Recommendations for Natural Interactivity and Multimodal Annotation Schemes

Laila Dybkjær
NISLab, University of Southern Denmark
Campusvej 55, 5230 Odense M
laila@nis.sdu.dk

Niels Ole Bernsen
NISLab, University of Southern Denmark
Campusvej 55, 5230 Odense M
nob@nis.sdu.dk

ABSTRACT
Standards and guidelines for creating natural interactivity and multimodal (NIMM) annotation schemes are becoming vital factors in ensuring usability and re-usability of annotation schemes as well as of the tools supporting the use of annotation schemes. This paper presents and discusses recommendations for the creation, documentation, representation, evaluation, selection, and adaptation of NIMM annotation schemes.

Keywords
Guidelines, annotation schemes, natural interactivity, multimodality.

1 INTRODUCTION

The field of natural interactivity and multimodal (NIMM) annotation covers spoken interaction, gaze, facial expression, gesture, body posture, use of referenced objects and artefacts during communication, interpersonal (physical) distance, etc., and combinations of any of these. Annotation (or coding) schemes in the NIMM area have so far been fairly anarchistic with little standardisation except for subareas, such as speech transcription and facial expression. However, standards and guidelines for creating NIMM annotation schemes more generally are becoming vital factors in ensuring usability and re-usability not only of the annotation schemes themselves but also of the tools which support the use of the annotation schemes.

This paper presents and discusses recommendations for the development and evaluation of NIMM annotation schemes in terms of five points addressed in the following five sections: how to create NIMM coding schemes; how to document NIMM coding schemes; how to represent NIMM coding schemes and annotations in a computer-readable format; how to evaluate NIMM coding schemes; and how to locate, select, and adapt an appropriate existing coding scheme. The proposed recommendations are heavily based on work done in the ISLE (International Standards for Language Engineering) NIMM Working Group [7], cf. [5].

2 CODING SCHEME CREATION

A coding scheme is designed to enable corpus tagging of instances of a particular class of phenomena expressed in one or several modalities. Coding scheme creation involves, at least, conceptual/theoretical work, tag set creation, and coding scheme testing and evaluation. Coding scheme creation often serves a particular initial purpose but this does not exclude that, once created, the coding scheme could benefit other coders and many different coding purposes.

The following rules of thumb address conceptual/theoretical work and tag set creation. Testing and evaluation is discussed in Section 5. The coding scheme creator should at least consider the following points:

- What is/are the coding purpose(s), what will the annotations be used for, etc.
- Which modality/modalities should be marked up;
- Which phenomena are of interest.
- Is the identified class of phenomena sufficient for the purpose(s) for which it is intended.
- Is the class of phenomena kept as general as allowed by the coding purpose(s).
- Often but not always, the class of phenomena to be coded is based on a theory which claims closure for the class, such as, for instance, that the class of phenomena includes all possible, different human facial expressions. This theory needs testing and validation.
- Sometimes the coding scheme is merely intended to capture a subset of some larger class of phenomena for some purpose, such as when speech transcribers use a subset of a larger set of transcription tags. In such cases, there should be clear rules for how to add new phenomena to the coding scheme, should that be needed later, so that these will be consistent with the already existing ones.
- Each phenomenon must be clearly exemplified and described, so that both the coding scheme creator and others are always able to decide, given a certain token in a corpus, whether or not that token is an instance of that phenomenon. This point is crucial to inter-coder agreement on how to apply the coding scheme to a given corpus, cf. Section 5. Lack of clarity and coverage in the description of phenomena translates into reduced inter-coder agree-

ment, reduced consistency of codings, and quickly into a coding scheme which is too unreliable for practical use.

- Each phenomenon must be assigned a syntactic tag whose presence in the corpus, or whose reference to a particular token in the corpus, indicates its presence.
- The tag set representing the relevant class of phenomena should preferably be defined using some kind of standard format for coding tool use, e.g. XML. The tag set to be interpreted by machine does not need to have the same format as the tag set used by the human coder, one-to-one correspondence is sufficient (se also Section 4).
- The tag set should be extensible following well-defined rules.

