# SYSTEMATIC USABILITY INSPECTION APPROACH FOR SONIFICATION APPLICATIONS

*Ag. Asri Ag. Ibrahim[1,2] and Andy Hunt[2]*

[1]Universiti Malaysia Sabah
Lock Bag No.2073, 88999 Kota Kinabalu,
Sabah, Malaysia
awgasri@ums.edu.my

[2]Electronics Dept, University of York
Heslington, YO10 5DD
York, United Kingdom
{aaai500, adh2}@york.ac.uk

## ABSTRACT

In previously reported research, most sonification designers have needed to develop at least a working prototype for user testing. The results are interpreted and analysed to look for possible problems and solutions to further improve the design. This paper introduces a new systematic usability inspection approach for the design of sonification applications design *before* they go to the initial development phase. This process gives an alternative for designers to evaluate their design, detect possible problems and improve the design before they start developing it. It uses two of our models - the Sonification Application (SA) model and the User Interpretation Construction (UIC) model. In this paper we discuss the steps of this process, which include preparing inspection materials, implementing inspection and managing the results.

[Keywords: Usability, Inspection, Applications, Sonification]

## 1. INTRODUCTION

Sonification is the representation of data using mainly non-speech sound for the purpose of communication and interpretation. In this paper we refer to the specific process for transforming the input data into sound as the sonification *technique*. There are many techniques currently available in data sonification and these are often categorised as parameter-mapping [1], model-based sonification [2], audification [3], auralisation [4] and so forth. These techniques are normally guided by the type of data to be presented and the required user tasks that the sonification can support. For example, the auralisation of programming language syntax for program debugging [4], the mapping of multi channel and time series data with different acoustic parameters for data exploration [5], the mapping of a real time stock market data with acoustic parameters for financial trade monitoring [6], and the audification of seismological wave data in earthquake prediction research [3] etc. We refer to each use of a sonification technique in a specific domain, data and task as the sonification *application*.

Until now the designer of a sonification application has needed to develop at least a working prototype and a user testing experiment to evaluate their design. Since User Testing is typically carried out at the stage when at least a working prototype has already been developed, it is quite costly and time consuming. This is especially true if the project involves a very tight schedule and deadlines. It will probably end up with a higher cost and a longer overall development time, particularly if it requires major changes. Much of this cost could be avoided if the major problems were detected in the early stages of design.

Because of the above problems, we believe that the field of sonification requires an alternative, not to replace but at least to enhance the evaluation techniques in order to predict anomalies or problems *before* the expensive development phase. Usability inspection can be such an alternative for evaluating sonification applications because it can be done towards the start of the development process, and without involving end users.

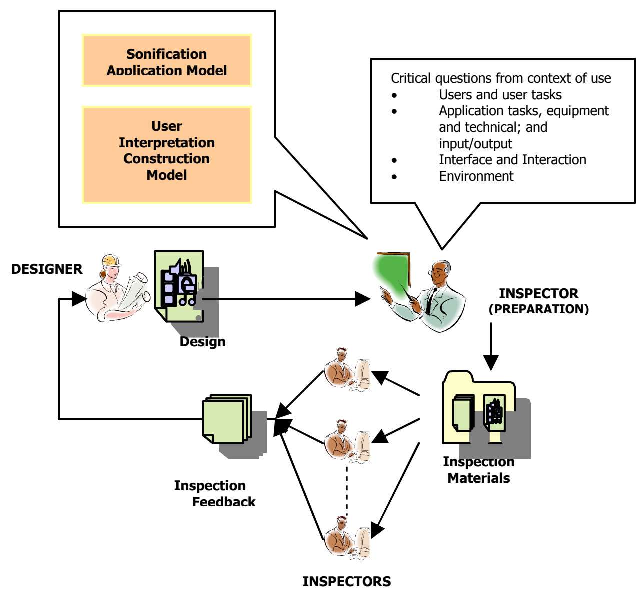


Figure 1: *Overview of Usability Inspection Approach for Sonification Applications*

## 2. USABILITY INSPECTION

Usability inspection is a common name for a set of methods based on having human ‘evaluators' inspect or examine usability-related aspects of a user interface [8]. It is an expert-based evaluation, which is carried out by human experts, and is normally implemented at the design stage before it goes to the implementation or development stage. It requires fewer participants (typically usability experts) than controlled end-user experiments. Examples of existing inspection techniques are Cognitive Walkthrough, Consistency Inspection, Pluralistic

Walkthrough, Standards Inspection, Heuristic Evaluation and Formal Usability Inspection [8].

