This paper describes ongoing work in the European NITE project on the development of a tool in support of annotation of natural interactive and multimodal data. The paper discusses the resources required for pursuing the vision of natural interactivity and provides an overview of existing natural interactivity data coding tools and projects. After discussing the target user groups of a NITE tool, the paper presents requirements to a visual coding tool interface followed by an early draft of the visual interface for the NITE coding tool.

1. Introduction

Within the broad area of natural interactivity, current trends are to move from the study of individual modalities to investigating modality combinations and studying interactions and interrelationships among modalities. A very significant amount of work still remains to be done in order to gain a better understanding of how humans actually communicate with each other and with machines (or systems). This knowledge is required in order to enable researchers and companies to produce high-quality, user-friendly natural interactive systems.

We are still far from having achieved the long-term vision of natural interactivity described in Section 2. However, a lot of effort is being invested in the field world-wide and indications are that the field is in rapid growth. Essential groundwork for being able to build the next generations of natural interactive systems involves the collection of quality natural interactivity corpora, corpus annotation at many different analytical levels as well as cross-level, and corpus analysis based on information extraction. As long as everything has to be done by hand, the processes of doing annotation, information extraction, and analysis continue to be slow and error-prone. To accelerate progress, therefore, general-purpose tools are strongly needed. This observation is not a new one. Several recent initiatives have as their goal the creation of some version of a general-purpose tool for the coding of, and the extraction of information from, natural interactivity corpora. So far, however, no project has succeeded in producing a really useful general-purpose tool for coding and analysing full natural interactivity data.

NITE (Natural Interactivity Tools Engineering) is one of the projects which currently address the challenge just described. NITE is a European HLT (Human Language Technologies) project which began its work in April 2001. The goal of NITE is to develop a workbench, or an integrated set of tools, for annotating and analysing full natural interactive communication among humans and between humans and systems. The annotated corpora can then be used and re-used to advance our understanding of complex natural interactive communicative behaviour, train natural interactive system components, etc.

As will be discussed below, developers of a general-purpose tool for coding and analysing natural interactivity data are faced with a large variety of user needs from a highly diverse user population. In view of the composition of this user population, partners in NITE are convinced that an essential condition for building a tool which could be successfully used by the majority of those working in the field, is to provide the tool with an easy-to-use, user-friendly interface.

This paper describes ongoing work in the NITE project towards building a general-purpose tool for coding and analysing natural interactive communicative behaviour. The perspective adopted is a partial one, as we will focus eventually on the user interface for the tool. Before describing the current state of work on the tool interface, we address some more general issues, including the vision of natural interactivity and its demands on resources in terms of data, coding schemes, and tools (Section 2), the state of the art in coding tools for natural interactive communication (Section 3) and the targeted users of the NITE tool (Section 4). Then follows a requirements specification for the visual interface of the NITE tool (Section 5) and a draft of the visual annotation interface (Section 6) Concluding the chapter, Section 7 discusses work still to be done.

2. Natural Interactivity: Vision and Needs

The long-term vision of natural interactivity is to enable systems to communicate, or exchange information, with humans in the same ways in which humans exchange information with one another, using thoroughly co-ordinated speech, gesture, gaze, facial expression, head movement, bodily posture, and object manipulation [Bernsen 2001]. The idea of multimodality is to improve human-system interaction in various ways by using novel combinations of (unimodal) input/output modalities [Bernsen 2002]. Natural interactivity is by nature (mostly) multimodal.

Evidently, natural interactivity is a long-term technological goal. Progress requires technical advancement in many areas, including, e.g., speech recognition, spoken dialogue processing, machine vision, computer graphics, multimodal input understanding and output generation, architectures, and system integration. Moreover, progress demands far better understanding of human communicative behaviour than we have at present.

