Work 41 (2012) 3494-3503
DOI: 10.3233/WOR-2012-0629-3494
IOS Press

Invisible computer for collaborative design: evaluation of a multimodal sketch-based environment

Aurore Defays[a*], Stéphane Safin[b], Françoise Darses[c], Anaïs Mayeur[c], Samia Ben Rajeb[d], Caroline Lecourtois[d], François Guéna[d] and Pierre Leclercq[b]

[a] *Laboratoire d'Ergonomie Cognitive et d'Intervention au Travail (LECIT), University of Liège, Belgium.*

[b]*Lab for User Cognition and Innovative Design (LUCID), University of Liège, Belgium.*

[c]*Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI), University of Paris-Sud, France.*

[d]*Atelier de Recherches en Informatique Architecture et Modélisation – Laboratoire d'Architecturologie et de Recherches Epistémologiques sur l'Architecture (ARIAM-LAREA), Ecole Nationale Supérieure d'Architecture de Paris La Vilette, France.*

Abstract. In the areas of design, especially in architectural design, collaboration has become an important challenge. The specialization of skills increase, work teams are more and more extensive and the geographic distance between them increases too. The economic and ecological stakes related to remote collaboration are an evidence. This context involves the need to support most efficiently possible remote working meetings. We present the Distributed Collaborative Digital Studio (DSDC), a tool designed to recreate, in distant situations, the context of copresence meetings. This shared environment is created in the "invisible computer" approach [11]. The idea is that the tool should disappear from user's consciousness. Indeed, creative design activities require some fluidity in their process. Therefore, any involuntary interruption created by the system can potentially brake creativity. In this perspective, we investigate specifically the "invisibility" of our environment. To do this, we propose a framework for the operationalization of the concept and a methodology to test the system invisibility. This methodology was applied through a case study consisting of a corpus of 12 hours of remote collaborative design sessions with the DSDC. We highlight the learning effects while using our system, conclude on its effectiveness and discuss our methodology.

Keywords: Computer Supported Cooperative Work, remote collaboration, invisibility, sketching, multimodality.

1. Introduction

In a wide range of activity sectors, collaboration has been intensified, notably in the design domains. Collective work is increasingly organized simultaneously (rather than sequentially as it used to be in the past). Moreover, design teams are often geographically distributed, and the need for distant real-time interaction is consequently emerging. A lot of effective systems are available for sharing information, but most of them are asynchronous (e.g. database server, email, etc.) or allow only partial interactions (e.g. phone or visioconference). Nevertheless, some systems or environments are emerging allowing rich distant interactions and are increasingly used in professional but also in educational situations.

At the early stages of design, called preliminary design, designers have to focus on their creative task rather than managing the computer tools. Therefore, the traditional CAAD tools are not the most useful tools at this design stage because they tend to limit the creativity [9] or to distract from the design activity itself [2]. Instead, sketching allows keeping some flexibility and natural interactions with the pen [1,5,7,10]. A natural interaction means an interaction not constrained by the tool. In fact, a natural tool or a

[*] Corresponding author. Email: aurore.defays@ulg.ac.be

tool easily used tends to disappear from users' consciousness in favor of the task to perform. Thus, sketching allows designers to focalize their attentionnal ressources and their cognitive efforts on the design process [6].

In this paper, we focus on a sketsh-based prototype : the Distributed Collaborative Design Studio (DCDS). It is a tabletop environment which allows users to import documents and annotate it, at distance and in real time, thanks to a digital pen. In addition to this graphical communication tool, a visioconference device conveys gazes, voice and gestures to create a whole multimodal environment (see Figure 2). . DCDS is specially meant for situations of collective design and aims to recreate the conditions of copresent meetings. We assume that multimodality of this environment will allow more natural mediated interactions between remote partners and will support collaborative design more effectively, making them closer to face-to-face meetings. DCDS has been conceived to be least intrusive in the activity. The philosophy of the environment is that it should "disappear" from the users' consciousness in favor of the real collaborative activity, i.e. it should become invisible, in Norman's sense [11].

Starting from the hypothesis that the use of a pen is always simple and natural is not necessarily true [12]. This ease of use must be objectified and measured using a suitable method. In this context, this article has two main objectives:

1. To develop a method for measuring the "invisibility" or "transparency" of our system. We investigate specifically whether the tool manipulation takes an important place in the collaborative activity, or if it allows users to concentrate on design. In other words, we try to see if the tool tends to disappear in favor of collaborative activity.
2. To test our system to see if users succeed in learning its functioning, if they use its full potentiality and if they succeed in using it more effectively over time. The ease of system use would confirm again that the system disappears effectively from their consciousness.

