INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

# OBSERVATION, ANNOTATION AND ANALYSIS OF DESIGN ACTIVITIES: HOW TO FIND AN APPROPRIATE TOOL?

**Rahi Rasoulifar[1], Brigitte Meillon[2] , Guillaume Thomann[1], François Villeneuve[1]**
(1) G-SCOP Laboratory, Grenoble Institute of Technology, (2) LIG Laboratory – University of Grenoble

## ABSTRACT

Recently, design researchers have been interested in observation-analysis approach to study design activities. This is how the annotation and annotation tools have been introduced into the design research. This paper describes the annotation process as an approach in design activity studies and proposes a sequential structure for the annotation process. The research objective is to characterize an ideal annotation tool, designed and adapted for design studies; with this objective in mind, the paper presents several design researches in the context of collaborative design and user-centred design, using video annotation tools, and explains how these tools have been employed. We describe the evaluation criteria for annotation tools and provide a functional comparison with the explained tools. We had explored the existing functionalities of different software to extract a list of desired characteristics of an ideal annotation tool for the design research and studies. Such an effort can provide a guideline in video annotation for design researchers.

*Keywords: video annotation, design activity, observation, analysis, annotation software*

## 1 INTRODUCTION

Observation of design activities takes a holistic view of the relationship between designer communication, designed artifacts, the socio-cultural context of the design situation, the position of the observers and the technique they use for the observation. While the observation-analysis studies are developed outside the design studies and mainly in psychology and natural interactivity sciences, their approaches and results did interest the design researchers in different aspects like argumentation analysis and human machine interactions. To be able to capture the information concerning the design interactions, different observation techniques were used. Among these various techniques, video recordings have been used more extensively in experiences. Accordingly, many tools and software were developed to facilitate the manipulation of streams and also to provide some digital annotations on videos.

Unlike the natural interactivity community, design interaction fully recognizes itself as a “design discipline”, in that the objective is to create a new system or to change existing interactive structures for the better [1]. There is an increasing interest in design community to observe and analyze design activities in industrial situations, student experiments and international research workshops. Although there is no generally accepted approach or research methodology, each study uses it’s own “observe - hypothesize - analyze – verify” method to obtain results. In this perspective, analysis used to be generally presented through the additional information and interpretation annotated on the media (ex. Video). *Annotation* and *coding* are the terms generally used for this process in the community.

Since the field of video annotation and analysis is growing rapidly in scope and importance, there has been an increasing need to expand, further develop and professionalize annotation tools for design research. Our research objective behind this paper was to explore the annotation activity in design studies in order to identify the logic of the process and the main steps. Then, we looked forward to an ideal annotation tool (preferably in form of software) to help design researchers in coding their observation and refine their analysis. Evaluating existing tools and the reported experiences in the design context and including what exists currently and what will be available could be useful in this process. Moreover, a guide comparing software programs' capabilities and performances and tool limits could give researchers useful information when they select a tool for future research.

In this paper, we will explore the background of observation, annotation and analysis, then review four annotation tools and their use in a design project. In section four, we define the main evaluation criteria of an annotation tool and compare the tools mentioned above. Finally, in section five, we discuss an ideal tool according to the identified design research requirements.

## 2 VIDEO ANALYSIS AND THE DESIGN PROCESS

Observation, annotation and activity analysis originated from the cognitive sciences such as psychology, sociology and natural interactivity. Observation process used to be direct, placing the researchers in the environment and their observation tools were simply pens and notebooks. The development of new technologies changed observation: video recording and analysis become the main observation method of many scientific researchers in these fields. Design as a human activity includes social aspects, so that the observation has become an important technique in design studies. Many design observation studies are published on various subjects such as: design communication [2], knowledge interaction [3, 4], validation hypothetic models [5, 6] and design education [7, 8].

Design activity as a data source provides a multitude of information about the design phases including design-thinking, decision-making, collaboration using intermediate objects and design evaluation. External activities can be observed, recorded, archived for analysis, etc. whereas external imperceptible activities are much harder to capture [9]. Researchers used to collect the design project archiving and reports. They used questionnaires, interviews and other types of surveys to communicate with the designers. The deeper and more complex the research question was, the more data were needed. The method, and consequently**,** tools for collecting and capturing the data bring new approaches to this research.