Figure 1 shows generally how our proposed inspection techniques work. The person who heads the inspection process is called 'chief inspector'. The chief inspector could in theory also be the designer himself, but an extra level of interaction and independence is gained by having a separate person do this task. The chief inspector uses the '**design**' to prepare the Inspection Materials. This is a package that contains descriptions of the application to be evaluated; steps and instructions for inspection; forms to write the encountered problems and so forth. Different inspection methods require different inspection materials. They are distinctive from each other in various aspects such as the purpose and focus of the method; the type of problems or anomalies the method addresses; and how the method guides the inspector to do the inspection. For instance, a Cognitive Walkthrough focuses on the goals and knowledge of a user while performing a specific task, whereas a Heuristic Evaluation emphasises a list of 'usability principles' to be followed as a guideline.

Several inspectors inspect the design using the inspection materials. As the output of this process, these activities produce qualitative results including the early identification of usability problems, anomalies, comments, suggestions and so forth.

The problems found by this process are then used to make recommendations on how to fix and improve the design. Nielsen [9] reported that on average, an inspector could detect around 20%, 40% or 60% of the problems, depending on whether the inspector is a novice (no expertise in either usability or the application domain), single-expert (knowledgeable in usability principles but not in the application domain) or double-experts (expert in both usability and the application domain).

Studies of usability inspection methodology have found that many usability problems are overlooked by user testing, however, such user testing also finds problems that are overlooked by inspection [10]. Therefore, the best result is obtained by *combining* both inspections and empirical user testing.

In this paper, we introduce our novel inspection technique which purposely allows the inspection of sonification applications. The core idea of our technique is to understand the design rationale of the sonification applications being inspected. As shown in Figure 1 above, we propose to critically analyse the design tasks and understand how users interpret the sound output through our two models; the **Sonification Application (SA) model** and the **User Interpretation Construction (UIC) model** [11].

## 3. OVERVIEW OF MODELS

Generally, to do the inspection, an inspector needs to understand the sonification application in the first place. This includes the *sonification technique* used, the *data to be converted* and the *objectives of the application*. The first question that we want to answer is 'How can we describe the application to the inspector?' This is done by considering the required data transformations and tasks to be carried out by the system and the user. These are dealt with in the following sections.

### 3.1 Three Data Transformations

As we defined earlier, the sonification technique is the specific process for *transforming* the input data into sound. Therefore, we propose a special emphasis on the transformation processes – how the data is transformed from its original form via an intermediate 'ready to play' form and then into the final sound. We now describe the three transformations.

1) Usually, time-dependent data (e.g. time series data, stock market data etc.) lends itself more readily for portrayal as sound. Unfortunately, not all data is in this form (e.g. images, equations, multi-dimensional lists etc.) and often needs to be changed into something that is more suitable for sound transformation. We refer to this as **data transformation.**

2) Let us assume that we are using a parameter-mapping technique where the data needs to be converted into some intermediate acoustic parameters. Examples of such acoustic parameters are amplitude, pitch, timbre and so forth. We refer to this conversion as **acoustic parameters transformation.**

3) The outputs from the above transformation are then converted to sound and listened by the users. The user might also be able to manipulate the output, such as repeating selected sounds in a loop, playing through the sound either faster or slower, forward or backward, or playing only a selected area etc. We refer to this process of manipulating the output from the acoustic parameters transformation as the **final sound transformation.**

All three of these transformations significantly influence the final sound output of the application, which needs to be interpreted by the user. Therefore, it is important to ensure that the most suitable transformations are used, and so we must consider these transformations in the usability inspection.

### 3.2 Three Levels of Task Involving Users and The System

Since the objective of the inspection is to find problems or anomalies that the user might encounter while using the application, we need to think about the eventual involvement of users and their interaction with the application. We categorise this interaction into three types of 'task' - namely the **user**, the **application** and the **interaction task**.

In general, tasks at the *'user'* level are those entirely performed by the user, independent of the system (the program). Tasks at the *'application'* level are those performed entirely by the system (to process, manipulate and transform data into sound) without any user involvement. Tasks at the *'interaction'* level are those performed by the user through interactions with the system.

### 3.3 Combinations of Task and Data Transformation to form the Sonification Application (SA) Model

We set these three levels of task in context, by coupling them with the three data transformation processes described earlier, giving a total of nine combinations of task and transformation.

For instance, an *'acoustic parameters user task'* would be an activity performed by users without interacting with the system and relating to acoustic parameters. An example of this might be the user judging the relative loudness level of two sounds (from two different data streams).

To take a different example combination, a '*Data interaction task*' would be performed by users through interaction with the system and which related to data changes. This could refer to activities such as selecting either to sort the data in ascending or descending order; or choosing which data dimension of multidimensional data is to be sonified.

The resulting 9 combinations of task and transformation tasks are shown together in Figure 2 as the horizontal and vertical axes respectively.

