Appendix G The COVEN Inspection Method

The CVE inspection method described here has been developed during the COVEN project. It is a step by step guide to perform an inspection of a CVE interface. The CVE Inspection method has been built on the traditional 2D Inspection method (Nielsen & Mack, 1994), and most notably, on top of the Inspection method for single-user VEs, which was developed by Sutcliffe and Kaur (1998).

G.1 Introduction

Usability inspection is aimed at finding usability problems in an existing design. Inspections are used at the stage in the usability engineering cycle after the interface design has been generated, and before testing with actual users. Inspecting an application will lead to:

- Overview of usability problems.
- Fixes and other redesign suggestions.
- Prioritise usability design activities.
- Aid in cost/benefit assessments.

In order to perform an inspection of a CVE design, first a short CVE user context analysis is created, identifying the main users and the situation of use. Next, a floor plan is created of the CVE space, marking the navigational paths through the spaces, and any system functions and objects. Subsequently, the expected or ideal navigational path and the ideal order of acting on interactive objects are marked onto the floor plan. Finally for each interactive task along the navigational path a task

analysis is performed, identifying each sub-task in the total task. Each task and sub task is subsequently inspected, using the appropriate interaction cycles. The forms used in this CVE Inspection encourage the inspectors to add design solutions to the design problems they identified, in order to give explicit advice where possible.

The next section (G.2) describes the principles of the CVE inspection process, including the interaction cycles. Section G.3 describes the actual inspection process with short examples from the COVEN project for each stage of the process. Section G.4 describes some conclusions about the inspection method as presented here.

G.2 CVE Inspection Process Explained

An analysis of the task domain is required in order to get a good understanding of goal of the application, the users and the type of tasks that the application is to support. Subsequently a floor plan is created of the CVE space, marking the navigational paths through the VE, and any system functions and objects. Next, the ideal navigational path and the ideal order of acting on interactive objects are marked onto the floor plan. Finally for each interactive task along the navigational path, a task analysis is performed, identifying the actions for each task and sub task needed to reach the user's goal. Each task is subsequently assigned one of the appropriate interaction cycles, and inspected using the questions that belong to each cycle. The inspection process can be summarised as follows:

- **The User Context Analysis**

An analysis of the user context is required in order to get a good understanding of the specific user needs, which one needs to keep in mind whilst doing the inspection.

- **The Virtual Space Analysis**

A floor plan is created of the CVE space, marking the navigational paths through the VE, any system functions, all objects, and any collaboration and interaction spaces.

- **The Task Analysis**

The navigational path, the ideal choices along the navigational path, the interactive objects, the ideal order of acting on interactive objects, and the collaboration positions are marked onto the floor plan.

- **The Interaction Cycle Analysis**

Each element identified in the task analysis can be assigned to the specific complementary interaction cycle.

- **The Inspection**

Each identified element is inspected using the appropriate interaction cycle and their associated questions about usability and task flow.

- **The Inspection Report**

For each problem that is found, a note is made in the inspection report, and a reference number is assigned. Each task identified in the application, is assigned to one of six task interaction cycles in order to guide the inspection. These interaction cycles can be summarised as follows:

1) System Initiative Cycle, where the user has to deal with the system temporarily taking control over the cause of events in the CVE, either because the user has caused this to happen or because the system has instructions to do so.

2) Normal task action Cycle 2D, where a user is interacting with 2D information in the environment in order to achieve a certain goal, such as text menu's or 2D pop-up displays.

3) Normal task action Cycle 3D, where a user is interacting with a 3D object in the environment in order to achieve a certain goal.

4) Goal Directed Exploration Cycle, where a user is searching for something known to be in the environment in order to achieve a certain goal.

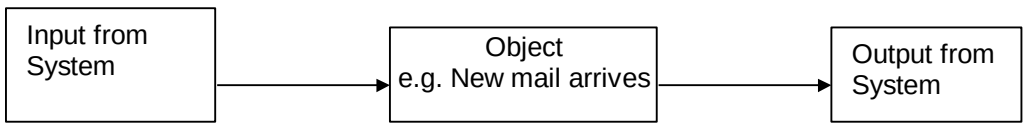
5) Exploratory Browsing Cycle, where a user is navigating through the CVE in order to achieve an understanding of the world layout, or world order.

6) Collaboration Cycle, where the user is interacting with other users in the CVE, either to collaborate on a certain task, or to socialize.

The interaction cycles have questions assigned to them, which represent the generic usability requirements for each type of task. The inspectors answer these questions during the inspection, and for each identified usability problem that is found, a severity rating is assigned by the inspectors (further explained below). These ratings signify the effect each usability problem will have on the overall usability of the application. Each problem found is also assigned a unique number, in order to be able to refer to each individual problem in the inspection report when necessary. The cycles, their definitions, and the associated questions are presented below.

