The NITE Workbench for Annotating Natural Interactive Behaviour

Niels Ole Bernsen, Laila Dybkjær, Mykola Kolodnytsky
NISLab,
University of Southern Denmark
{nob, laila, mykola}@nis.sdu.dk

Abstract

This paper addresses the emergence of much-needed general-purpose natural interactivity coding tools. In particular, the NITE (Natural Interactivity Tools Engineering) project is currently completing three toolsets for the annotation and analysis of natural interactive communication data. The tools share a common requirements specification but are otherwise quite different. The paper briefly presents the state of the art in the field, discusses the common NITE specification, compares the three NITE tools, and describes in more detail the tool developed by the authors, i.e. the NITE Workbench for Windows or NWB.

1 Introduction

Natural interactive systems, by any name, are proliferating in research today and will soon, in various forms, find their way into industrial products. Research into natural interactivity pursues the long-term goal of allowing human users to communicate with computer systems in the same situated ways in which humans communicate with one another, i.e. through speech, gesture, gaze, facial expression, body posture, object manipulation as part of communication, etc. In prototypical situated communication, humans exchange information whilst sharing the same physical and temporal context as well as sharing the emerging context of discourse. In most cases, their communication is extremely well coordinated, so that speech, gesture, etc. all contribute to communicating the same combined message. This coordination is, to a large extent, spontaneous and unconscious both in generation and reception, which is why we, despite the fact that we conduct natural interactive communication to a massive extent every day, remain largely ignorant of its mechanisms of coordination as well as of the patterns of communication behaviour which emerge as a result.

So far, fledgling natural interactive systems have primarily turned into a commercial paradigm in the form of spoken dialogue systems. However, spoken dialogue systems are far, indeed, from being full natural interactive systems. These systems, which often communicate with their human users over the telephone, have no notion of the physical and temporal context of the communication, remain ignorant of all non-speech communicative events, and are equally incapable of producing any of these. Moreover, today's commercial spoken dialogue systems are insensitive to large parts of the received input speech signal, such as the parts which signal prosody and emotion and which are often intimately coordinated with the non-speech behavioural elements of natural

interactive communication. Similarly, these systems are still mostly incapable of generating context-dependent prosodical and emotional output.

The trail towards increasingly natural interactive systems is currently being blazed by new, emerging research communities which combine hitherto separate software disciplines, such as interactive speech systems, computer graphics, computer vision, and machine learning, as well as theoretical studies of human gesture, facial expression, conversation, etc. One such community is the extremely broad one of multimodal interaction which, natural interactivity being multimodal, includes natural interactive communication as a sub-set together with other sub-sets, such as the GUI (Graphical User Interfaces) paradigm, and an unknown amount of other sub-domains (Bernsen 2002). Another, more well-defined, community is that of spoken dialogue-capable animated interface agents (Cassell 2000). What these communities have in common is the need to better understand natural interactive communication in humans in order to make machines understand and generate the same forms of communication.

To better understand natural interactive communication in humans, what is needed are (i) high-quality data on human natural interactivity, (ii) an indefinite number of new coding schemes for annotating this data, and (iii) general purpose coding tools which can dramatically facilitate data annotation, new coding scheme creation and exploration, data analysis and scientific generalisation, and, ultimately, help us understand the huge complexity of natural interactive communication. Arguably, among (i) through (iii), general-purpose coding tools are the most important because their availability will make it so much more efficient to exploit carefully created new data resources and new coding schemes aimed at annotating hitherto unknown aspects of natural interactivity.

2 Natural Interactivity Coding Tools

The state of the art in natural interactivity coding tools is described in the ISLE NIMM (International Standards for Language Engineering, Working Group on Natural Interactivity and MultiModality) Report D11.1 (Dybkjær et al. 2001a), available from isle.nis.sdu.dk under reports. The report reviews some 12 completed or ongoing coding tools projects and concludes that although several of the tools are meant to be general-purpose, none of them cover natural interactivity coding in general. During the two years which have passed since the report was written, some of the reviewed tools have matured and new tools projects have been launched. The Anvil coding tool, available from http://www.dfki.de/~kipp/anvil/, which was fairly new two years ago has been improved and has, although it is not general-purpose, gained quite some use (Kipp 2001). The general-purpose Annotation Graph Toolkit has begun to be used (Bird and Liberman 2001), cf. http://agtk.sourceforge.net/. New tools have emerged which, like Anvil and NXT (see Section 4.2), are based on Java and XML, such as TASX (Milde 2002), available from http://tasxforce.lili.uni-bielefeld.de/. In addition, important general-purpose natural interactivity coding tools results have

emerged from the EU project NITE (Natural Interactivity Tools Engineering, (nite.nis.sdu.dk)). In NITE, and based on a common specification for a general-purpose natural interactivity coding tool, three software development strands are now reaching completion.

