

# Users' quest for an optimized representation of a multi-device space

Dzmitry Aliakseyeu · Andrés Lucero ·  
Jean-Bernard Martens

Received: 10 October 2008 / Accepted: 15 March 2009 / Published online: 25 June 2009  
© The Author(s) 2009. This article is published with open access at Springerlink.com

**Abstract** A plethora of reaching techniques, intended for moving objects between locations distant to the user, have recently been proposed and tested. One of the most promising techniques is the Radar View. Up till now, the focus has been mostly on how a user can interact efficiently with a given radar map, not on how these maps are created and maintained. It is, for instance, unclear whether or not users would appreciate the possibility of adapting such radar maps to particular tasks and personal preferences. In this paper, we address this question by means of a prolonged user study with the Sketch Radar prototype. The study demonstrates that users do indeed modify the default maps in order to improve interactions for particular tasks. It also provides insights into how and why the default physical map is modified.

**Keywords** Interaction techniques · Map · Spatial · Reaching · Large-display systems · Multi-display systems

---

D. Aliakseyeu (✉)  
Philips Research Europe,  
HTC34, 5656 AE Eindhoven, The Netherlands  
e-mail: dzmitry.aliakseyeu@philips.com

A. Lucero  
Nokia Research Center, Visiokatu 1,  
33720 Tampere, Finland  
e-mail: andres.lucero@nokia.com

J.-B. Martens  
Department of Industrial Design, Eindhoven University  
of Technology, Den Dolech 2,  
5600 MB Eindhoven, The Netherlands  
e-mail: j.b.o.s.martens@tue.nl

## 1 Introduction

Thanks to the rapidly reducing cost of display and network technologies, situations in which many different devices with heterogeneous display sizes interact together are becoming commonplace. Often these environments present a mixture of personal devices such as Personal Digital Assistants (PDAs), tablet and laptop PCs, and shared devices such as large displays.

In a device-cluttered space, such as the one shown in Fig. 1, the tasks of identifying a particular device and facilitating the transfer of objects from one device to another, also referred to as multi-device (display) reaching, becomes frequent. Therefore, alternative techniques for performing such interactions have lately received a fair share of attention.

A number of interaction techniques have been developed that aim at intuitive and efficient reaching between different devices. In a recent study, Nacenta et al. [15] found that the Radar View, a technique based on the use of a reduced map in which the user can pinpoint the desired destination, performed significantly better than related techniques like the Pantograph [10, 15] and Pick-and-Drop [17]. Their results suggest that Radar View might be a very efficient technique for multi-device reaching.

Map-based techniques such as Radar View [15] have the potential to support the intuitive system identification and interaction without necessarily requiring physical proximity to the system they interact with (although they might profit from it). The success of map-based techniques relies on being able to associate a physical device with its representation on the map. However, how this association is accomplished and maintained has, as far as we know, never been studied in detail. Usually, this process is hidden behind a “smart system” (a black box) that knows at any



**Fig. 1** Environment used in the “Feeding Boris” experiment

moment what should be presented on the map, including how and where these objects should appear.

In this paper, we report on a user study that explores whether or not users would appreciate the possibility of adapting such radar maps to particular tasks and personal preferences. Or, in other words, if users are given freedom to modify the Radar View representation in real time, will they strive to optimize this representation? If so, which criteria will be used to motivate changes?

The study was done using the Sketch Radar prototype [1]. With it, a user is able to control how and what information is presented on the radar at any time. The default representation of a device on the radar map can be acquired in a direct and explicit way. In the current prototype this is accomplished using a barcode reader that allows identifying the device by means of a barcode label. This representation only needs to be acquired once. Subsequent interactions with the representation of the device on the map can be used to change the default appearance, and additional information such as text and sketches can be added. The users are free to adjust the map in such a way that it fits better to a particular task or to their preference.

A short pilot study showed that some users adjusted the default physical map when told that they will be required to repeat some prescribed tasks that they had done before. The goal of the current study is to determine whether or not such behavior is also observed in a more open (less prescribed) task environment and during prolonged use. We wanted to use a natural setting where people would be engaged in an activity over an extended period of time. We also wanted our participants to focus on the activity supported by the tool rather than on the interface with the tool itself. Therefore, we opted for a game setting to conduct our user study.

The remainder of the paper is organized as follows. First we describe related work, then the user study, and finally report our results.

## 2 Related work

The related work can be subdivided into three parts: (1) multi-display reaching and interaction techniques for large displays, (2) interaction techniques that allow connecting to and identifying devices, and (3) remote control techniques.