Tang proposed recording design activities in his Observe-Analyze-Intervene methodology [10]. Video recordings provide maximal data on the subject and the situation. It can be replayed and reinterpreted, and give access to behaviors and interactions that could have been missed by direct observation. It can reduce observers' bias and corroborate better the results [11]. Thus, video recording is being used widely for research on design in research workshops. Technology advancement facilitates the use of video streaming. Real-time audio and video recordings, known as protocol analysis, have widespread use in academic research. For instance, in an international workshop held in Delft, The Netherlands, researchers analyzed video recordings of designers working on an engineering product design [12]. Moreover, the thematic of a workshop held in Lulea, Sweden, concerned capturing the design-activities of a group of students in various situations and practices.

Video recording analysis requires an analytical framework. That framework is based on a theory which explains the classes of phenomena and their interrelations [13]. Based on this framework, a coding scheme should be developed to categorize and classify the happenings. Annotation is consists in adding data synchronized with the video stream and allows analysis of the happenings captured by the video recording.

Numerous studies have demonstrated that annotation is an important part of human reading behavior in both printed and digital environments [14]. Annotation in the electronic environment requires special support due to limited media features. Many tools were developed to support the requirements of researchers and some studies were carried out to evaluate the functionality and advantages of their annotation tools [15, 16]

An increasing demand concerning better tools for handling and analyzing video requires special attention, thus the new tools would have appropriate functionalities and meet users' needs. However, comparatively little research has been conducted to understand the specific needs of researchers for annotating videos, in order to develop tools to support their needs.

Surprisingly, there are too many tools developed for video annotation, since almost all the tools were created, used, tested and multiplied without connection with design research. In order to be successful in the design context, video analysis tools must support a wide variety of design tasks and analysis styles, and support design objectives. We need to gain new insights in the way the designers progress in the design process. Little is known about the use of such tools in specific behavioural contexts and more still needs to be explored. The annotation process is explained in the next section.

## 3 CONFIGURATION, CODING AND ANALYSIS

In its very basic form, the annotation process consists of three steps: configuration, coding and analysis. One can claim that the analysis is not an internal annotation process, but what we are explaining here is the process used to learn more about the design actors, objects and activities during a design situation.

While the observation is the entry of this process, we argue that the observation can be influenced by the researcher knowledge and experience in annotation. Observation provides raw material for the research; the richer the video capture is, the more profound the analysis could be.

### 1. Configuration (the analysis framework)

Based on the kind of data an analysis seeks to show, the type of input data and its categorization should be defined accordingly. This categorization has two main parts: spontaneous happenings and their interpretation. A happening is a set of actor, activity and object that can be clearly recognized, such as: "Alain draws a sketch" or "Bob asks a question". On the contrary, an interpretation comes from the study hypothesis like "problem explanation" or "understanding new solution". In other words, this classification called the *framework* includes explicit and implicit classification.

A happening can be defined as an *event* by an occurrence time, or as an *interval* by the start and stop times. Framework presents spontaneous happening through the subjects, objects and actions. For example, commonly in design studies, such a framework presents typical elements including actors, intermediate objects and interaction between actors. This categorization can be as detailed as the research needs and is supposed to follow a logical classification. Figure 1 shows the typical examples of classification.

<table>
<tr><td>Actor 1</td><td rowspan="3" colspan="2">Activity 1, Activity 2, Activity 3</td><td>Phase 1</td><td rowspan="3" colspan="2">Happening 1, Happening 2, …</td></tr>
<tr><td>Actor 2</td><td>Phase 2</td></tr>
<tr><td>Actor 3</td><td>Phase 3</td></tr>
<tr><td>Activity 1</td><td rowspan="3">Actor</td><td rowspan="3">object</td><td>Object 1</td><td rowspan="3">Actor</td><td rowspan="3">Activity</td></tr>
<tr><td>Activity 2</td><td>Object 2</td></tr>
<tr><td>Activity 3</td><td>Object 3</td></tr>
</table>

*Figure 1-Various types for coding scheme, according to the observation focus*

Configuration is the act of modifying the software options to their nature, number and chief characteristics to realise the framework classification. The result of configuration provides a guidance table called *coding scheme.* The coding scheme characteristics depend on the annotation software, including the level of details, the interrelation between elements and being flexible or predefined.

This is the most essential and conceptual step for the annotation. However, the annotation process is iterative; rarely a first coding scheme is capable to classify and distinguish all the happenings of an observation.

### 2. Coding

Coding is the software facility to assign a defined happening from the coding scheme to the video stream. Thus, by coding, the user marks – event or interval – on the stream and adds the appropriate data. Technically, the coding is performed by stopping the stream and entering a comment or clicking on the coding scheme to assign an item. Depending on the annotation software, coding could be restricted to the coding scheme or be open to create and add a new class. Some software allow the coding scheme to be changed once the coding is started while the others don't. This capability helps the annotation to evolve during the process instead of restarting each time.