To summarise the above brief coding tools state of the art, the natural interactivity and multimodality research community has shown itself to be responsive to the strongly increasing needs for general-purpose coding tools. However, having worked in the field of annotation tools for several years, starting with the MATE (Multi-level Annotation Tools Engineering) tool for annotating spoken dialogue at multiple levels of abstraction as well as cross-level (http://mate.nis.sdu.dk), it seems important to state that one should not be carried away by the emergence of general-purpose natural interactivity coding tools. For several annotation purposes, such as orthographic or phonetic transcription, relatively mature tools are already available, such as, for orthographic transcription, Transcriber (http://www.etca.fr/CTA/gip/Projets/Transcriber/), and, for phonetic transcription, PRAAT (http://fonsg3.let.uva.nl/praat/), among others. The field of natural interactivity coding is far too complex to be catered for by a single general-purpose annotation tool. If particularly strong coding tool capabilities are required for some fragment of natural interactivity coding, current general-purpose tools are hardly up to meeting the challenge from special-purpose tools. However, if general-purpose natural interactivity understanding-through-coding is at stake, there is no alternative to the tools listed in the previous paragraph.

This remainder of this paper will focus on the NITE coding tools family. We first present and discuss the common NITE specification for a general-purpose natural interactivity coding tool (Section 3). Section 4 describes commonalities and differences among the NITE coding tools. Section 5 focuses on the tool which we are developing ourselves, i.e. the NITE Workbench for Windows or NWB. We describe the look-and-feel of NWB, how it meets the NITE tools specification, and discuss the important issue of how to visualise coding and query results.

3 The NITE Common Specification

The NITE common tool specification was prepared initially in ISLE Report D11.2 (Dybkjær et al. 2001b), available from isle.nis.sdu.dk under Reports. Based on the tools survey mentioned in Section 2 above, ISLE Report D11.2 proposed preliminary requirements for a general-purpose natural interactivity coding tool. These requirements were turned into a complete set of tool requirements in NITE, cf. the addendum to NITE report D1.1 (Bernsen et al. 2002a). Instrumental to the development of a common NITE requirements specification was the use case of annotating, cross-level and cross-modality, the full set of natural interactive communication behaviours involved in conversational turn-taking. The following list broadly describes the common NITE requirements specification.

1. *A flexible and open architecture which allows for easy addition of new tool components by advanced software-literate users.* Even a general-purpose tool is not likely to cover the needs of all target users. Thus, it is important that users can make their own additions to the tool and that APIs are made available which support the addition of new tool components.
2. *Separation of user interface from application logic and internal data representation.* The internal data representation should be separated from the user interface via an intermediate logical layer so that the former two layers can be modified separately.
3. *Adequate support for viewing, listening to, and controlling the presentation of raw data, including video, audio, and the graphically represented speech signal.* There should be adequate support for navigating and searching raw data according to time information. Strong support for raw data control is particularly essential for data segmentation. Users need to be able to segment the raw data files at millisecond level and at frame level in order to investigate, e.g., the temporal coordination between word-level or sub-word-level orthographic or phonetic speech segments and coordinated gestures, gaze actions, or facial expressions. Once segmented, the raw data files should be searchable based on the segmentation.
4. *The most common raw data file formats must be accepted by the tool,* such as wav, au, mp3, avi and mpeg. The raw data file, all annotations of the raw data file, and all queries of raw data and annotated files, should, directly or indirectly, refer to a common timeline. The common timeline, thus, should "glue" together any view of the raw data, its annotations, and, when required, the results of any query made.
5. *The launch version of the toolset should include a number of best practice coding schemes* for annotation at different levels of abstraction, of different modalities, and of interaction across levels and modalities. The purpose of this requirement is two-fold. Firstly, users might want to use state of the art coding schemes for their own purposes. Secondly, the included coding schemes serve to illustrate the coding scope of the tool.
6. *The launch version of the toolset should include and demonstrate a pre-standard markup framework* for describing the best practice coding schemes included and serving as guidelines for describing any new coding schemes to be added. The markup framework should describe coding schemes in a uniform way which makes them comprehensible and easy to use. The NITE markup framework (Dybkjær et al. 2002) is an extension of the markup framework developed for multi-level and cross-level annotation of spoken dialogue in MATE (Dybkjær et al. 1998). As demonstrated in the ISLE Reports surveying existing natural interactivity raw data resources (Knudsen et al. 2002b) and coding schemes (Knudsen et al. 2002a), the research community is generally too lax in providing the raw data and coding scheme meta-data required by other users in order to decide if some available resource meet their needs. This hampers raw data

and coding scheme re-use, and the NITE markup framework has been designed to encourage raw data and coding scheme creators to provide the minimum information necessary for resource re-use.

