# Wizardry in Distributed Participatory Design

## From Design to Implementation

Malin Wik(✉) and Akhona Khumalo

Karlstad University, Karlstad, Sweden
malin.wik@kau.se

**Abstract.** Many participatory design methods and techniques assume that the designer and the participant are in the same location. In this study, we explore methods for Distributed Participatory Design. Combining the Wizard-of-Oz technique with oral discussions, in three iterations, we allowed users to actively participate, over distance, in designing a solution for their Events Management and Booking System. A video prototype was captured and used as a specification to communicate the system requirements. System development issues that arose were captured in a log and used to explore the effectiveness of such a specification in the development process. The results show that using wizardry in distributed participatory design is a viable method for allowing active participation and that video prototypes can sufficiently communicate system requirements.

**Keywords:** Distributed participatory design · Wizard of Oz · Video prototyping · Requirements specification

## 1 Introduction

Distributed Participatory Design (DPD) offers great opportunities. It reduces the need for travel and can bring design expertise to the participants and bring content and field expertise to the designer. DPD also poses challenges, but these can be overcome with tools that support collaboration in the essential channels of interaction design, namely the user interface.

This paper will discuss a case demonstrating both sides of the coin. The study furthermore explores the advantages and disadvantages of using the outcome of the DPD sessions, namely, an interactive prototype as the requirements specification (or at least to demonstrate the specifications) and the effectiveness of such a specification in the resultant developed solution.

### 1.1 Background

The original participatory design projects in the 1970s involved the workers of one organization as workplaces went through the process of digitalization [1]. Today, organizations may need information systems and groupware that allow them to interact and cooperate with branches in different geographical locations or even with other organizations.

M. Kurosu (Ed.): HCII 2020, LNCS 12181, pp. 172–186, 2020.
https://doi.org/10.1007/978-3-030-49059-1_13

The field of Computer-Supported Cooperative Work (CSCW) deals with the issues of cooperation over time and space, and the field of Information Systems and Software Engineering deals among other things with the issues of distributed development and outsourcing [2]. But the design of the groupware allowing these collaborations to happen is also a question for the field of Human-Computer Interaction. The participatory design processes of today should allow for active participation of users in different geographical locations and, which might be even more challenging, from different organizations [3]. Many of the methods and techniques applied in participatory design processes, however, assume that the designer and the participant is in the same location.

One key issue of Distributed Participatory Design is thus mitigating the geographical distance, without removing the design process and the participants from the context of use [4]. There have been some advances on the matter, but still the available literature on distributed participatory design (DPD) is limited.

## 2 Related Work

### 2.1 Distributed Participatory Design (DPD)

Some studies within Distributed Participatory Design have focused on how to enable participation via distance. [5] for example report on a project in the late ’90s where email and web-based prototypes were used to facilitate collaboration between users and developers in a distributed development project of groupware. New designs were communicated using web-based prototypes which the users, after receiving email notification of a new prototype made available, were asked to test. Mailing lists were used for the majority of the communication such as feedback on prototypes, questions from the developers to the users, decision making [5].

In [6] a tool, DisCo, for asynchronous and distributed co-design between children and adults was developed and tested. The participants were asked to design a reading game using the tool. The authors found that the participants demanded more from the tool than participants in analogous co-design sessions had done when it came to sketching their designs. Furthermore, the authors found the co-design sessions needed more facilitation, meaning the participants needed clear instructions of when to do what.

The study reported by [7] explored how to “remotely discuss, develop and test GUI prototypes with users and stakeholders” using a web-based prototyping tool (the technique, GUI-ii, will be introduced in Sect. 2.4).

[8] recently report on a distributed participatory design in a crowdsourced systems development context, but seem to focus on the distribution and participation of the developers rather than the future users.

[4] explored requirements for a web-based groupware rendering distributed participatory design but did not explore techniques for distributed participation in themselves. Another study that is within the DPD field, but do not include methods for distributed participation is reported in [9], in which the authors report on their participatory design methods used in a distributed and inter-contextual setting.

### 2.2 Requirements Specifications

In 2003 [10] discuss the difficulties of requirements engineering for multimedia systems. While *multimedia system* may sound a bit outdated, multimedia in that work refers generally to the "extrovert parts" of a system and the important role they play – i.e. the interactive graphical user interface (GUI) of a system. The authors argue system requirements should be defined in ways that make them easy to verify and measure, but note that interactive systems contain many properties that are difficult to explicitly express and measure. [10] recommend the multimedia requirements to be visualized rather than written, for example through interactive prototypes.