Furthermore, the annotation tool sometimes provides a space for adding non-categorized data during the coding. For example, in the case of verbal annotation, the user may prefer to write down the whole sentence or to extract a phrase expressed in the film.

Almost all annotation software use a time line to provide the coding space. This idea comes from the video edition software like Adobe Premiere or Windows movie maker. In some cases, there is no visualization of the timeline (horizontal line), but the each coding tag appears one after the other in a column. While the horizontal presentation of the timeline is simpler to understand and more ergonomic, some find the other one more functional in practice.

The other important concept in coding support is what we call *coding channel*. Coding channel is the coding dedicated to one category of coding scheme. It means the user can decompose the happening into different independent channels (naturally following the coding scheme) and focuses only on one category. For instance, the user defines the verbalisation channel and codes the categories of the verbalization such as asking a question, answering a question, making a suggestion, aside from the other happenings. Thus, with the coding channel support, the user can decompose the annotation into different channels (like verbalisation, hands movement, use of objects, etc.) and code them separately.

The main advantage of the coding channel is that such a support provides the possibility of annotation using several people with different expertises. This allows to go deeply into subcategories and to code the happening as much as possible.
Only a few annotation software provides such a facility. Technically, this option also allows the same channel to be coded by different persons, in order to evaluate the reliability of the coding procedure. Few recently promoted software have a feature to show the agreements and disagreements of the different user coding.
By the way, this step is the most time consuming part of the process. Coding a five minute movie, even a low detailed coding scheme, can take almost a week. As mentioned, it is very common to start with a first coding scheme and evolve it during the coding. It costs a lot of time and patience, particularly when the annotation software don't support the coding scheme modification and oblige to restart the coding.

## 3. Analysis

Once the coding is done, all annotations are recorded in a log file, with each item placed in a different column identified by the time code and added comment. Thus, the analysis can have the outputs such as tables of frequencies, duration, interaction matrix and transition matrix. The log file can be directly viewed, exported to the data analysis software (e.g. SPSS) or manually modified.
Only a few software have some basic statistical analysis tools (like graphs) while the others just make an export of data. The export format is very different from one software to another, and there is no standard on this issue. In most of cases, the user needs to manipulate the data himself to extract some analytical results.
Qualitative analysis has been used in design study issues much more than quantitative analysis, which somehow explains the lack of motivation in tool development. In most studies, a graphical representation of the annotation or a result table has been supposed adequate.
Reliability analysis is the solution to avoid observers' bias. According to the study goals, the annotation should be done by 3 or 5 different persons and a certain percent of agreement supposed to be reached. A few software provide systematically this facility by comparing records of different coders and list the agreements and disagreements between them.

One excellent type of export provided by one annotation tool is the video extract export. This means the user can select a coding type, like "actor A uses a blackboard" and the software exports a video file of the montage of all the scenes where actor A was coded as using the blackboard. This software provides also a multi-criteria selection like "actor A uses blackboard" and "actor B takes notes", and so extracts a movie made of the concatenation of all the sequences concerned by the correlation of these events. This facility is very helpful in qualitative studies and analysis.

These three successive steps build the body of the annotation process. The entry to this process is the observation capture, in different forms of recordings. Since the focus of this paper is on video recordings, the last issue to discuss here is the data import function of annotation software.

### 4. Data entry

One of the biggest drawbacks of many annotation tools is that they are very limited in data import and accept one or very few video formats. Considering the movie format converting and encoding configuration as a professional knowledge which the researcher is not supposed to deal with, the data import is a serious limitation for usage.

Moreover, most of annotation tools do not support multi-streaming and can import just one video. This limit is also important, because researchers need more and more to have multiple views on the design session to capture more about what is happening.

## 4 OBSERVATION ON DESIGN STUDIES

In this section, four experiences of annotation in the context of design study are reviewed. The studies have been selected from different contexts, such as collaborative design and user-centred design. The cases have been selected due to their relevance to the design research in the regional design community. Thus, each annotation tool is explained through a study case to provide a clear picture of how and why the tool was used. Some advantages and disadvantages are listed. The goal of this review was not to compare these software and select the best one, which by the way is not possible, but to show the capabilities and limits in design research. Nonetheless, explaining the annotation tools through the annotation experiments seems to provide a good overview of tools' capabilities and possibilities.

### 1. VideoGraph

Videograph is a free tool, windows platform, which enables the construction of observation categories and rating scales which the viewer can use as a "measuring instrument" to analyze the contents of the video.