7. *Support for easy addition and use of new coding schemes,* including coding schemes which are being developed through exploratory annotation. This requirement is of key importance to any general-purpose natural interactivity coding tool. The importance of the tool is not due, primarily, to the fact that it already includes several state-of-the-art coding schemes (cf. Point 5 above). Rather, its main importance is due to the possibility of adding new, possibly exploratory, coding schemes and test these through coding of existing or new raw data resources.
8. *Annotation (including transcription) support at different analytical levels, for different modalities (speech, facial, gesture, posture, etc.), and for annotating complex relationships (long-range dependencies, partial inclusion, clustering, etc.) within-level as well as across levels and modalities.* This is a crucial requirement to a general-purpose tool. Arguably, no natural interactivity coding tool could be claimed to be general-purpose without meeting this requirement. In order to proceed towards full understanding of coordinated natural interactive communication, we need to be able to link phenomena at different levels of abstraction as expressed in a single modality or in different modalities, and coordinated across modalities, such as a spoken deictic expression coordinated with a pointing gesture, or a prosodic cue coordinated with a particular sense of a word. Moreover, natural interactive communication exhibits an as yet unknown range of abstract communicative patterns, such as the long-range dependencies involved in co-reference or cause-and-effect, which need exploration supported by a general-purpose tool.
9. Today, annotation at certain levels, such as (morpho-)syntactic annotation, can be done semi-automatically or automatically. *To the extent that it is possible to (semi-) automate natural interactivity data annotation processes, this should be supported by the toolset.* Similarly, *to the extent that it is possible to (semi-) automate data analysis, this should be supported by the toolset.* Automation should be supported in two ways: (i) via the possibility to add (through an API) additional components for automatic annotation and analysis, and (ii) via the use, as far as possible, of standard data formats allowing easy import and export of annotations for subsequent processing by other tools. This point aptly expresses the NITE development philosophy: even if useful and justifying the resources invested into its development, no general-purpose natural interactivity coding tool could, or should, do everything its users might need. Rather, to the extent that existing software packages, such as statistics packages or other specialised tools, could complement what can be done with the tool, this should be supported through data import/export facilities.

10. *Powerful functionality for query, retrieval, and extraction of data from annotated corpora, and a minimum of tools for data analysis, possibly including simple statistical tools.* Powerful query and information retrieval is needed for the user to exploit the annotated data for an open-ended range of purposes. The query tool interface should make it easy to specify even fairly complex query expressions. Results must be presented to the user in a sensible, intuitive and easy-to-use manner. In NITE, we view coding file querying as part of the coding process, whereas sophisticated multi-purpose data analysis is the task of complementary tools, cf. Point 9 above.
11. *Adequate visual presentation of annotated data and data query results.* The toolset should enable visualisation of timeline information and synchronised views of different layers of annotation and different modalities. The toolset should come with a set of predefined but flexible and versatile visualisations. Availability of options for customising views on annotated data are desirable. As any coder knows, appropriate visual presentation of the results of codings and coded data queries is crucial to workflow efficiency. Different coders have different visualisation requirements, partly due to personal preferences, partly due to coding task demands. Any usable and useful general-purpose coding tool should be able to accommodate a significant range of different user requirements to the visualisations offered by the tool.
12. *Possibility of importing and re-using existing data resources via conversion tools.* Today, as demonstrated in the ISLE surveys of data resources, coding schemes, and coding tools (cf. above), there is widespread use of proprietary systems and formats for natural interactivity data coding. This constitutes a major obstacle to creating a standard coding tool which still allows users to exploit the data resources they have built using other tools and in whatever format they have found appropriate. The only practical solution would seem to be to offer a way of adding tools for converting from the user's data format to the formats used by the NITE coding tools. It should be easy to write converters and add them since it is not practically feasible to incorporate every conceivable converter from the outset.
13. *Possibility of exporting, by means of conversion tools, coded data resources and query results for further processing by external tools, such as statistics software packages. In particular, given the growing importance of XML, XML export functionality is desirable.* This requirement hardly needs justification.
14. *Most importantly, perhaps, the tool must be robust, stable and work in real time even with relatively large data resources and complex coding tasks.* It is important for general-purpose tools developers to take into account the fact that, should their tool become used more widely, some users will want to use the tool for heavy-duty coding tasks involving gigabytes of raw data, mega-bytes of data codings, and many coders who want to exchange coding results.