**Capturing System Requirements on Video.** [11] suggest that using video representations provide a means to capture and communicate interactive aspects of a system, such as dynamic content and usage scenarios.

[12] ascertain the effectiveness and efficiency of videos in requirements elicitation and validation and find that under time pressure videos clarify requirements better than use cases. They do not explore the use of videos in design and development but propose these processes could benefit from video documentation.

[13] emphasise that the main focus of videos is the design statement, not the quality of the videos. The authors add that the ease of availing videos digitally has made them more attractive for expressing and documenting interaction design ideas.

### 2.3 Active Participation

Already in 1991, [14] noted the importance of what they label *envisionment*. By this concept, the authors refer to a process that allows the future user to actively use a prototype and influence it. This, in contrast to a user attending a demonstration, reading a system description or reviewing prototypes of different fidelity, let the user experience the system design. The breakdowns can lead to changes in the design and thus in the future system. *Envisionment* is important in revealing issues that would, otherwise, not be revealed until the final system is in place [14].

*Envisionment* is similar to what [15] term *genuine user participation*. Without genuine user participation, the users are viewed as informants, asked only to share their view of their work in interviews, surveys or perhaps by involving them in final system testing. Instead [15] propose the participation should be a mutual learning process between a user and IT designer, and that the users should have a say in what affects their working conditions. How people participate is just as important as who participate.

In this paper, we refer to *envisionment* and *genuine user participation* as *active participation* for the sake of simplicity. The active participation of future users has in the present study been encouraged by utilizing GUI interaction interviews (GUI-ii), which is a method explained in the following subsection.

### 2.4 Graphical User Interface Interaction Interviews (GUI-ii)

Graphical User Interface interaction interviews (GUI-ii) is a method which can be used over distance to explore a graphical user interface, often in the form of an explorative and interactive prototype [7].

By using GUI-ii, the fact that the designer is a human can be fully appreciated. The participant and the designer (who could just as well be another participant or a system developer) can interact as in a face-to-face interview or rather a design workshop, but via distance with the interactive graphical user interface between them [7] but also facilitated by voice communication. This way discussions about the design are carried out in direct relation to the interface, where both participant and the designer can see and follow each other's interactions, instead of having to link discussions to external artifacts [5].

### 2.5 Tools for Distributed Participatory Design

There are numerous prototyping tools at hand for the designer creating prototypes. The problem is, however, that a majority of the tools do not incorporate structures allowing *active participation,* which is vital to the participatory design process. In some tools the designer can decide on the interaction design beforehand by linking elements together, making the prototype appear as functioning. Very few of these tools, however, allow the designer to view the participant's interactions, make swift changes in the interaction or graphical design and let the participant interact with the prototype with inputs beyond clicks. To circumvent these limitations the Wizard-of-Oz tool Ozlab [16], developed at Karlstad University, was used to create the explorative prototypes and conduct the GUI-ii sessions.

**Ozlab: Interactive Prototyping and Testing Tool.** When using a Wizard-of-Oz setup, the prototype appears functional enough for the participant to perform her tasks and achieving her goals [16, 17]. In reality, all system responses are controlled by the human experimenter, the wizard, who can follow all interactions of the participant and may rapidly modify the interface in the meantime, evaluating changes directly. Allowing for swift changes in interaction design is an advantage of using a human wizard instead of preprogrammed responses in the prototype.

The participants get to see directly how their suggestions would look and feel. [18] argue this direct response give the participants a feeling of control as well as it increases their engagement and motivation to continue participating in the system development process.

## 3 The Study

In three iterations with five participants in each iteration (n = 3 × 5), GUI-ii was employed in the design of an Events Equipment Booking and Management system. The stakeholders of the system had the opportunity to interact with an Ozlab-based prototype of the system and give oral feedback while doing so. The designer was in Sweden while the participants were in South Africa.

This study is limited to synchronous participation since the method used, i.e. the Wizard-of-Oz technique, is based on the interaction between two humans. Synchronous participation may pose a challenge if the participant and the designer are in varying time zones. However, it worked out well during this study since the time difference between Sweden and South Africa is 0–1 h (depending on Daylight Saving Time).

