A Technique for Evaluating Shared Workspaces Efficiency

Antonio Ferreira and Pedro Antunes

Department of Informatics of the University of Lisbon, Portugal
{asfe,paa}@di.fc.ul.pt

Abstract. We propose a technique based on human-performance models to evaluate the efficiency of shared workspaces, where individual and collaborative actions are intertwined. We apply the technique to an illustrative case and report that it: 1) facilitates the fine-grained analysis of workspace collaboration; 2) provides time predictions about collaborative actions; and 3) enables quantitative comparisons of alternative designs via multi-dimensional team performance estimates. The technique may be used to complement existing practice and knowledge with the ability to make quick measurements and calculations without users or functional prototypes, thereby enabling faster design iterations.

1 Introduction and Motivation

CSCW usability evaluation is a challenging endeavor for researchers and practitioners because current methods and techniques impose significant constraints motivated by the number of participants in the evaluation processes and by the required control over variables related to the group, the task, the context, and the technologies [1].

In this paper our research interest is in reducing the cost and complexity of evaluating shared workspaces efficiency, thus enabling more design iterations and allowing for the emergence of more successful designs. Collaboration in shared workspaces entails high levels of interdependence and workspace awareness because continuing individual actions often limit the options and affect the outcomes of the other team members, and vice-versa [2]. For this reason, small design decisions (the low-level details of individual and collaborative actions, usually performed in very dynamic contexts) have much greater impact in workspace collaboration than in other contexts, where the focus may be on more abstract activities such as group decision making.

Several techniques from the HCI (Human-Computer Interaction) field—and thus focused on single user interactions—already reduce complexity and give attention to details. For example, the GOMS (Goals, Operators, Methods, and Selection Rules) family of techniques [3] relies on human-performance models to analyze fine-grained usability problems. From these techniques, we are particularly interested in the KLM (Keystroke-Level Model) [4,5], because it is relatively simple to use and has been successfully applied to evaluate the efficiency of many single-user designs [3].

In this paper we propose a technique, based on earlier research on the benefits of using human-performance models [6], to provide additional insights about workspace collaboration, not covered by other evaluation techniques. Some advantages of this technique emerge from the following characteristics of human-performance models:

Ferreira, A. and P. Antunes (2007) A Technique for Evaluating Shared Workspaces Efficiency. Computer Supported Cooperative Work in Design 3: 10th International Conference, CSCWD 2006, Nanjing, China, May 3-5, 2006, Revised Selected Papers. W. Shen, J. Luo, Z. Lin, J. Barthès and Q. Hao. Lecture Notes in Computer Science, vol. 4715, pp. 15-31. Heidelberg, Springer-Verlag.

- Afford studying alternative designs without the participation of users or the development of prototypes, which may reduce design time and effort;
- Elucidate the assumed capabilities and mechanisms of the human processing system, which may be instrumental to develop more useable CSCW tools;
- Offer quantitative predictions of human performance, which may be used to make design decisions based on quick measurements and calculations;
- Address the fine-grained details of workspace collaboration, which may be used to optimize overall team performance.

In Sect. 2 of this paper we apply our evaluation technique to an example of workspace collaboration, compare two alternative designs, and discuss benefits and limitations; in Sect. 3 we confer related work, and Sect. 4 is on contributions and future work.

2 Illustrating Example

The technique proposed in this paper will be described and explained by means of its application to an example of workspace collaboration. The example refers to a collaborative game where multiple players draw either vertical or horizontal connections between adjacent pairs of points in a board. The game is over when the board is filled with connections, but players must observe this rule: if a player, e.g. Sophie, is an expert in drawing vertical connections, then she must consider adjacent pairs of points that contain, at least, one horizontal connection to a third point. The behavior of an expert in horizontal connections, e.g. Charles, is analogous.

For illustration purposes, the board is characterized by a square arrangement of contiguous cells, numbered 1 to 9, and by an initial state that contains at least one horizontal and vertical connection lines (see Fig. 1).