1) System Initiative Cycle

System Initiative Cycle	The user has to deal with the system temporarily taking control over the cause of events in the CVE, either because the user has caused this to happen or because the system has instructions to do so.

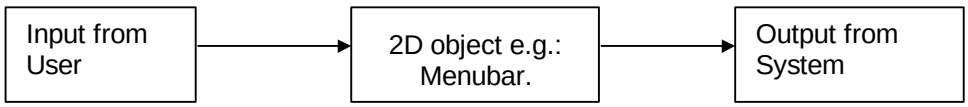


Task: *Function Description*

Generic Task Scenario for System Initiative Cycle	**Potential Problem and Design Solution**	**Severity Rating**	**Ref. Numb**
i: Is it clear to the user that the system has taken control?			
ii: Can the user resume control at any point and is the appropriate action clear?			
iii: Are the effects of system actions visible and recognizable?			
iv: Are the system actions interpretable?			
v: Is the end of the system action clear?			
vi: Is there an obvious next action to perform for the user, now that this task has ended?			

2) Normal task action Cycle 2D

Normal task action Cycle 2D	The user is interacting with 2D information in the environment in order to achieve a certain goal, such as text menu's or 2D pop-up displays.



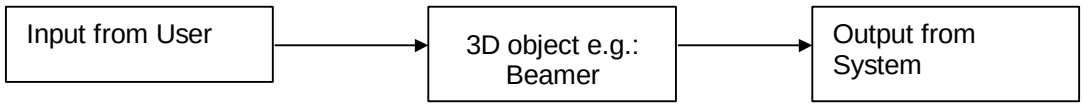
Task: *Function Description*

Generic Task Scenario for a Normal Task Action 2D Cycle	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Will the users be trying to produce whatever effect the action has?			
Ii: Will users be able to notice that the correct action is available?			
Iii: Once a user finds the correct action at the interface, will they know that it is the			

right one for the effects they are trying to produce?			
Iv: After the action is taken, will users understand the feedback they get?			
vi: Is there an obvious next action to perform for the user, now that this task has ended?			

3) Normal task action Cycle 3D

Normal task action Cycle 3D	The user is interacting with a 3D object in the environment in order to achieve a certain goal.



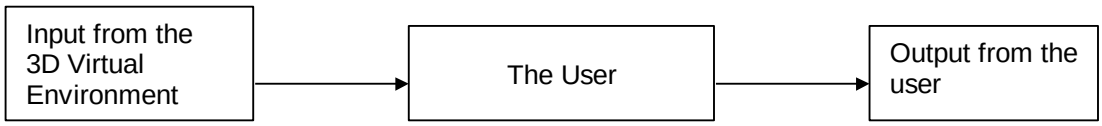
Task: *Function Description*

Generic Task Scenario for a Normal Task Action Cycle	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Can the user form or remember the task goal?			
ii: Can the user specify an intention of what to do?			
Iii: Are the objects or part of the environment necessary to carry out the task-action (users new intentions) visible?			
iv: Can the objects necessary for the task action be located?			
v: Can the users approach and orient themselves to the objects so the necessary action can be carried out?			
vi: Can the user decide what action to take and how?			
vii: Can the user carry out the manipulation or action easily?			
viii: Is the consequence of the users action visible?			
ix: Can the user interpret the			

change?			
x: Is it made clear to the user what the next correct/needed action could be?			
Xi: Is there an obvious next action to perform for the user, now that this task has ended?			

4) Goal Directed Exploration Cycle

Goal Directed Exploration Cycle	The user is searching for something known to be in the environment in order to achieve a certain goal.



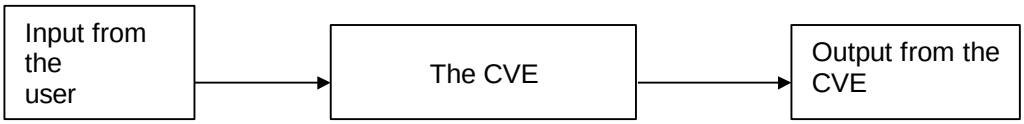
Task: *Function Description*

Generic Task Scenario for Goal Directed Exploration	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Does the user know where to start looking?			
ii: Can the user determine a pathway towards the search target?			
iii: Can the user execute movement and navigation actions?			
iv: Can the user recognize the search target?			
v: Can the user approach and orient themselves to the objects so the necessary action can be carried out?			
vi: Can the user decide what action to take and how?			
vii: Can the user carry out the manipulation or action easily?			
viii: Is the consequence of the users' action visible?			

ix: Can the user interpret the change?			
x: Is it made clear to the user what the next correct/needed action could be?			