We present, in the next paragraph, our system DCDS, followed by the presentation of the notion of invisibility, then we present the methodology we have created to try to measure the system invisibility. And then, we present our results that we discuss before concluding.

2. The DCDS

Our prototype, Distributed Collaborative Design Studio (DCDS) is composed of

- a **hardware** part – the Design Virtual Desktop – which consists of an electronic A0 table with a suspended ceiling equipped with a projection system offering a large working surface. An electronic stylus allows the drawing of virtual sketches onto this surface.
- a **software** part – SketSha (for sketch sharing), which is a shared drawing environment allowing several virtual desktops to be connected to the same drawing space (see figure 1). Various functionalities, such as importation of CAD plans and bitmap images, a panel of coloured pens (and an eraser), a navigation functions (zoom, pan, rotate) and a layers palette, are proposed through graphical widgets manipulated with the stylus. This software captures the strokes that compose the sketch and shares them between the different distant locations (through a classic internet connection) in real time.
- a 24 inches **display** with an integrated camera and a videoconferencing commercial module, that allow the participants to see and talk to each others, in an almost 1/1 scale, during a real-time meeting (see figure 2 for the whole environment).

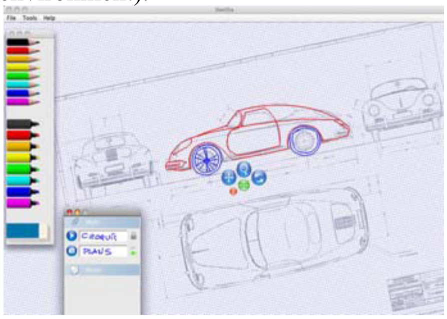

Figure 1 : SketSha Interface.

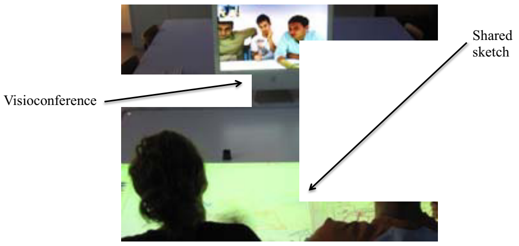

Figure 2 : Distributed Collaborative Digital Studio.

This environment aims to recreate at distance the conditions of copresence. It has proven to be efficient in supporting design activities, in professional and educational settings. More information on this environment can be found in [4,8,17,18].

3. Invisible computer, control levels and automatisms

Our system is based on the paradigm of the invisible computer [11]. According to this paradigm, the computer becomes "invisible" when it disappears from the consciousness of the user.

Most tools for complex activities (including creative or design activities) generate, while using them, two sub-tasks to be performed jointly :

- A target task focused on the object of the activity, in this case, the collaborative activity : problem-solving activities, communications, group management, etc.
- An additional task focused on the management of tool use. These are actions that are meant to manage the computer system behaviors, and are out of the strict framework of the main task. They can potentially be disruptive because they can mobilize resources at the expense of collaborative design activity.

The principle of invisible computing is to allow the user to focus (almost) exclusively on the target task. Thus, the manipulation of a tool called invisible may not constitute in itself a task, which may disrupt the activity. Rather, it may be a part or an extension of the target task. The challenge of new technologies, especially in creative domains, is that they should support activities without causing any interruption and without creating new "noise" tasks (not oriented towards the main activity).

This disappearance from the consciousness is not a state, but a dynamic process. This concept is to be linked with action control levels [15,16]. Thus, this disappearance relies on low level of control, i.e. when the system use becomes based on automatisms or constructed action schemes, appropriated and interiorized by the user [14].

This concept of invisibility is very important for the design of objects or computer systems, but is not very operationalized in terms of ergonomics and evaluation. This is the objective of this study.

4. Methodology

We based our observations and data collection on a video corpus of about twelve hours of a remote collaborative design activity. We observed the design of a building renovation, containing several flats, by a team of three professional architects from an architectural firm of Liege. These professional were working at distance : two of them were located in Liege (Belgium) and one in Paris (France). This building design was a real project of this architectural firm.

They collaborated with the DCDS system which allows synchronous activities. The collaboration work consisted of four sessions (during four days) of two hours and half, except the last one which lasted one hour longer. After these four work sessions, we realized interviews in order to sound out users' impressions about the ease of system use and its capacity to be "transparent" or "invisible" during the collaborative work.

