Annotating and Measuring Multimodal Behaviour – Tycoon Metrics in the Anvil Tool

Jean-Claude Martin (1, 2) , Michael Kipp (3)

(1) LIMSI-CNRS, BP 133, 91403 Orsay, FRANCE
martin@limsi.fr http://www.limsi.fr/Individu/martin
(2) LINC - IUT de Montreuil (Université Paris 8), 140 rue de la Nouvelle France, 93100 Montreuil, FRANCE
(3) DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, GERMANY
kipp@dfki.de http://www.dfki.de/~kipp/anvil

Abstract

We demonstrate how the Tycoon framework can be put to practice with the Anvil tool in a concrete case study. Tycoon offers a coding scheme and analysis metrics for multimodal communication scenarios. Anvil is a generic, extensible and ergonomically designed annotation tool for videos. In this paper, we describe the Anvil tool, the Tycoon scheme/metrics, and their implementation in Anvil for a video sample. A new Anvil feature, motivated by the Tycoon scheme, is presented: non-temporal annotation objects – an important concept, we argue, of general interest. We also outline future plans for automatizing Tycoon metrics computation using Anvil plug-ins.

1. Introduction

The manual annotation of recorded multimodal behaviour on video may have several goals for researchers building multimodal systems: exploring the expected combination of input modalities, collecting training data for classifiers, extracting behaviour rules for modeling an embodied agent or evaluating system performance. Apart from annotated corpora there is a need for algorithms that process such annotations, measuring observed multimodal patterns in order to synthesize the observed behaviour into meaningful figures. Such measures can tell you how redundant the behaviour of a user is, how much s/he relies on one or the other modality or how much s/he switches between modalities. This is useful for summarizing observed human behaviour in a way that allows you to assess the need of multimodality in a particular scenario or to evaluate an existing system by looking at the relationship between multimodality and efficiency and/or acceptance. For the computation of such measures we need suitable tools that allow efficient data collection on one hand and that allow the integration of computational modules on the other hand.

In this paper, we outline the Tycoon framework for measuring multimodal behaviour in HCI research. We use the Anvil tool for the coding of a concrete study on multimodal human-machine interaction. Anvil is a generic multiple-level video annotation software developed for gesture research.

We start by describing the Anvil tool, then the Tycoon metrics and its requirements on annotation tools. We explain how the Anvil tool was extended and how it was applied on a video sample of multimodal communication. We conclude by describing how we intend to bring closer Tycoon and Anvil by exploiting the plug-in facilities.

2. Anvil

Anvil[1] is a generic annotation tool for videos (Kipp 2001a, 2001b). It allows manual coding on multiple levels and the insertion of cross-level and within-level links which makes it a suitable tool for multimodal communications research. One can see it as a graphical notepad for parallel events such as speech and gesture. It is highly flexible because of its XML data encoding and various import/export facilities (PRAAT, XWaves, SPSS). It is extensible because of its new plug-in architecture where Java programmers can connect custom-designed modules easily. A set of annotations can be conveniently maintained and browsed with Anvil's project tool, mass data can be exported to SPSS readable format.

The Anvil system is written in Java using the Java Media Framework (JMF), and runs under Windows, Linux, Solaris and Macintosh OS X. It is freely available for research purposes.

2.1. Anvil Concepts and GUI

Transcribing behavior on various layers or levels is common practice since the 70's (cf. Ehlich 1992). In Anvil, a layer is called a *track*. In a track, the user inserts so-called track elements (e.g. a word, sentence, gesture), each one containing typed attributes. Attribute types are: String, Boolean, Number, ValueSet (a set of user-defined tokens) and Link (links are explained in Section 2.2). *Primary* tracks refer directly to the timeline of the video insofar as each element has a start and end time. *Secondary* tracks refer to an arbitrary other track, the so-called reference track: the span of a secondary track element is then defined by a start and end element of the reference track. For instance, a "sentence" track would contain elements whose span are defined by a start/end element of the "words" track.

All tracks, attributes and attribute types must be defined by the user before annotation. This is essentially what one would call an annotation or coding *scheme*. In Anvil, the coding scheme takes on the form of an XML file, called *specification file* (see Figure 2).

Ergonomics is a much neglected factor in tool design, though it is highly relevant to mass data annotation to reduce costs of creation (training time and coding time) and revision of corpora. In this regard, Anvil's graphical user interface (GUI) allows a fast and efficient access to all the tracks (see Figure 4). The annotation board (lower left window) grants a timeline view on the tracks where time is translated to the x-dimension. A playback line shows the position of the video and can be dragged to

[1] Annotation of Video and Spoken Language

control video playback. With the help of this line, together with popup menus and keyboard short-cuts the user can quickly add and modify track elements. The audio waveform can be displayed on demand, the pitch contour can be imported from PRAAT, to support linguistic annotation.