A number of interaction techniques have been developed to improve interaction in multi-display environments. Pick-and-Drop [17] is one of the first techniques proposed for multi-display reaching. The user can ‘pick up’ an object from the workspace of one system by touching it with a digital pen or any other suitable device, and then ‘drop’ the object anywhere in the workspace of a second system by repeating the touch action in the desired target location. Pick-and-Drop implies that users perform the physical action of moving from one system to the other.

### 2.1 Movement amplification techniques

Techniques like push-and-throw, pantograph and flick are based on transporting the user’s screen cursor from one device to the screen of another device. Throw [10, 15, 23], Pantograph [10, 15], and Flick [13] are all based on the amplification of user movements. The required precision in the user actions of course increases linearly with the amplification used. Unlike the case of Pick-and-Drop, users can stay in a fixed location, provided of course that they can observe the effect of their actions on the remote screen.

### 2.2 Radar views [15]

The Radar technique uses a reduced representation (a map) of the surrounding environment. When the pen touches an object, the map appears. The user can place the object at a desired location by moving the pen to that target location. The Radar View is hence similar to the World in Miniature [21], but in two dimensions. Again, users do not need to physically move to access a remote system, but the required precision of their actions increases when more devices need to be represented within a radar map of fixed size and resolution.

### 2.3 Sketch Radar [1]

The common implementation of the Radar View is based on the physical positions of interacting devices. This imposes limitations on how the map is acquired and managed. The Sketch Radar tries to solve these by allowing a user to control how and what information is presented on the radar at any time. The representation of a device on the radar map can be acquired in a direct and explicit way. In the current prototype, this is accomplished by means of a

barcode reader that reads a device's barcode. Therefore, it solves a key problem of the existing Radar interaction technique by providing an easy and quick way to manage one or more maps of available devices. The Sketch Radar prototype was used for the experiment reported in this paper.

Another example of a system that uses the radar metaphor and addresses how physical devices can be arranged on a map is ARIS [4, 5]. ARIS uses an iconic map of a space as part of an interface for performing application relocation and input redirection.

The success of map-based techniques such as the Radar View [15] relies on being able to associate a physical device with its representation on the map. Or in other words, Radar Views support stimulus-response compatibility (SRC). SRC was introduced in 1953 by Fitts et al. [9]. It was shown that the speed and accuracy of responding are dependant on how compatible stimuli and response are. Duncan [8] has studied spatial SRC and found that if spatially distributed stimuli (lights) and responses (buttons) have compatible arrangement subjects were able to response faster than when the arrangement was incompatible. However, the effect of SRC is unclear when more complex tasks need to be solved.

It was also shown that the spatial organization of displays allows efficient access to them, in the sense that it outperforms existing tree- or list-based approaches (such as File explorer or Favorites in Internet explorer) [7, 20].

Jones and Dumais [11] are questioning the utility of a spatial metaphor over a symbolic one. Their evaluation showed that spatial organization alone provides weaker retrieval cues than semantic labels; however, the combination of the two enhances performance.

Next to the above interaction techniques that were specifically developed for multi-display reaching, there are a number of large wall and tabletop interaction techniques that can be adopted for the purpose. Drag-and-Pop [2] and Push-and-Pop [6] are examples of techniques that use semantic information to assist users in their interactions, by bringing potential targets within reach.

A second class of interaction techniques, such as SyncTap [18], Proximal Interactions [19], InfoPoint [12], or GesturePen [22], aim at identifying devices in a direct and explicit way, usually with the intention of establishing a connection with (or between) them.

A number of applications have been developed to use a PDA as a mediator between stationary computers and other devices, or as a (remote) control for distant devices, especially those devices that do not possess their own controls, or devices that do not have a display. Examples of such techniques are Semantic snarfing [14] and the Personal Universal Controller [16].

### 3 User study

#### 3.1 Background

The original implementation of Radar Views is based purely on the physical position of interacting devices. This raises several questions/issues:

1. Which devices should be presented on the map? Should all devices be equally prominent?
2. How does the nature of the task and user preferences affect the map?
3. How do users deal with the fact that the map needs to be presented on a screen with limited size and resolution?
4. How should devices be represented (for instance, how can horizontal and vertical screens be represented on a single planar map)?
5. What are the boundaries of the map?