The whole activity on the SDCD was videotaped and we made our video analysis with ELAN[1] software. Our analysis focuses exclusively on the system use. So, we did not investigate the design activity itself, but only actions on the DCDS system. We consider two elements:

1. The **actions on the system** (the use of certain functionalities) and their evolution in order to study how users use the potentiality of system over time. We test the invisibility of our system by its capacity to disappear in favor of collaborative activity.
2. The **verbalizations** related to the **system use** in order to investigate the "weight" that the software takes in the user's consciousness and especially the evolution of this weight according to their learning over time.

4.1. Actions on the system analysis

Initially, we analyzed in detail the types of actions performed on the system. Ours aims are on one hand to understand whether the system is used to its full potentiality (and thus test the software *utility*) and on the other hand, if the functionalities are used correctly (*usability*). We also studied whether, while getting used to the system, there is an evolution in the type of functionalities used.

The number of functionalities available in the software being relatively small, we have listed them exhaustively (zoom, pan, import an image, save project, etc.) and counted their use for each user and for each work session. The evolution of the use of the different functionalities over the time informs us of a possible learning of the system use. Indeed, if the functionalities are increasingly used, that means that

[1] http://www.lat-mpi.eu/tools/elan/

users exploit the maximum potential of the system. Conversely, a decrease in their use could reveal a user frustration. In this case, the functionalities are no longer used because they are evaluated as inefficient or too complex to use by the designers.

We have excluded from the analysis, all "design" activities for two main reasons. On the one hand, the drawing activities are, in our view, only acts of communication or design, and on the other hand, it's difficult to calculate standards for this kind of activities without taking a totally "mechanistic" view of activity.

4.2. Verbalizations analysis

We identified all verbalizations oriented towards the system use and excluded those which are strictly related to collaborative design. We distinguish several categories, some of them are inspired from Blavier and Nyssen's ones [3].

1. **Orders or requests (ORDERS).** These verbalizations involve explicit demands from an actor to his remote partner to perform some actions on the system. For example, "Can you change the color of pen?" or "Can you zoom a little more on the plan? ".
2. **Confirmations on the system use or on the system state (CONFIRMATIONS)**. These are the verbalizations stating that the remote partner has received and understood the message. For example, "Ok, I'll zoom", "Yes, I understand, I will change that", etc.
3. **Explicitations of actions on the system (EXPLICITATIONS).** These verbalizations are directly related to the current use of functionalities, for example, "I'm changing the layer" or "I'm erasing."
4. **Planning of the system use (PLANNING).** These verbalizations explicit future actions on the system to ensure a good coordination : "We should take a separate layer for the ground floor" or "I will try to refocus the plan".
5. **Identification of a user error (ERROR IDENTIFICATION).** This type of verbalizations focuses on the prevention of an error, its detection, its understanding or its recovery. Those errors can be of different nature : based on a lack of knowledge, an erroneous rule or an wrong execution (Reason 1990). For example, "Oops", "I lock the layer overwise ...".
6. **Verification of correspondence of views (CORRESPONDENCE OF VIEWS).** These verbalizations are used to check or certify that the remote partner has the same view as he. For example, "I think the plan is smaller on your side" or "Are we on the correct layer? ".
7. **Identification of a system problem or confusion with the system (SYSTEM PROBLEM).** These verbalizations concern problems that are not attributable to the user but to the system. These include a problem of sound, visibility, or a bug. For example, "We don't know what's happening but we can't hear you".
8. **Questions or explanations on the system functioning (QUESTIONS).** This category includes all the questions asked by users to their remote partners or observers. It also includes the answers or explanations on the system functioning. For example, "Is it possible to import an image of the last day? " or "Can we go back and cancel? "
9. **Judgments on the system (JUDGMENTS)**. This type of verbalizations is related to judgments or evaluations (positive or negative) about the system. For example, "This software is very useful!" or "It's a pity that we haven't the possibility to cancel". We distinguish positive and negative evaluations.

All these verbalizations are centered on the system use. Other verbalizations related to design process or collaborative activity (which are the essential part of activity) have not been taken in account.

4.3. Statistical analysis

Given the nature of the data, we use mainly descriptive statistical data analysis. Our main analysis consists in testing the link between the several categorical data (types of functionalities and verbalizations) and the time (Day_1 to Day_4). We use the Pearson Chi^2 test to measure the independence of theses variables.

4.4. Functionalities

We analyzed four elements : the average number of functionalities used (per hour) for each day, the link between types of functionalities and time, the diversity in functionalities used (heterogeneity of its use) and the efficiency of its use.

The average **number of functionalities used** (per hour) increases from the first to the second day, remains stable till the third day and decreases a little bit the last day (Day_1 : 105 ; Day_2 : 130 ; Day_3 : 128 ; Day_4 : 112). These results show an intensive use of the system : already on the first day, there is about two actions per minute which concerns other functions than drawing.