What is needed to better understand human communicative behaviour is, *first*, quality acoustic and

visual data. For a recent survey of natural interactivity and multimodal data resources world-wide, see the ISLE (International Standards for Language Engineering) NIMM (Natural Interactivity and Multimodality) European Working Group report on NIMM data resources [Knudsen et al. 2002a]. *Secondly*, a wide range of coding schemes are needed for coding details of human-human and human-system communication, each capturing a particular class of phenomena at some level of analysis, such as speech prosody or the facial expression of emotion. Although coding schemes already exist for annotating various aspects of spoken, facial and gestural information, several schemes need further investigation to improve their theoretical soundness and make them usable by machines. Moreover, new coding schemes are needed for phenomena which have not been systematically investigated yet. For a recent survey of natural interactivity and multimodal corpus coding schemes world-wide, see the ISLE NIMM Working Group report on NIMM coding schemes [Knudsen et al. 2002b]. *Thirdly*, a major characteristic of human communication is behavioural coordination. For instance, we sometimes use eyebrow movement to accompany prosodic stress on a particular word in order to communicate that this word is intended to have a particular lexical meaning rather than another. The coordination present in human communication across levels and across modalities remains poorly understood. Its understanding requires scientifically well-founded coding schemes for the different classes of phenomena involved as well as for their interrelationships. *Fourthly*, we need tools to support working with data and coding schemes. Several tools already exist each of which supports annotation and analysis of some aspect(s) of acoustic and (static or dynamic) graphical data. However, none of these tools cover the full range of aspects present in natural interactive communication, and many of them are research tools with severe deficiencies from the point of view of practical use. For a recent survey of natural interactivity and multimodal corpus annotation tools world-wide, see the ISLE NIMM Working Group report on NIMM coding tools [Dybkjær et al. 2001a]. The three ISLE reports mentioned above are summarised in [Dybkjær and Bernsen 2002].

3. State-of-the-Art

Annotation of spoken dialogue data has emerged as an important field of research during the past 10-15 years. A key factor driving this development, although by no means the only factor involved, is the need for annotated data for the development and evaluation of interactive speech systems, such as spoken language dialogue systems and spoken translation systems. As the sophistication of interactive speech systems increased, so did the need to better understand spoken interaction. Spoken language is the core communication modality in standard *situated* communication, having developed to efficiently serve human-human communication in shared space, time, and situation, physical and otherwise. The tremendously rich speech signal reveals as much about the speaker's personality and mental states as it informs and directs the interlocutor(s).

In the field of spoken dialogue corpus annotation, level-specific coding tools gradually emerged - for morphosyntactic annotation, co-reference annotation, dialogue acts annotation etc., as described in the MATE (Multi-level Annotation Tools Engineering) project report on the state of the art in spoken dialogue annotation tools [Isard et al. 1998]. All of those tools, however, were either completely level-specific or very limited as regards their multi-level coding capabilities. To our knowledge, the MATE Workbench [mate.nis.sdu.dk] which appeared in 2000 is still the only fully multi-level and cross-level spoken language dialogue coding tool around. However, this tool still has important limitations, such as being fragile and without an appropriate user interface for the average user.

Situated natural human communication involves not only speech but a whole series of modalities in addition to speech. This is reflected in the fact that researchers and technology developers are now moving beyond spoken human-system dialogue towards the long-term goal of achieving natural interaction between humans and machines. Animated graphical interface agents capable of some amount of spoken dialogue, humanoid robots with similar capabilities, audio-visual speech recognition, and combined speech and gesture input understanding systems all illustrate the emerging trend towards the development of increasingly natural human-system interaction. This trend towards the integration of spoken dialogue into more complete natural interactive systems has emphasised the need for efficient natural interactivity coding tools.