The guidelines above are closely connected with coding scheme documentation and coding scheme formats, as discussed in Sections 3 and 4.

3 CODING SCHEME DOCUMENTATION

Experience shows that many coding schemes are poorly documented, which makes their retrieval and re-use very difficult. There is not yet any standards as regards which kind of documentation (meta-data) to include with a coding scheme. The MATE [8] and NITE [9] projects have proposed the concept of a coding module which extends the notion of a coding scheme with documentation that should be sufficient for colleagues to understand and use the coding scheme, cf. [3, 4]. At the same time, this documentation is structured in a way which makes it easy to search through if available on the web. The contents of a coding module is listed below:

- Name of coding module
  (E.g. my_gestures.)
- Author(s) of coding module
  (E.g. Tom Jones.)
- Version
  (E.g. v1.2.)
- Notes
  (References to literature, validation information, comments, etc.)
- Purpose of the coding module
  (Description of the purpose for which the coding module was first created.)
- Coding level(s) covered by the coding module
  (E.g. dialogue acts, hand gesture, nose wrinkles, …)
- Description of data source type(s) required for use of the coding module
  (E.g., an orthographic transcription may be a precondition for applying a particular coding scheme.)
- Explanation of references to other coding modules
  (If the coding module assumes that there are references to other levels of markup then these references should be explained.)
- Coding procedure
  (Description of how the coding module should be applied to a corpus in order to produce a reliable coding. The coding procedure is important to ensure the reliability of the coding and thus to its quality. The coding procedure should include, cf. [4]:
  - Description of the coders: their number, roles and required training.
  - The steps to be followed in the coding.
  - Intermediate results, such as temporary coding files.
  - Quality measures (the non-satisfaction of which may require re-coding).
- Coding example showing the coding scheme markup in use
  (This could be a snippet from an annotated file or a constructed example. The purpose is to give users of the coding module an idea of what the markup looks like when applied.)
- Clear description of each phenomenon, example(s) of each phenomenon
  (The descriptions provided are essential to a clear and sufficient explanation of how each concept-tag pair should be applied during markup. Any uncertainty left by the descriptions and examples provided will translate into unreliable coding, inter-coder disagreement, etc.)
- A markup declaration, possibly hierarchically ordered, of the tags for the (individually named) phenomena which can be marked up using the coding module
  (The tag set declaration can be presented in several different ways, e.g., as a DTD, cf. Section 4.)

It takes time to create and document good coding schemes but we believe it is worth the effort. Don't expect that anyone will be able to reliably use a "coding scheme" which only consists of, e.g., a tag set and a sparse description. You may have been able to use it yourself at creation time having it all in your head, but if you want to return to it just a few month later it will not be that easy even for you.

4 CODING SCHEME REPRESENTATION

This section addresses which formats to use for coding scheme representation. We need to distinguish between computer-readable formats and human-readable formats.

As for computer-readable formats, there is a strong trend today towards using XML. Coding scheme definitions are very often provided via an XML DTD (Document Type Definition) or via XML Schemas. We recommend to follow this de facto standard since XML, DTDs and Schemas are machine-readable, extensible, and widespread. Also for annotated data, XML is widely used. This means that using XML for this purpose as well will facilitate the exchange of annotated corpora and the use of tools based on XML corpus representation. It should be noted, however, that XML is only syntax. Translation of tags into the set used by a specific tool may be needed in order to use that tool. Usu-

ally this is still much less work compared to translating a home-grown language into XML, e.g. the same parser tools can be used. For more information on XML, see, e.g., [10].

Whereas XML DTDs and Schemas are excellent for computers, they are less easy to read and write for humans. If tool support is available when one makes a markup declaration, it may be possible to use a format which is more friendly and easy for humans without special programming skills. Behind the user interface, the tool may then, e.g., convert the markup declaration into an XML DTD. To the user, however, the markup declaration may just be in terms of, e.g., well-defined form-filling. The special XML tags are then added by the tool behind the scene.