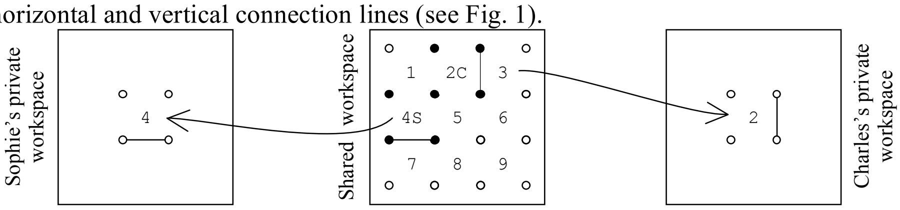

Fig. 1. Cell reservations and ownership letters (2 players shown)

The game features a shared workspace for displaying a public up-to-date view of the board, and private workspaces where players can connect cell points. To simplify our analysis, we restrict player interactions to a mouse with a button.

In order to connect points, players must first reserve the points by selecting and dragging the corresponding cell into the private workspace. Later, the modifications on the cell will be made public when the cell is moved back to the shared workspace.

To minimize inadvertent selections of reserved cells, the shared workspace provides awareness by displaying a letter, next to the cell number, that identifies the current owner (see Fig. 1). Additionally, the collaborative game impedes concurrent reservations of the same pairs of adjacent points. For example, if two players select vertically or horizontally neighbor cells (or the same cell), and simultaneously try to reserve them, then only one player will accomplish the cell reservation, while the other is notified that the cell cannot be reserved.

It is expected that the cells remain reserved for a small amount of time due to the expertise of the players and their eagerness to accomplish the shared goal.

To demonstrate why this case concerns workspace collaboration we can consider that the team must work in harmony in order to quickly connect all pairs of adjacent points: the more horizontal connections exist, the more vertical connections can be drawn, and vice-versa. Conversely, if one player stops drawing connections, the other player will soon also stop. In other words, the actions of the team members (the players) are intertwined, this being a distinctive feature of workspace collaboration [2].

2.1 The Proposed Technique

Step 1: Characterizing goals and actions. The technique begins with the characterization of the collaborative environment in terms of goals and actions. In the collaborative game, players pursue *individual goals*: to draw connection lines as fast as possible. At the same time, they are conscious of team performance towards the *shared goal*: to quickly connect all adjacent points in the board.

Complementarily, team work results from a combination of individual and collaborative actions. *Individual actions* correspond to drawing vertical and horizontal connections, which, due to their similarity, can be generically identified by `DRAW`. *Collaborative actions* are related to moving a cell from the shared into the private workspace, and vice-versa. These actions, named `RESERVE` and `RELEASE`, involve the shared workspace and are required to coordinate work and prevent conflicts.

Step 2: Detailing actions. The technique proceeds with detailed descriptions of the individual and collaborative actions that characterize the collaborative environment. Table 1 shows the details of the actions that players can perform in the game.

Table 1. Individual and collaborative actions

Action	Description
`RESERVE` (collaborative)	The player: 1) locates a cell in the shared workspace; 2) presses the mouse button over the cell; 3) moves the mouse cursor to the private workspace; and 4) releases the mouse button
`DRAW` (individual)	The player: 1) locates a cell point in the private workspace; 2) presses the mouse button over the point; 3) moves the mouse cursor to the adjacent point in the cell; and 4) releases the mouse button
`RELEASE` (collaborative)	The player: 1) locates a cell in the private workspace; 2) presses the mouse button over the cell; 3) moves the mouse cursor to the shared workspace; and 4) releases the mouse button

In a shared workspace, the individual and collaborative actions are entwined and under the control of the CSCW tool, which means that their design can influence individual, and especially, team performance.