This tool have been used for a research study, following the scope of the international workshop DTRS7, which was held in London in September 2007 [17]. The data obtained from the workshop includes: DVDs of four meetings, transcriptions, materials used during the meetings, role description of all participants along with the seating plan. The aim of this research was to see if it was possible to keep the trace of the design meetings by intermediary objects. As the first objective, the researcher needed to become familiar with the data and to have an overview of the meeting. Then, a deeper analysis was realized in order to identify the intermediary objects and their activity level.

A coding schema of all objects observed during the meeting was developed. Each object was labelled "active" as long as it was used or produced by any one of the participants. By coding the video, it became possible to see which object types were more active than others, at different phases of the meeting (see Figure 2)

Videograph was also used for a detailed analysis of a relevant meeting sample to identify all kinds of visible interactions that engineers could have with an object. So, there were two analyzing axes. The first one was the objects which are active on the analyzed period, and the second, the interactions of the actors with those objects.

From this experiment, Videograph is easy to use; it is possible to redefine, modify or delete coding variables within the program while it is running. Using different colors for each category makes the coding and also analyzing process easier. It is possible to put some notes on the time interval, which helps the practice of coding. Once the coding is finished, the data sets are shown graphically on the screen and can be transferred to a spreadsheet software like Excel. This Excel format gives Start and End point of coding sequences. However, there is no possibility to make sub-categories. The software does not support the multi-streaming. The extracted data are not easy to manipulate.

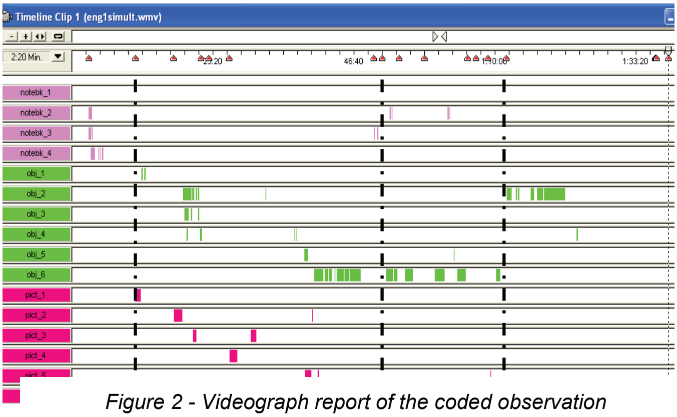


*Figure 2 - Videograph report of the coded observation*

## 2. Anvil

Anvil is a research tool for audio video annotation, written in JAVA language by Michaël Kipp in 2001 [18]. This tool has been implemented to study multimodal information such as gesture, speech or any other visual or auditory signal coming from a digitized audiovisual stream. Anvil's overall design is object oriented. The coding scheme has to be written in an XML specification file, according to a formal description of the tracks, elements, attributes and their possible values.

In a research project on design of a surgical simulator in Laboratory of Informatics of Grenoble (LIG), Anvil was selected for annotation and analysis of the surgeon's hand movements. The goal of the research was to decompose the operational techniques into the simple hand gestures and movements.

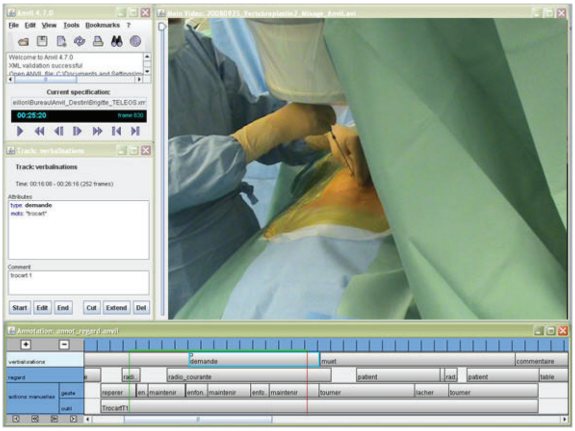


*Figure 3 - Anvil coding screen*

In Anvil, the concept of coding is defined by temporal "tracks", for instance a gesture, verbalizations, and gaze. Tracks can be independent or related to a "reference track". Each track contains several "elements" (e.g. question, request, and order for the verbalization track) and each element can hold a number of "attribute-value" pairs. It's also possible to define non temporal tracks (called "sets"), to provide a list of objects that can be linked to an element. For example, the Object "set" (such as radio, patient, table, colleagues) can be linked to the Gaze "element" in an operating room. Figure 3 shows an extraction of the Anvil interface.