To simplify our analysis, we regroup all functionalities into 5 supra-categories : tool changes, layer manipulations, view changes, file manipulations and interface management.

The association between the types of functionalities used and the variable "time" is significant ($Chi^2(12)=77.89$, $p<.0001$). We found only one linear evolution: an increase of the number of files manipulations (save project, load project, import a picture) from day 1 to day 4. For the other types of functionalities, we find rather a "sawtooth" pattern for major part of them during the collaborative sessions. However, These results suggest that the nature of functionalities used do not vary substantially with learning from one day to another, but is rather adapted to the current tasks.

As we reported, we have listed these functionalities exhaustively : users have the possibility to use 22 different functionalities. If we compare the total **number of *different* functionalities used** : we move from 13 different functionalities for the first day to 16 for the second, and 19 and 18 for the last days. Upon first use, users already use the 50% of the functionalities available (see Figure 3). Then, they use more and more over time to reach 85% at the end. Some functionalities are not used (i.e. export an image, remove a layer, empty the trash, etc.) either because these advanced functionalities require more expert skills, or because these ones are not useful or too complicated to use. These results suggest a clear learning curve. As the designers feel comfortable with the use of the system, they tend to exploit it in a more exhaustive way.

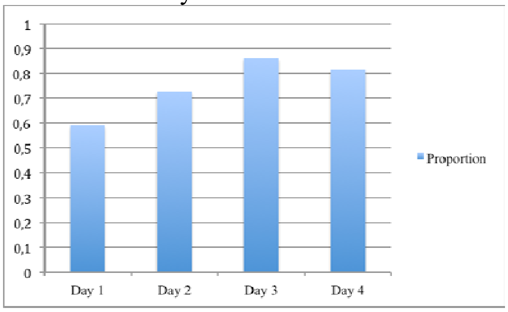

Figure 3 : Heterogeneity of functionalities types (proportion of functionalities actually used relative to possible functionalities).

Regarding the **efficiency of the functionalities** use, two functionalities are particularly problematic : pan and zoom. For these both functionalities, there are two types of strategies : a "one-click strategy", consisting in handling with a single drag and drop movement starting from the button which engages the manipulation, or a "two-click strategy" consisting in clicking once on the manipulation button and then performing the manipulation on the drawing itself. We observed that the two-click strategy is much less efficient than the one-click[2].

Our results show that the "two-click" strategy tends to disappear over the days both for the pan (Figure 4) to for the zoom functionality (Figure 5).

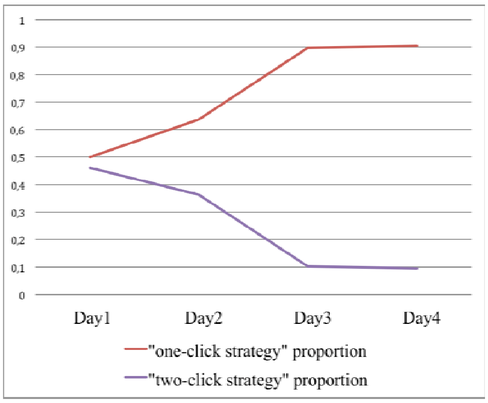

Figure 4 : Pan – proportion of each strategy.

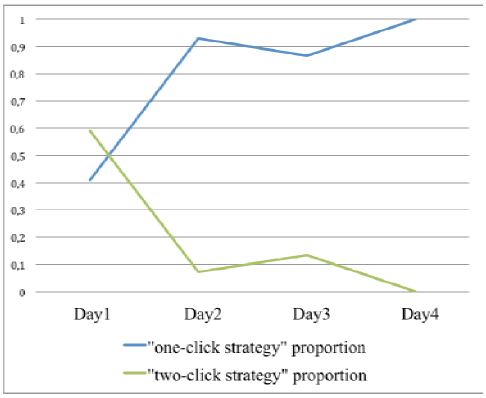

Figure 5 : Zoom – proportion of each strategy.

Over time, users are turning to the most efficient strategy. This "economy" of actions indicates a better understanding of the system working and a more efficient use of it although some functionalities seem to require more experts' skills.

[2] We calculated the efficiency by dividing the number of actions observed to reach the goal and the number of actions theoretically required to reach the same goal.

Globally, these results suggest that, in their learning phase, users do not use many functionalities which are not very different. Users tend, during their familiarity with the system, to repeat the same learned and familiar actions rather than to explore other options. After two collaborative sessions, when they start to master the system, they do not use more functionalities in terms of absolute numbers, but the heterogeneity of functionalities used increases. This exploration of other functionalities indicates that the system potentiality is better used over time, and highlights a learning by the users and a more efficient system use.