Participants were emailed a consent form and instructions. Both the consent form and instructions were discussed with the participant, and participants were asked to give consent verbally before a session commenced and then email the signed consent form. One selected company employee was requested to collect the physically signed consent forms and email scanned copies in the first and third iterations. In the second iteration, the participants were asked to email the consent forms themselves. This was because printing facilities were not always readily available at the start of a session.

After the three iterations a video prototype was created. Thereafter the implementation of the system began, using the video prototype as a requirements specification.

The whole process is described in some detail in the following paragraphs.

### 3.1 Initial Requirements Gathering and Creation of Prototypes

At the beginning of the study, preliminary informal interviews were held with two of the company employees in an open-ended manner to get an introduction to the company's existing work process, its shortfalls and what the desired system should fulfill. The two participants were the warehouse manager and technical manager at the company. These were considered best suited because they are constantly at the center of the process of briefing, requesting and issuing equipment for events. Following the interviews, three 'rough' designs were put together using Ozlab. These were developed keeping in mind the guidelines discussed by [19] for navigating the interface, organizing the display, getting user's attention and facilitating data entry.

### 3.2 Iteration 1

Five participants were requested to participate in the GUI-ii sessions to further understand what the ideal system should entail. They could share what they thought would and would not work in ensuring great interactivity for them while helping them execute their work tasks in a better way. The system had to present them with a list of events equipment grouped according to sections (departments), displayed on different pages. The participant would select the desired items and capture the quantities required to make a booking. Recommendations would be shown when some equipment is selected. This feature would assist the capturer to remember items that usually work together, that may otherwise be forgotten.

The first iteration of GUI-ii sessions included discussions on which of three types of proposed designs appealed to the stakeholders. Each participant used all three designs, simultaneously discussing by telephone and gesturing by mouse cursor on the computer screen what they would or would not prefer. Participants were called by the designer at the start of the session and, following introduction to the study and gaining consent to proceed, explained the intent and process of the session and what the participant's role was in the design process. Each session was an hour long.

The participants were requested to make a booking by finding and selecting 15 specified items (of which 2 were pop up recommendations based on other selected equipment) from each of the three different designs. Thinking out loud was encouraged. This provided a means to understand what was going on in their minds as they worked through the designs and allowed for comparison of what people said against what they did - how they carried out the tasks and whether they had ease or difficulty finding the required items.

Participants were tasked with finding a list of equipment for a new booking, opening a previous booking to process a return of equipment and discussed further the need for keeping a history log. The order of designs to be tested was given to the participants in varying sequences. Upon completing interacting with each design, the participants answered a questionnaire that helped communicate and give feedback about what they would like altered and added to make the design more suitable. After the three designs had been evaluated, the participant was asked to select their most preferred design. All participants were stakeholders from the company.

The designs were all well received with the preferred system (one that allowed them to select items using checkboxes) leading by 3:2 to another (which allowed them to select by clicking on an image of the relevant item). A third option allowed the participants to make their selection by using a quantity drop-down box next to each equipment item. Suggestions during discussions included rearranging items to improve the layout and ease of finding equipment. Examples were categorizing items, arranging them in alphabetical order, arranging them from left to right rather than from top to bottom and using colors for different categories. Participants tended to rate the first design option high and then sounded a little hesitant about their previous ratings after the subsequent designs had been tried out.

### 3.3 Iteration 2

The second iteration was preceded by the designer giving feedback on the findings of the previous GUI-ii sessions and presenting the design that will be used going forward. Here we also discussed and weighted other requirements that had surfaced. The design that had been selected as most favorable in iteration one was updated and explored in iteration two. Five participants took part in the GUI-ii sessions of this iteration by following instructions that required them to carry out various given tasks. These included carrying out an event booking, an equipment-hire, an equipment-return and accessing various other system aspects to discuss proposals and expectations of how these would work. Participants were given a call at the beginning of the session and stayed on the call for the duration of the session.