Considering the state of the art in natural interactivity coding tools, we find a variety of home-grown, limited-functionality, special-purpose, and level-specific coding tools from among which somewhat more general tools are beginning to emerge. The ISLE NIMM Working Group report on of natural interactivity and multimodal corpus annotation tools [Dybkjær et al. 2001a] describes twelve tools which support annotation and analysis of (spoken) dialogue, facial expression, gaze, gesture, and/or bodily posture, etc., and possibly cross-level or cross-modality issues as well. Figure 1 shows the reviewed coding tools and tool projects. In Figure 1, *Tool* is the name of the tool or project reviewed. *NIMM (Natural Interactivity and Multimodality) aspects addressed* are the NIMM aspects which a particular tool explicitly claims to support. *Brief tool/project description* is a brief description of the tool or project, including a web address.

A couple of tools had not been implemented at the time of the review. However, the tool concepts presented by the projects aiming to develop the tools were found sufficiently interesting for including a project description in the ISLE NIMM survey, e.g., because the project has standardisation among its goals. Two tools are professional (commercial), i.e. The Observer and SyncWriter. The rest are research tools (or projects). MATE is a limiting case in another sense, because the MATE Workbench only supports spoken dialogue and text annotation. The tool is included because of its advanced properties for multi-level and cross-level annotation, which may show the way towards building a general-purpose natural interactivity coding tool. The CLSU Toolkit coding tool is for output generation only. Finally, so far, at least, the SmartKom project is a user rather than a provider of NIMM coding tools.

Tool: Anvil. *NIMM aspects addressed:* Speech and gesture. *Brief tool description:* Annotation of video and language data. A Java-based tool for annotating digital video files. See www.dfki.de/~kipp/anvil

Tool: ATLAS. *NIMM aspects addressed:* No tool was available at the time of review. *Brief project description:* Architecture and Tools for Linguistic Analysis Systems. See www.itl.nist.gov/iaui/894.01/atlas

Tool: CLAN. *NIMM aspects addressed:* Text, speech and gesture. *Brief tool description:* Computerised Language Analysis. A program designed specifically for analysing data transcribed in the format of the Child Language Data Exchange System (CHILDES). Transcriptions can be linked to audio or video files. See childes.psy.cmu.edu

Tool: CSLU Toolkit. *NIMM aspects addressed:* Speech, TTS and facial expression. *Brief tool description:* Center for Spoken Language Understanding Toolkit. A suite of tools including the Rapid Application Developer, BaldiSync (for facial animation), SpeechView, OGIsable (an annotation tool), speech recognition tools, and a programming environment (CSLUsh). OGIsable is the only annotation tool included. It allows the user to attach properties to a text before it is spoken (via the Festival TTS engine), e.g. to synthesise facial expression synchronised with speech output. See cslu.cse.ogi.edu/toolkit/

Tool: MATE Workbench. *NIMM aspects addressed:* Speech and text. *Brief tool description:* Multi-level Annotation Tools Engineering. A Java-based tool in support of multi-level annotation of spoken dialogue corpora and information extraction from annotated corpora. See mate.nis.sdu.dk

Tool: MPI tools: CAVA and EUDICO/Computer Assisted Video Analysis and European Distributed Corpora. *NIMM aspects addressed:* Speech and gesture. *Brief tool description:* Both tools support annotation of audio-visual files and information extraction. See www.mpi.nl/world/tg/CAVA/CAVA.html, www.mpi.nl/world/tg/lapp/eudico/eudico.html

Tool: MultiTool. *NIMM aspects addressed:* Speech and gesture. *Brief tool description:* MultiTool was developed in a project on a Platform for Multimodal Spoken Language Corpora. A Java-based tool in support of the creation and use of multimodal spoken language corpora (audio and video). See www.ling.gu.se/multitool

Tool: The Observer. *NIMM aspects addressed:* Gesture and facial expression. *Brief tool description:* A professional system for the collection, analysis, presentation, and management of video data. It can be used to record activities, postures, movements, positions, facial expressions, social interactions or any other aspect of human or animal behaviour as time series of tagged data. See www.noldus.com/products/index.html?observer/

Tool: Signstream. *NIMM aspects addressed:* Speech, gesture and facial expression. *Brief tool description:* Signstream was developed as part of the American Sign Language Linguistic Research Project. A database tool for analysis of linguistic data captured on video. See web.bu.edu/asllrp/SignStream