Based on these questions we have formulated our main research question as follows:

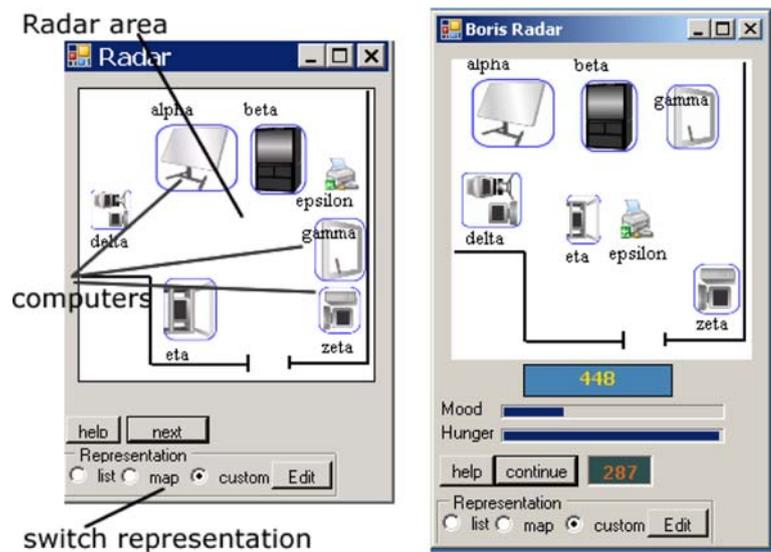
Given the freedom to modify the Radar View representation in real time, will users strive to optimize this representation? If so, which criteria will be used to motivate changes (nature of the task, prior knowledge of the environment, spatial location, etc.)?

In order to address this question we have performed a prolonged user study that consisted of two parts split over several days: the first part consisted of several controlled sessions in which participants performed preset tasks and the second part was an unconstrained gaming situation.

The Sketch Radar prototype was used in the study (Fig. 2). It allows using preset (physical) maps, user-created maps and simple lists for interacting in a multiple-device environment (the Sketch Radar is described in more details in [1]). For example, in a new environment it is usually wise to start with a map that is based on the physical position and size of the devices. After some time, an environment becomes more familiar and tasks become clearer. This may lead the user to readjust the positions, sizes, and representations of the devices that are represented.

For example, frequently used devices may be increased in size and placed closer to the center of the map. Also, by allowing users to add "sketches" (lines, text) to the map, they can add elements that further strengthen the association between a specific map and a particular task. This flexibility makes the Sketch Radar useful in different situations, ranging from interaction in an unfamiliar space where a close correspondence with the physical arrangement is needed to identify individual devices, to frequent and long-term usage, where the physical space is well

**Fig. 2** Sketch Radar main window (left); Sketch Radar in game mode (right)



known and users can profit from a map that is specifically tailored to their purpose. It is expected that this diminishing importance of “physical” correspondence will go hand-in-hand with growing user knowledge about the task and space.

In the next section, the game that was used for the second part of the study is introduced.

### 3.2 Game description

Feed Boris is a Tamagotchi-like game and was inspired by the Feeding Youshi game presented in [3].

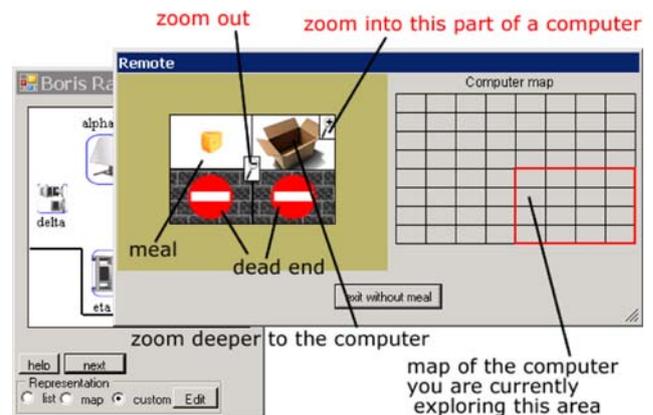
The main goal of the game was to feed a virtual cat called Boris. Boris is continuously traveling between different computers to find a “safe” hiding place. Depending on the players’ actions Boris would become hungry or unhappy, which in turn determines his most likely hiding place. For example, if he is happy and hungry he will look for more open places so that he can easily be found. When he is unhappy, on the other hand, he is likely to hide so that it might be more difficult to find him. Both hunger and happiness were defined based on how a player fed Boris, i.e., the hunger level of Boris was calculated based on the meal’s nutritious level and frequency of feeding, while the happiness was determined by the diversity of meals (if a player offers Boris the same kind of meal all the time he will refuse to eat it and quickly become unhappy). Players could observe the current status of both parameters at any time (Fig. 2, right). However, the level of hiding behavior was not visible, so players had to learn to associate this to the level of hunger and happiness, during the course of the game.