Step 3: Predicting execution times. The technique proceeds with an evaluation of efficiency using the KLM (Keystroke-Level Model) [4,5]. In this model an action (e.g., each action in Table 1) is converted into a sequence of mental and motor operators whose execution times have been quantified and validated in psychological experiences [4,7]. An important KLM requirement is that modeling applies to expert error-free behavior only. This is met in the collaborative game since the players are highly trained in drawing connections and in using the shared workspace.

To illustrate the conversion from a detailed textual description into a KLM representation, consider the `RELEASE` action in Table 1. In steps 1 and 2, player Sophie locates a worked cell in her private workspace; this is converted into the `M` operator. Then, she moves the mouse cursor over the cell, a `P`, and presses the mouse button, a `K`. In step 3 she moves the mouse cursor to the shared workspace, an operation that is translated into a `P`, without a preceding `M` since there is no need to find the workspace. In step 4 Sophie releases the mouse button, `K`. The total predicted time for the execution of the `RELEASE` action is obtained by adding the individual times of the KLM operators, which for `MPKPK` gives 1.2 + 1.1 + 0.1 + 1.1 + 0.1 = 3.6 seconds.

Interestingly, all actions in our case are essentially a sequence of `MPKPK` operators, hence the predicted times are the same. This suggests that the required human skills for drawing a connection between two points are very similar to those needed for moving a cell between workspaces, which seems plausible if we consider Fitts's Law, the sizes of the objects, and the distances between them [4].

The previous time estimates apply to actions as if they were unrelated. To reveal goal achievements—individual and shared—in a collaborative environment we need to realize how work is produced with the CSCW tool. In the next step we will analyze individual behavior and then proceed to an evaluation of team performance.

Step 4: Focusing on the individual goals. In the collaborative game, and given an appropriate cell in the shared workspace, each player carries out individual goals by following one of two possible sequences of actions, shown in Table 2. Sequence `S1` corresponds to drawing a single connection in a cell. The sequence of actions `S2` applies to cases where two connections can be drawn in the same cell.

Table 2. Sequences for achieving individual goals

S#	Actions	Time (s)	Collaborative	Individual
`S1`	1) `RESERVE` 2) `DRAW` 3) `RELEASE`	3.6 + 3.6 + 3.6 = 10.8	7.2/10.8 = 67%	3.6/10.8 = 33%
`S2`	1) `RESERVE` 2) `DRAW` × 2 3) `RELEASE`	3.6 + 3.6 × 2 + 3.6 = 14.4	7.2/14.4 = 50%	7.2/14.4 = 50%

Table 2 is very interesting because it shows that the collaborative actions, `RESERVE` and `RELEASE`, are more costly (7.2s or 67% of total predicted time) than the individual action of drawing a connection line, `DRAW`, that characterizes sequence `S1`. It is therefore likely that the CSCW designer admits that players will avoid such situation and instead prefer sequence `S2`, due to its lower collaboration overhead (50%).

Step 5: Focusing on the shared goal. In this step we analyze team performance. We start by defining a *goal unit* as a conceptual metric for assessing progress in terms of the shared goal. In the collaborative game, the shared goal is reached when all line connections have been drawn on the board, which gives a total of 24 goal units.

We continue the analysis with a characterization of the sequences of actions along three dimensions which we think are intrinsic to workspace collaboration. The *productivity* dimension measures the number of goal units produced per time unit. The greater the value, the faster the team may progress towards the shared goal. In single-user software design this dimension measures individual efficiency. However, with workspace collaboration team efficiency cannot be determined by simply combining individual efficiencies; we try to capture this with the other two dimensions.

The *opportunities* dimension is related to the intertwined nature of workspace collaboration: if a team member stops, then soon the team will also halt, eventually never reaching the shared goal. This suggests that collaboration among team members is bound by opportunity dependencies created by the achievement of individual goals. The measurement unit for this dimension is new goal unit opportunities potentially created per time unit. The greater the opportunities, the faster the team may progress.