5) Exploratory Browsing Cycle

Exploratory Browsing Cycle	The user is navigating through the CVE in order to achieve an understanding of the world layout, or world order.



Task: *Function Description*

Generic Task Scenario for Exploratory Browsing	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Can the user determine a pathway for movement?			
ii: Can the user execute movement and navigation actions?			
iii: Can the user recognize objects in the environment?			
iv: Can the user interpret identity, role and behaviors of objects?			
v: Can the user remember important objects or locations?			
vi: Can the user form a mental map of the explored environment?			

6) Collaboration Cycle

Collaboration Cycle	The user is interacting with other users in the CVE, either to collaborate on a certain task, or to socialize.

Task: *Function Description*

Generic Task Scenario for Collaboration	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Can the user locate the other user(s)?			
ii: Can the user recognize the identity of the other user(s), tell the other users apart?			
iii: Are the communication channels between the users effective?			
iv: Are the actions of the other user(s) visible and recognizable?			
v: Can the user act on a shared object while keeping the other user(s) in view?			
vi: Can the user easily switch views between the shared object, other locations/object of interest and the other user(s) (sweep from one to the other)?			
vii: Can the user get an overview of the total shared space and all other users in it?			
viii: Can the user tell when there are interruptions in the attention of the other user(s) to the CVE?			

Severity Ratings

During the Inspection severity ratings between 0 and 4 are given for each problem found. These ratings can be regarded as recommendations in terms of the urgency for redesign and development, and they can be used for guidance when developing a plan of action. The severity of a usability problem is a combination of five factors:

- The frequency with which the problem occurs: Is it common or rare?
- The impact of the problem if it occurs: Will it be easy or difficult for the users to overcome?
- The persistence of the problem: Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?
- The design concept: Does this design concept radically decide and/or exclude user options?
- If a metaphor is used: Is the metaphor appropriate and consistent?

The following rating scale is used to define the severity of usability problems:

0. I don't agree that this is a usability problem at all
1. Cosmetic problem only: need not be fixed unless extra time is available on project
2. Minor usability problem: fixing this should be given low priority
3. Major usability problem: important to fix, so should be given high priority
4. Usability catastrophe: imperative to fix this before users test the system.

G.3 The inspection Process

The successive steps which have to be taken to perform the inspection are described below. Examples are given by presenting small parts of the COVEN "Usability Inspection of the London Demonstrator" by Steed (Del, 3.6).

G.3.1 User Context Analysis

The User Context Analysis is a necessary element of the Inspection, because it helps to clarify the specific user needs and requirements, which the Inspector has to keep in mind while inspection the application. Clear descriptions of the user goals, anticipated user groups, a description of a typical user and user groups, and a user task description are derived by filling in the forms below. To illustrate the type of descriptions this will supply, each form contains an example.

User's initial Goals

User Goal	**Give a short description of the user goal, which the application is to support.**
London Demonstrator	Potential conference attendees wish to plan their visit to the conference center. This involves reviewing accommodation choices, looking at transportation facilities, examining conference facilities, planning routes, rehearsing presentations. The potential attendee expects to be able to book meetings with the local conference organiser and/or plan to rendezvous with other potential attendees.

Anticipated User Groups

User Groups	**Give a short description of any different types of user**

	groups, which are expected to use the application.
London Demonstrator	Conference attendees. Local conference organisers.

User Description

User Description	**Give a short description of the users and their expected background.**
London Demonstrator	User will be familiar with 3D graphics systems, and be, at least, an in-frequent visitor to on-line virtual environment services. User will be expected to be familiar with 3D navigation and interaction through their experience with similar systems.

User Group Description

User Group Description	**Give a short description of the user groups.**
London Demonstrator	Several classes of user can be determined: ♦ Planner - creates and enters data to application. On-line editing of persistent world ♦ Organiser - knowledgeable about facilities. Can act as a guide to the virtual world and assist planners in review and verification of on-line data ♦ Visitor - has little or no knowledge of event or data sources. Expects to be able to access an organiser or at least leave messages for them.

User Task Description

User Task Description	**Give a short description of the task the users are to perform.**

London Demonstrator	The user's task will be to enter the application either individually, expecting to chance upon others or to review known data sources, or as a member of a group responding to a planned meeting call. It is expected that group visits will be planned in advance using the individual mode of access using built-in meeting planning tools.

G.3.2 The Virtual Space Analysis

To perform the virtual space analysis a floor plan is created of the CVE space, marking the navigational paths through the VE, any system functions, all objects, and any collaboration and interaction spaces. This floor plan helps to create a quick overview of the major tasks in the application, which will be used in the inspection. However, analysing the lay-out of the CVE has several functions, all of which check the design of the application for flow of interaction:

- It creates an insight in the most likely path users will take.
- It highlights which path is the desired path.
- It highlights in which order users are expected to interact with each object.
- It highlights which navigational cues are available or missing in the environment.