Tool: SmartKom. *NIMM aspects addressed:* No tool available. *Brief project description:* A large-scale project which aims to merge the advantages of spoken dialogue-based communication with the advantages of a mixture of graphical user interfaces and gesture and mimetic interaction. SmartKom uses tools developed elsewhere: in Verbmobil for audio annotation, Anvil for mimics and gesture coding. See www.smartkom.org

Tool: SyncWriter. *NIMM aspects addressed:* Speech and gesture. *Brief tool description:* A professional tool for transcription and annotation of synchronous "events" such as speech and video data. See www.sign-lang.uni-hamburg.de/software/software.html

Tool: TalkBank. *NIMM aspects addressed:* Speech (Transcriber), text, speech and gesture (CLAN), see above. *Brief tool description:* A project which aims to provide standards and tools for creating, searching, and publishing primary materials via networked computers. No tools had been developed at the time of review. See www.talkbank.org

Figure 1. Brief overview of the 12 natural interactivity and multimodality coding tools and projects reviewed in [Dybkjær et al. 2001a].

Speech seems to be the key modality addressed by all the tools in Figure 1 but one (The Observer), i.e. by nine tools. Gesture is addressed by seven of the ten tools for which there was software to review. Facial expression is addressed by three tools only.

Among the tools reviewed, the MATE workbench is by far the most advanced tool as regards markup of spoken dialogue. It can handle, in principle, annotation at any analytical level and comes with a number of example coding schemes for different annotation levels, such as dialogue acts and co-reference. Most of the other tools reviewed are capable of addressing speech annotation in a multimodal context as well. However, these tools either do not go beyond the transcription level or they offer, at most, single-level annotation of, e.g., dialogue acts according to a built-in annotation scheme (Anvil).

Several of the reviewed tools for handling gesture annotation could probably –with more or less effort – be extended to handle markup of facial expression as well. When it is not mentioned in Figure 1 that a gesture annotation tool supports facial expression annotation, this is typically because the tool does not include a coding scheme for facial annotation. It may be noted here that if some coding scheme is hard-coded into a tool, it is not necessarily easy to add new coding schemes to the tool.
The Observer provides support for annotation of gesture as well as facial expression because it is quite easy to add new annotation schemes using the interface offered by the tool for this purpose. However, in order to add new coding schemes, one has to comply with the general markup framework provided by the Observer, which imposes important limitations on the structure of the coding schemes that can be added. Among the reviewed tools only two other tools (SignStream and the CSLU Toolkit) claim to support annotation of facial expression. For the CSLU Toolkit, the annotation support is meant for output generation of an animated speaking face.

4. NITE and its Target Users

In many ways, NITE pursues the same objectives as its predecessor HLT project MATE (Multi-level Annotation Tools Engineering, mate.nis.sdu.dk). The main difference

is that NITE goes beyond spoken dialogue coding and analysis to full natural interactivity data annotation and analysis. The NITE objectives thus are: to develop a markup framework; identify, or develop, a number of natural interactivity best practice coding schemes to be described following the markup framework; and build a general-purpose natural interactivity annotation and analysis toolset which includes those coding schemes and supports the addition of new ones within the general boundaries of the markup framework.

It is evident from the efforts and aspirations behind the tools listed in Figure 1, that there is a strongly felt need for general-purpose annotation and analysis tools among many different communities. Current needs for annotated natural interactivity data span academic research on all aspects of natural interactivity, including, e.g., human communicative behaviour, prosody, linguistics, psychology, anthropology, disappearing languages and cultures, human factors, and research prototype development of many different kinds of interactive systems that include some amount of natural interactivity. Similar needs characterise the emerging commercial development of limited-capability natural interactive systems. Based on those needs, annotated natural interactivity data resources are being used for a range of purposes, including, e.g., information gathering, coding scheme research, component training, component evaluation, systems design and development, embodied agent design, audio-visual speech recognition, and automatic person identification.