4 Three NITE Toolsets

Based on the common NITE specification presented in Section 3, the three NITE coding tools development strands are The Observer, the NITE XML Toolkit (NXT), and the NITE Workbench for Windows (NWB). Whereas The Observer is a commercial tool, the core NXT software is Open Source and NWB is freeware. The possibilities for making NWB Open Source are being investigated. The core NXT software can be downloaded from http://www.ltg.ed.ac.uk/NITE/. NWB can be downloaded from http://nite.nis.sdu.dk/download/. As will become apparent from the tools descriptions and comparisons in this section, The Observer, NXT and NWB interpret the common NITE specification in significantly different ways, aiming at different groups of users, cf. (Soria et al. 2002).

4.1 The Observer

The Observer is a widely used Windows-based commercial software package for general behavioural annotation based on video raw data. Existing versions of The Observer (Noldus et al. 2000, Cadée et al. 2002, www.noldus.com) have not been aimed at supporting natural interactive communication coding at all. The development aim of the Noldus company behind The Observer is to exploit the common NITE specification in order to extend The Observer's capabilities, so that the tool can be used for natural interactivity coding to a limited but important extent. The Observer, in other words, does not aim to support unrestricted general-purpose natural interactivity coding.

The strategy for extending The Observer's capabilities is to equip new releases of the software package with a number of functionalities that will enable the tool to be used for natural interactivity coding. Some important functionalities for inclusion in 2003 are:

- XML file export.
- Support analysis of multiple code layers (nested analysis).
- Decode and play back two synchronized MPEG streams.
- Encode and retrieve timing information from two synchronized MPEG streams.

Since the Observer can already be used for some amount of coding of facial expression, gesture, gaze, etc., the described software additions will enable The Observer to be used to quite some extent for the coding of natural interactive behaviours. The Observer already offers facilities for annotated data query and query results display, including limited statistics functionality and export to formats accepted by standard statistics packages.

Compared to the research prototypes NXT and NWB, The Observer possesses the robustness to be expected from a commercial tool.

4.2 NITE XML Toolkit

The Observer basically differs from NXT (Carletta et al. to appear, http://www.ltg.ed.ac.uk/NITE/) and NWB in its limited-purpose coding aims as regards natural interactive behaviours. NXT, in its turn, basically differs from The Observer and NWB by targeting a limited user group, i.e. those users who want to be able to have control of the XML-based coding by using XML schemas for coding scheme specification, and of the display of coding and query results by using XSLT stylesheets. Thus, unless NXT already includes the coding scheme and the corresponding display facilities which a user wants to use, the user must start by programming the coding scheme and its on-screen display. By contrast, The Observer and NWB are both Windows-based tools which offer an easy-to-use, familiar coding interface to those users who just want to explore natural interactivity without having to do any programming to get started. Both tools offer various coding file and query results visualisation customisation options through menu selection.

Among the NITE toolsets, NXT is the direct descendant of the MATE Workbench developed in NITE's predecessor project MATE (Multilevel Annotation Tools Engineering, see (McKelvie et al. 2000) and http://mate.nis.sdu.dk). With its internal XML representation and engine, NXT also joins the growing number of Java-based, platform-independent, broad-purpose or general-purpose natural interactivity coding tools aimed at a special user group, which includes Anvil and TASX, and which has a competitive relationship with the US-produced Annotation Graph Toolkit. NXT includes special-format, purpose-built functionality for query, information extraction, and analysis of coding files. NXT of course enables import and export of files in XML format. XML file export is enabled in The Observer and NWB as well. As for raw data handling, NXT shares with Anvil two (non-Open Source) plug-ins developed in NITE. The first plug-in is a graphical visual markup tool which enables the user to insert markup in a video, e.g. as a green box surrounding a moving hand. The second plug-in is a sonogram tool which creates sonogram/spectograms for media files.

4.3 NITE Workbench for Windows

Among the NITE toolsets, NWB (Bernsen et al. 2002b, Bernsen et al. 2003, http://nite.nis.sdu.dk/download/) is the only tool which aims to deliver general-purpose natural-interactivity coding support for ordinary users who do not want to do any programming in order to explore and exploit the full complexity of natural interactive communication. The NWB's Windows platform provides a familiar interface to most users. Through this interface, users can easily enter new coding schemes, perform single-level, multi-level, cross-level, and cross-modality annotation, and query annotated data files. The tool's relational Access database for data storage provides a powerful SQL query interface for many users, a query interface which will later be furnished with an easy-to-use superstructure for non-

SQL-literate users. For the moment, NWB comes with a set of explained, frequently used SQL queries.

5 The NITE Workbench in Detail

Figure 5.1 shows a four-layered conceptual view of the NWB. Everything in the application layer has been developed in NITE together with the binary resources in the development tools layer, and the database structure and XML export in the physical store layer. The NWB user provides the project files, raw data video and audio files, database contents, and exported XML documents, all in the physical store layer. Apart from the binary resources already mentioned, everything else in the development tools layer and the Operating System layer are the development resources used to develop NWB. The six boxes in the Application Layer provide an operational view of the NITE tool specification presented in Section 3.