The *restrictions* dimension reflects a possible negative outcome of coordination in shared workspaces: the prevention of conflicts and duplicate efforts (positive outcomes) may slow down or even impede the work of other team members. Restrictions are measured in inaccessible goal units times the duration of the sequence of actions. This unit of measurement emphasizes fast and unobtrusive execution of individual goals: the greater the restrictions value, the slower the team may progress, because team members will probably spend more time waiting to proceed.

We can now evaluate team performance based on the analysis of the sequences of actions `S1` and `S2` along the three dimensions (see Table 3). Once more, a goal unit (gu) corresponds to one connection. The main time unit, for convenience, is minutes.

Table 3. Team performance for the initial design

S#	Productivity	Opportunities	Restrictions
`S1`	1 gu / 10.8 s = 5.5 gu/min	2 gu / 10.8 s = 11.1 gu/min	1 gu * 10.8 s = 0.18 gu.min
`S2`	2 gu / 14.4 s = 8.3 gu/min	5 gu / 14.4 s = 20.8 gu/min	1 gu * 14.4 s = 0.24 gu.min

The predictions in Table 3 show that `S2` is more productive than `S1`, because `S2` takes 14.4s to draw 2 line connections—thus the 8.8 gu/min—in contrast with 5.5 gu/min of `S1`. `S2` also compares favorably with `S1` in creating new individual goal opportunities for the other team members: 20.8 versus 11.1 gu/min. The logic behind the number of opportunities for each sequence of actions is illustrated in Fig. 2.

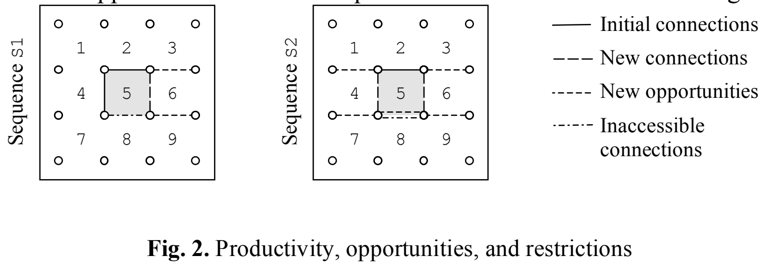

Fig. 2. Productivity, opportunities, and restrictions

Using sequence S1 only one vertical connection line can be drawn by Sophie in cell 5, which, in the best case, opens two new opportunities to Charles since he will be able to draw two horizontal connections: the top and bottom lines in cell 6. The missing bottom horizontal line in cell 5 is *not* an opportunity because it was already available via the left vertical connection in cell 5. Actually, this bottom connection is inaccessible to the other players while Sophie is running S1. In sequence S2 up to 5 opportunities can be created after the left and right vertical lines are drawn in cell 5.

The only dimension where S1 is preferable to S2 is the restrictions to the work of other team members. The lower 0.18 gu.min of S1 versus 0.24 gu.min of S2 is caused by its faster predicted execution time, 10.8 versus 14.4s, since the number of inaccessible goal units during the execution of the sequence of actions is the same in both cases: a single line connection drawing (the bottom horizontal connection in cell 5).

The data in Table 2 and Table 3 provide a basis for doing comparisons with other designs. In the next section we evaluate a design alternative using the same technique.

2.2 Evaluating a Design Alternative

Our design alternative for the collaborative game features multiple cell reservations/releases, and the display of awareness information while team members *select* cells in the shared workspace. The motivation is twofold: a) the impact of collaborative actions on individual goal execution decreases with the number of connections that can be drawn consecutively; and b) selecting cells in the shared workspace is faster than reserving cells, which means that awareness information will be more up-to-date.

The new features introduce changes in the *collaborative* actions that characterize the work environment: two novel actions are used for selecting single and multiple cells, **SELECT_1** (a simple click on a cell) and **SELECT_N** (a click and drag movement over consecutive cells); additionally, the reservations and releases, **RESERVE_B** and **RELEASE_B**, are now a bit simpler to reflect the fact that players don't need to search for a cell or cells that they have just selected (cell selections always precede cell reservations or releases). Table 4 shows the new KLM models and predicted times.