As it was mentioned, the coding scheme should be developed separately in an XML file. Although some templates are provided in the software library, the XML coding is not obvious to make for a

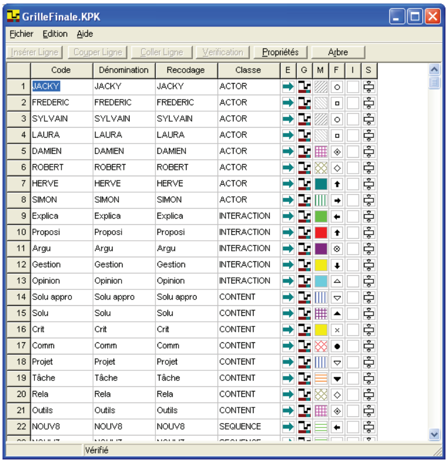
design researcher. The graphical user interface of Anvil is user friendly and it's quite easy to perform annotations through guide windows once the specification file is written.
An important feature of Anvil software is the possibility to gather several annotation files in a project (the concerned annotations files must have the same coding specification file) and search for a specific track over all the files. The output file resulting from an annotation process is called the "annotation file" and is provided in XML language.

Anvil is a convenient tool for video annotation. It is platform independent and based on XML, so that people can use all the free tools provided for the manipulation and conversion of XML files to analyze their annotation data. Anvil gives the possibility to register external "plug-in" and some of them are available on the Web.
However, the input video format in Anvil is very limited and unfortunately the common MPEG format is not supported.

## 3. Actogram (Kronos)

Actogram is a Windows software for chronological observation treatment. In this software, the events are considered like the state generator for a class, thus the categorization is up to the user based on the description protocol (framework) [19]. Actogram is selected for a research project in ICAR research center, Lyon, France, in which the dynamics of argumentation was investigated in a design situation. The main goal of the project was to identify the convergence factors to a new solution. The corpus of a design experiment was selected for the annotation and analysis.

Actogram was used in this research aiming to find the interaction between different kinds of objects like people, tools, in order to describe what is happening during the session. Thus, by making the coding scheme and the coding, Actogram helped to decompose the complex situation into the detailed description. Once the coding completed, the researchers made some hypothesis about the patterns of happenings. They used the Actogram for first analysis for the hypotheses evaluation. Figure 4 shows an example of the software workspace.



|  | Code | Dénomination | Recodage | Classe | E | G | M | F | I | S |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | JACKY | JACKY | JACKY | ACTOR | ➡ |  | ▨ | ○ |  |  |
| 2 | FREDERIC | FREDERIC | FREDERIC | ACTOR | ➡ |  | ▧ | ▫ |  |  |
| 3 | SYLVAIN | SYLVAIN | SYLVAIN | ACTOR | ➡ |  | ▨ | ○ |  |  |
| 4 | LAURA | LAURA | LAURA | ACTOR | ➡ |  | ▧ | ▫ |  |  |
| 5 | DAMIEN | DAMIEN | DAMIEN | ACTOR | ➡ |  | ▦ | ◈ |  |  |
| 6 | ROBERT | ROBERT | ROBERT | ACTOR | ➡ |  | ▩ | ◇ |  |  |
| 7 | HERVE | HERVE | HERVE | ACTOR | ➡ |  | ■ | ↑ |  |  |
| 8 | SIMON | SIMON | SIMON | ACTOR | ➡ |  | ▥ | → |  |  |
| 9 | Explica | Explica | Explica | INTERACTION | ➡ |  | 🟩 | ← |  |  |
| 10 | Proposi | Proposi | Proposi | INTERACTION | ➡ |  | 🟥 | ↑ |  |  |
| 11 | Argu | Argu | Argu | INTERACTION | ➡ |  | 🟪 | ⊙ |  |  |
| 12 | Gestion | Gestion | Gestion | INTERACTION | ➡ |  | 🟨 | ↓ |  |  |
| 13 | Opinion | Opinion | Opinion | INTERACTION | ➡ |  | 🟦 | △ |  |  |
| 14 | Solu appro | Solu appro | Solu appro | CONTENT | ➡ |  | ▥ | ▽ |  |  |
| 15 | Solu | Solu | Solu | CONTENT | ➡ |  | ▦ | ▲ |  |  |
| 16 | Crit | Crit | Crit | CONTENT | ➡ |  | 🟨 | × |  |  |
| 17 | Comm | Comm | Comm | CONTENT | ➡ |  | ▩ | ● |  |  |
| 18 | Projet | Projet | Projet | CONTENT | ➡ |  | ▥ | ▽ |  |  |
| 19 | Tâche | Tâche | Tâche | CONTENT | ➡ |  | ▤ | ▼ |  |  |
| 20 | Rela | Rela | Rela | CONTENT | ➡ |  | ▩ | ◇ |  |  |
| 21 | Outils | Outils | Outils | CONTENT | ➡ |  | ▦ | ◈ |  |  |
| 22 | NOUV8 | NOUV8 | NOUV8 | SEQUENCE | ➡ |  | ▤ | ← |  |  |