To increase chances for consistency of annotation, Anvil can automatically generate coding manual pages in HTML from documentation tags in the specification file. Also, during coding the user can pop up online documentation windows describing the usage of attributes and values.

The ergonomic design of Anvil has never been quantitatively evaluated but it would be a desirable future task to compare different tools in terms of efficiency and easy usage.

2.2. Anvil Links

Relations between elements of different tracks can have different forms. The hierarchical or subsumption relationship is captured by Anvil's notion of primary and secondary tracks. Primary track elements are anchored in time, secondary track elements are anchored using another track, subsuming a number of consecutive elements in this other so-called *reference* track. The relationship between elements consists of implicit links between secondary track element and the respective reference elements.

What we actually call a *link* in Anvil an explicitly visible pointer or rather a set of pointers. They are stored in an attribute of a track element. Anvil knows three attribute types dealing with links:

1. MultiLink: The attribute can take links to any element of any track.
2. MultiLink(**T**): The attribute can take links to any element of track **T**.
3. ReciprocalLink(**A**): the attribute can take links to any element **E** of any track; backlinks will be automatically inserted in attribute **A** of **E**.

2.3. Anvil Plug-in Interface

The plug-in interface implements a concept inspired from database design and transferred to linguistic annotation by Bird and Liberman (2001). The idea is to isolate the part of a tool that represents and maintains the annotation and to open this part (i.e. classes and methods) to other programmers via an application programmers' interface (API). In database design this part is called the *logical level*, as opposed to the physical level (data files, XML or ASCII) and the application level (visualization, coding, browsing, analysis tools). Anvil offers such an API for Java programmers whose programs can access methods for the creation, modification and storage of annotations, in other words: it allows access to Anvil's internal object structure of annotations. On top of this "minimal functionality" suggested by Bird/Liberman (2001), Anvil offers methods to access its GUI, so the user can connect his/her own graphical components to Anvil's GUI. For synchronizing a plug-in with the video, Anvil provides an event-based interface.

A Java program can be registered as a plug-in via Anvil's GUI. After registration, the plug-in will appear in Anvil's "tools" menu and can be started by mouse-click.

Possible applications for plug-ins are bootstrapping modules, spectrographic analysis[2], spatiotemporal annotation using graphical overlay[2], statistical analysis modules. Tycoon metrics will be computed using a plug-in (see Section 6).

3. Tycoon

3.1. Tycoon Definitions

Tycoon is a framework for the study and development of multimodal systems (Martin et al 2001). Although it has been used for software development, we will focus on its use for measuring the multimodal behavior of recorded human sujects.

Tycoon includes a typology of basic modality independent *types of cooperation* between modalities. These types of cooperation are defined as follows. "Equivalence" means the recorded subject sometimes uses one modality (e.g. speech) and some other times another modality (e.g. hand gesture) to convey the same information. "Specialisation" means the subject always use the same modality to convey a specific kind of information. "Complementarity" means the subject uses different modalities to convey different chunks of information which have to be merged. "Redundancy" means the subject uses different modalities to convey same chunks of information which have to be merged. Rates are associated to measure the use of each of these types of cooperation.

The framework also includes the notion of *referenceable objects.* Such an object might be a graphical object the user is able to refer to using speech or gesture.

3.2. Tycoon Metrics

The Tycoon metrics are composed of the computation of the following rate of use of possible types of cooperation: specialisation, equivalence, complementary / redundancy.

Annotation of the features (type, color…) of the objects the subject has refered to, as well as monomodal referential behavior via the association (i.e. amount of the object's features provided via speech, ambiguity of deictic gestures).

They were first manually applied in a multimodal Wizard of Oz experiment (Cheyer et al 2001). Then a sample coding scheme has been specified as a XML DTD and a Java software for parsing such annotations and computing preliminary version of these metrics was developed and applied to video samples (Martin et al 2001). Yet, annotation was still done manually with a text editor on the basis of the observation of digital video tapes using a VCR (ie. without any timestamp and real automatic means to check the link between annotation and video of for searching isolated multimodal patterns).

4. Anvil meets Tycoon: Annotation of References to Non-temporal Entities

The generic layout of the Anvil tool allows for customization to the Tycoon framework in order to adress the two issues of (1) efficient manual annotation and (2) automated computation of Tycoon metrics.