The initial proposals and the changes that had been made were discussed during the session and thinking out loud was encouraged. The wizard responded to certain events as the participants interacted with the prototype and jotted down what was discussed. Due to lack of availability, the participants of the second iteration were not from the company, but the design in this iteration had been introduced to the company participants. Consequentially though, this allowed us to ascertain if the system was usable, easy to learn, follow and understand, appealing, and pleasant to work on even for someone who was not familiar with the company's current practices. The sessions, therefore, focused less on the intricacies of the functionality within proper context and granularity with regards to details than must be availed in the system. Participants included three people who were in the same career field (Sound Engineering/Events Management).

Each participant answered a questionnaire at the end of the 20-minute session. All participants said they had a pleasant experience; the majority found the tasks easy to complete but pointed out that a small amount of training would be ideal, nonetheless. In comments, the system was appreciated for its simplicity and conciseness but in terms of suggestions of what would heighten their experience, they suggested adding aesthetic details for visual appeal. A slightly challenging aspect some participants mentioned was that they did not know what the items were and in which section/category they could be found. This was despite showing in brackets next to the item in the instruction sheet which section they could be found.

### 3.4 Iteration 3

The third iteration of GUI-ii sessions presented the participants with a more refined version of the design that had been explored in the second iteration. Suggestions that the participants had put forth during iteration one and two had been incorporated in the design. More color, images, and effects to make the design more interesting were added without impacting on the functionality. Five company participants took part in the GUI-ii sessions. The participants were given tasks to complete and yet again encouraged to openly discuss their experiences as they proceeded because they were on call with the designer for the duration of the session. Following clarification received after the sessions in iteration two, the third iteration presented a version of events booking that better resembles the current process flow.

Unlike the previous sessions, where the participant would make an entire booking selecting equipment items from various departments/sections on their own, the third iteration demonstrated how this would be regulated using permissions. This meant a participant would execute only tasks that were relevant to their department and see an event booking process progress as it would normally be carried out with various users giving their input in the same booking by adding their department's requirements for the event booking until it was signed off. This, to a small extent, also gave the participants an audit trail of that booking, making it easier to know who to liaise with, in the case of dependencies, clashes or other changes. A sign off is done once all equipment requests have been received from various departments and the warehouse is ready to release the booked equipment.

All participants were, however, still able to fully process an equipment-hire from start to finish, process a return of equipment and access bookings lists and history screens. At the end of each session, participants answered a questionnaire that was used to establish if we have maintained simplicity and ease-of-use after introducing further changes and whether all expected functionality was present. The participants stated the prototyped system was easy to use. 3/5 mentioned a small amount of training would be recommended. The aesthetics added to the design were effective in making the design more attractive and all participants were satisfied with how well the prototype functionality represented what is required of the system. Keeping a design simple, showing all the relevant information without having to look for it, making it easy to accomplish tasks with a few clicks, allowing the user to see and be able to edit what they have done, and adding a bit of color and images proved to be a pleasing product.

### 3.5 From Prototype to Requirements Specification

A phased approach was decided upon and the first three design iterations would focus on providing a way to book equipment for events and pure (dry) hire, a way to handle returns of previously booked equipment and a way to manage inventory by keeping information on what the company has in the warehouse, what is booked out, when it is due back, what conditions the equipment is in and whether equipment is available for the next booking. This phase also includes a booking and equipment search functionality. The second phase would focus on the ability to provide a comprehensive history, keep audit logs of user activities and provide performance information such as what equipment is most hired, high demand periods, etc.

Ozlab prototypes are dependent on the human experimenter responding to the participants' input to appear as fully working. This poses a challenge when it comes to the requirement's specification. The final prototype was thus captured in a video to communicate the interaction design to the participants and as a requirement specification for the system to be built. The video prototype was done by using a screen recording tool, Camtasia (as it happened, version 4 was used for the recording and version 8 for editing), to record the Ozlab prototype in a browser. The prototype was recorded from the view of a user, and it appeared functioning since the designer responded to the input of another designer acting as participant. A voice-over explaining the functionalities seen in the video was added. The video prototype was divided into four parts: Event booking (17 min), Equipment hire (6 min), Lists and Returns (5 min) and Inventory Management (3 min).

The video prototype was shared on a link to a cloud storage space (Box) for the company to review. A video conference, via Zoom, was then held to discuss whether the video prototype gave a true picture of how they expected the system to work. The video prototype was accepted, a few additional requirements were highlighted, and a go-ahead given for development to commence.