We can conclude that, even if the software is not completely transparent at first sight, users get very quickly used to its functioning to achieve optimal use. These results are confirmed by our interviews : users claim to be more comfortable with the functioning logic of DCDS over time and this tool tends to disappear from their consciousness. Users can thus concentrate their resources on the collaborative activity without be constrained by this tool.

4.5. Verbalizations

The average number of verbalizations focused on the system varies along a sawtooth pattern over time (Day_1 : 111 ; Day_2 : 72 ; Day_3 : 125 ; Day_4 : 91). On the first day, there are many system-centered verbalizations, which decrease on the second day. Then, they grow again on the third day and decrease once again on the last day.

Regarding to the types or natures of verbalizations (as the categories presented above), we have considered the total proportion of each category for the whole collaborative activity (for the four work sessions). And then, we have examined in detail the evolution of proportions of each verbalization category over time.

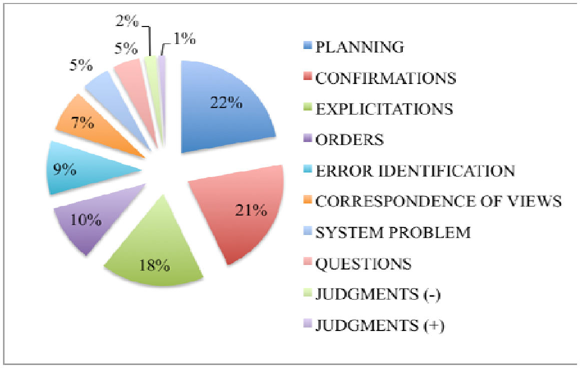

Figure 6 : Proportion of types of verbalizations related to the system use (whole activity).

Globally (for the four work sessions), the most frequent verbalizations are planning of the system use (253/1145 or 22%, see figure 6), followed by confirmations (240) and explicitations (202) of actions on the system (respectively 21% and 18%). Other types of verbalizations occupy a less important proportion of the verbalization on the system activity. These results indicate that the verbalizations about the tool use are more centered toward the collaborative activity (planning, confirmations, explicitations) instead centered toward the system learning (questions, explanations, etc.) or toward errors management or partners' adjustments. We can assume that the major part of verbalizations concerns the collaborative activity.

The association between the types of verbalizations mobilized and the time variable is significant ($Chi^2(27)=132,70$, $p<.0001$). Concerning the evolution of type verbalizations over time, we obtained three patterns of results : a linear decrease of certain types of verbalizations, an linear increase of other ones and non-linear changes over time for the others.

We found a decrease of verbalizations related to the learning of the system use as questions, error identification and negative judgments (see Figure 7).

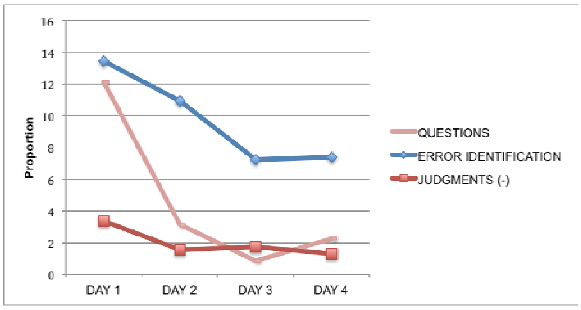

Figure 7 : Proportion of QUESTIONS – ERROR IDENTIFICATION – JUDGMENTS (-) for each work session.

QUESTIONS. On the first day, users ask a lot of questions or explain many times how the system works (12% of total verbalizations). The second day, the questions or explanations on the system functioning decline hardly (until only 3%) and still decline a little bit on the third day (1%).

ERROR IDENTIFICATION. On the first day, there are also a lot of verbalizations related to the identification and the avoidance of errors. As for the last type of verbalizations (questions), error identifications decline on the second day (from 13% to 11%) and still decline on the third day (7%). However, error identifications remain stable between the third and the last day.

JUDGMENTS (-). Users often judge the system more negatively early in their learning compared to the end of their work sessions (from 3% to 1%).

We found an increase of verbalizations related to collaborative activity (planning, confirmations, orders) and positive evaluations of the system (Figure 8).