Vérifié

*Figure 4 - Actogram coding screen*

This software is simple to use, and particularly suitable for making the coding scheme. In the coding phase, the right-click function helps the process to be fast. The software supports the multi streaming and shows the coding scheme patterns during the coding phases. For the analysis, the result presentation is very easy to parameterize, and is not restricted. The software is very useful in pattern research.
However, the software didn't show enough stability and happened sometimes to close suddenly. Stream entry has a strict format and coding scheme entry has some difficult obligations. The video is

not linked up to the coding, thus the user should replay manually and find the appropriate time to see the happening. The coding page doesn't have the timeline presentation which can be considered as a deficiency**.**

## 4. The Observer XT

The Observer XT® is a commercial product from Noldus Information Technology, Netherlands, available only on Windows platforms. This software is a professional software package for the collection, analysis and presentation of observational data and has the capability to define independent variables and link them to a specific observation. In a research project on design process of innovative surgical instruments, the surgeon-designer collaboration was studied during the development [20]. That study went through the research steps: the observation in the operating room, data capturing, and analysis and interpreting of the recordings. The main goal of the study was to extract and analyze the communication between the surgeon and the engineer in the operating room, for which the discussion, use of design artifact and use of operating room equipments had to be annotated.

The Observer XT was selected according to the primary qualification criteria of multi-stream support and analyzing capabilities.

Coding scheme in The Observer has three categories: Subject, Behavior, and Modifier. There are no limits in making subcategories, and the (sub)category elements can be grouped, as shown in the Figure 5. For example, "Dr. Smith moves his right hand down during the operation pushing down a screw" can be coded as follows:

Dr. Smith > Surgeons > Subject
Push down > Right hand movements > Hand movements > Behavior
Screw > surgical tools > Modifier

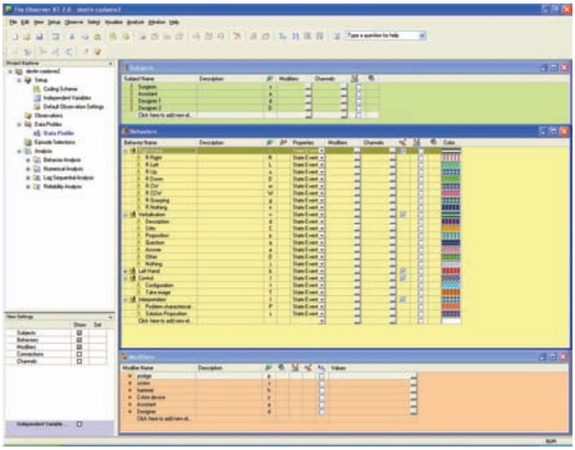


*Figure 5 - The Observer coding scheme configuration*

In order to annotate an observation from an emulation in the operating room, an analytical framework has been developed and the software was configured accordingly. Subjects were divided into surgeons and engineers, so the persons' names were entered in the subcategories. Behavior was divided into gestural movements, control movements and the surgeon's verbalizations. Thus, a grammar of possible and detailed gesture was developed for the first category. Control gesture was subcategorized into visual and tangible gesture. And the verbalization was subcategorized into description, critique, proposition, question and answer.

The Observer provides some facilities for coding, with simple keyboard shortcuts or by clicking on the coding scheme table while the movie stream is playing. Nonetheless, coding at this level of detail took a large amount of time.

Concerning the analysis part, the observer generates descriptive statistics and graphs instantly, formatted the way it is needed, which is very helpful. It calculates descriptive statistics. A system called filtering provides the possibility of analyzing a group of annotations, totally free in configuration.

One very interesting feature of this software is to make clips of those parts of video and data on demand. This extract has a great value in collaborative working and takes an intermediate tool role in group. However, having many functions makes the usage complex and depending on the very correct configuration. The commercial price of the software is high and not accessible to any research group, and as there is no trial version, it is not easy to make the decision to buy and base the research on The Observer XT.

Reviewing these cases helps to see the big picture of an annotation process in a design research. In the next section, the general evaluation criteria for an annotation tool are provided and, accordingly, the software explained above are compared by their functionalities.