In summary, we explain sonification applications through a blend of the data-to-sound transformation processes and the tasks by which the user and the system interact within them. This is the basis of our new Sonification Application (SA) model [11]

### 3.4 The User Interpretation Construction (UIC) model

Since the main output for sonification applications is sound, we need to consider the human auditory system and its capability of interpreting sound into useful information. As a result, we need to address the question 'How will the user *interpret* the output sound?'

This interpretation occurs from the first contact between the sound and the user's ears. This is called 'sensation' and deals with the more basic aspects of experience such as 'how loud does the sound appear to be?'[7]. It is followed by 'perception', which is how we form a conscious representation of the outside environment [7]. Examples of such questions are 'how far away is it?' or 'how large is it?'

We categorise the interpretation process into three levels: selection, reasoning and hypothesizing.

1) The **selection** level is a discriminating process where the user listens out for something that might concern them. This is more to do with filtering and attending only to significant things that we call *conditions*. A condition is a mode or state of the data or sound at particular time. For example, a sound that changes its average pitch from low to high might be considered as producing two different conditions, 'low pitch' and 'high pitch'. At the selection level, the user will be listening out for a change in pitch level, as this might indicate a significant change in the data being portrayed.

2) **Reasoning** is the activity where users construct, arrange or put together several of the above conditions to form a statement or *premise*. Basically, at this level, the user tries to use some or all of the available conditions to make a logical judgement. For example, if a sound had the conditions of 'high pitch' and 'panned fully right' it should be possible for the user to 'picture' this as a certain position in a two-dimensional pitch-panning space. A second sound which had the same degree of panning but a lower pitch, might be considered to be lower than the first one (i.e. directly underneath) in this 2D space.

3) At the **hypothesizing** level, the user tries to make sense, conceptualize or conceive the significance of the above premises by relating them with their knowledge, previous experience or even using their instinct. The combination of several premises forms a hypothesis. A hypothesis is a proposed explanation of a phenomenon based on several reasoning premises. For instance, in the above example, different positions of sound in the 2D pitch-panning space can be used to represent different positions of the mobile phone's joystick in 2D space.

These three activities (selection, reasoning and hypothesizing) are important for the user to interpret the output into more useful information. Therefore, we use these three activities as the basis of our second model to explain how users interpret the output of sonification applications. We call this model the User Interpretation Construction (UIC) model.

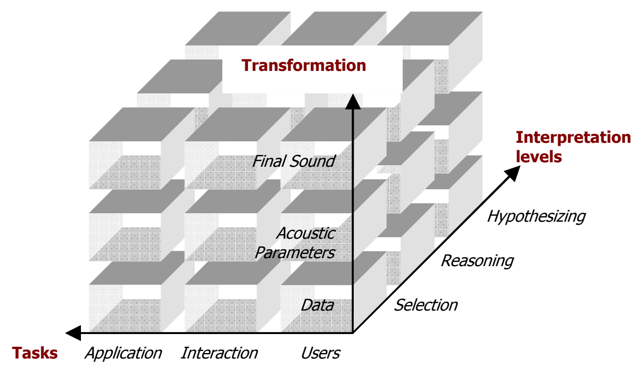


Figure 2: *Focus areas of the Proposed Systematic Usability Inspection Approach*

As we mentioned earlier, both the transformations and the tasks significantly influence the final output. Therefore, for the purpose of inspection, we propose to look at how these tasks in each transformation processes can influence users in their interpretation process. We visualize this as the third dimension called 'Interpretation levels' in Figure 2.

Based on Figure 2, the 'Transformation' and the 'Tasks' axes explain what the sonification application offers to the users to accomplish their tasks, whereas the 'Interpretation level' axis explains what the users will understand and interpret from the output of the sonification application. The application is said to be effective if the user's intended tasks can be accomplished with high accuracy and completeness. This will have happened if the user gets a useful mental representation from the sound. This can be achieved if the intended structure of the data *(explained through the Transformation and Tasks axes*) and the perceptual structure of the sound (*explained through interpretation levels axis*) coincide. Therefore, it is important to ensure that every box in Figure 2 which contains information from the three dimensions has no design problems. Each of these boxes will be the focus of our systematic usability inspection.

## 4. USABILITY INSPECTION APPROACH FOR SONIFICATION APPLICATIONS

The main approach of our inspection technique is to analyse the design rationale of the sonification application and its influence on how the users interpret the output sound. The information about the application is described from the designer's point of view and is gathered through an interaction or interview with them. The information includes all the three tasks (*users, interaction and application tasks*) in each transformation process (*Data, Acoustic Parameters, and Final Sound transformations*).

The design's influence on each of interpretation levels is inspected by questioning and rationalizing every transformation processes based on the situation and contexts where the application will be used or operated. This is important as most products are designed and developed for a specific context of use. This context of use provides inspectors with a guideline about the application such as, 'for whom the application is designed?', 'where it will be used?' and 'what it will be used for?'