### 3.6 From Video Prototype to System

In order to try to capture what, if any, issues arise when using a video prototype as a demonstration of a requirements specification the designer, who also developed the system was asked to keep record of the development process in a log.

The log had five columns: "Development Area and Date", "Describe the problem, questions, issue", "How did you (try to) solve the problem?", "Classify the issue (if possible)" and "Reported by whom? (note others than you)".

A frequent reoccurring contact between the designer and the participants at the company was advised but proved difficult to withhold. Emails tended to get long waiting times before response. The employees are constantly out of office. Communication was both through formal and informal means. i.e. emails and calls and WhatsApp/text messages.

# 4 Results

The results reported here have been gathered from reports from the video prototyping process, from the log book kept during the development process and from reports from the GUI-ii sessions.

## 4.1 The Video Prototyping Process

The designer noted five issues regarding the video prototyping. All issues pertained to how to produce the videos in an efficient manner.

During the video prototyping phase, the designer also noticed four things had been overlooked by both the participants and the designer during the GUI-ii sessions: the possibility to enter two different types of information was missing in the prototypes, that a certain information (return date) was not added automatically in the prototype but should be, and that the possibility to enter certain details (flag as unavailable) was not present in the prototypes.

During the video prototype meeting, the participants also pointed out a few more issues that had been overlooked. For example, the designer had used "Equipment hire" to describe the process when a client contacts the company to hire equipment without needing them to handle the setup. The company, however, use "Dry hire" to describe this type of process.

## 4.2 Records in the Log

In total the developer noted 44 records in the log. The records form a timeline of the development. For analysis, all records were numbered and the developer's own classifications was used to categorize the records. The developer used varying classifications and thus the records was analyzed to recognize if records under different developer classifications were pertained to the same type of issue. The different types of issues are reported below.

**Developer Know-How.** At least 17 records were logged that were classified as issues related to developer know-how. These records related to issues such as how to programmatically solve a problem, miscalculations resulting in faulty values, web browser compatibility issues, to issues related to moving from local to remote hosting.

The issues raised in these records were general development issues and therefore did not suggest any relation to design, requirements gathering, specification or whether the prototype had any shortfalls.

**Uncertainty of Developer as a Result of Given Data.** Eight of the records have to do with uncertainties of the developer as a result of the inventory data produced and shared by the company. For example, the supplies were inconsistently named, some was categorized while other lacked categories and inventory was described in terms of "most" instead of actual count.

While the designer included real inventory in the prototypes, it was first when the database was to be created that the ambiguities was introduced. Thus, the ambiguities and what each decision would entail regarding the interaction design was not depicted in the prototypes and evaluated by the participants in GUI-ii sessions. The company was instead asked to provide the developer with the "correct" data, and engage in more talks in order to clarify uncertainties.

**Prototype Misrepresentation.** One record had to do with misrepresentation of reality in the prototype. Though this did not mean that in reality what the prototype promised could not be done, it highlighted that some things could, in a prototype, appear much simpler than they are to develop in a real system.

**Missed Elements and Functions in Prototypes.** Five records pertained to uncertainties arising as a result of not having been explored (sufficiently) during the prototyping stage. Prototypes need not represent entire functionality of the product. This means certain elements or functions, despite having been discussed as requirements, were taken for granted that they will be present in the developed system though not explicitly presented in the prototype. In some instances, the functionality is presented in simple terms without intricately going into detail. During development, however, as was in our case, these may present unforeseen complications.

**Other Kinds of Records in the Log.** The remaining records had various classifications relating to development considerations that would not have been possible to ascertain from the design process and resultant prototype. Some of these were database structure, data security considerations, the use of special characters and delays resulting from unavailability of end-users.

### 4.3 Other Findings

These findings are experiences extracted from other reports of the GUI-ii sessions conducted in the three iterations.

**Wizardry During Interactivity.** Using Ozlab in conjunction with calls to the participants allowed us to better understand what the participant wanted without having to spend time coding the desired functionality. Because the designer, acting as a wizard, could see the mouse moves and clicks the participants were making, the participants would use mouse gesture to point out areas of improvement in the design while explaining on the telephone call. Participants were aware of the designer seeing everything they were doing on the computer but not the wizardry that came into play in terms of the responses they would see when they performed specific actions.