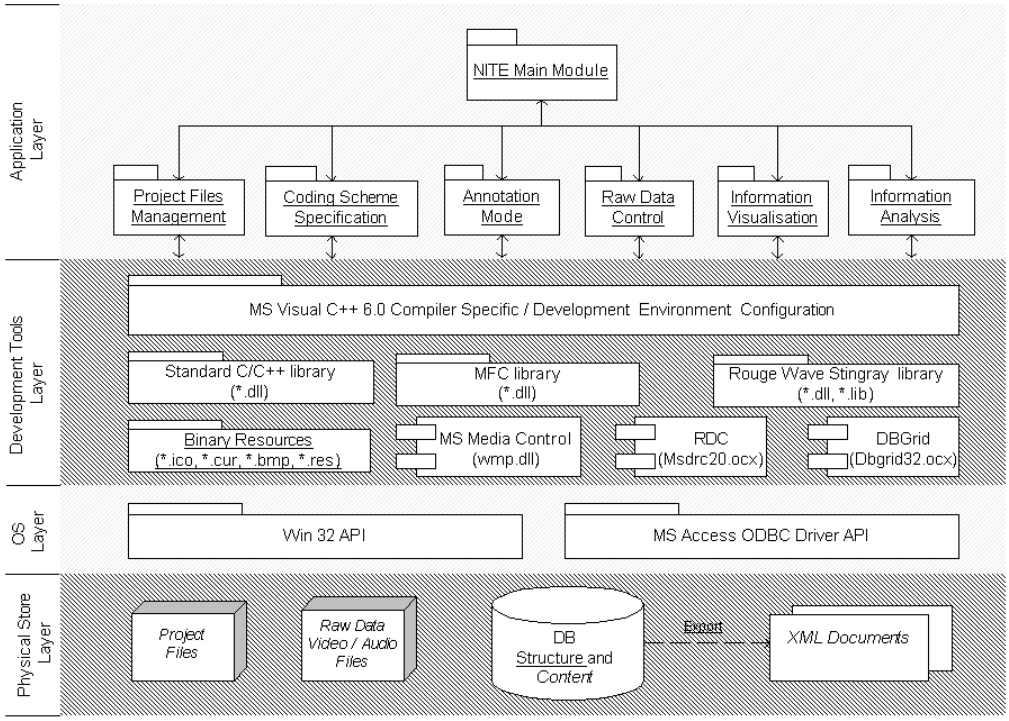

Figure 5.1. Conceptual view of the NWB technology architecture. Underlined text means developed in NITE. Italicised text means produced by the users.

The third, and first public, release of the NWB, i.e. NWB3, will be made available on 31 July 2003 at http://nite.nis.sdu.dk/download/. The first two releases were only made available to a smaller group of test persons. The software release comes with a User Manual describing in detail how to set up and work with the NWB as well as Release Notes explaining the differences between the previous and the present NWB releases. At the time of writing, NWB2 user testing is taking place at various sites across Europe and it is too early to report on the

outcome. Instead, in this section, we will present the look-and-feel of NWB2, discuss to which extent the NWB3 under development meets the requirements specification discussed in Section 3, and discuss some key issues of present and future NWB development.

5.1 NWB2 Look-and-Feel

Figure 5.2 shows orthographic transcription with NWB2. The NITE Floorplan Corpus video data file is in the upper left-hand corner and is controlled by the control board in the upper right-hand corner. Orthographic transcription is entered using the small window at the bottom right. Transcription tags are entered using the tag palette in the lower left-hand corner. The result of the transcription coding, i.e. timestamps, speaker ids, transcription tags, and transcription text, is shown in the coding file window in the middle.