The Sketch radar [1] was modified to accommodate the study. The radar map stayed the same as in the original implementation [1], but instead of files, different kinds of

meals (nine in total) could be found on the computers. The remote control function allows exploring computers in order to find different kinds of meals or to find and feed Boris (Fig. 3).

Every time the meal was given to Boris the player was rewarded with scores. The scores were calculated based on happiness, hunger, and the nutritious level of the given meal. In addition, the scores were constantly added or subtracted depending on the current happiness level.

The exploration of a computer with Sketch Radar is done through a hierarchical (tree-like) interface (Fig. 3). By tapping-and-holding the pen on one of four regions, the selected region is opened up into the next level of the hierarchy. The player starts at the top level of the hierarchy and can zoom into different parts of the hierarchy. There are three levels to the hierarchy. The amount of zoom required matched with Boris’s hiding behavior. More specifically, level one implies that Boris is at the topmost level of the computer, so that no zooming action is required



**Fig. 3** Sketch Radar: remote control interface

to find and feed the cat. Level 4 signifies that Boris is hiding at the deepest level, so that three consequent zooming actions will be required to locate him.

The computers that play a part in the game were not directly accessible, they only provided visual information (i.e., only the displayed output of the computers is available). For example, the player could find out where the cat is by either exploring computers one-level-at-a-time through Sketch Radar or by checking all levels at the same time on the screen of the computers (Fig. 4). However, to feed the cat the player needed to use Sketch Radar.

A TabletPC with the Sketch Radar prototype software was used to access and explore the different computers, to gather food and to feed Boris.

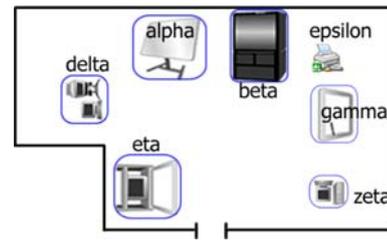
In order to examine the effect of the specific task both Boris's movements and the meals locations were non-random. For example, Boris would only hide on 3 of the 10 computers, and specific kinds of meal would only appear on specific computers. During the first part of the study, participants were receiving different hints (for example, "Boris usually hides on computers with large screens" or "Boris has found a new hiding place its computer Theta.")

### 3.3 Apparatus

The test started in a single room which contained multiple devices with which the participant needed to interact: two PCs with turned-on displays (Zeta and Delta), one PC with the display turned off (Eta), one tabletop display (Gamma), one printer (Epsilon) and two wall displays (Alpha and Beta) (see Fig. 5). All devices were clearly labeled with their respective names. During the course of the study two new rooms were introduced, each room contained a single PC with a display (Theta and Kappa).



**Fig. 4** Information displayed on the screen of one of the computers. The size of the meal or the cat shows how much the player has to zoom in to reach it (in this specific case Boris requires one zoom action, while the fish requires three consecutive zooms), the position shows the part of the computer that the player needs to zoom in to



**Fig. 5** The layout of the first room. Icons reflect actual appearance of the devices

### 3.4 Participants

The experiment was conducted with seven participants (two females and five males) between the ages of 23 and 35. All participants had previous experience with graphical user interfaces, but not with Sketch Radar. The environment where the study took place was familiar to all participants. The participants were tested individually.

### 3.5 Tasks

The experiment consisted of three parts: tutorial, controlled sessions, and free-form game.

In the first part, participants performed multiple training tasks with the Sketch Radar application on the TabletPC, following a map builder tutorial. The duration of this first part varied across participants from 30 to 60 min.

The second part lasted for 3 days and included one 20- to 40-min session per day. On the first day, participants received the TabletPC with a preloaded physical map of the first room. All systems were presented equally on the map (in terms of geometrical size) in a position that closely corresponded to their actual physical position within the room. The participants were also positioned inside the same room. Their task consisted of feeding Boris the cat with specific meals. During the first session, participants performed 20 trials, and were not allowed to modify the map. Before every trial a hint was given, for example "Boris prefers to hide on large computers," or "This kind of meal is very rare and always well hidden." Immediately after the session, participants were asked to modify the map and to create their own representation of the environment using the knowledge that they had acquired while performing the tasks (i.e., having gained experience with where Boris usually appeared and where meals were most likely to be found, etc.). In the second and third sessions, one or two additional rooms that were positioned further down the corridor from the first room, were introduced, respectively. Participants were free to start from the default physical map, their own map that they had created before for the first room, or a name-based list representation. They were allowed to keep modifying the map at any time.