Table 4. New collaborative actions

Action	KLM Model	Time (s)
SELECT_1	MPKK	2.5
SELECT_N	MPKMPK	4.8
RESERVE_B	KPK	1.3
RELEASE_B	KPK	1.3

The data show that the 2.5s of **SELECT_1** is lower than the 3.6s of the previous **RESERVE** action (cf. Table 2), meaning that players should experience less time dealing with coordination conflicts. On the other hand, the time to reserve a single cell increases because now it takes a **SELECT_1** followed by **RESERVE_B**, with a total of 3.8s. We consider this tradeoff acceptable because the time to recover from a reservation conflict is, at least, an order of magnitude greater than the extra 0.2s.

In Table 5 we analyze the new sequences of actions for achieving individual goals.

Table 5. New sequences of actions

S#	Actions	Time (s)	Collaborative	Individual
`S3`	1) `SELECT_1` 2) `RESERVE_B` 3) `DRAW` 4) `SELECT_1` 5) `RELEASE_B`	2.5 + 1.3 + 3.6 + 2.5 + 1.3 = 11.2	7.6 / 11.2 = 68%	3.6 / 11.2 = 32%
`S4`	1) `SELECT_1` 2) `RESERVE_B` 3) `DRAW` × 2 4) `SELECT_1` 5) `RELEASE_B`	2.5 + 1.3 + 3.6 × 2 + 2.5 + 1.3 = 14.8	7.6 / 14.8 = 51%	7.2 / 14.8 = 49%
`S5`	1) `SELECT_N` 2) `RESERVE_B` 3) `DRAW` × `n` 4) `SELECT_N` 5) `RELEASE_B`	4.8 + 1.3 + 3.6 × `n` + 4.8 + 1.3 = *total*	12.2 / *total* `n` = 1 → 77% `n` = 2 → 63% `n` = 3 → 53% `n` = 4 → 46%	3.6 × `n` / *total* `n` = 1 → 33% `n` = 2 → 37% `n` = 3 → 47% `n` = 4 → 54%

As expected, if players can *only* select single cells, they will probably prefer reserving those in which they can draw two connection lines using sequence `S4`, in detriment of `S3`. This is because in `S4` the overhead of collaborative actions, 51%, is lower than the 68% in `S3`. However, if players see an opportunity for reserving multiple cells at once, then they will likely use sequence `S5` when *at least* four connections (`n` ≥ 4) are doable in those cells, because the impact of collaborative actions is *at most* 46%, this being unmatched by any of the sequences `S3` and `S4`.

Table 6 shows the new team performance values afforded by the alternative design, for the sequences of actions **`S3`**, **`S4`**, and for three variants of **`S5`**, which are illustrated in Fig. 3.

Table 6. Team performance for the alternative design

S#	Productivity (gu/min)	Opportunities (gu/min)	Restrictions (gu.min)
`S3`	5.4	10.7	0.19
`S4`	8.1	20.3	0.25
`S5` a)	9.0	18.0	1.8
`S5` b)	10.6	17.8	3.4
`S5` c)	11.7	19.0	6.2

The first rows in Table 6 represent the sequences of actions, `S3` and `S4`, which are less restrictive and offer good opportunities, albeit with lower productivity. The last rows describe the more productive variants of sequence `S5`, but which are the most restrictive and offer only normal opportunities to the other team members.

We end the analysis of the design alternative by noting that the `S5` variants in Fig. 3 are ideal cases and that actual team performance depends upon the evolving state of the board. However, an exhaustive analysis of `S5` variants is clearly unmanageable. By focusing our attention on ideal cases of `S5` we can create a reasonable basis for evaluating and comparing team performance towards the shared goal.