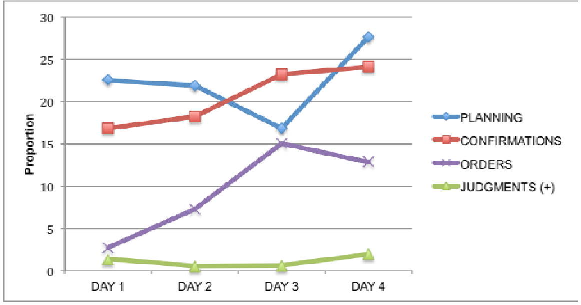

Figure 8 : Proportion of PLANNING – CONFIRMATIONS – ORDERS – JUDGMENTS (+) for each work session.

PLANNING. We have seen that verbalizations related to planning of the system use took a major part of total verbalizations. However, they remain stable at the beginning of users' learning (Day$_1$: 23% ; Day$_2$: 22%) and decrease a little bit on the third day. We observe a strong increase of this verbalization category on the last day (from 17% to 28%).

CONFIRMATIONS. Concerning the second important category of verbalizations (21% of total verbalizations), we found a linear relationship over time. The number of confirmations on the system use or on the system state (i.e. verbalizations stating that the remote partner has received and understood the message) increase more and more each day (Day$_1$: 17% ; Day$_2$: 18% ; Day$_3$: 23% ; Day$_4$: 24%)

ORDERS. Concerning the verbalizations related to orders (or actions to perform on the system by distant partner), we found also a linear relationship over time. Indeed, we observed few orders at the beginning of users' learning (Day$_1$: 3% ; Day$_2$: 7%) and a strong increase of this type of verbalizations on the third day (15%). Then, they remain constant.

JUDGMENTS (+). There are a little more favorable judgments about the system and its functionality over time but this tend is small.

For other verbalization categories, we did not find a linear evolution pattern (see Figure 9).

EXPLICITATIONS. On the first day, there are few verbalizations which explicit actions performed on the system. On the second day, we observe a peak of this type of verbalizations (from to 14% to 23%). Then, these verbalizations tend to decline over time to come back around the initial situation. This result is probably due to the evolution of the number of actions performed on the system over time. Indeed, if we compare the curve of this verbalization category and the curve of the average numbers of functionalities used, they are almost identical.

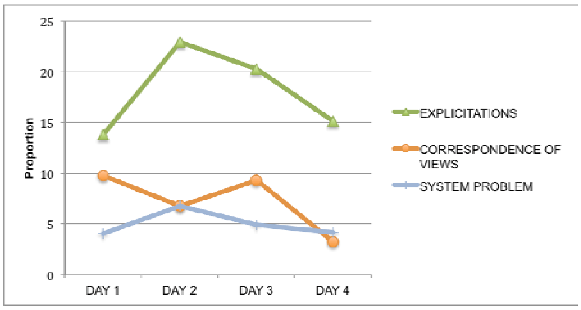

Figure 9 : Proportion of EXPLICITATIONS – CORRESPONDENCE OF VIEWS – SYSTEM PROBLEM for each work session.

CORRESPONDENCE OF VIEWS. The verbalizations related the correspondence of partners' views does not change linearly over the days. In fact, we find rather a "sawtooth" pattern during the collaborative sessions : at the beginning, users check many times the correspondence of their views on the first day (10%), with a little decrease on the second day (7%), a little increase on the third day and a strong decrease on the last day (from 9% to 3%). So, the general tendency for this type of verbalizations is decreasing.

SYSTEM PROBLEM. Globally, verbalizations related to the detection of problems from the system or related to confusions about the system state remain stable over time. There is only a little increase of this verbalization type on the second day (7%). Because the system working is relatively stable and the number of bugs is relatively constant, this result can probably be explained by the fact that users have learned and interiorized the system use. So, they can directly perceive a gap between the system expected state and the observed one. They are aware that the problem comes from the system and that system behaviors are not due to their own mistake.

Globally, these results suggest that, at the beginning of learning, there are more questions or explanations on the system working. The evolution of such verbalizations clearly emphasizes the users' learning phases or their familiarization with the system. In parallel, there are also more verbalization related to the identification and the avoidance of errors, probably because they are committed more often at the beginning. At the beginning, users also often need to check if they share the same view of their workspace. This result reflects the need of mutual adjustment emerging between partners.

Over time and according to users' learning, there are more verbalizations which explicit actions performed (EXPLICITATIONS) or to be performed on the system (ORDERS). Indeed, users take more time to inform their remote partner of what they are doing. On the other hand, users begin giving orders or requests to their remote partners. This result shows that coordination begins being developed between the partners. In fact, the actions to be performed become organized and there is a genuine distribution of tasks between partners. Moreover, we noted a constant increase of confirmations on the system use or on the system state.