When creating floor plans one can use the following questions as a guideline:

I. Where does the user enter the space?

II. What information or objects are visible?

III. Which objects should the user be exploring?

IV. What is the next logical action from the users' point of view?

On the floor plan one should mark the desired path the user should take, the interactive objects, the collaboration or interaction places and any system functions (see figure G.1).

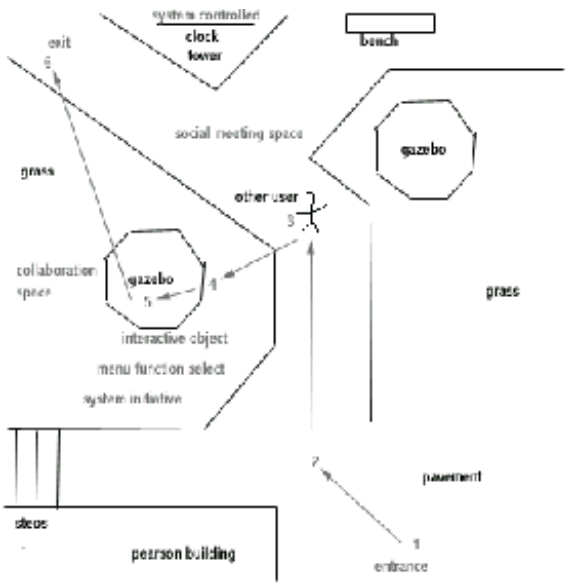


Figure G.1: Floor plan of planned CVE space.

Figure G.1 is an example floor plan for the London Demonstrator. In this example the desired path consists of six 'steps', from entrance to exit. The interactive object in this case is a gazebo used for private communication. The system function in this example is a clock showing the actual real-world time. The social function of the space is indicated by a bench and the gazebo's, which are placed there as metaphors for social meeting. Other objects are placed for decoration, which is an important element for

the creation of a cognitively immersive experience. The functions and objects are further analysed in the Task Analysis section.

G.3.3 Task Analysis

Task analysis is a detailed breakdown of what the user is required to do in terms of actions and cognitive processes to achieve a task goal. The process is divided into two phases. First the major tasks within the application are listed. Next these major tasks are broken down into sub-tasks, until one has a good overview of the task flow involved. The floor plan created in the Virtual Space Analysis is used as a guideline for task order and subsequent task decomposition. The following example illustrates the task analysis process.

Task Analysis	**Tasks, system functions, and 3D objects should be identified from the scenario description, they may be listed as general activities, and more specific activities subsumed under general ones.**
London Demonstrator	The functionality of the London demonstrator and the underlying platform and standard user-interface are broken down as follows: ♦ Communication ♦ Way-finding ♦ Data visualisation ♦ Meeting rooms **Communication:** There are several styles of communication within the application: ♦ Avatar Representation and Shared Interaction

	♦ "Simple" interface ♦ Spatialised speech ♦ Text communication **Avatar Representation and Shared Interaction:** Each user has an avatar. This avatar can be customised and it is usual for the general design of the avatar to be uniform for a particular site (perhaps with the site's logo), but for each individual to customise the accessories. Each avatar also bears a name tag. When the user speaks the avatar produces "audio waves" - a visual indication that they are speaking. It is also possible to correlate perceived direction of audio with actual position of avatar, leading to easy avatar identification. When the user interacts with an object, a ray appears joining the participant's eye to their selection point. This is useful for identifying which person is manipulating an object, since the direction gaze of their avatar is usually ambiguous. Speech and interaction activities are also reflected in the "Simple" interface. **"Simple" Interface:** The simple interface is a panel listing all the current users in the application. Each line of the panel has an icon representing that user's current actions (Moving/Talking/Interacting). Lack of any action for a long period generates an Asleep ("Zzz") icon next to the user's name. The participants name is active. Upon selecting the name, the user is presented with a menu of three actions: Talk To, Go To, Look At.

	These activate the text tool, teleport the user to in-front of the other and turn the user to face the other, respectively. *Etc.*

G.3.4 Interaction Cycle Analysis

The tasks, system functions and 3D objects identified in the Task Analysis (section G.3.3), should now be linked to one or more of the 6 interaction cycles as described in section G.2.