The inspection approach encompasses six main steps, starting from the initial preparation to the actual inspection by a team of inspectors and finally the analysis of results. The steps include:

Inspection Materials Preparation Phase:

- Step 1: Describe user goals and tasks; and the application's context of use.
- Step 2: Analyse tasks of the sonification application
- Step 3: Define goals of each Interpretation level for each transformation process.

Inspection Implementation Phase:

- Step 4: Inspect and find design problems or anomalies based on the Inspection Materials

Analysis Phase

- Step 5: Manage and Analyse Inspection Results

To show how this process works, we explain each of the above steps through an example of inspecting the Sonification of the movement of a joystick on a hand-held phone. The joystick is used to create alphanumeric characters as an optional method of text entry. The user moves the joystick to create a character, and every single movement will be converted into sound. The sound is used to assist users in learning how to move and create characters using the joystick correctly without having to look at the text display.

### 4.1 Step 1: Describe user Goals and Tasks, and the application's Context of Use.

In general, an application is developed to help users achieve their goals and to carry out certain tasks successfully. Therefore, it is important for the inspector to know what these goals and tasks are that the user wants to achieve.

Figure 3 shows an example of the goal, tasks and sub-tasks of the sonification of handphone joystick movement. All these goals and tasks are from the user's points of view. This information should be gathered by the designer even before they start designing the application.

```
Goal 1: To write alphanumeric characters through a handphone
  |        joystick
  |__Task 1.1: Write an alphanumeric character
       |___Task 1.1.1: use thumb to move joystick to create the
       |               character
       |___Task 1.1.2: listen to whether or not the movement
       |               is creating the correct character
       |___Task 1.1.3: Confirm that the correct character
                       has been created
```

Figure 3: *Example of goal and users tasks*

Besides the users' goals and tasks above, the contexts of where this application will be used and operated are also described. The examples of context of use are the characteristics of the main users of the application and the kind of situation and environments where the application will be used.

Generally, we use the following four contexts of use- namely (1) users and user tasks; (2) application tasks, equipment and input/output; (3) Interface and Interaction; and (4) Environment; as shown in Figure 1 above [13][14]. These contexts are important for the inspector to know so that the application is assessed fairly and appropriately. This information also provides contextual validity of any problems or anomalies found by the inspector. Further examples of these context are shown below:

1. User and User Tasks
   - User personality, experiences, knowedge, cognitive system, skills, motor system and perceptual system.
2. Application Tasks, Equipments and input/output
   - Application Task – e.g. task flexibility, frequency, reliability etc.
   - Equipment and Technical – e.g. hardware, software, reference materials, network etc.
   - Input and output – depends on the transformation processes as described in the Data-Task State Diagram.
3. Interface and Interaction
   - Interaction – as described in the Data-Task State Diagram
   - Interface and graphical user interface – e.g. graphical windows, menu etc.
4. Environment
   - Physical environment – e.g. condition of the place where system is to be used, noise level, location etc.
   - Organization and social environment – e.g. group working, assistance etc.

As an example for our sonic joystick application, in the context of 'users and user tasks' (in the sub context of users' perceptual system) - '*The users are presumed to have a normal auditory perceptual system*'. By stating this, the inspector does not need to consider the user who has hearing problems. This gives the inspector an inspection scope and enables them to evaluate the application practically.

### 4.2 Step 2: Analyse tasks of sonification application

The tasks of the sonification application are analysed through our new Tasks-Data State diagram as shown in Figure 4. It is a combination of *tasks analysis* and *state of data*. It shows how the data is transformed into different states and the tasks that are involved in the transformation. In previous research in visualization, Chi [15] introduced a Data State Model, however this diagram does not show tasks and interaction between users and the application. CCT (ConcurTaskTrees) [16] is an example of a good task diagram for explaining the interaction between the applications and the users. But this diagram does not show how the data is changed and converted into its different state. Our diagram uses concepts from both these diagrams to analyse the effect of users, the application and their interaction towards the transformation of the data into its final state.