Additionally, the wizard could see what actions seemed harder or took longer even when the participant did not verbalize it. We note however, that this could not be measured as several other issues could come into play and these could not be ruled out. In some instances, where these actions are accompanied by verbal utterances that indicate frustration, this can constitute helpful information in assessing usability.

**Iterative Prototype Progression.** The iterations facilitate the evolution from one point of the design to the next, making befitting alterations informed by the outcomes of the previous sessions. Participants appreciated seeing their suggestions implemented and getting to experience them as part of a working prototype. Because participants participated one on one with the designer, each one's voices were heard. This allowed for most commonly mention considerations to be accommodated and the lesser ones to be explored further by adding those into future discussions with other participants. This was often followed by many realizing things they may otherwise have overlooked. Another reason to keep all suggestions in mind arose from the fact that different users have different needs that may need to be catered for in order to fulfil their individual (or even departmental) task specifications.

**Nature of Discussions.** The nature of discussions during and immediately post the GUI-ii sessions was relaxed and open-ended. In thinking out loud, which participants were encouraged to do, no structure can be enforced. Using telephone and Ozlab also had the limitation that neither party on either end of the computer had a visual of the other. Often these two parties did not know each other apart from being informed about participating.

Participants often expressed how they felt about their experiences even before they had to respond to a questionnaire.

**Requirements Change.** Though preliminary discussions and GUI-ii sessions covered several of the system requirements, some requirements were highlighted at a later stage in the design process. These changes were introduced by participants who had participated in all the discussions and sessions and expressed satisfaction.

Examples in our study included that it was only after the second iteration that it was brought to the designer's attention that the system should cater for incremental bookings and only after the third iteration and during the review of the video prototype that it was brought to the designer's attention that an additional group of users should be included that will be responsible for and limited to starting bookings.

**Video Prototype as a Requirements Specification.** Using a video of the prototype as a requirements specification proved beneficial to the designer, developer and participants. Playback made it easy to review scenarios and what was adopted in the specification. It was also convenient for the developer to refer back to and easily understand what the developed solution must fulfil.

We point out that a developed solution may not be a perfect replica of the video prototype but should provide as closely matched a system if requirements have not changed drastically and the prototype itself had been approved as a signed-off specification.

**Commitment of Participating Organization.** We fully appreciate that companies have the highest priority in keeping their business running. The techniques used in this study of a participatory nature and the key prerequisite for stakeholders to actively participate. This goes beyond participating in a GUI-ii session, for example, to includes commitment to the project, timely communication response and participant availability. Committing to project and the developed product may be quick and easier. Commitment to the process requires time and availability.

## 5 Concluding Discussion

Initially, we set out to discuss and demonstrate methods for Distributed Participatory Design (DPD) and the use of interactive prototypes as a requirements specification (or demonstration of such). Before us, [5, 6] used asynchronous methods for participatory design over distance. In our study, we used the synchronous method called GUI-ii, as demonstrated and proposed in [7]. GUI-ii enabled us to ensure the *active participation* of participants as stressed by [14, 15] over distance.

Without moving forward to the development of the system designed in the GUI-ii sessions, it would not be possible to say as much of the efficiency of the sessions. We decided to create a visual representation, a video prototype, to communicate the interaction design and agreed-upon requirements with the participants. The video prototype was furthermore used as a requirements specification during development. This process was captured in a logbook.

**Few Prototype Misrepresentations.** In this section, we will bring up something that we have not seen discussed as much in the literature. We think, however, that the category “Prototype misrepresentation” is interesting, although probably not specific for digital prototypes as misrepresentations of implementable system/reality could be introduced in a paper prototype as well.

The majority of the records in the development log was development related. This is expected as the focus shift from interaction design to how to implement the system programmatically. What was a bit unexpected, however, was the few instances of “Prototype misrepresentation” noted in the log.

Many prototyping tools allow the designer to suggest all sorts of design and interaction paradigm regardless of how difficult or even impossible the suggestions are to implement. This holds for Ozlab as well. One might expect that transcending from prototype to implemented system would induce many problems related to prototype misrepresentation (of implementable system/reality).

Interestingly enough, only one record in the log was related to prototype misrepresentation. It should be noted, however, that the designer is also a knowledgeable programmer, why some pitfalls could have been avoided. In any way, the few instances of prototype misrepresentation and the double roles of the designer at least suggest that engaging the system developers in the prototyping process could be a good idea as emphasized in [7], at least if the developers do not immediately shy away from unconventional solutions.