Class of Function	Sub-Function	Description	Interaction Cycle
Avatar Representation	Identify others	Look around	**5. Explore**
	Identify interactor	Look around	**5. Explore**
Access Simple interface	Select name of participant	Observe activities of other participants	**2. Cycle 2D**
	Select relevant action	Talk to, go to, or look at other participant	**2. Cycle 2D**
Speech	Identify speaker	Look around	**5. Explore**
		Simple interface icon	**2. Cycle 2D**
	Interact	Interact with other participant(s)	**6. Collaborate**
	Enter private group (explicit)	Enter gazebo	**4. Goal**
		Manipulate walls	**3. Cycle 3D**
	Exit private group (explicit)	Leave gazebo	**4. Goal**
	Enter private group (implicit)	Enter room	**1. System**
	Exit private group (implicit)	Leave room	**1.System**

Trouble Shooting

Allocating a certain actual task existing in a CVE, to one of the 6 interaction cycles is sometimes difficult. It has to be noted that choices of allocation of task cycles to application tasks are informed by a clear understanding of the task cycles and their

respective definitions. In testing the method during development in the COVEN project, several problems and solutions have been identified (see Table G.1):

Problem	**Cause**	**Solution**
None of the task cycles seems to fit.	The task has not been broken down into sufficient dept.	Go back to the task-analysis step of the Inspection method and break the task down into subtasks until you reach the level of atomic sub-tasks. Than re-assign the task cycles.
More than one task cycle seems to fit.	The task has not been broken down into sufficient dept.	Go back to the task-analysis step of the Inspection method and break the task down into subtasks until you reach the level of atomic sub-tasks. Than re-assign the task cycles.
More than 1 inspector is inspecting the application jointly and we can not seem to agree on which task cycle seems best.	The task has not been sufficiently clarified.	Engage in a discussion with all relevant team-members to clarify the task. Than re-assign the task cycles.
No matter how long we discuss disagreement over a task cycle allocation, we can not seem to find an agreement.	Some tasks may be interpretable as more than one task cycle.	Apply all task-cycles that seem relevant and include all usability problems in the final report.
Only a few task cycles seem to apply in my Inspection.	Some applications make limited use of all task-cycles.	Make sure you understand and remember the definitions of all 6 task cycles.
I feel I haven't got time to inspect all tasks in the application on the level of detail that the method seems to demand.	You may not have been allocated sufficient time to perform an inspection, or you may have chosen the wrong method for this point in the development process in which you are involved.	Prioritize a major task you would like to be as user-friendly as possible and inspect that first. Record how much time you needed, and estimate the time needed for an inspection of the rest of the application. Document your results to show the effect of the inspection method and the time-schedule

		involved.

Table G.1: Problems and solutions for allocating interaction cycles to tasks in the CVE.

G.3.5 Performing the Inspection

For each task, system function and 3D object identified in the task analysis, the Inspector steps through the task hierarchy, asking the questions belonging to the interaction cycles. For each question the answer is noted down in the column 'Potential Problem and Design Solutions'. The Inspectors are encouraged to write down any design solutions they can think of in this same column. The next column 'Severity Rating', is used to note a number between 0 and 4 to indicate the severity of the problem found. The final column is used to give a unique reference number to the problems, so that in the subsequent report it will be easier to refer to specific problems. Now follow some examples for each cycle.

System Initiative Cycle

Task: *Communicate: Speech: Enter Private Group (implicit)*

Generic Task Scenario for System Initiative	**Potential Problem and Design Solution**	**Severity Rating**	**Ref. Numb**
i: Is it clear to the user that the system has taken control?	No. But not a problem	0	
ii: Can the user resume control at any point and is the appropriate action clear?	NA?	0	
iii: Are the effects of system actions visible and recognizable?	No. Not obvious the group is private audio	3	1
iv: Are the system actions interpretable?	NA?	0	
v: Is the end of the system action clear?	NA?	0	

Task: *Communicate: Speech: Leave Private Group (implicit)*

Generic Task Scenario for System Initiative	**Potential Problem and Design Solution**	**Severity Rating**	**Ref. Numb**

i: Is it clear to the user that the system has taken control?	No. But not a problem	0	
ii: Can the user resume control at any point and is the appropriate action clear?	NA?	0	
iii: Are the effects of system actions visible and recognizable?	No. Not obvious the group is now public audio	3	2
iv: Are the system actions interpretable?	NA?	0	
v: Is the end of the system action clear?	NA?	0	

Normal task action Cycle 2D

Task: *Communicate: Speech: identify Speaker (Simple interface icon)*

Generic Task Scenario for a Normal Task Action 2D	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Will the users be trying to produce whatever effect the action has?	Yes	0	
Ii: Will users be able to notice that the correct action is available?	Yes	0	
Iii: Once a user finds the correct action at the interface, will they know that it is the right one for the effects they are trying to produce?	Yes	0	
Iv: After the action is taken, will users understand the feedback they get?	Yes - the icons are obvious though they lag behind action in the virtual world significantly	1	4