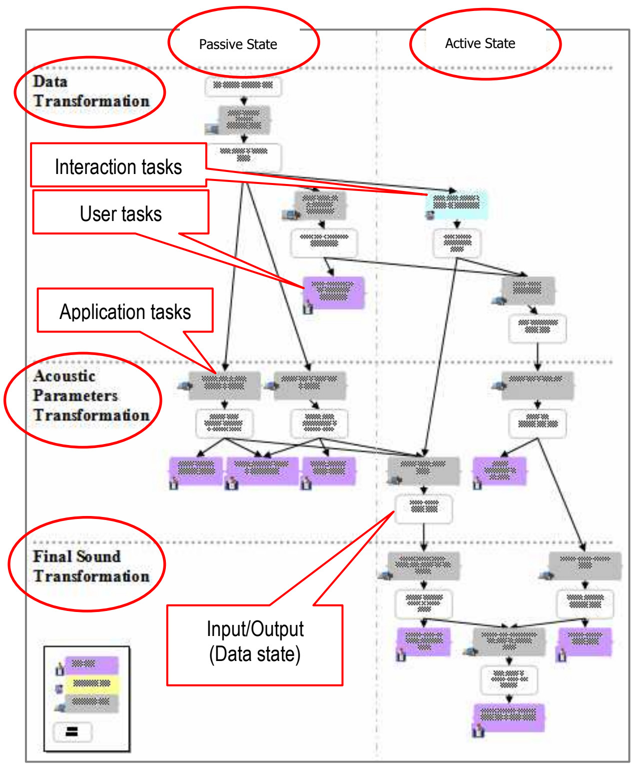


Figure 4: *Example of a Task-Data State Model for a Sonification Application*

Without a good understanding about the design of the application being evaluated, it is quite difficult for the inspector to detect any problems or anomalies. Therefore, it is important to describe the application in as easy and informative a way, and in as much detail as possible to the inspectors. We propose to describe the sonification application by using our Task-Data State diagram because the diagram:

- is easy to understand and use, therefore it improves communication between designers and inspectors.
- gives the inspectors an overview of the application about how the data is transformed from its original condition into a final sound.

As shown in Figure 4, the Task-Data State diagram consists of the three tasks (user, interaction and application tasks) which are placed in one of three transformations processes (data, acoustic parameters and final sound transformation) and two 'application states' (passive and active states). Each task has inputs and outputs except for some user tasks, such as understanding or perceiving. The flow of this data changing processes is shown through arrows that going 'in and out' of the tasks as well as the input/output.

In this diagram, we use application states (Passive/ Active) to describe situations where some sonification applications are in silent mode and only produce sound if there are interactions by the user. For example, in model-based sonification [2], the data is transformed into a physical model where the user can explore it by striking-interaction (as part of the design model) which results in a sound. This sound is taken as the sonification and is presented to the user as a sonic feedback of each interaction. In this diagram, we describe these two situations by dividing the tasks either as 'passive state' or as being in the 'active state'. Any tasks that are involved in creating or changing the final sound while the application is running are described as active state. Tasks are considered to be in the passive state if there was no interruption and they were not involved in changing the final sound output while the application is running such as monitoring application where the user is only required to listen to the sound.

From this diagram, the inspector can observe several useful pieces of information for the inspection process, for example:

- the involvement of users in the production of the final sounds
- the different states of the data before it is finally converted into sound
- the flow how the application transforms the data into different states
- detail tasks of data conversion in each transformation process
- the flexibility of the application in reproducing different sound outputs
- designer's presumption towards what the users should know and do with the application

### 4.3 Step 3: Define Goals of each Interpretation level for each transformation process

The stated goal (in the inspection materials) for each transformation process lets the inspector knows what the designer wants to achieve for each transformation. These 'transformation goals' are gathered from the designer. The goals for 'every interpretation level of each transformation' give the inspectors an idea of what the user should interpret as the application's output. These 'interpretation goals' can be constructed by the chief inspector together with the designer. Therefore, together, both these sets of goals give inspectors the designer's point of view on what the application offers and what the user should do and know.

Basili et al. [12] in their Goal Question Metric (GQM) approach describe goals in these four terms; purpose (what is it required to do?), issues (what qualities are important for the goal to be achieved?), objects (what is involved in carrying out this goal?) and viewpoint (who makes the judgments whether the goal is successfully achieved?). We will be using these terms in defining the goals for the inspection. This may seem rather abstract, but let us take a look at an example. The goals for the phone application for the acoustic parameters perspective are as follows:-

***Goal for Acoustic Parameters Transformation***

*[What is this transformation required to do?]*
***Purpose**: to create the sound's position for all nine joystick contact points (from the data transformation process) by mapping every single point with one of the three pitch levels (mapped to vertical position) and one of three panning degrees (mapped to horizontal position).*

*[What qualities are important for success?]*
***Issue**: to produce a correct sound representation (using 2D pitch-panning space) of the position of the joystick (based on its 9 contact points in 2D space).*

*[What is involved?]*
***Object/process**:*
*(1) All nine joystick positions [contact points of joystick movement by user]*
*(2) Three levels of pitch [low, mid, high]*
*(3) Three degrees of panning [left, centre, right]*

*[Whose point of view is this goal?]*
***Viewpoint**: Designer (application point of view)*