At the end of work sessions, there is a significant difference in verbalizations related to the planning of system use. That testifies again to this new work organization. Furthermore, the past tendency to check the correspondence of views has decreased, indicating that users no longer need to check whether they have the same view as if they had acquired the knowledge of what their partner could see. So, they can concentrate their attention on the planning and achievement of actions.

In conclusion, the nature of the verbalizations change over time : at the beginning, they are more focused on the functioning of the system itself (questions, errors, judgments, etc.) and become more focused on coordination and collaboration between partners at the end. Therefore, we can say that these indices show an disappearance of the tool in favor of communication and collaborative work : the focus is no more on the tool, but on the collaborative activity and on tasks to perform.

5. Discussion and perspectives

This article has two main objectives. First, it aims to develop a method for measuring the “invisibility” or “transparency” of our system. We investigate specifically whether the tool makes actually place for the communications oriented to design, so if the tool tends to disappear from users’ consciousness in favor of collaborative activity. Secondly, it aims to test usability of our system. We investigate if users succeed in learning its functioning, if they use its full potentiality and if they succeed in using it more effectively over time.

In our analysis, we consider two elements and their evolutions: the actions on the system (functionalities used) and the verbalizations related to the system use.

The average numbers of functionalities used follow an inverse “U” curve. Indeed, designers use few functionalities at the beginning, then, they use more and more functionalities, and there is a little decrease at the last day. In parallel, the functionalities used are more and more heterogeneous : the designers start by using half the proposed functions and finish by using more than 80%. Users tend to learn progressively to use the different functionalities. As soon as users master major functionalities, they select the most relevant ones or the most useful to perform a particular task. Thus, we notice, at the end, a decrease of average number of functionalities used and a little decrease of their heterogeneity. Moreover, we observe a difference in the types of functionalities used over time, even if, except for the files manipulations, we did not find linear evolution.

Regarding to verbalizations, we found a sawtooth pattern over time. The average number of verbalizations decreases on the second day, there is a peak on the third day, and it decreases once again on the last day. Results have shown that the average number of verbalizations centered to the system use varies between the different sessions, but the nature of verbalizations also varies over time. In fact, users progressively focus their verbalization activity from the system functioning to their collaboration work. These results attest that the tool becomes, in a way, the extension of the task and it disappears progressively from the users’ consciousness.

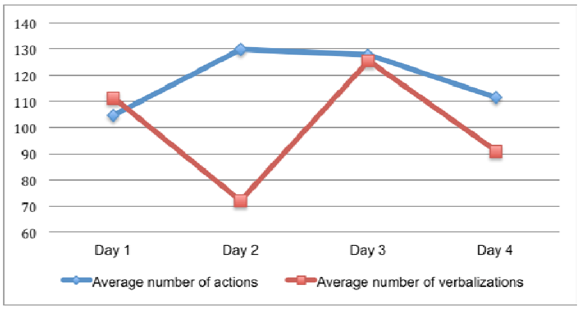

Figure 10 : Comparaison between average number of actions and average number of verbalizations

If we compare the average number of functionalities used and the average number of system-centered verbalizations over time, we observe that these elements do not vary along the same way (see Figure 10).

These patterns of results correspond to the different users' learning phase.

On the first day, designers use few functionalities and often the same ones. In parallel, users verbalize a lot about how the system working, they explicit a lot the actions performed, they often check the correspondence of their views. These strategies can be qualified as novice ones. The system takes an important place in collaborative activity.

On the second day, we notice a strong increase of numbers of functionalities used. In parallel, the number of system-centered verbalizations decreases hardly. These results suggest a phase of "stabilization" of the learning: actions begin to be automated and require less verbalizations, and especially less verbalizations directly oriented to the system use. In parallel, the users choose the best strategies to deal with complex functionalities.

On the third day, the number of functionalities used remains stable, but their heterogeneity increases. We also observe a new strong increase of verbalizations. This seems to show that the users, being comfortable with the system use, try to exploit it more intensively. This leads to the emergence of new difficulties, shown by the increased number of verbalizations. In this third phase, it is like if users were coming back to the initial situation of learning. Nevertheless, there are more orders and confirmations, which indicate an that specific coordination procedures and task distribution between remote partners begin to emerge. Orders and confirmations are indispensable for an accurate cooperative work and to construct common situation awareness.

On the last day, the number of functionalities used, their diversity and the number of verbalizations oriented to the system use both decrease. Users have learned to master certain functionalities and automate their use, as in second day.

These different phases, along two cycles, can be explained by the assimilation and accommodation processes described by Piaget [13]. To use a system, human develop action schemes which are constructed with the repetition (learned and interiorized schemes). People try to use their schemes in various situations (assimilation) and change their schemes if necessary to deal with new situations (accommodation). New technologies transform our communication, cooperation and coordination modes.