It appears that users of natural interactivity annotation tools may be divided into three groups, as follows.

The first user group includes people who need a tool which allows them to do their tasks of annotation, information extraction, and analysis without bothering about the internal design, data representation formats, and workings of the tool. Typically, these users are experts in their area, such as the understanding of the integration of speech and facial expression in human communication, and they consider the annotation tool simply as a vehicle for making their work more efficient and its results more useful and more widely available. Considering the many different disciplines whose practitioners are potential users of a general-purpose tool, it seems likely that this first user group is the larger by far compared to the two following user groups.

The second user group includes users who have become so used to data coding formalisms, such as SGML or XML, or who experience that existing editors are not good enough for their purposes, that they feel most comfortable if they can edit the coded data directly. Thus, if, e.g., XML is being used for internal data representation, they want to have access to the XML files, possibly via an editor, even if the internal data representation is meant for the computer rather than for the user.

The third user group includes users who have the programming skills and the motivation to add new tool functionalities to an existing tool provided that the tool makes this possible. To accommodate these users, and given the fact that no natural interactivity coding tool will ever include all conceivable useful functionality, the best option is to equip a general-purpose toolset with an open architecture which enables the addition of new functionality.

Although NITE aims to support all three user groups, we shall focus only on the first-mentioned user group in the remainder of this paper, since this is the user group which needs an ordinary and easy-to-use visual interface. NITE support for the second and third user groups above is described in [Soria et al. 2002].

5. Requirements to a Future Tool

Before proceeding to describe current work on the NITE tool's user interface, we would like to briefly present the NITE consortium's *core functionality* philosophy. According to this line of thinking, current tool developments in the field of natural interactivity should be observant of the fact that multiple parallel activities are already going on in the field: in tools for orthographic and phonetic transcription, statistics packages, data representation formats, metadata standards, etc. There is no reason for a NITE tool to replicate existing tools or functionalities when these are already satisfactory. Rather, NITE should focus on the core functionalities involved in providing a tool for full natural interactivity data annotation and analysis. If this task can be solved to the satisfaction of the intended users of the tool, those users can be relied upon to use other, already existing tools for the many other things they have to do when working with their data and which are not being provided by the NITE tool. What this requires is for the NITE tool to incorporate, to the extent possible, emerging standards for data representation, such as XML, enable data import from, and export to, already existing tools, such as advanced statistics packages, etc.

Based on the actual user needs expressed in the tools reviewed in [Dybkjær et al. 2001a], the tool requirements established on this basis in [Dybkjær et al. 2001b], and taking into account that our focal user group consists of those who simply want a tool in order to carry out their annotation and analysis tasks efficiently and without bothering about programming and internal tool data representations, the following would seem to be the core requirements to the tool's audio-visual interface. The tool should:

1. support annotation of natural interactive communication at any analytical level through the use of an existing or a new coding scheme;
2. enable users to specify new coding schemes either by editing an existing coding scheme or by adding one from scratch;
3. enable information extraction and analysis of annotated data.

With these basic requirements in mind, the first step in the design of a visual interface for the NITE tool has been to make a series of design decisions concerning the general layout of the visual interface to ensure that those requirements can be met in a coherent and user-friendly way. We want the interface layout to be similar to what users are likely to be familiar with from other programs. This will reduce the learning curve for novice tool users as regards basic layout and functionality. Thus, it has been decided to have a common top-screen menu line that provides access to all other system functionality, cf. points 1-3 above, including basic options such as copy, paste, save and print. Some options will always be available. Other options are specifically related to one of the requirements above and can only be selected when the

user is actually doing the task addressed in the requirement.

So far, we have developed the tool's interface based on analysis of the first requirement above, i.e. annotation support which is discussed in more detail in Section 5.1.

5.1. Annotation support

The NITE visual interface aims to support annotation of any kind of phenomena involved in natural interactive communication. To this end, the following requirements to the annotation user interface have been specified.