## 5 DISCUSSION AND SUGGESTIONS

At the time of this research, several interesting video annotation software were found and among them some selected to be presented in the paper. Although selecting an annotation tool for a research is highly depending on the focus and the objectives of the research, having some evaluation criteria could help finding the best match. In this section, a list of evaluation criteria for annotation tools is presented in five main categories: Configuration, Coding, Analysis, Data entry and General characteristics of the software. Table 1 shows these criteria and typical values in detail.

*Table 1 - Annotation criteria range for existing software*

| Main criteria | Sub-criteria | Negative | Positive |
|---|---|---|---|
| **Configuration** | Subcategory level | 2-level limited | Unlimited |
| | Modifiability during the coding | Not possible | Real-time |
| **Coding** | Coding shortcuts | No shortcuts | Keyboard and mouse |
| | Time line demonstration | Not available | Dynamic |
| | Punctual event coding | Not possible | Possible |
| **Analysis** | Data export | Struggle numbers | Standard format |
| | Integrated analysis tool | Not available | Graphs, tables, etc. |
| | Coding reliability | Not available | Multi-user reliability control |
| **Data entry** | Video format and codec | Limited | Various format |
| | Multi-stream entry | Not available | Up to 8 streams |
| **General characteristics** | Software stability | Unstable | Stable, data recovery |
| | Operating system limitation | Limited to Mac OS | Windows and Mac OS |
| | Usage ergonomics | Non standard | Standard |
| | Sample version | Not available | Available |
| | Price | High price | Free |

A qualitative comparison of the software explained in section four is provided by using the evaluation criteria mentioned above. The evaluation is based on opinion of researchers on each software, which has been gathered thanks to the authors' contacts (interviews and simple questionnaires). Figure 6 shows the evaluation graph.

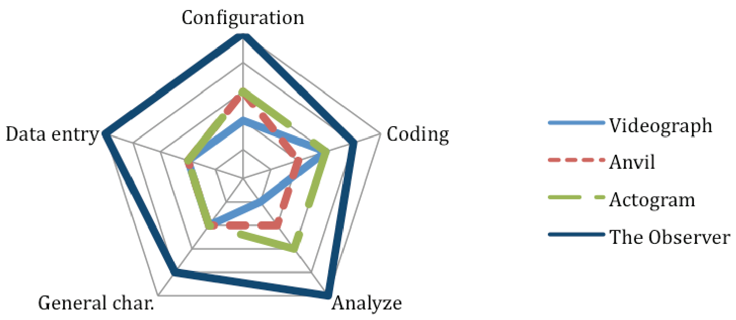


*Figure 6 - Comparison of four annotation software*

So far, video annotation software were used for different design research objectives, but no study explored the specific requirement of design research of an annotation tool. Such a study should investigate the user requirements and the technical capabilities of software developments. Meanwhile, there are some functions listed below as result of our experiments in this field. Nonetheless, listing these requirements could demonstrate an image of the ideal software for the design studies and can be evaluated and improved by the researchers in this field. Here is a list of design requirements, shown in table 2.

*Table 2 – Main criteria for an ideal annotation tools*

| Configuration requirements |
|---|
| Flexible and non-limited level of subcategories<br>Standard for coding scheme<br>Library of coding scheme that can be used for different annotation<br>Modifiability of coding scheme during the coding with trackback the evolution |
| **Coding requirements** |
| Simple and ergonomic interface, improvement in specific hardware<br>Extended techniques for linking the coding to the video stream<br>Automatic coding (ex. using pattern recognition) |
| **Analysis** |
| Features for reliability control of coding process (different persons, different channel)<br>Tools for simple analytical reports like graphs, tables, etc.<br>Export data compatibility with the common statistic analysis software |

## 6 CONCLUSION

This research has tried to explain the annotation process by giving a definition and by looking in practice, through the design studies where an annotation tool has been used. The aim of this study was to determine the characteristics of existing annotation software through the design studies and to explore the possible functions of an ideal annotation tool for design research.

We review different annotation tools and extract a list of evaluation criteria for the functionality of these software. It seems clear that selecting an annotation tool is highly dependent on the project objectives, researchers' expectations and qualifications, and also on the financial support of the project. Making the technical comparison between these various tools wouldn't have provided information better than what can be easily found on internet. Nonetheless, in this study, we tried to

make a functional comparison of selected annotation tools, aiming at providing a decision making help for researchers in this field.
Finally, since there are too many annotation tools, our selection was limited on the authors' former experiences and studies from the regional design community (design research department in Grenoble and Lyon). We therefore look for several uses of experiences on the same annotation tools, to find out more about the tool functionality and relevance.