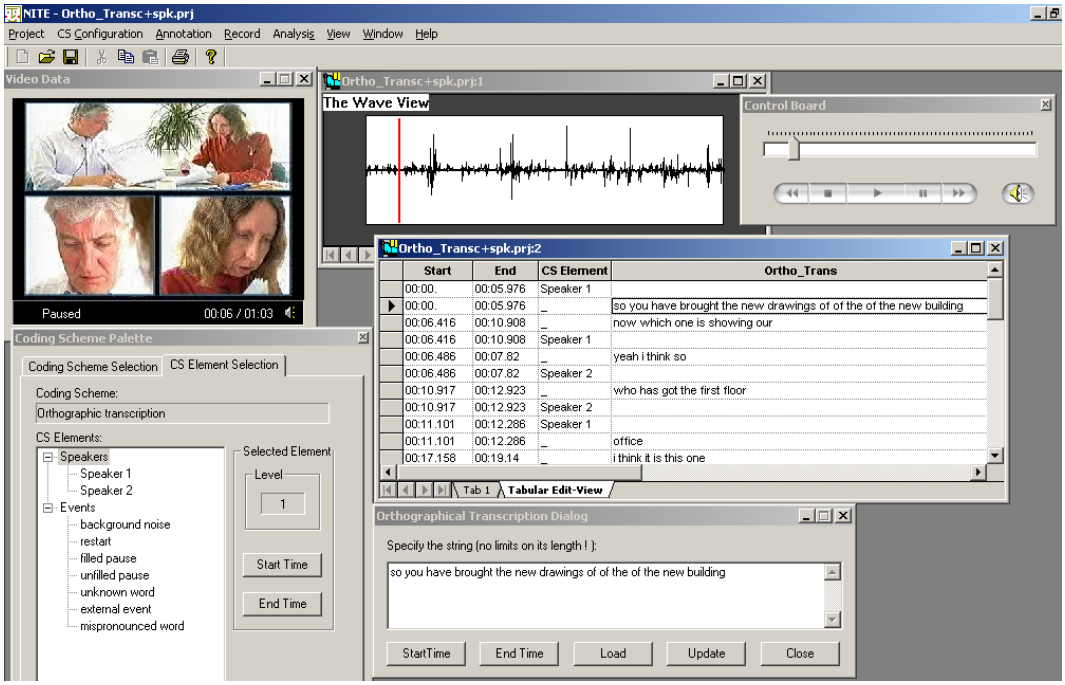

Figure 5.2. Coding with NWB2.

Figure 5.3 shows semi-complex coding of the NITE Floorplan Corpus. In addition to timestamps for all tags inserted in the two left-most columns, the markup includes (a) speaker ids in the third column, (b) simple orthographic text transcription in the fourth column, (c) exploratory tagging of the corpus using a new speech and non-speech deictics coding scheme in the third column, and, also in the third column, (d) coding using a speech-non-speech deictics coordination coding scheme and showing the occurrences of coordinated use of, e.g., a spoken deictic expression and a pointing gesture. The theoretical implications of this demo example for the coding scope of the NWB are discussed in Section 5.3. It

may be pointed out here that it is possible to view the coding in Figure 5.3 in different ways, such as showing speakers in one column, coding tags per coding scheme applied in separate columns as well, swap columns for better viewing, etc.