***Goal of Interpretation Levels in Acoustic Parameters Transformation***

***(1) Selection***
***Purpose**: to detect different levels of pitch and degree of panning*
***Issue**: ability of user to differentiate distinct pitch levels and different degrees of panning*
***Object/process**:*
*(1) Three levels of pitching [low, mid, high]*
*(2) Three degrees of panning [left, centre, right]*
***Viewpoint**: User (designer's point of view)*

***(2) Reasoning***
***Purpose**: to deduce different positions based on different combinations of pitch and degree of panning*
***Issue**: ability of user to deduce and judge (correctly) the sound's positions which are represented by different levels of pitch and panning*
***Object/Process & viewpoint:** as above*

***(3) Hypothesizing***
***Purpose**: to interpret that pitch represents the 'horizontal coordinate' and panning represents the 'vertical coordinate' and there are three coordinate values each (producing 9 coordinate points) to represent a 2-dimension location of sound in 2D space.*
***Issue**: ability of user to identify roughly those coordinates in space*
***Object/Process & viewpoint:** as above*

### 4.4 Step 4: Inspect and find design problems or anomalies based on the Inspection Materials

As mentioned earlier in this paper, the core idea of our technique is to understand the design rationale of the sonification application. Step 3 gave an idea of what the designer wanted to give the users, and their assumptions and expectations of what the user should understand. In other words, we try to understand how the designers rationalize the design of their application. If the designers' assumptions and expectations are inappropriate, the application might cause problems for the user. The questions are, can we know whether or not the assumptions or expectations are good and correct? Is it possible for us to detect such problems prior to development?

The inspection process will critically inspect the goals of every transformation and its interpretation levels as well as the tasks required to achieve them. For the purpose of inspection, we propose to ask questions on these goals and tasks through the four different contexts as in Step 1. The inspector needs to follow inspection procedure and uses a given inspection materials. The Inspection Materials is a package containing the necessary documents for inspections, such as 'inspection instructions' and 'problem writing forms' that will be given to the inspectors. The Inspection Procedure explains the rules and regulations on how to do the inspection.

#### *4.4.1 Inspection Materials*

The inspectors will be provided with an Inspection Materials package containing information about the how the inspection should be done, and a description of the application being assessed. These include:

1) Inspection instructions
2) List of Users Goals and Tasks [refer to Step 1]
3) Description of Context of use [refer to Step 1]
4) Task-data state diagram [refer to Step 2]
5) List of goals [refer to Steps 3]
6) Feedback Form where the inspectors can write the problems.
7) Any related and necessary documents such as Severity level reference, sketches of graphical user interface etc.

#### *4.4.2 Inspection Procedure*

Below are procedures to be followed by inspectors during inspection session.

*For each **context of use**, do the following;*
*{*
*For each **goal** (Transformation, Selection, Reasoning and Hypothesizing) do the following;*
*{*
*Work through the design (using the **Task-Data State Diagram**);*
*{*
*Create questions based on the framework below:*

*In terms of <<context of use that you are currently referring to e.g. User's practical experience>> and based on <<the processes and objects of goals that you are currently referring to e.g User Task, data states etc.>> do you foresee any problems…:*

*a) for the application to carry out the <<**purpose**>> and achieve the <<**issue**>> of the transformation process?*
*b) for the user to do the <<**purpose**>> and achieve the <<**issue**>> of the Selection for the transformation?*
*c) for the user to do the <<**purpose**>> and achieve the <<**issue**>> of the Reasoning for the transformation?*
*d) for the user to do the <<**purpose**>> and achieve the <<**issue**>> of the Hypothesizing for the transformation process?*

*While stepping through the design, use the question above and its related information from the context of use descriptions as your guideline to:*

*a) Look for any possible failure scenarios that influence the usability of the application (describe the effects of the problem).*
*b) Identify and detect any possible cause of the problems or anomalies of your scenarios above that may hinder the effective, efficient and satisfying use of the application (describe the cause of the effects).*
*}*
*}*
*}*

All the problems found will be rated by inspectors using a severity level (1 to 4) (adapted from Nielsen [17]). This rating is applied to prioritize the problems encountered. This is especially useful in deciding which problem is most critical and thus needs to be resolved first. The levels are:

1 = cosmetic problem - only needs to be fixed if 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 product can be released

The inspectors need to repeat the same activities for other transformations and their interpretation levels.

Below are several examples of questions which can be generated to inspect the phone joystick application from the STEP 1.

***Transformation****: Data*
***Level****: Reasoning*
***Context****: User [Knowledge]*
***Question****: Could the user's prior knowledge help them to match correctly the 'moving direction of the joystick' with the 'predefined direction needed to represent the alphanumeric character'?*

***Transformation****: Final Sound*
***Level****: Interpretation*
***Context****: User [Perceptual System]*
***Question****: Could the user perceive the combination of different positions and directions of sounds as the shape of an alphanumeric character?*

### 4.5 Step 5: Manage and Analyse Inspection results

In this step, all the problems will be gathered and classified by the chief inspector. The classification is based on the three transformations (data, acoustic parameters or final sound). Each problem will be listed with its frequency and severity level. Frequency shows how often the same problems are detected by different inspectors. The severity level is based on the average (if the frequency is more than one).