Participants were positioned inside the first room during the second session, and outside of it (in a closed room, where they were not able to see the screens of the computers) in the third session. The tasks to be performed were similar to the tasks in the first session.

The third part of the experiment was the actual game. It also lasted for 3 days (with 15–30 min playing sessions every day). Users started from the maps and knowledge that they had acquired from the second part of the experiment. The game involved all three rooms. Users were free to choose where they wanted to be physically, but all of them chose to play the game from within the first room (which contained most of the systems). The goal of the game was to acquire as many points as possible by feeding Boris, in a given time. Participants were aware of the fact that the one who collected the maximum score would get a prize.

After every session a short interview was conducted in order to evaluate the participants' perception of the game environment. In the first part participants were asked to describe those computers that shared task-related properties using computer names, locations, etc. For example, "Please describe computers where you usually can find Boris." In the second part, they were interviewed about why they chose a specific representation (such as list or map). In case they had used a modified radar map, they were interviewed about all modifications that they had made to the map.

### 3.6 Results

The evaluation showed that users indeed changed the layout of the map to make it more suitable for the particular task that they needed to perform. Most of the participants (5/7) only adjusted the map before and after test sessions, but not during the session itself.

By the end of the experiment, all participants had created their own representation, only two participants used the preset physical map during the first part of the experiment, but changed it after the first game session. All other participants switched to their own representation after the first session of the first part.

There are some more specific observations that were made during the experiment:

(1) Physical location provides strong external cues, while custom-made representations which are often based on internal cues that might be forgotten or changed, need repetitive usage to be remembered. Between sessions, some participants (3/7) had forgotten about acquired patterns of cat and food behavior. Therefore, their own representation created during a previous session did not make sense to them anymore, and even caused confusion. In such cases, participants either returned to the physical map or created a new representation from scratch.

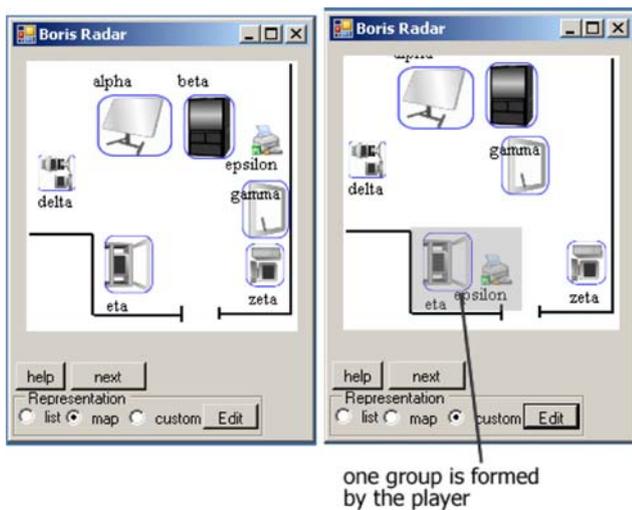
(2) In the post interview where participants were asked to describe computers that shared the same task-related property, the description usually relied on properties provided in the game hints (6), names (3), look (2) or/and location on the map (2). For example, if the provided hint stated that "Boris is hiding on computers with large displays," the most common answer to the question: "Where does Boris usually hide?", would be "Large computers Alpha, Beta, and Gamma." After the last sessions, most of the participants completely moved to the hint-based property, so the answer on the above question became "Large computers."

(3) If to the known group of computers (for example, "Large computers where Boris hides") a new computer is added ("This is a new computer Boris also can hide here"), even without giving it any specific properties, it will acquire the properties of the group. So first time it will be referred as a "new one," and after that it will usually be referred together with the rest of the group so "Large computers Alpha, Beta, Gamma, and Theta [new computer]." This new computer Theta that is actually physically small is placed in the group of "large computers" which no longer corresponds to the physical size but more to the fact that Boris can be found on them. Therefore, "large computers" evolve from being a property of the computer to becoming a label. This was observed with four out of seven participants.

(5) When placed in a second separate room, only one participant moved from a physical to a purely task-oriented map. Others commented that if they would start from the separate room it might be quite possible that they would adjust the map for the first room more drastically. But since the first room was well-known and they had started the experiment in it, they had already built some mental map of it that provided them with rich cues.