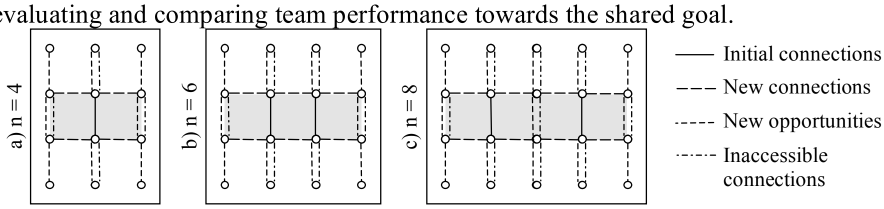

Fig. 3. Analysis of three variants of sequence `S5`

2.3 Using the Technique to Compare Designs: The Big Picture

We now describe how the proposed technique can be used to compare the two design alternatives. Fig. 4 shows the impact of collaborative overhead in total predicted time versus the proportion of time for doing individual actions. The values are sorted by collaborative overhead to facilitate the detection of the sequences of actions that are more costly to perform in the shared workspace.

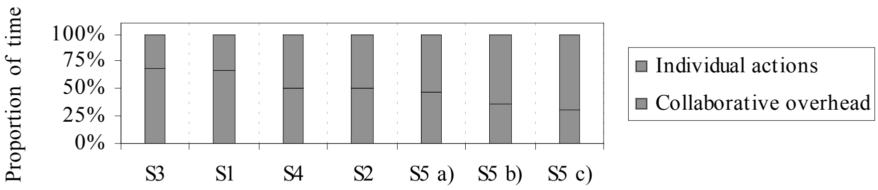

Fig. 4. Summary of collaborative overhead

The data in Fig. 4 show that the two pairs of sibling sequences, `S3/S1` and `S4/S2`, have similar proportions of collaborative overhead, and that the variants of `S5` have the best proportions of individual actions in total predicted time. These results seem to indicate that the alternative design is preferable to the first design, even more so because, intuitively, collaborative overhead has a negative effect in team performance.

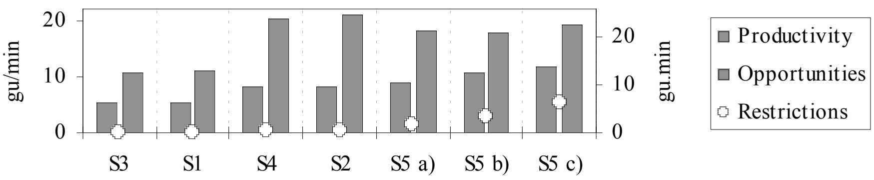

Fig. 5. Summary of team performance

To show that the intuition is *wrong*—at least in this case—we state this proposition: lower proportions of collaborative overhead for achieving individual goals lead to higher team performance towards the shared goal. Now, consider the succession of `s5` variants, with equal ordering in Fig. 4 and Fig. 5. Reading left to right, the proportion of collaborative overhead steadily decreases while the productivity increases in a symmetrical way, the opportunities remain almost constant, and the restrictions raise at a higher rate. So, contrary to the proposition, the lower the proportion of collaborative overhead in the variants of `s5` the *slower* the team progresses towards the shared goal because its team members will probably spend more time waiting to proceed.

Given this somewhat puzzling scenario the designer must find an optimal equilibrium between individual goals and the shared team goal. Were this equilibrium could be is the subject of further work. At the moment the big picture is still getting clearer.

3 Related Work

The use of human-performance models in the CSCW context is very rare, and mostly inexistent for workspace collaboration. DGOMS (Distributed GOMS) [8] is an extension of GOMS that allows group tasks to be decomposed until individual subtasks are reached. A communication operator is then used to coordinate individual tasks executed in parallel, meaning that this method does not address workspace collaboration, but coordinated work. A similar approach is also suggested in a study of GOMS applied to a team task [9], where several users with individual roles were to monitor a display while coordinating their actions via a shared radio communication channel.