Our observation suggest that people try to manage the new technology on the first day (assimilation). On the second day then adapt themselves to the situation and manage it quite well (accommodation). O the third day, they create an imbalance by exploring more functionalities (assimilation). In the last day, they start to master the environment and reduce their actions to the essential. We can say that, in assimilation phase, communication and coordination schemes are not modified by the insertion of a new technology. New technologies incorporate in existent schemes. In accommodation phase, there is a profound modification of communication and coordination schemes created by the insertion of the new technology.

Our observations also suggest a kind of "homeostasis" between the degree of confidence and the exploitation of the system. When people seem to master the software, they try new possibilities and new functionalities. Our study should be extended on a longer period of time, to see if these movements of assimilation-accommodation continue until a total mastering of the system.

The concept of invisibility of a system seems therefore not relative to a quantitative propriety, but rather a qualitative propriety of the system. As users learn to use the system, this latter does not take less importance in the activity, but it takes a different nature. Its potentialities are better exploited by users over time and the verbal management of the use of the system evolves in verbal coordination of the users' actions. Even if the system is not completely invisible at the beginning, we can say that it becomes more and more invisible along the natural users' learning curve, freeing resources for collaborative activity.

Acknowledgments

This work is funded by a grant from the Research Council of the University of Liège, and by the ANR project "Cocréa", financed by the National Research Agency of France. The authors wish to thank the participants to the study.

References

[1] Aliakseyeu, D. (2003). A Computer Support Tool for the Early Stages of Architectural Design. PhD Thesis, Eindhoven, The Netherlands.

[2] Aliakseyeu, D., Martens, J. B., Rauterberg, M. (2006). A computer support tool for the early stages of architectural design, Interacting with Computers, 18, 528–555.

[3] Blavier, A. & Nyssen, A.-S. (2010). Étude de l'impact des nouvelles technologies sur les modes de coopération des chi-

rurgiens par l'analyse des communications sur le terrain. Le Travail Humain, 73(2), 123-140.

[4] Elsen, C. & Leclercq, P. (2008). "SketSha" or the sketch power to support collaborative design. Lecture Notes in Computer Science, Cooperative Design, Visualization and Engineering, 20-27

[5] Goel, V., (1995). Sketches of thought, Cambridge, Massachusetts, The MIT Press.

[6] Goldschmidt, G. (1999). The backtalk of selfgenerated sketches, Gero, J. S. & Tversky, B. (eds), In Visual and spatial reasoning in design, Sydney, Key Centre, University of Sydney, pp. 163-184.

[7] Gross, M., Yi-Luen Do, E. (1996). Ambiguous intentions : a paper-like interface for creative design. In Proceedings of UIST'96, November 1996, Seattle. ACM Press, pp. 183–192.

[8] Kubicki, S., Bignon, J.-C., Leclercq, P. (2008). Cooperative Digital Studio IT-supported cooperation for AEC students. In Proceedings of 25th International Conference on Information Technology, Universidad de Talca, Chile.

[9] Lawson, B. (1999). 'Fake' and 'real' creativity using computer aided design: some lessons from Herman Hertzberger. In: Proceedings of Creativity & Cognition'99, October 1999, Loughborough, ACM Press, pp. 174–180.

[10] McCall, R., Ekaterini V. & Joshua Z. (2001). Conceptual design as hypersketching. CAAD Futures'01, Kluwers, Doordrecht.

[11] Norman, D.A. (1998). The invisible Computer. MIT press.

[12] Norman, D.A. (2010). Natural User Interfaces are not Natural. Interactions, 17(3). ACM.

[13] Piaget, J. (1975). L'équilibration des structures cognitives : Problème central du développement, Paris : Presses Universitaires de France.

[14] Rabardel, P. (1995). Les hommes et les technologies: approche cognitive des instruments contemporains. Paris : Armand Colin.

[15] Rasmussen J. (1986). Information Processing and Human-Machine Interaction, Amsterdam, North-Holland.

[16] Reason, J. (1990). L'erreur humaine. Paris: PUF.

[17] Safin, S., Delfosse, V., & Leclercq, P. (2010). Mixed-reality prototypes to support early creative design. In E. Dubois, P. Gray & L. Nigay (Eds). The Engineering of Mixed Reality Systems. London : Springer, pp. 419-445.

[18] Safin, S. & Leclercq, P. (2009). User studies of a sketch-based collaborative distant design solution in industrial context. Proceedings of CDVE 2009. The 6th International Conference on Cooperative Design, Visualization and Engineering. Luxembourg, Septembre.