(6) Four common steps in the evolution of custom-made representations (or maps) could be identified:

1. The physical maps are only slightly distorted. The icons that represent those devices are slightly resized and repositioned to make movements shorter. No specific grouping is made. (5/7)
2. The map is moderately distorted (Fig. 6). Some grouping is made. For example, computers where food appears more often are grouped together. However, participants try to maintain as much as possible a correspondence to physical location. (5/7)
3. The map is strongly distorted (Fig. 7b). Only the computers that have screens and that are located in the first room retain a position that correlates strongly with the actual physical location. Computers that do not have a screen are positioned freely based on different properties that varied from participant to participant.



**Fig. 6** Physical (left) and modified (right) map. The custom map is moderately distorted, with only one group (computers that have only one hiding place and one type of food)

Computers that were originally outside of the first room were positioned freely, although still kept outside of the room boundaries. (6/7)

4. The map is completely distorted (Fig. 7c). Computers are grouped based on certain properties, no correspondence with physical location. However, some order-based spatial relationships between computers are retained. Despite the fact that the actual location does not longer matter, relative relationships have remained (such as this computer is to the left, right or in front of that computer). (4/7)

(7) During the experiment, all devices with screens were constantly displaying information about their status. The same information was available through the Sketch Radar, but in order to obtain this information, participants needed

to go through several steps. We observed that during the game participants very often instead of exploring the device representation on the TabletPC were first checking the content of surrounding displays, locating the cat or needed type of food, and only then accessed the food or cat through the Tablet. They would only start to look for the cat through the TabletPC if it was not visible on any of the screens. We believe that is why most of the participants did change the map, but also tried to partly keep some references to the physical location of devices.

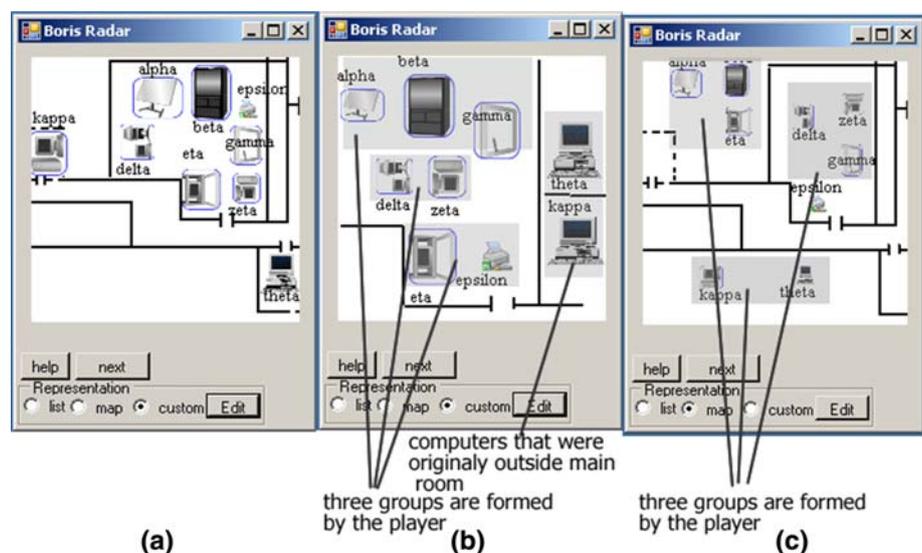
The speed that this transformation occurred with varied between participants (Fig. 8). Some participants skipped steps in between. Two participants immediately after the first session created custom-made representations that were moderately distorted. One participant moved back to the physical map, used it for two consequent sessions and then jumped to the strongly distorted representation (Level 3).

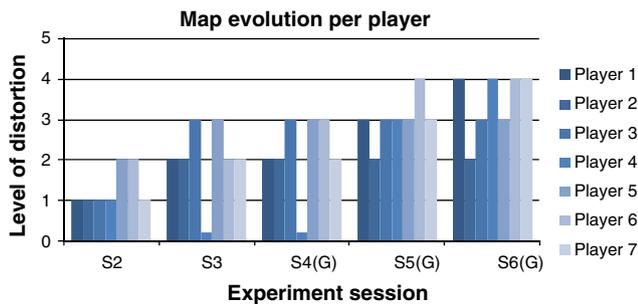
(8) While creating their own representation participants only adjusted location (7/7) and size (6/7), and have not used any other features of the Map Builder, such as sketching or adding text. Several participants commented that they were thinking of adding some labels, but none of them did.

(9) Participants usually grouped computers based on the kind of food they provide, the amount of clicks needed to reach a specific kind of food (so they would first group together “shallow”/discrete computers that require only one click to get a food, and that do not have a zooming-in possibility (5/7), the next group will be the group with computers that require maximum amount of clicks (2/7)), how often the computers are visited by Boris (6/7), if the computers have a screen or not (7/7), and if the computer is located inside or outside of the room (7/7).