**Making Requirements Explicit Supported Mutual Learning Between Developer and Participant.** Some functions and wording were overlooked by the designer and the participants in the GUI-ii sessions. One example was the designer’s use of “Equipment Hire” instead of “Dry Hire”. Perhaps this was overlooked by the participants during the first three iterations in an attempt to be polite, and/or the video prototype made it evident that the wrong term was used as the term “Equipment hire” was repeated in the voice over in the video (as well as in the GUI).

By explicitly communicating the systems requirements via interactive prototypes and video prototypes and allowing active participation, the participants could see how the designer had interpreted their expressed needs (as proposed by [10, 11]) and corrected any misinterpretations and mutual learning was supported as emphasized by [15].

The designer also became aware of missed functions and details while reviewing the video prototype. This might be thanks to how focused one must be on the GUI and the interaction design when editing videos and adding voice over explaining everything visible on the screen.

We thus argue video documentation is beneficial for the design and development, as [12] suggested.

Awareness of overlooked functionality and details may mean that additional requirements are added. These requirements may be crucial or require major changes. Indeed, this is welcome as such valuable information may lead to the design most appropriate to the user and since the prototypes were not programmed, the effort to correct mistakes was minimal.

**Informal Discussions Helped in Highlighting Work Process Frustrations.** Using a relaxed/informal discussion and introducing both the study and the role they are expected to play played a positive role in getting the participants to open up and share their thoughts about what the system should entail to properly address their needs. This often allowed them to open up also about their current work process frustrations which provided information about how the system could be designed to alleviate these. Because questionnaires were given verbally in such an open communication setting, ratings were often given and substantiated which enriched the data collected. Building rapport with participants to encourage them to open up and share information is a well-known success factor in interviewing [20] and seems to hold for GUI interaction interviews as well.

**Active Participation in DPD is Difficult Without Internet Connection.** Now, the following issue has not been addressed in this study yet but should be noted nonetheless. The methods for the distributed participatory design used in this study, as well as in [5, 6] rely heavily on the participants' access to an internet connection. Sharing video prototypes, even though made easier by digital means [13], is also difficult without an internet connection. Indeed, the oral discussions during the GUI-ii sessions were held over the telephone, but the exchange of views in the graphical user interface and the wizardry behind the non-programmed prototype would not be possible without a web-based tool such as Ozlab [7, 10, 16]. Without it, the participants' interactions with the prototypes would have to be recorded and then shared with the designer in some other way – thus losing the possibility of not only relating the discussion directly to the interface but also the possibility to swiftly respond to the participants' desires.

From our study, we see that it is possible to conduct participatory design over distance, but that it relies on the use of the internet. Therefore, it might be difficult to ensure active participation as stressed by for example [14, 15] over distance in the most secluded and rural parts of the world.

## 6 Prospects for HCI and Participatory Design

This study does not include the process of introducing the system to the company and its work processes. There is a risk that more overlooked functions and requirements will emerge once the system is in use.

The participants were able to share their views not only in discussions together with their colleagues (after reviewing the video prototype, for example) but also one-on-one during the GUI-ii sessions. By using GUI-ii, the participant was not only allowed to express his opinions regarding the prototype but also to experience design based on his opinions through use. Another way forward that could be interesting to pursue is to explore the possibilities of the participants to be co-designers over distance (as in [7]) and even let them act as wizards.

Although our study includes not only a demonstration of the distributed participatory design process but also the use of video prototypes in development, we have not seen that our account diverges much from the body of literature in the field of participatory design and, the rather limited body of literature in distributed participatory design. To us, this shows that conducting participatory design and development over distance is not something to shy away from. Using a technique such as the Wizard-of-Oz technique in combination with oral discussions contributes to better clarity in understanding what the user requires and geographical distances between participant and designer can be mitigated.

What remains to do is, however, to connect distributed participatory design with the distributed agile development and outsourcing body of literature.