Task: *Communicate: Text: Receive Message (Text dialogue dismiss)*

Generic Task Scenario for a Normal Task Action 2D	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Will the users be trying to produce whatever effect the action has?	Yes	0	
Ii: Will users be able to notice that the correct action is available?	Yes	0	
Iii: Once a user finds the correct action at the interface, will they know that it is the right one for the effects they are trying to produce?	Yes	0	
Iv: After the action is taken, will users understand the feedback they get?	Yes	0	

Normal task action Cycle 3D

Task: *Communicate: Speech: Enter Private Group (explicit): Close Walls*

Generic Task Scenario for a Normal Task Action	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**

i: Can the user form or remember the task goal?	Probably they may or may not know the gazebo is the place to go, and then it is not obvious how to close the blinds, though the action is simple	2	41
ii: Can the user specify an intention of what to do?	Yes, this follows from their understanding the action, (??)	0	
Iii: Are the objects or part of the environment necessary to carry out the task-action (users new intentions) visible?	Yes	0	
iv: Can the objects necessary for the task action be located?	Yes	0	
v: Can the users approach and orient themselves to the objects so the necessary action can be carried out?	Yes	0	
vi: Can the user decide what action to take and how?	Yes	0	
vii: Can the user carry out the manipulation or action easily?	Yes	0	
viii: Is the consequence of the users action visible?	Yes	0	
ix: Can the user interpret the change?	Yes	0	
x: Is it made clear to the user what the next correct/needed action could be?	Yes	0	

Task: *Communicate: Text: Review Message Board: Review Board Message*

Generic Task Scenario for a Normal Task Action	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Can the user form or remember the task goal?	Yes	0	
ii: Can the user specify an intention of what to do?	Yes	0	
Iii: Are the objects or part of the environment necessary to carry out the task-action (users new intentions) visible?	Yes	0	
iv: Can the objects necessary for the task action be located?	Yes	0	
v: Can the users approach and orient themselves to the objects so the necessary action can be carried out?	Yes	0	
vi: Can the user decide what action to take and how?	Yes	0	
vii: Can the user carry out the manipulation or action easily?	Yes	0	
viii: Is the consequence of the users action visible?	Yes	0	
ix: Can the user interpret the change?	Yes	0	
x: Is it made clear to the user what the next correct/needed action could be?	Yes	0	

Goal Directed Exploration Cycle

Task: *Communicate: Speech: Enter Private Group (Explicit): Enter Gazebo*

Generic Task Scenario for Goal Directed Exploration	Potential Problems and Design Solutions	Severity Rating	Ref. Numb
i: Does the user know where to start looking?	If they know the talking booths are in UCL (i.e. no)	1	86
ii: Can the user determine a pathway towards the search target?	Yes	0	
iii: Can the user execute movement and navigation actions?	Yes	0	
iv: Can the user recognize the search target?	Probably	1	87
v: Can the user approach and orient themselves to the objects so the necessary action can be carried out?	Yes	0	
vi: Can the user decide what action to take and how?	Yes	0	
vii: Can the user carry out the manipulation or action easily?	Yes	0	
viii: Is the consequence of the users action visible?	Yes	0	
ix: Can the user interpret the change?	Yes	0	
x: Is it made clear to the user what the next correct/needed action could be?	Yes	0	

Task: *Communicate: Speech: Exit Private Group (Explicit)*

Generic Task Scenario for Goal Directed Exploration	Potential Problems and Design Solutions	Severity Rating	Ref. Numb
i: Does the user know where to start looking?	Yes	0	
ii: Can the user determine a pathway towards the search target?	Yes	0	
iii: Can the user execute movement and navigation actions?	Yes	0	
iv: Can the user recognize the search target?	Yes	0	
v: Can the user approach and orient themselves to the objects so the necessary action can be carried out?	Yes	0	
vi: Can the user decide what action to take and how?	Yes	0	
vii: Can the user carry out the manipulation or action easily?	Yes	0	
viii: Is the consequence of the users action visible?	Probably, the talking booth is behind them	1	88
ix: Can the user interpret the change?	Yes	0	
x: Is it made clear to the user what the next correct/needed action could be?	Yes	0	

Task: *Way-Finding: Overview Map: Navigate to destination (in-direct)*

Generic Task Scenario for Goal Directed Exploration	Potential Problems and Design Solutions	Severity Rating	Ref. Numb
i: Does the user know where to start looking?	Maybe, needs to look around	0	
ii: Can the user determine a pathway towards the search target?	Not always, but this is part of the task	0	
iii: Can the user execute movement and navigation actions?	Yes	0	
iv: Can the user recognize the search target?	Yes	0	
v: Can the user approach and orient themselves to the objects so the necessary action can be carried out?	Yes	0	
vi: Can the user decide what action to take and how?	Yes	0	
vii: Can the user carry out the manipulation or action easily?	Yes	0	
viii: Is the consequence of the users action visible?	Yes	0	
ix: Can the user interpret the change?	Yes	0	