1. Raw data perception and inspection. Raw data must be made perceptually accessible to the user at will. The user must be able to look at the video whenever necessary and switch it off at will. The user must be able to listen to the audio track whenever necessary and switch it off at will. The audio track and the video track must be controllable independently of one another. The same applies to the viewing of other kinds of raw data, such as log-files and graphical representations of acoustic information (the latter is considered raw data for present purposes even if this might be contended). Acoustic raw data, video raw data, and graphical representations of acoustic information should all come with a visible time-line. It must be possible to navigate back and forth in the raw data based on the timeline. It should be possible to open several video raw data windows at the same time, for instance in order to inspect complex long-range dependencies.

2. Data annotation. Appropriate facilities must be present for annotating any aspect of the video, including free-form comments on what goes on in the video as well as use of standard annotation schemes for spoken dialogue, facial expression, emotion, gaze, gestures of all kinds, lip movements, bodily posture, actions, etc. Annotation is also taken to include orthographic transcription of the acoustics. Phonetic transcription remains an open issue in NITE.

3. One annotation at-a-time. We should not expect serious users to annotate according to several annotation schemes simultaneously. This means that the structure of one palette/one annotation scheme/one layer of annotation active for annotation would be a valid one. Palettes are described in Section 6. However, since we are after finding new regularities which may well be cross-level or cross-modality, it should be possible to link several annotation levels during annotation.

4. Resolution levels. Given the fact that the phenomena to be annotated in trying to understand natural interactivity may not only be quite complex per time unit, as it were, but also temporally extended or even linked to an indeterminate extent, it must be possible to view transcriptions and annotations at different levels of resolution, from viewing few-seconds-duration cross-level, cross-modality annotations close-up to viewing minutes-long stretches of transcription and annotation birds-eye. This imposes additional requirements onto the representation of levels and links.

5. Editors. It should be possible to open simple editing windows at any time. These windows may be used for, e.g., inserting additional observations related to the annotation process, or doing preliminary experiments with new coding schemes which have not yet been specified to work with the tool.

6. Tag palettes. On-screen palettes must be able to refer to (or mark up) single-level phenomena as well as cross-level links.

7. Within-level tagging. It should be possible to clearly label time-aligned entities during annotation. Labelling of entities, if not done free-form, could be done by selecting from a palette elsewhere on the screen. The palette would contain all possible tags belonging to a particular coding scheme. It should be possible to link the following temporal entities to the common timeline: points in time (no duration), timed entities of any length, such as sub-words, filled or unfilled pauses, words, phrases, communicative acts, as well as larger-duration communication units of any kind. It should be possible to visibly link to the common timeline the following types of within-level relationship: long-range dependencies, such as co-references, cause and effect, and event conditions.

8. Cross-level, cross-modality tagging. It should be possible to visibly link annotations at different levels and of different modalities, such as linking facial expression annotation to transcription, or linking gesture annotation to speech act annotation. It should be possible to tag these links. This will enable researchers to establish highly innovative coding schemes for coordinated cross-level, cross-modality clusters of communication phenomena.

9. Number of displayed annotation levels. It must be possible to show up to ten annotation levels simultaneously on the screen. Given the fact that showing as many as ten annotation levels may be a relatively rare occurrence, allowance could be made for screen extensions to make this possible, for instance through scrolling. However, it must be possible to inspect a reasonable number of annotation levels on the screen without scrolling.

10. Perceptible time-alignment. Annotation levels (including orthographic transcription) should visibly share a common timeline. This means that, independently of the way in which those annotation levels are being represented, the common timeline corresponding to the audio and/or video and/or graphical representation of acoustic events, must visibly "run through" all annotations.

11. Display management during coding. It should be possible to remove individual links as well as all links between two different levels of annotation. If all links are removed temporarily, this may be done by saving the file with the links before removing them. It should be possible to re-order annotation representations on the screen. Re-ordering may affect cross-level links. Links should be preserved but a certain clutter may be unavoidable.