The inspectors could also organize a meeting to discuss the feedback or problems they have found, especially to confirm the severity level. There is also the possibility that the problem they have reported is actually not a problem at all. This uncertainty might occur especially when a problem is found only by one inspector and is rated with severity level 1 (cosmetic problem). Through this meeting, inspectors can decide the final list of problems to be given to and considered by the designer. This is important in order to clarify the reliability of the classification method, as different people might have different opinions on which problem is the most serious.

All feedback will be given to the designer and can be used as a guideline for improving their design before progressing to the development phase.

## 5. EXPERIMENT & RESULTS

An experiment to test this technique was conducted with 20 subjects who are currently taking Masters courses at The University of York. 10 of the students are currently studying Music Technology and another 10 are studying Software Engineering.

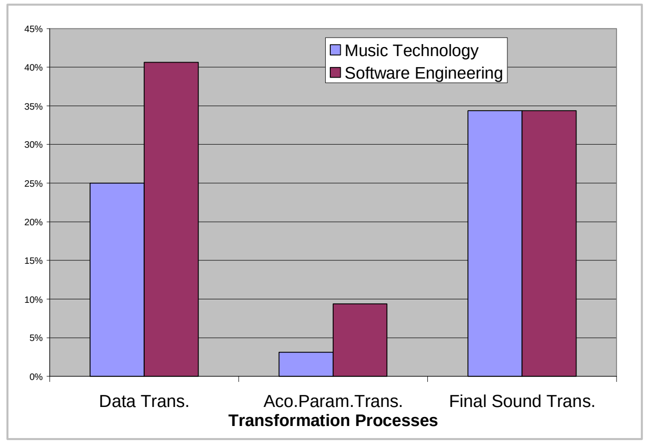


Figure 5: *Percentage of problems in three different transformation processes that found by inspectors from different background*

Figure 5 shows a total of 32 problems or anomalies detected by the subjects, of which 44% were classified as related to 'data transformation, 9% as related to 'acoustic parameters transformation' and 47% as related to 'final sound transformation'. This suggests that the technique is able to detect problems or anomalies that are specifically related to auditory display in addition to those related to the graphical interfaces. In fact, more than 50% of the problems were related to sounds.

The Music Technology students detected only 63% of the 32 total problems while the Software Engineering students detected 84% of the 32 total problems. The chart also suggests that having a background in music technology does not actually help with the detection of more problems or anomalies related to sounds or acoustics. The Software Engineering students detected more problems in data transformation. This might be because of their previous knowledge in the area of usability for graphical user interfaces.

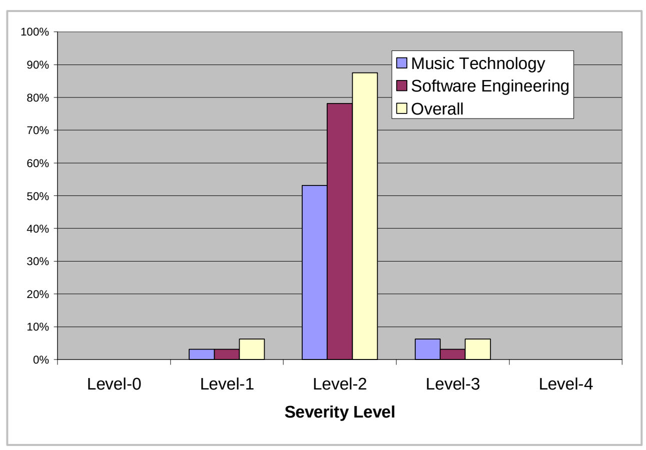


Figure 6: *Percentage of problems based on severity level for different background of inspectors*

Figure 6 shows the problems in five severity levels; namely level 0, 1, 2, 3 and 4. These levels are used to give a rough idea to the designer of how critical and important the problem is. The frequency of the problem can also be used as a guideline to their level of importance. If a problem is found by many inspectors, it is more likely to be important. Therefore, the frequency value can be used to rank the problems within the same severity level.

In this experiment (as in Step 5), a general list of problems was given to all inspectors for severity level rating. Referring to Figure 6, it shows that 88% of the problems were found in level 2; and only 6% in level 1 and 3. Software engineering students seem to have detected more problems in level 2 compared to music technology students.