Exploratory Browsing Cycle

Task: *Communicate: Speech: Identify Speaker: Look Around*

Generic Task Scenario for Exploratory Browsing	Potential Problems and Design Solutions	Severity Rating	Ref. Numb
i: The user determines a pathway for movement.	Yes	0	
ii: The user executes movement and navigation actions.	Yes	0	
iii: The user recognizes objects in the environment.	Yes	0	
iv: The user interprets identity, role and behaviors of objects.	Audio waves are not totally obvious as a speaking icon	1	89
v: The user remembers important objects or locations.	Yes	0	
vi: The user forms a mental map of the explored environment	Yes	0	

Task: *Data Visualisation: Access Data: Manipulate Visualisation: View values (free movement)*

Generic Task Scenario for Exploratory Browsing	Potential Problems and Design Solutions	Severity Rating	Ref. Numb
i: The user determines a pathway for movement.	Yes	0	
ii: The user executes movement and navigation actions.	Maybe not, the scale of the viztool is vast. Automatically scale world	2	90
iii: The user recognizes objects in the environment.	Yes	0	

iv: The user interprets identity, role and behaviors of objects.	Yes	0	
v: The user remembers important objects or locations.	Yes	0	
vi: The user forms a mental map of the explored environment	Yes	0	

Collaboration Cycle

Task: *Navigation: Move Through VE as a Group: Decide Destination*

Generic Task Scenario for Collaboration	**Potential Problems and Design Solutions**	**Severity Rating**	**Ref. Numb**
i: Can the user locate the other user(s)?	Yes.	0	
ii: Can the user recognize the identity of the other user(s), tell the other users apart?	The name tags don't always face the user.	1	47
iii: Is there an indication of mutual awareness?	No, text and audio messages are not confirmed. Sound can be very bad and no indication of the loss is given.	3	48
iv: Are the actions of the other user(s) visible and recognisable?	No. Selection and picking are not indicated explicitly e.g. if an object is selected it is not possible to tell who selected it if several people are looking at that object. No pointing gestures except for looking at something	3	49
v: Can the user act on a shared object while keeping the other user(s) in view?	Only in certain situations.	0	
vi: Can the user easily switch views between the shared object, other locations/object of interest and the other user(s) (sweep from one to the other)?	No since the navigation sweep must be specified manually. Maybe a viewpoint "hotspot" feature could be added, with spots corresponding to nearby users and active objects.	2	50
vii: Can the user get an overview of the total shared space and all other users in it?	No, no overview map provided,	2	51
viii: Can the user tell when there are interruptions in the attention of the other user(s) to the CVE?	No, avatar posture can convey the wrong impression.	2	52

G.3.6 The Inspection Report

The inspection report consists of all tasks as scored and analysed for the usability problems. Generally, the type of usability problems found can be summarised to emphasize aspects of concern to the designers, the usability engineers, the managers of the project, and the scientists in the project, respectively. Depending on the type of summary the inspection report can serve several different functions. Firstly, there is

the actual feedback report on the usability problems found, which should go to the persons who will redesign the application. Secondly, a feedback report can be made of reoccurring usability problems, which should go to the persons who developed the application so that they may learn about the weak points in their design techniques. Thirdly, a feedback report on the iterative improvement of the successive versions of the application, which should go to the managers of the project so that they may learn of the cost/benefit involved in the respective iterations of redesign. Finally, there is a summative report on the recurrence of the type of usability problems found, which point to research issues on areas where the best design solutions are simply not known yet due to novelty of the technology under inspection. Examples of the different feedback reports are presented in more detail below.

Feedback Report to the Designers

The first use of the inspection report is to prioritise all the usability problems found during the inspection in a table, in order to inform and direct the redesign of the application. This table is of most direct concern for the designers of the application. The usability problems are prioritised based on the severity rates given to the problems found. This prioritisation should help the design team to structure the redesign tasks and to estimate time needed for the redesign effort.

The report should list a table, such as the one below (table G.2), listing the cycles, the associated functions of the application, the type of problem found (that is: which particular step of the interaction cycle the problem belongs to) with a short summary. Column 4 states shortly whether a design solution has been given by the Inspector or

whether the entry is merely flagging a problem. Finally, the severity rating and the problem reference numbers are stated.