We now refer to three methods developed for CSCW that share our purpose of reducing the complexity and cost of CSCW usability evaluation. They are: Collaboration Usability Analysis (CUA) [10], Groupware Walkthrough [11], and Groupware Heuristic Evaluation [12], and, all based on a common framework called "mechanics of collaboration." It is interesting to contrast the CUA and human-performance model approaches; both analyze tasks via hierarchical decomposition but CUA reduces collaboration tasks to the mechanics performed by users in shared workspaces (such as writing a message or obtaining a resource) while human-performance models decompose tasks at a much lower level of detail, for instance, keystrokes. Single keystrokes are most times unrelated to collaborative work—notably when group decision making is involved—which is a strong argument in favor of high-level approaches such as CUA. However, in this paper we hypothesize that the designer of shared workspaces may find it necessary to optimize the effort applied by users in low-level tasks.

4 Conclusions and Future Work

In this paper we show how an analytical technique that is based on human-performance models and three dimensions of team performance—productivity, opportunities, and restrictions—can be used to inform the design of shared workspaces. We also show how the technique can be used to provide quantitative indications of which design alternatives may be more beneficial to team performance.

In our view, shared workspace designers should complement existing practice and knowledge—based on high-level task analysis or depending on inspections performed by multiple usability experts—with the ability to make quick measurements and calculations about shared workspaces efficiency. Our motivation is based on the century-old need to measure before improving as well as on the evidence that faster evaluation enables more design iterations.

Research described in this paper is a preliminary step in the direction of exploring shared workspaces efficiency with human-performance models. As it is, the technique calls for external validity and more work is needed to better understand how it can be combined with other existing techniques and methods.

Acknowledgements

This work was partially supported by the Portuguese Foundation for Science and Technology, Project POSC/ EIA/57038/2004.

References

1. Fjermestad, J., Hiltz, S.: An assessment of group support systems experimental research: methodology and results. Journal of Management Information Systems **15**(3) (1999) 7–149
2. Nunamaker, J., Briggs, R., Mittleman, D., Vogel, D., Balthazard, P.: Lessons from a dozen years of group support systems research: A discussion of lab and field findings. Journal of Management Information Systems **13**(3) (1997) 163–207
3. John, B., Kieras, D.: Using GOMS for user interface design and evaluation: Which technique? ACM Transactions on Computer-Human Interaction **3**(4) (1996) 287–319
4. Card, S., Moran, T., Newel, A.: The psychology of human-computer interaction. Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA (1983) 259–311
5. Card, S., Moran, T., Newell, A.: The keystroke-level model for user performance time with interactive systems. Communications of the ACM **23**(7) (1980) 396–410
6. Antunes, P., Ferreira, A., Pino, J.: Analyzing shared workspaces design with human-performance models. Proceedings of the twelfth international workshop on Groupware, Medina del Campo, Spain. Lecture Notes in Computer Science 4154 (2006) 62–77
7. Olson, J., Olson, G.: The growth of cognitive modeling in human-computer interaction. Human-Computer Interaction **5**(2&3) (1990) 221–265
8. Min, D., Koo, S., Chung, Y., Kim, B.: Distributed GOMS: An extension of GOMS to group task. Proceedings of the 1999 IEEE conference on Systems, man, and cybernetics, Tokyo, Japan (1999) 720–725
9. Kieras, D., Santoro, T. Computational GOMS modeling of a complex team task: Lessons learned. Proceedings of the 2004 conference on Human factors in computing systems, Vienna, Austria (2004) 97–104
10. Pinelle, D., Gutwin, C., Greenberg, S.: Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. ACM Transactions on Computer-Human Interaction **10**(4) (2003) 281–311
11. Pinelle, D., Gutwin, C.: Groupware walkthrough: Adding context to groupware usability evaluation. Proceedings of the SIGCHI conference on Human factors in computing systems, Minneapolis, Minnesota, USA (2002) 455–462
12. Baker, K., Greenberg, S., Gutwin, C.: Empirical development of a heuristic evaluation methodology for shared workspace groupware. Proceedings of the 2002 ACM conference on Computer supported cooperative work, New Orleans, Louisiana, USA (2002) 96–105