(10) In addition to grouping, some participants reduced the distances between computers to improve movement

**Fig. 7** **a** Physical undistorted map; **b** the custom map is strongly distorted: four groups are formed by the player (computers that have only one hiding place and one type of food, computers with large screens, small computers, and two computers located outside of the starting room); **c** the custom map is completely distorted: three groups are formed by the player (computers with large screens, two computers located outside of the main room, and small computers together with the tabletop computer)





**Fig. 8** Level of map distortion on every session, for every player (during first session all players used the physical map). Level 0 means original physical map

time, and some changed (usually increased) the size of computers to more efficiently use empty space.

Figure 8 illustrates how the map evolved during the course of experiment. It is clearly visible that after session 4, three out of seven participants have reached a stable representation that they have no longer modified. The post-questionnaire showed that the main reason why no changes were made is that they had experienced the representation extensively, so that any change to this established relation could cause a confusion and therefore reduce the performance. An interesting observation, in terms of scores, is that participants who kept the representation stable during the whole game part (sessions 4–6) collected more scores at the end. Only participants who were not satisfied with their results changed the representation during the game sessions.

Based on these results we can conclude that during prolonged usage of a modifiable Radar View representation, users do strive to optimize the representation based on the task and personal preferences. The nature of the task is the main criterion for motivating the change; other less important criteria are the location of devices, the amount of available space, the visibility of devices, and the type of devices.

However, it is still unclear if the new representation is more efficient than a physical location-based representation. It also remains difficult to derive how exactly and why tasks affected the change.

### 3.7 Design guidelines

Based on the results of the study, we can formulate the following guidelines for building reaching interaction techniques that are based on a map-like representation:

- If the number of computers is small and they all have observable screens and the interaction occurs only inside the represented area, a simple physical mapping such as the iconic map in ARIS system [5] would be the best representation.

- If the interaction occurs outside of an environment, even in the case when the users know the environment, it is wise to use a representation that allows better task-oriented interaction. However, the mapping should be very clear to the users so they can easily remember it.
- In mixed environments, a tool that allows some adjustments of the map is most appropriate.
- In situations where available space is limited, the exact spatial locations of devices can be sacrificed to looser, order-based, relations.

## 4 Discussion

### 4.1 Mobility

Mobility was not addressed in the study. However, it is an important aspect that might influence the perception of the map and behavior of users. There can be two situations: one where the user is moving and another where some device(s), that are part of the environment, are mobile. If a matching physical representation is used, then the position of the device can be dynamically updated and displayed on the map. However, if the representation of the environment does not match physical locations (for example, when adjusted in accordance to the task) positioning of the mobile device might be problematic. Different approaches might be used to resolve this issue, for example, the mobile device can be represented on the map as another static device, or the system can automatically position the device based on its distances from other devices represented on the map (for example, a mobile device can be shown next to the static device that is currently closest).

### 4.2 Effect of the task

In this study, all participants had the same tasks and experienced the same cat and food behavior. Therefore, it is more difficult to measure the effect of the task. A second group of participants that would have different cat and food behavior would help to measure the effect of a task more precisely.

### 4.3 Multi-user

Another aspect that is clearly relevant for multi-device environments that we consider here is multi-user collaboration, either co-located or not. Although it is allowed to have multiple Sketch Radar devices operating within a single environment, where participants can even exchange radar maps, it is less clear how conflicts should be handled

and how performance and appreciation should be measured.

#### 4.4 Privacy

Our experiment did not address privacy issues that are also involved in multi-device operations. In case nonaccessible devices show up in the radar maps, the most straightforward response would be to simply remove or minimize them. Using a different representation for systems that are only accessible for reading might also be an option.

### 5 Conclusions and future work

One of the most promising reaching techniques is the Radar View. We performed a user study that explores whether or not users would appreciate the possibility of adapting radar maps to particular tasks and personal preferences and if so, which criteria would be used to motivate changes. A modified version of the Sketch Radar prototype, that provides an easy and quick way to manage one or more maps of available devices, was used for implementing the experiment.

The study confirmed that users indeed modify the map for different reasons, namely type of computers, relation between computers defined by the task, visibility of the computers, spatial relation, and order of computers. Since no explicit performance measures are available it is still unknown if an altered representation is more efficient than a representation purely based on the physical locations.