[2] visit http://www.dfki.de/nite/

The Tycoon scheme led to an important extension of Anvil, available in its latest version 3.6: the *set*. A set is similar to a track in that it can keep annotation elements but these elements lack any temporal information. This is of general interest because it allows an intuitive encoding of real-world objects and abstracts relations that can be linked up with temporally anchored track elements.

Tycoon's coding scheme contains a set of world objects which can be referenced by speech or gesture. These objects differ from words and gestures in that they usually exist for the duration of the whole discourse, whereas words and gestures have a start and an end point in time. Though it would be possible to create a whole track for such an object and let the object fill the length of the track, this solution seems cumbersome and unintuitive. Instead, one could let words and gestures refer to an ID (a simple string). Then what else would this ID be but a representation of the object, certainly a crude and improvised one? One that does not allow to encode properties of the object like we can in track elements.

A better solution is to introduce a new track type that contains non-temporal elements. Or to be precise, a track is generalized to be a *container* of elements. A new subtype of this container is what we call the *set* (Figure 1).

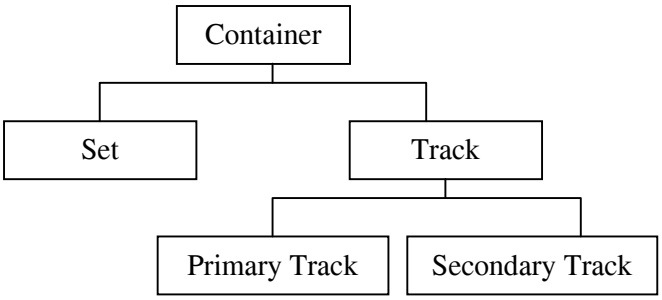

Figure 1: Container classes.

Set elements can now be used to represent real-world objects and locations. They contain typed attributes just like track elements where (non-temporal) properties can be stored. Track elements like words and gestures, which are time-anchored, can refer to these via links.

Another application is the representation of abstract concepts, e.g. the congruence relation of postural elements (mirror, anti-mirror, no relation). For this, the set elements could contain links to the temporally anchored postural elements.

This feature is certainly of general interest as many multimodal scenarios deal with real-world objects. It offers a convenient way of representing co-reference. The more complex the real-world objects become, the clearer the advantage of this feature as opposed to using simple IDs. In Anvil's GUI, a set is displayed as a table where each line contains one element and the columns contain the different attributes (see Figure 4, lower right window).

5. Application to the annotation of a video

We have applied the extended Anvil tool to a video of one student explaining to other students a schema drawn on the blackboard (see video still in Figure 4). The explanations are about the internal and external components of computers.

In this corpus sample, the video-taped student makes use of several modalities: speech, hand gesture, gaze. He refers with these several modalities to several objects drawn on the blackboard: central unit, screen, keyboard, loudspeakers, printer, scanner, webcam.

The questions we were willing to tackle with the annotation of such a corpus were the following. What is the relationship between each monomodal segment and the features of the refered object? How much multimodal is the subject's behavior? What is the rate of use of each modality and modality combination ? Can the cooperation between these modalities be qualified of equivalence / complementary / redundant / specilisation?

As it can be seen in Figure 4, we have specified three tracks in the annotation scheme: spoken words, hand gesture and gaze. For Anvil, the scheme was defined in an XML specification file (Figure 2). Besides classical track specification for each modality, an objects set specification is also used. These objects have the following features: label, size, shape, position. Note that all of these features remain constant for the duration of the video which is why they can be encoded in the non-temporal set type (cf. Section 4). In order to enable the future computation of multimodal metrics, a ReciprocalLink has been specified as an attribute to the hand gesture track to allow the linking of annotations of hand gestures to objects drawn on the screen (Figure 3).