Cycle	Function	Type of Problem	Design Solution	Severity	Reference Number
System Initiative Cycle	Receive Message	ii) resume control	Yes	3	1
Normal Task Action 2D Cycle	Identify Receivers	i) trying to produce the action	Yes	2	2
		ii) correct action noticeable	Yes	4	3
		iv) feedback	Yes	2	4
	Send Message: Composition	ii) correct action noticeable	No	1	5
		iii) correct action identifiable	Yes	1	6
		iv) feedback	Yes	4	7

Table G.2: Designers feedback report.

This summary table of the complete inspection report is useful to gain a rapid overview of the redesign issues involved. It will help the design team to discuss and prioritise the redesign tasks, and allows a breakdown of the work ahead. Some redesign tasks are major redesign of visual aspects, while others may be fine-tuning of the system functionality, and anything in between.

Feedback Report to the Usability Engineers

The usability engineers may use the same report to identify area's were the designers are going to need specific guidance on redesign of the application. During the

Inspection, inspectors are encouraged to provide redesign advice or solutions, but it is not always possible to provide this immediately. For instance, the fourth column of the table below (Table G.3), indentifies one usability problem without a design solution; an indication of a potential difficult and/or time consuming redesign task (printed in bold face).

Cycle	**Function**	**Type of Problem**	**Design Solution**	**Severity**	**Reference Number**
System Initiative Cycle	Receive Message	ii) resume control	Yes	3	1
Normal Task Action 2D Cycle	Identify Receivers	i) trying to produce the action	Yes	2	2
		ii) correct action noticeable	Yes	4	3
		iv) feedback	Yes	2	4
	Send Message: Composition	ii) correct action noticeable	**No**	1	5
		iii) correct action identifiable	Yes	1	6
		iv) feedback	Yes	4	7

Table G.3: Design solutions for usability problems tracked in column four.

Additionally, usability engineers can use the third column ‘Type of Problem’, to identify recurring weak area’s of the design. For instance, if there is a high incidence, throughout the inspection report of usability problems with ‘noticing the correct action’; this might mean a total rethink of techniques used to draw users attention to interactive elements of the application is needed.

Feedback Report to the Managers of the Project

A useful summary table for the managers of the project would be one similar to the example below. The columns labelled 'Severity 4, Severity 3, Severity, 2, Severity 1' have a score for how many of problems with each severity were found. By adding each successive Inspection summary to the table, one gains a rapid overview of the rate of improvement of the applications, and the areas in which improvements have taken place. Severity 4 problems generally indicate considerable reprogramming tasks, which take time and are likely to introduce new problems on lower severity levels in a future inspection. Severity 1 problems are generally cosmetic problems, these problems although not serious, could still have an impact on selling the product.

Cycle Function	Severity 4	Severity 3	Severity 2	Severity 1	Totals
Inspection 1 'Initial Demonstrator'	10	1	5	10	26
Inspection 2 'Online Demonstrator'	7	3	5	3	18
Inspection 3 'Final Demonstrator'	3	2	0	2	7

Report to the Researchers in the Project

By analysing the type of usability problems found during an inspection, analysing consecutive inspection reports produced during the development cycle of an application, and by comparing the usability problems found in the inspection of one application with inspection results from other applications, patterns in the type of usability problems that present open issues in usability design, can be found. The COVEN inspections showed that generally the inspection uncovered usability problems in three categories:

- System problems including lack of functionality, performance and display quality.
- Interface problems that concern the acts of navigating, and picking of objects.
- Application specific problems concerning the actual actions and meaning of objects within the environment.

For each of these categories new interaction paradigms are being created for CVEs by researchers and developers of this field. It is however, by no means clear yet which of these interaction solutions are the best ones.

G.4 Conclusions

This inspection method for CVE is different from inspection methods for 2D applications in that it uses:

- A virtual space analysis, to capture the freedom of choice of interaction in a typical CVE space.
- Interaction cycles, to capture the wide variety of tasks and the intertwining of tasks, typical to CVEs due to 'interruptions' of other CVE participants and the CVE system

This inspection method for CVEs is different from the inspection method for single user VEs in that it uses:

- A collaboration task cycle, to enquire into collaborative tasks.

- A strong emphasis on task flow between sub task in the total CVE interaction task; because during the COVEN project it became clear that it is not easily obvious what a CVE participant should or could do next inside the CVE.

The method as described here has been used during the COVEN project and enabled the usability engineers of the project to identify a large number of usability problems, together with detailed usability redesign solutions (Del. 3.6). Other researchers have used this method on other CVEs than the COVEN platform and found similar large amounts of usability problems, with detailed suggestions for redesign directly derived from the interaction cycles. Although performing an inspection is time consuming, it can be performed by one person alone, if nessacary, and it will when applied in a systematic but flexible manner, generate an overview of the occurance, the type and severity of usability problems for each step of the user's task.