We recommend the development of tools which facilitate easy indication of markup declarations and support the use of an underlying standard representation format.

5 CODING SCHEME EVALUATION

Coding scheme evaluation follows coding scheme creation and documentation. The purpose of evaluation is to test the quality of the coding scheme and the results produced by using the coding scheme as intended. Precise and informative evaluation results provide very useful information to those looking for an existing coding scheme to use, cf. Section 6.

The coding scheme should be applied according to the prescriptions in the coding procedure, cf. Section 3. Thus, e.g., the annotators must have the background and expertise recommended and the number of annotators prescribed must be used to ensure the quality of the coding.

The ease-of-use and reliability of the coding scheme may be measured by:

- asking coders their opinion (interview, questionnaire);
- checking if different coders use tags consistently;
- measuring the time taken to code;
- measuring the quality of the annotations, cf. below.

Similarly, the ease-of-use of coding tools may be evaluated by asking coders their opinion and by measuring the time it takes them to code. Measuring the quality of codings is also relevant for tools evaluation if markup is done semi-automatically or automatically.

Coding scheme quality is a research area of its own. A coding scheme may be evaluated by:

- comparing different corpus samples coded by means of the scheme to assess *coverage;*
- comparing the results produced by different coders to assess *inter-coder reliability;*
- comparing the results produced by the same coder on the same corpus sample at different times, for instance with a one-week delay, to assess *consistency*.

Coding scheme quality may be evaluated:

- qualitatively through discussion of the choices made by coders when they differ;
- quantitatively through scoring measures.

A frequently used method to compare the results produced by different coders (inter-coder agreement) is called *kappa*:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

P(A) is the proportion of times that the coders agree and *P(E)* is the proportion of times that they are expected to agree by chance. A problem with this method is that there is no sound interpretation of which kappa values are good enough. Moreover, kappa presupposes independent events which is far from always the case in NIMM contexts, see also [2].

Two other measures, precision and recall, may be used if there is an 'authoritative source' to which the codings may be compared. *Precision* expresses the proportion of the occurrences found that have been correctly coded:

$$\text{precision} = \frac{\text{found - incorrect}}{\text{found}}$$

Found represents everything that was marked by the coder. *Incorrect* represents the incorrect markups made by the coder, as determined by the authority.

Recall expresses the proportion of occurrences that have been found:

$$\text{recall} = \frac{\text{all - missing}}{\text{all}}$$

All represents all occurrences present in the corpus, as determined by the authority, and *missing* represents those occurrences that were not identified by the coder [1].

We recommend that:

- any evaluation made of a coding scheme is referenced from the documentation of the coding scheme, so that it is easy to find;
- evaluation methods used and the evaluation process are clearly described;
- evaluation results are clearly documented.

6 CODING SCHEME SELECTION AND ADAPTATION

We have discussed recommendations related to the creation, documentation, and evaluation of coding schemes. However, it is of course much easier if there is already a well-documented and evaluated coding scheme available somewhere which fits the needs one may have. It is better still if this coding scheme comes with tools support.

No matter if one is going to create a coding scheme or select an already existing scheme, one should consider the issues listed in Section 2. Moreover, one should know who will be doing the coding, i.e. which level of expertise is available for this task.

When this is done, we recommend to look for an existing coding scheme which satisfies the identified constraints before a possible decision is made to create one's own coding scheme. Locating existing coding schemes is not necessarily easy to do for the moment since there are many sources which one may consult, including, e.g., survey reports, proceedings of conferences such as LREC, the ELRA/ELDA website [6], and free-style web search.

The checking of which coding schemes exist and what they are meant to be used for could be greatly facilitated if coding schemes are:

- well-documented, following the recommendations in Section 3;
- available on the web in the form of collections maintained at a small number of sites.