During the inspection, the inspectors were required to describe the effects and the cause of the potential problems. Several examples are given below:

Example 1:
***Failure story (effect of the problem):***
*It is difficult to memorize the predefined alphanumeric characters especially for older people.*
***Problem (cause of the problem):***
*The number of movements for certain characters is quite high; you may need to reduce it.*

Example 2:
***Failure story (effect of the problem):***
*The sequence of single sounds could be played too fast or too slow for certain users, which make it difficult for them to perceive the sound direction.*
***Problem (cause of the problem):***
*The users are not allowed to set their preferable sound speed.*

Example 3:
***Failure story (effect of the problem):***
*The user may forget what they have written if they listen to the sound only after they release the joystick especially if the character took many steps or movements to create. It is also difficult to detect where the error is.*
***Problem (cause of the problem):***
*The application plays the sounds only after the user releases the joystick (or after the event)*

During the inspection, there were also several problem effects that the inspectors did not really know or quite sure its cause. In this case, it is the job of the chief inspector to identify the cause of the problem. Therefore, it is an advantage in this technique to ask the inspectors to create a failure story in the first place. This indirectly helps in detecting more problems even though the inspector does not really know which part of the application design causes them.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a new systematic usability inspection approach for sonification applications. The approach is based on our two new models called Sonification Application (SA) model and User Interpretation Construction (UIC) model. This approach gives the option for a designer to check their design before the development phase. This could save time and the cost of running an early user test. The idea of this approach is to primarily understand the design rationale and critically ask questions through it. In order to do this, we have explained five steps which include preparing inspection materials, implementing inspection and managing the result. In this paper, we also introduced briefly our Task-Data State model which is used to explain and describe how the data changes with the involvement of users.

Based on this inspection technique, the designer will be able to receive feedback about the design in the form of problem type, number of problems, severity level of problems, and sometimes a solution or suggestion for the designer to consider. We found that by using this technique, we can check thoroughly the design starting from the raw data transformation through to the final sound for the user. Issues such as data insufficiency, sound density, sound structure, perception problems, environment influences, interaction etc. can be addressed through several critical questions. For the future work, we will do a comparison study between our new approach with an existing usability inspection approach such as the Walkthrough or Heuristic Inspection techniques.

## 7. REFERENCES

[1] Kramer G., 1994. *Auditory Displays: Sonification, Audification and Auditory interfaces.* Proceedings Volume XVIII, Santa Fe Institute, Studies in the Science of Complexity, Addison-Wesley Publishing Company

[2] Hermann, T., 2002. *Sonification for Exploratory Data Analysis.* PhD Thesis, University of Bielefeld, Germany

[3] Dombois, F., 2001. *Using Audification in Planetary Seismology*. Proc. 2001 International Conference on Auditory Display (ICAD)

[4] Vickers, P. & Alty, J.L., 2000. *Musical Program Auralisation: Empirical Studies*. Proc. 2000 International Conference on Auditory Display (ICAD)

[5] Pauletto, S. & Hunt, A., 2004. *A Toolkit for Interactive Sonification*. Proc. 2004 International Conference on Auditory Display (ICAD)

[6] Janata, P. & Childs, E., 2004. *Marketbuzz: Sonification of Real-time Financial Data*. Proc. 2004 International Conference on Auditory Display (ICAD)

[7] Coren, S., Ward, L.M. and Enns, J.T., 1999. Sensation and Perception, Fifth Edition. Harcourt Brace College Publisher, USA.

[8] Shneiderman, B., 1998. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, 3rd edition.

[9] Nielsen, J., 1992. Finding Usability Problems Through Heuristic Evaluation. ACM CHI'92 Conference Proceedings, pg 373-380.

[10] Nielsen, J. & Mack, R.L., 1994. *Usability Inspection Methods*. John Wiley & Sons, Inc.

[11] Ibrahim, A.A.A. & Hunt, A., 2007. An HCI Model for Usability of Sonification Applications. TAMODIA 2006, LNCS 4385, Springer-Verlag, pp. 245 – 258.

[12] Basili, V.R., Caldiera, G. and Rombach, H,D., 1997. The Goal Question Metric Approach. Proc. of The International Conference on Human-Computer Interaction

[13] Bevan, N. & & Macleod, M., 1994. *Usability Measurement in Context*. Behaviour and Information Technology, 13, 132-145.

[14] Maguire, M., 2001. Context of Use within usability activities. Int. J. Human-Computer Studies (2001) 55, pg. 453-483.

[15] Chi, E.H., 2001. Expressiveness of the Data Flow and Data State Models in Visualization Systems. CHI 2001

[16] Paterno', F., Mancini, C. and Meniconi, S, 1997. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Proceedings of the IFIP TC13 Int. Conf. Human-Computer Interaction.

[17] Nielsen, J., 1993. *Usability Engineering.* Academic Press. Morgan Kaufmann. UK. p.103.