In our future work, we will firstly extend our selection to have more design research cases using annotation and eventually annotation software, and help to build an adequate requirement list. Therefore, we investigate the possible evolution in order to reach a well defined architecture for such a software tool. In the next step, we will set up the development of a design activity annotation responding to the defined criteria.

## ACKNOWLEDGEMENT

We gratefully acknowledge our colleagues Serap Arikoglu and Jean-Laurent Cassier, for their help in project review. We would also like to thank Bernadette Hunter for her review and relevant comments.

## REFERENCES

[1] Daniel, F. Design-oriented human-computer interaction. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003 (ACM, Ft. Lauderdale, Florida, USA).

[2] Medland, A.J. Forms of Communications Observed During the Study of Design Activities in Industry. pp. 243 - 253 1992 (Taylor & Francis).

[3] Kitamura, Y. and Mizoguchi, R. Ontology-based systematization of functional knowledge. pp. 327 - 351 2004 (Taylor & Francis).

[4] Ahmed, S. and Wallace, K.M. Identifying and supporting the knowledge needs of novice designers within the aerospace industry. pp. 475 - 492 2004 (Taylor & Francis).

[5] Jarrett, J. and Clarkson, P.J. The surge-stagnate model for complex design. pp. 189 - 196 2002 (Taylor & Francis).

[6] McAdams, D. Identification and codification of principles for functional tolerance design. pp. 355 - 375 2003 (Taylor & Francis).

[7] Eder, W.E. and Hubka, V. Curriculum, pedagogics and didactics for design education. pp. 45 - 61 2005 (Taylor & Francis).

[8] Croll, P. *Systematic Classroom Observation*. 1986 (RoutledgeFalmer).

[9] Pedgley, O. Capturing and analysing own design activity. *Design Studies*, 2007, 28(5), 463-483.

[10] Tang, J. *Toward and Understanding of the Use of Shared Workspaces by Design Teams*. Ph.D. Dissertation Thesis, 1989 (Stanford University).

[11] Carrizosa, K., Eris, Ö., Milne, A. and Mabogunje, A. Building the Design Observatory: A core instrument for design research. *International Design Conference*, 2002 (Dubrovnik).

[12] Cross, N., Christiaans, H. and Dorst, K. *Analysing Design Activity*. 1996 (John Wiley & Sons, Chichester, UK).

[13] Dybkjær, L. and Bernsen, N.O. Data, Annotation Schemes and Coding Tools for Natural Interactivity. *International Conference for Spoken Language Processing, ICSLP'2002*, 2002,

217-220 (Denver, CO, USA).

[14] Fu, X., Schaefer, J.C., Marchionini, G. and Mu, X. Video Annotation in a Learning Environment. *69th Annual Meeting of the American Society for Information Science and Technology (ASIST)*, 2006 (Austin, USA).

[15] Bernsen, N.o., Dybkjær, L. and Kolodnytsky, M. An interface for annotating natural interactivity. In Smith, J.v.K.a.R.W., ed. *Current and New Directions in Discourse and Dialogue*, 2003 Dordrecht: Kluwer).

[16] Harrison, B.L. and Baecker, R.M. Designing Video Annotation and Analysis Systems. *Graphics Interface '92 Conference*, 1992, 157-166 (Morgan Kaufmann Publishers Inc, Vancouver, British Columbia, Canada).

[17] Arıkoğlu, E.S., Blanco, E. and Pourroy, F. Keeping Traces of Design Meetings through Intermediary Objects. In Lloyd, P., ed. *About: Designing - Analysing Design Meetings*, 2009.

[18] Martin, J.-C. and Kipp, M. Annotating and Measuring Multimodal Behaviour - Tycoon Metrics in the Anvil Tool. *Third International Conference on Language Resources and Evaluation (LREC)*, 2002.

[19] Kerguelen, A. The Kronos Software : a tool for work activity analysis. *13th Triennal Congress of International Ergonomics Association*, 7, 1997, 240-247 (Tampere, Finlande).

[20] Rasoulifar, R., Thomann, G., Villeneuve, F. and Caelen, J. Proposal of a New Design Methodology in the Surgical Domain. *ICED*, 2007 (Paris).

Contact: Rahi Rasoulifar
G-SCOP Laboratory
Integrated Design
46, Ave Félix Viallet
38031, Grenoble
France
+ 33 4 76 82 05 50
+ 33 4 76 57 46 95
rahi.rasoulifar@gmail.com
www.g-scop.fr/~rasoulir