6. The Audio-Visual Annotation Interface

Based on the requirements presented in Section 5, the NITE tool's user interface will be a uniform visual interface which offers the kinds of customisation known from many other programs. As those requirements suggest, the interface will otherwise have to be a quite complex one because it has to cater for the presentation of raw data, multi-level and cross-level annotation, free-style comments, annotation analysis, annotation comparison, search and inspection of search results, as well as basic interface operations such as file saving, printing, deletion, overview, duplication, import and export. In this section,

we describe and illustrate the current draft annotation interface.

In order to provide the user with a uniform way of doing multi-level and cross-level annotation of natural interactivity data as well as to enable several ways to display annotated files, the user interface for data annotation is composed from the following five main components:

1. the *main window* which contains the main menu, the title, etc. (Figure 2);
2. the main window *toolbar* which contains the changeable (contents-sensitive) set of buttons (Figure 2);
3. a changeable amount of *panels* of the *i*-th class of phenomena to be annotated – 1 up to 10 panels (Figure 3), cf. Section 5.1 point 9;
4. a changeable number of *raw data windows* displaying the different types of raw data (Figure 4), cf. Section 5.1 point 1;
5. the common *control board* for controlling the active raw data window(s) (Figure 5), cf. Section 5.1 point 1.

In addition, numerous palettes (dialogue boxes) with built-in controls will be provided for the user to work with different coding schemes, for instance in order to insert or delete tags, visualise tags, etc., cf. Section 5.1 points 2, 6, 7, 8, and 11.

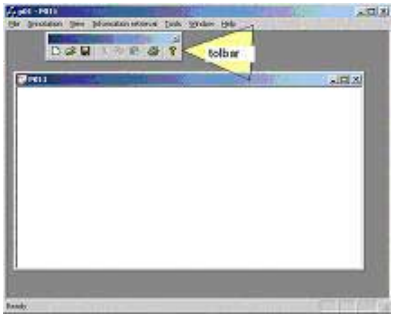

Figure 2. NITE main window with toolbar.

Figure 3. Annotation panels (up to ten) embedded in the main window.

Based on the interface concept outlined above, the following two steps will be needed to accomplish an annotation:

1. *select* a class of phenomena to annotate by selecting a particular coding scheme. This will cause the list of tags belonging to the selected coding scheme to appear on the screen as a coding palette together with a panel in which to insert the annotations, cf. Section 5.1 points 2, 3 and 6;
2. *edit* (insert/delete) time markers on the time-line in the appropriate annotation panel. Markers will visualise the tags according to the chosen tag set on the annotation palette, cf. Section 5.1 points 6, 7, 8, 10 and 11.

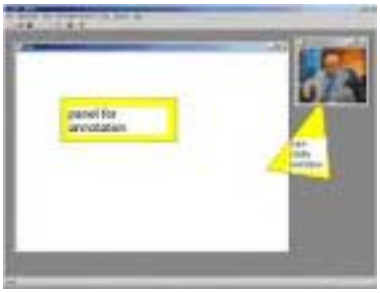

Figure 4. Raw data window.

Figure 5. Control board for controlling the active raw data window.

The result may look as outlined in Figure 6. This approach allows us to provide a uniform style of work with the annotation tool. For any level of annotation and any coding scheme, the user will perform the same set of actions: choosing a class of phenomena (or a coding scheme), choosing the appropriate button (the appropriate tag) from the coding palette, and insert the marker of the tag onto the time-line on the panel.