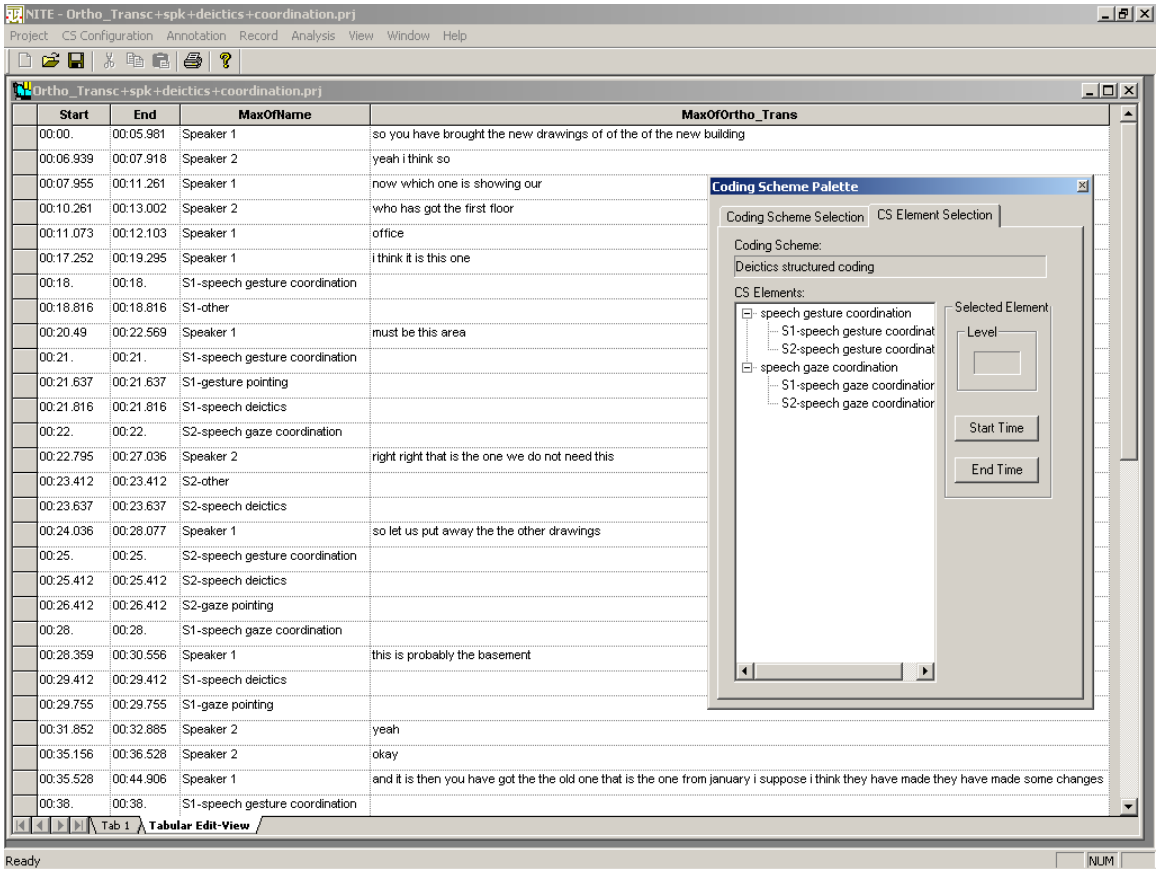

Figure 5.3. Multi-level, multimodal, and cross-level coding.

Space limitations do not allow us to illustrate how to enter a new coding scheme into NWB2. Query results are visualised in the same general way as the coding results in Figure 5.3.

5.2 NWB3 and the NITE Requirements Specification

In this section, we selectively focus on the points in the NITE requirements specification (Section 3) which could still be enhanced compared to the functionality that NWB3 will have. Thus, NWB3 basically meets all the NITE requirements which are not discussed below. Numbers refer to the NITE requirements specification.

3. *Adequate support for viewing, listening to, and controlling the presentation of raw data.* NWB3 offers millisecond precision in controlling the audio signal, enabling, e.g., word-by-word transcription, but still needs frame-by-frame control of the raw data video signal.

9. *To the extent that it is possible to (semi-) automate natural interactivity data annotation processes and data analysis, this should be supported by the toolset.* It is possible to add new components to NWB3 for (semi-) automated data annotation. However, given the time of the NWB3 release, no components have yet been added. As for data analysis (semi-) automation, the NITE philosophy is to enable XML coding file and query analysis result export in order to facilitate the use of other tools for the kinds of data analysis which cannot be executed within NWB3 by using SQL queries (cf. also Point 10 in the NITE requirements specification).
10. *Powerful functionality for query, retrieval, and extraction of data.* As mentioned in Section 4.3 the query interface to the workbench still has to be furnished with an easy-to-use superstructure for non-SQL-literate users.
11. *Adequate visual presentation of annotated data and data query results.* As argued in Section 3, coding file and query file visualisation is a key usability issue in any annotation tool. We discuss the issue further in Section 5.3 below. NWB3 offers a range of customisation options, such as table and text manipulation, and colour and font display customisation.
12. *Possibility of importing and re-using existing data resources via conversion tools.* While NWB3 enables XML coding file and query file export, file import remains an open issue. It is, of course, possible to build converters for converting proprietary data formats to the internal NWB3 format, but we have not yet analysed any real example of this task.
14. *The tool must be robust, stable and work in real time even with relatively large data resources and complex coding tasks.* NWB3 is still, of course, more buggy than a commercial product and we continue to debug the software. We do not expect major problems with NWB3's handling of large data resources. So far, however, the tool has only been used with small resources, so it remains to be seen if the tool scales up as expected.

5.3 Coding and Query Visualisation in Post-NWB3 Releases

So far, indications are that NWB's relational database data model is capable of supporting full general-purpose natural interactivity annotation of unlimited complexity, including multi-level, cross-level, and cross-modality coding of all known structural phenomena. Note, however, that, at present, all codings, and all query results as well, are visually presented in a *symbolic, time-stamped, tabular view* (see Figure 5.3). A main next challenge in NWB development is to complement this view with an *informationally equivalent, semi-analogue, and only partially time-stamped view.* Information equivalence means that the user can freely switch between the former, *tabular* view, and the latter, *analogue* view. Semi-analogue (Bernsen 2002) means that some of the coded items viewed, such as relative duration of overlapping utterances from different speakers, can be perceived directly rather than having to be read from the symbolically presented durations in Figure 5.3. Some current coding tools, such as Anvil and TASX, offer this view only. Indirect time-stamping means that, once some phenomenon, such as a spoken word, has been time-stamped with, say, start and end time, then

subsequent coding referring to that phenomenon, such as POS tagging of words and phrases, will not have to be time-stamped but simply inherit the time-stamps from the time-stamped phenomena which are being structure coded.