Documentation following the recommendations above (Section 3) would also greatly facilitate comparison of different coding schemes.

If one or several coding schemes are found which could be candidates for selection, we recommend to consider at least the following criteria before selection is made, and to weight the criteria according to their importance in the specific case:

- Coding scheme documentation.
- Coding scheme evaluation.
- Coding scheme extensibility, if applicable (Section 2).
- Coding scheme adaptability.

By *extensibility* we mean that new tags and their conceptual descriptions can easily be added. Extensibility becomes easier if the coding scheme includes a description of how this should be done. *Adaptation* of a coding scheme may include coding scheme extension but may also include other forms of changes to the original scheme, such as partial replacement of the tag set, a different coding procedure, or other/more coding files referenced. Whether adaptation - which is typically a larger operation than extending a scheme –is the right choice, depends at least on:

- how many changes are needed to make the coding scheme fit one's purpose;
- how easy it will be to make the adaptation; and
- what will be gained from making the adaptation compared to creating a new coding scheme.

Ease of adaptation depends on the coding scheme itself as well as on the available documentation.

The gain by making adaptation may range from not having to create an entirely new coding scheme and not having to do the coding scheme documentation from scratch, to getting access to tools support which may greatly facilitate the annotation and analysis process. If the gain is small, it may, in fact, pay off to create a new coding scheme instead, one which completely fits one's purposes. Available tools support, on the other hand, is a great advantage and may make adaptation the optimal choice.

7 CONCLUSION

(De facto) coding scheme standards mainly exist for speech and text annotation, especially in the area of transcription, and for media production-related issues. For other NIMM sub-areas, no real standards seem yet to exist. The standards which do exist have typically been brought forward by projects or international groups of people with a shared interest in some area, and sufficient need and momentum to get the consensus-building process started. Most existing standards are accompanied by supporting software, which makes them even more attractive to use since their use is facilitated by the software.

The recommendations for NIMM annotation scheme development and evaluation presented in this paper are based on best practice studies made in the European NIMM Working Group in the ISLE project. We hope that they may serve as a basis for further work in the NIMM annotation area, eventually leading to standardisation.

8 ACKNOWLEDGEMENTS

Much of the work reported above was carried out in the ISLE project. We gratefully acknowledge the support by the European Commission's HLT Programme. We would also like to thank Malene Knudsen, Joachim Llisterri, Maria Machuca, Jean-Claude Martin, Catherine Pelachaud, Montse Riera, and Peter Wittenburg for their contributions to ISLE report D9.2.

9 REFERENCES

1. Bernsen, N.O., Dybkjær, H. and Dybkjær, L. *Designing Interactive Speech Systems. From First Ideas to User Testing.* London, Springer Verlag 1998.
2. Dybkjær, H. and Dybkjær, L. Measuring Transaction Success in Spoken Dialogue Information Systems. *Proc. of the Nordtalk Symposium on Relations between Utterances*, Copenhagen, 2002, 110-131.
3. Dybkjær, L., Bernsen, N.O., Carletta, J., Evert, S., Kolodnytsky, M. and O'Donnell, T. The NITE Markup Framework. *NITE Report* D2.2, 2002.
4. Dybkjær, L., Bernsen, N.O., Dybkjær, H., McKelvie, D. and Mengel, A. The MATE Markup Framework. *MATE Report* D1.2, 1998.
5. Dybkjær, L., Bernsen, N.O., Knudsen, M.W., Llisterri, J., Machuca, M., Martin, J.-C., Pelachaud, C., Riera, M. and Wittenburg, P. Guidelines for the Creation of NIMM Annotation Schemes. *ISLE Report* D9.2, 2003.
6. ELRA: http://www.elda.fr/
7. ISLE NIMM: isle.nis.sdu.dk
8. MATE: mate.nis.sdu.dk
9. NITE: nite.nis.sdu.dk
10. XML: www.w3.org/XML