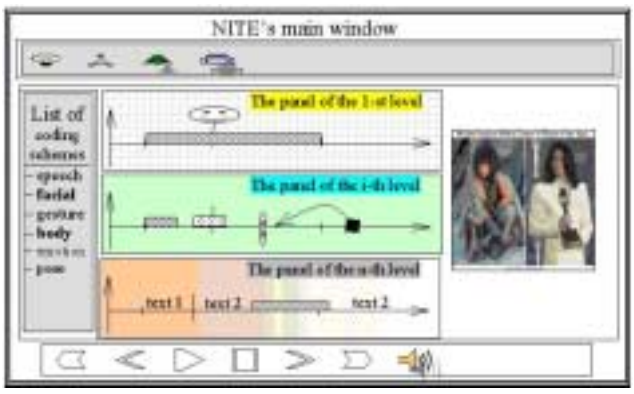

Figure 6. Tags visualised on the time line.

Points 4 (resolution levels) and 5 (editors) of Section 5.1 are not illustrated in the figures in this section. Resolution levels can be set by allowing the user to zoom in and out as well as choosing among different types of

visualisation, i.e. musical score and ordinary running text format.

An editor is a window like the raw data windows. The editor window will allow the user to enter pure text and save it in a file. If feasible, it should also be possible to link between comments in the editor window and the corresponding time-aligned tags in the annotation file.

7. Future plans

The NITE annotation user interface is currently being implemented in C++. We are presently developing detailed specifications of the user interfaces for adding new coding schemes and performing information extraction and analysis. In parallel, work is going on towards specifying the NITE markup framework, including the NITE metadata representation, as well as to identify a core set of import/export facilities from/to tools of high relevance to NITE tool users.

8. Acknowledgements

We gratefully acknowledge the support of the NITE project by the European Commission's Human Language Technologies (HLT) Programme.

9. References

Bernsen, N. O.: Multimodality in language and speech systems - from theory to design support tool. In Granström, B. (Ed.): Multimodality in Language and Speech Systems. Dordrecht: Kluwer Academic Publishers 2002 (to appear).

Bernsen, N. O.: Natural human-human-system interaction. In Earnshaw, R., Guedj, R., van Dam, A. and Vince, J. (Eds.): Frontiers of Human-Centred Computing, On-Line Communities and Virtual Environments. Berlin: Springer Verlag 2001, Chapter 24, 347-363.

Dybkjær, L., Berman, S., Bernsen, N. O., Carletta, J., Heid, U. and Llisterri, J.: Requirements Specification for a Tool in Support of Annotation of Natural Interaction and Multimodal Data. ISLE Deliverable D11.2, 2001b.

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, 2001a. View or download from isle.nis.sdu.dk

Dybkjær, L. and Bernsen, N. O.: Natural Interactivity Resources - Data, Annotation Schemes and Tools. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), 2002 (to appear).

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, 1998. View or download from mate.nis.sdu.dk

Knudsen, M. W., Martin, J.-C., Dybkjær, L., Ayuso, M. J. M, N., Bernsen, N. O., Carletta, J., Kita, S., Heid, U., Llisterri, J., Pelachaud, C., Poggi, I., Reithinger, N., van ElsWijk, G. and Wittenburg, P.: Survey of Multimodal Annotation Schemes and Best Practice. ISLE Deliverable D9.1, 2002b. View or download from isle.nis.sdu.dk

Knudsen, M. W., Martin, J.-C., Dybkjær, L., Berman, S., Bernsen, N. O., Choukri, K., Heid, U., Mapelli, V., Pelachaud, C., Poggi, I., van ElsWijk, G. and Wittenburg, P.: Survey of NIMM Data Resources, Current and Future User Profiles, Markets and User Needs for NIMM Resources. ISLE Deliverable D8.1, 2002a. View or download from isle.nis.sdu.dk

Soria, C., Bernsen, N. O., Cadée, N., Carletta, J., Dybkjær, L., Evert, S., Heid, U., Isard, A., Kolodnytsky, M., Lauer, C., Lezius, W., Noldus, L. P. J. J., Pirrelli, V., Reithinger, N. and Vogele, A.: Advanced tools for the study of natural interactivity. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), 2002 (to appear).

Websites
ISLE: isle.nis.sdu.dk
MATE: mate.nis.sdu.dk
NITE: nite.nis.sdu.dk