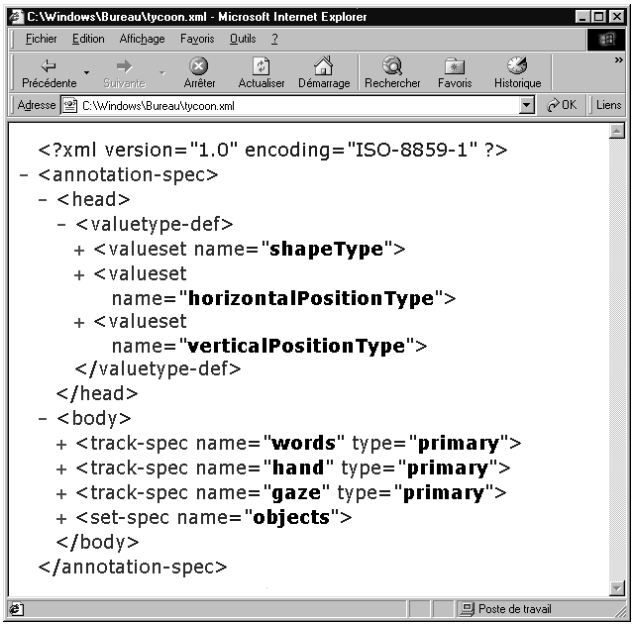

Figure 2: the XML specification of annotation format.

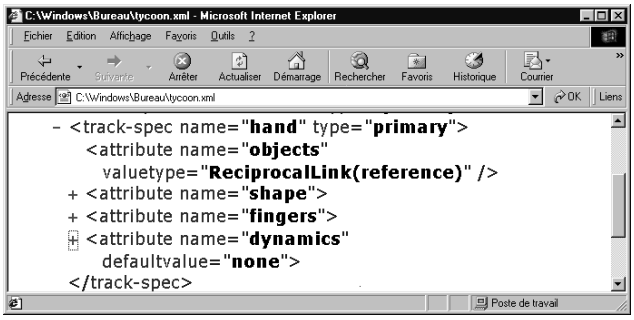

Figure 3: The hand track contains a link towards the objects set.

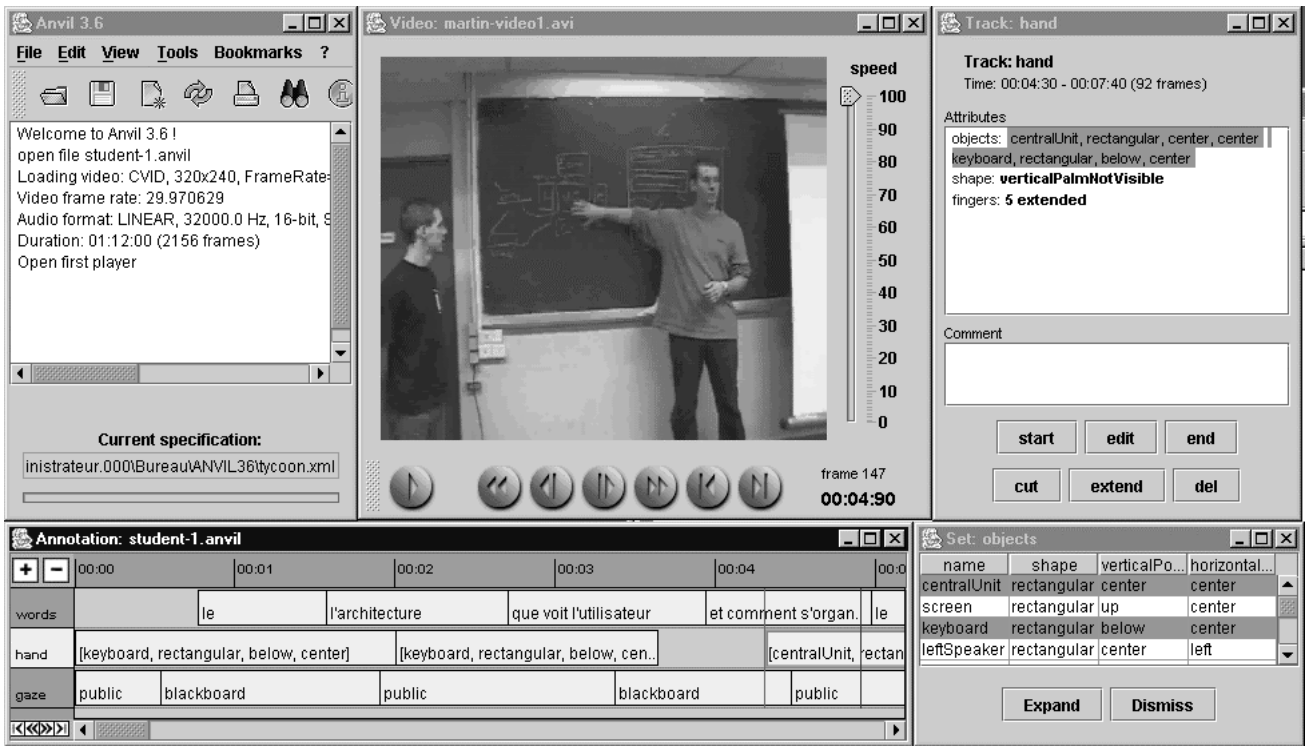

Figure 4: Screenshot of the annotated example. The annotation scheme (lower left window) contains 3 tracks (spoken words, hand gestures, gaze). The new Anvil features enables the annotation of objects referred in gesture and speech (lower right window).