## References

1. Bannon, L., Bardzell, J., Bødker, S.: Introduction: reimagining participatory design-emerging voices. ACM Trans. Comput.-Hum. Interact. **25**, 1–8 (2018)
2. Iivari, N.: Coordinating, contributing, contesting, representing: HCI specialists surviving distributed design. In: De Angeli, A., Bannon, L., Marti, P., Bordin, S. (eds.) Design of Cooperative Systems, pp. 207–223. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-33464-6_13
3. Gumm, D.C., Janneck, M., Finck, M.: Distributed participatory design–a case study. In: Proceedings of the DPD Workshop at NordiCHI (2006)
4. Lazarin, C.A.J., Almeida, L.D.A.: Distributed participatory design web-based groupware: gathering requirements through braindraw. In: Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems, pp. 1–10. ACM, New York (2016)
5. Farshchian, B.A., Divitini, M.: Using email and WWW in a distributed participatory design project. ACM SIGGROUP Bull. **20**, 10–15 (1999)
6. Walsh, G., et al.: DisCo: a co-design online tool for asynchronous distributed child and adult design partners. In: Proceedings of the 11th International Conference on Interaction Design and Children, pp. 11–19. Association for Computing Machinery, Bremen (2012)
7. Pettersson, J.S., Wik, M., Andersson, H.: GUI interaction interviews in the evolving map of design research. In: Paspallis, N., Raspopoulos, M., Barry, C., Lang, M., Linger, H., Schneider, C. (eds.) Advances in Information Systems Development: Methods, Tools and Management, pp. 149–167. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-74817-7_10

8. Kautz, K., Bjerknes, G., Fisher, J., Jensen, T.: Distributed participatory design in crowdsourced information systems development. In: 29th Australasian Conference on Information Systems, pp. 1–12. Australian Chapter of the AIS (2018)
9. Obendorf, H., Janneck, M., Finck, M.: Inter-contextual distributed participatory design. Scand. J. Inf. Syst. **21**, 2 (2009)
10. Molin, L., Pettersson, J.S.: How should interactive media be discussed for successful requirements engineering? In: Burnett, R., Brunström, A., Nilsson, A. (eds.) Perspectives on Multimedia: Communication, Media and Information Technology, pp. 69–96. Wiley, New York (2003)
11. Karras, O., Unger-Windeler, C., Glauer, L., Schneider, K.: Video as a by-product of digital prototyping: capturing the dynamic aspect of interaction. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), pp. 118–124. IEEE, Piscataway (2017)
12. Brill, O., Schneider, K., Knauss, E.: Videos vs. use cases: can videos capture more requirements under time pressure? In: Wieringa, R., Persson, A. (eds.) REFSQ 2010. LNCS, vol. 6182, pp. 30–44. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14192-8_5
13. Bogdan, C., Ertl, D., Falb, J., Green, A., Kaindl, H.: A case study of remote interdisciplinary designing through video prototypes. In: 2012 45th Hawaii International Conference on System Sciences, pp. 504–513. IEEE, Washington (2012)
14. Bødker, S., Grønbæk, K.: Design in action: from prototyping by demonstration to cooperative prototyping. In: Greenbaum, J., Kyng, M. (eds.) Design at Work: Cooperative Design of Computer Systems, pp. 197–218. Lawrence Erlbaum, Hillsdale (1991)
15. Bødker, K., Kensing, F., Simonsen, J.: Participatory IT Design: Designing for Business and Workplace Realities. MIT Press, USA (2004)
16. Pettersson, J.S., Wik, M.: The longevity of general purpose Wizard-of-Oz tools. In: OzCHI 2015: Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction, pp. 422–426. ACM, New York (2015)
17. Kelley, J.F.: An empirical methodology for writing user-friendly natural language computer applications. In: Proceedings of CHI 1983, pp. 193–196. ACM, New York (1983)
18. Larsson, N., Molin, L.: Rapid prototyping of user interfaces in robot surgery—Wizard of Oz in participatory design. In: Nilsson, A.G., Gustas, R., Wojtkowski, W., Wojtkowski, W.G., Wrycza, S., Zupančič, J. (eds.) Advances in Information Systems Development, pp. 361–371. Springer, New York (2005). https://doi.org/10.1007/978-0-387-36402-5_31
19. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies For Effective Human-Computer Interaction. Pearson Education, Bengaluru (2010)
20. King, N., Horrocks, C., Brooks, J.: Interviews in Qualitative Research. Sage, New York (2019)