6 Outlook

The rapidly growing fields of natural and multimodal interaction are coming closer to the satisfaction of a fundamental need, i.e. the need for general-purpose tools for the annotation and analysis of natural interactive communication. We invite the reader to download or otherwise access the tools discussed in this paper and to kindly let the tool developers know about any problems with, or new requirements to, the tools.

7 Acknowledgements

We gratefully acknowledge the support of NITE by the European Commission's Human Language Technologies (HLT) Programme. Thanks to all NITE partners for their collaboration.

8 References

8.1 Literature

Bernsen, N. O.: Multimodality in Language and Speech Systems - from Theory to Design Support Tool. In Granström, B., House, D. and Karlsson, I. (Eds.): Multimodality in Language and Speech Systems. Dordrecht: Kluwer Academic Publishers 2002, 93-148.

Bernsen, N.O., Cadée, N., Carletta, J., Dybkjær, L., Evert, S., Heid, U., Isard, A., Kolodnytsky, M., Lauer, C., Lezius, W., Noldus, L., Reithinger, N., and Vögele, A.: Lists of Requirements. NITE report D1.1 Addendum, September 2002a.

Bernsen, N. O., Dybkjær, L. and Kolodnytsky, M.: The NITE Workbench - A Tool for Annotation of Natural Interactivity and Multimodal Data. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'2002), Las Palmas, 2002b, 43-49.

Bernsen, N. O., Dybkjær, L., and Kolodnytsky, M.: An Interface for Annotating Natural Interactivity. In J. v. Kuppevelt and R. W. Smith (Eds.): Current and New Directions in Discourse and Dialogue, Dordrecht: Kluwer 2003 (to appear).

Bird, S. and Liberman, M.: A Formal Framework for Linguistic Annotation, in: Speech Communication 1-2, 2001, 23-60.

Cadée, N., Meyer, E, Theuws, H. and Noldus, L.: The Observer Video-Pro: A Versatile Tool for the Collection and Analysis of Multimodal Behavioral Data. Proceedings of the LREC'2002 Workshop on Multimodal Resources and Multimodal Systems Evaluation, Las Palmas, May 2002, 53-55.

Carletta, J., McKelvie, D., and Isard, A.: A generic approach to software support for linguistic annotation using XML. In G. Sampson and D. McCarthy (Eds.): Readings in Corpus Linguistics. London and New York: Continuum International, (to appear).

Cassell, J., Sullivan, J., Prevost, S. and Churchill, E. (Eds.): Embodied Conversational Agents. Cambridge, MA: MIT Press, 2000.

Dybkjær, L., Bernsen, N.O., Dybkjær, H., McKelvie, D. and Mengel, A.: The MATE Markup Framework. MATE Report D1.2, November 1998.

Dybkjær, L., Berman, S., Bernsen, N. O., Carletta, J. C., Heid, U., and Llisterri, J. (2001). Requirements Specification for a Tool in Support of Annotation of Natural Interaction and Multimodal Data. ISLE Report D11.2, July 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 Report D11.1, January 2001a.

Dybkjær, L., Bernsen, N.O., Carletta, J., Evert, S., Kolodnytsky, M. and O'Donnell, T.: The NITE Markup Framework. NITE Report D2.2, 2002.

Kipp, M.: Anvil - A Generic Annotation Tool for Multimodal Dialogue. Proceedings of Eurospeech 2001, 1367-1370.

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, 2002a.

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, 2002b.

McKelvie, D., Isard, A., Mengel, A., Møller, M.B., Grosse, M., and Klein, M.: The Mate Workbench: An Annotation Tool for XML Coded Speech Corpora. Speech Communication 33(1-2), 2000, 97-112.

Milde, J.-T.: Creating Multimodal, Multilevel Annotated Corpora with TASX. Proceedings of the International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems. Copenhagen, 2002, 120-126.

Noldus, L. P. J. J., Trienes, R. J. H., Hendriksen, A. H. M., Jansen, H., and Jansen, R. G.: 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 and Computers, 32, 2000, 197-206.

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 LREC'02, 2002, 357-363.

8.2 Websites

Anvil: http://www.dfki.de/~kipp/anvil/
Annotation Graph Toolkit: http://agtk.sourceforge.net/
ISLE NIMM: http://isle.nis.sdu.dk
NITE: http://nite.nis.sdu.dk
MATE: http://mate.nis.sdu.dk
NWB: http://nite.nis.sdu.dk/download/
NXT: http://www.ltg.ed.ac.uk/NITE/
PRAAT: http://fonsg3.let.uva.nl/praat/
TASX: http://tasxforce.lili.uni-bielefeld.de/
The Observer: http://www.noldus.com/
Transcriber: http://www.etca.fr/CTA/gip/Projets/Transcriber/