Spoken words have been manually annotated using the PRAAT[3] tool and imported in Anvil. Gaze and hand gestures have been manually coded in Anvil. Using Anvil's graphical user interface during the annotation process, hand annotations have been linked to elements of the object set (Figure 5).

6. Conclusions and future directions

In this paper we described the Tycoon framework by a sample annotation that was done using the Anvil annotation tool. Tycoon's annotation scheme motivated the extension of Anvil by non-temporal entities such as referred objects. We believe that this extension as well as Tycoon's metrics of measuring multimodal behavior are of general interest to the multimodality community.

In the future, we intend to use the Anvil's plug-in interface to integrate the Java software module computing Tycoon metrics already used in (Martin et al. 2001).

Yet, some issues still need to be solved in order to fully integrate Tycoon and Anvil. In our previous work on human-computer interaction, multimodal segments were manually and easily segmented (ie. they were one section in a XML annotation). In the current state of the Anvil/Tycoon cooperation, the output annotation file is divided into tracks each containing annotations in a temporal order. Manual solutions (adding a multimodal segment track with links to each modality annotation) and automatic solutions (using timestamps to group annotations into logical multimodal segments) will be tackled.

Regarding the metrics themselves, they could be moved from discrete metrics into continuous metrics. In order to compute continuous rates of redundant ot complementary behavior, the Tycoon frameworks requires the attribution of a salience value during multimodal reference. We will study how the current Anvil/Tycoon can be extended to enable the assignment of a value to links between annotations and objects.

In order to reach a better coverage of temporal features of multimodal behavior, the hand gesture annotation could be extended with the annotation of preparation, stroke and retraction phase. The resulting tool will also be tested with other corpora.

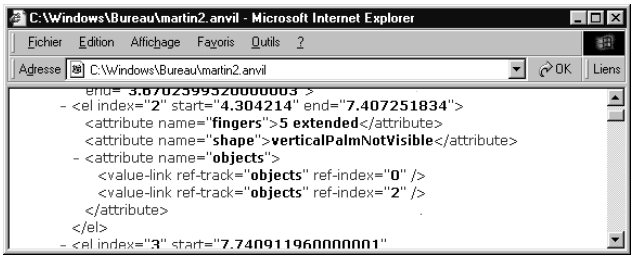

Figure 5: Output file of Anvil where one of the hand annotation has been linked to 2 potential refered objects (objects of index 0 and 2 in the object set).

[3] developed by Paul Boersma and David Weenik. For information, contact Paul Boersma: paul.boersma@hum.uva.nl

7. References

Bird, S. & Liberman, M. (2001) A Formal Framework for Linguistic Annotation, in: *Speech Communication* **1-2**, pp. 23-60.

Cheyer, A., Julia, L. & Martin, J.C. (2001) A Unified Framework for Constructing Multimodal Experiments and Applications, in: *Cooperative Multimodal Communication*. Bunt, H., Beun, R.J. & Borghuis, T. (Eds.). Second International Conference, CMC'98, Tilburg, The Netherlands, January 1998. Springer. LNAI2155.

Ehlich, K. (1992) HIAT – a Transcription System for Discourse Data, in: *Talking Data: Transcription and Coding in Discourse Research*, J.A. Edwards & M.D. Lampert (eds.), Hillsdale: Erlbaum, pp. 123-148.

Kipp, M. (2001a) Anvil - A Generic Annotation Tool for Multimodal Dialogue, in: *Proceedings of Eurospeech 2001*, Aalborg, pp. 1367-1370.

Kipp, M. (2001b) From Human Gesture to Synthetic Action, in: Proceedings of the Workshop on "Multimodal Communication and Context in Embodied Agents" (Agents-2001), pp. 9-14.

Martin, J.C., Grimard, S., Alexandri, K. (2001) On the annotation of the multimodal behavior and computation of cooperation between modalities, in: *Proceedings of the workshop on " Representing, Annotating, and Evaluating Non-Verbal and Verbal Communicative Acts to Achieve Contextual Embodied Agents ",* May 29, Montreal, Fifth International Conference on Autonomous Agents. pp 1-7 http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-46/