In the future, we plan to run several studies in which we want to collect quantitative results, more precisely measure the effect of the task, and compare the performance in different environments with different representations. Based on the results of these studies we hope to formulate guidelines for (automatic) generation of environment representations that would efficiently facilitate the task of reaching.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

### References

1. Aliakseyeu D, Martens J-B (2006) Sketch Radar: a novel technique for multi-device interaction. In: Proceedings of HCI'2006, Vol. 2, British HCI Group, pp 45–49
2. Baudisch P, Cutrell E, Robbins D, Czerwinski M, Tandler P, Bederson B, Zierlinger A (2003) Drag-and-pop and Drag-and-pick: techniques for accessing remote screen content on touch-and pen operated systems. In: Proceedings of Interact 2003. IOS press, Amsterdam, pp 57–64
3. Bell M, Chalmers M, Barkhuus L, Hall M, Sherwood S, Tennent P, Brown B, Rowland D, Benford S, Capra M, Hampshire A (2006) Interweaving mobile games with everyday life. In: Proceedings of CHI 2006. ACM Press, New York, pp 417–426
4. Biehl JT, Bailey BP (2004) ARIS: an interface for application relocation in an interactive space. In: Proceedings of graphics interface, pp 107–116
5. Biehl JT, Bailey BP. A toolset for constructing and supporting iconic interfaces for interactive workspaces. In: Proceedings of Interact 2005. Springer, Berlin, pp 699–712
6. Collomb M, Hascoët M, Baudisch P, Lee B (2005) Improving drag-and-drop on wall-size displays. In: Proceedings of GI 2005, pp 25–32
7. Czerwinski M, van Dantzich M, Robertson GG, Hoffman H (1999) The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3D. In: Proceedings of Interact '99. IOS press, Amsterdam, pp 163–170
8. Duncan J (1977) Response selection rules in spatial choice reaction tasks. In: Dornic SVI (ed) Attention and performance. Erlbaum, New Jersey, pp 49–61
9. Fitts PM, Deininger RL (1954) S-R compatibility: correspondence among paired elements within stimulus and response codes. *J Exp Psychol* 48:483–492
10. Hascoët M (2003) Throwing models for large displays. In: Proceedings of HCI2003. British HCI Group, pp 73–77
11. Jones WP, Dumais ST (1986) The spatial metaphor for user interfaces: experimental tests of reference by location versus name. *ACM Trans Office Inform Syst* 4(1):42–63
12. Kohtake N, Rekimoto J, Anzai Y (2001) InfoPoint a device that provides a uniform user interface to allow appliances to work together over a network. *Pers Ubiquitous Comput* 5:264–274
13. Moyle M, Cockburn A (2002) Analyzing mouse and pen flick gestures. In: Proceedings of SIGCHI-NZ, pp 266–267
14. Myers B, Peck CH, Nichols J, Kong D, Miller R (2001) Interacting at a distance using semantic snarfing. In: Proceedings of UbiComp'2001. ACM Press, New York, pp 305–314
15. Nacenta MA, Aliakseyeu D, Subramanian S, Gutwin CA (2005) A comparison of techniques for multi-display reaching. In: Proceedings of CHI 2002. ACM Press, New York, pp 371–380
16. Nichols J, Myers B (2003) Studying the use of handhelds to control smart appliances. In: Proceedings of ICDCS'03, pp 274–279
17. Rekimoto J (1997) Pick-and-drop a direct manipulation technique for multiple computer environments. In: Proceedings of UIST 1997. ACM Press, New York, pp 31–39
18. Rekimoto J, Ayatsuka Y, Kohno M (2003) SyncTap: an interaction technique for mobile networking. In: Proceedings of mobile HCI 2003
19. Rekimoto J, Ayatsuka Y, Kohno M, Oba H (2003) Proximal interactions: a direct manipulation technique for wireless networking. In: Proceedings of Interact'03. IOS press, Amsterdam
20. Robertson G, Czerwinski M, Larson, K (1998) Data mountain: using spatial memory for document management. In: Proceedings of UIST 1998. ACM Press, New York, pp 153–162
21. Stoakley R, Conway M, Pausch R (1995) Virtual reality on a WIM: interactive worlds in miniature. In: Proceedings of CHI 1995. ACM Press, New York, pp 265–272
22. Swindells C, Inkpen K, Dill J, Tory M (2002) That one there! Pointing to establish device identity. In: Proceedings of UIST 2002, ACM Press, New York, pp 151–160
23. Wu M, Balakrishnan R (2003) Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: Proceedings of UIST 2003, ACM Press, New York, pp 193–202