INFORMATION RETRIEVAL WITH AUDIO FEEDBACK FROM MULTI-DIMENSIONAL SPACES

by

Paulo Pacheco

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

God

I Am the Absolute Essence, I Am Above Nature,
Omniscient and Omnipresent, I am the Universal Mind,
I Am the Originary Cause, I Am the Omnipotent Father,
I Am Distinct and I Am the Whole, I Am Ambivalent.

I Am Outside and Inside, I Am Above and Below,
I Am the Whole and the Part, I involve everything,
Being the Divine Essence, I Reveal Myself Creation as Well,
And I Breathe in My Work, being the Whole and the Fraction.

I Am in your depths, always to keep you,
Because I Am your Existence, your Reason to be,
And I Speak in your interior, and also in your exterior,
I Am in the brain and in the heart, Because I Am the Lord.

So come to My Temple, return therefore to Me,
I Am in you and in the Infinite, I am Principle and I am End,
From My Mind you are children, you will always be gods,
And, marching to the Truth, you will break your crosses.

Do not give yourself in to mysteries, enigmas and rituals,
I want Truth and Virtue, nothing of "isms" and such,
That from Me part the Laws, and, when you grow up in them,
In My Facts you will grow up, in order to have My Glories.

I don't Come and I don't Go, I Am the Eternal and the Present,
I have always Been and I always will be, in you, the Patent Divine Essence,
Your presence is in Me, and I want it plane and grown,
Above simulations, glorifying in Me, the Eternal Life.

Abandoning the overdue an morbid paths,
Which remind idolatrous times and dusty paganisms,
Search for Me in the Inner Temple, in Virtue and Truth,
And joined to Me, you will have, in Me, the Glory and the Freedom.

I have always Been, I always Am, and I will always Be, in you, the Source of Mercy,
Waiting for your Holiness, in the Integral Consciousness,
Because I don't want forms and phoniness, but conscious children,
Collaborating children of mine, through the Union of Our Minds.

Osvaldo Polidoro

Information Retrieval with Audio Feedback from Multi-Dimensional Spaces

Masters of Science, 2002

Paulo Pacheco

Department of Computer Science

University of Toronto

Abstract

This thesis focuses on the exploration of information retrieval with audio feedback from multi-dimensional spaces. Practical contributions were made to the area of human-computer interaction and information visualization. During this process of exploration, four prototypes were designed and implemented and, as a result, a new tool to manipulate multi-dimensional spaces was created: the Phaser tool. Evidence of not better efficiency but more pleasure when using graphical interfaces compared to a text based tree interface is presented in the particular case of using audio as part of the information to be accessed. The results were obtained from a very practical everyday task: music browsing.

Table of Contents

1. Introduction

It is a well known problem to represent a large amount of information in the limited space a computer monitor screen provides. In order to solve this problem, a number of techniques have been developed, allowing the user to move around in a space that does not fit in the screen (panning) or examining in more detail whatever has been possible to represent in such a limited space, in less detail (zooming, semantic zooming – [Per93] and [Bed94]).

In parallel, in order to store information in a computer system, information data is stored in databases. The more knowledgeable the user is about the information data, the easier it is to retrieve information from a database system, once most of the existing computer systems require the user to provide precise information for the interaction.

It becomes difficult to retrieve information when the user has a limited knowledge about the domain. A number of subjects reflect this situation. It is very common, for example, knowing about the symptoms of a disease, like headache or fever, but having absolutely no idea about what kind of disease is that about, viruses or bacteria involved, etc. As another example, try to remember your favorite toy when you were a child. You might remember lots of different characteristics in detail, especially physical features, like color, shape or weight, but you certainly wouldn't know anything about what a computer system would most likely need in order to retrieve information about that toy, like the brand or the commercial name of the product.

A more user friendly system should be able to provide some feedback to the user as he or she interacts with it, so that one is able to know whether or not he or she is in the right track. Returned empty sets from a naive query would be avoided or, at least, not only predicted and communicated to the user, but also the cause of the empty set result identified.

This thesis comes out from the research and development of a multidimensional space architecture and interaction design or mSpace and covers the HCI and Information Visualization fields. My responsibilities in the mSpace project were the design of models and development of prototypes to aid the mSpace exploration and understanding (with respect to user efficiency when interacting with the mSpace) when my independent research started towards the investigation of information visualization (by information visualization we consider "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition" [Car96]).

In order to investigate information visualization in the mSpace and in addition to our limited knowledge domain examples we can consider a musical space, an example of a domain that can be used in the mSpace architecture. When we consider the musical space we can realize that there are a number of music properties, such as composer, period, arrangement, etc., many of them not known by the user. We probably can remember lots of different songs and melodies from our childhood but we may not recall by whom or when they were composed.

A number of existing tools on the web do not support finding pieces of music without the use of queries (which requires very precise input from the user) and displaying connected musical information in an enjoyable and effective way. A music browser application that avoids the use of queries and keywords and displays musical information spatially arranged in a more connected and meaningful way has been designed and implemented.

The classical music domain was chosen among other knowledge domains because it contains a series of attributes non-existing in other domains: *music* is a non-visual entity and temporal (it has to be experienced over time); *classical* because it is part of the occidental culture, being easily recognizable for its unique characteristic. These unique attributes particular to the musical domain allowed us to investigate questions which wouldn't be possible in other domains, like the need of using a visual cue for a non visual domain and how to bookmark an entity that varies in time and therefore may offer different impressions to the same user.

Despite our non-visual domain, our investigation in information visualization addressed a visual interface, in order to check how current research fits in such a domain. Specifically by making use of our application, we investigated the user behavior upon a non-query visual-spatially based application. The research includes three prototypes and a pilot study, where a new tool has been created in order to manipulate multi-dimensional spaces and an experiment that compares a text tree based interface to a GUI, both using the same tool.

1.1 Thesis Statement

By the use of visual spatial distribution of different attributes of the musical information space, real-time animated feedback, zooming, selective aggregation and highlighting, it is more effective and enjoyable to visualize elements and its attributes, learn and visualize the connections among similar elements (elements that share some common attributes), locate pieces of music without the use of queries, which requires the knowledge of keywords, and understand the musical domain lexicon.

2. Related Work

My research investigates the major information visualization problems faced by users who browse music in the World Wide Web:

- Finding pieces of music according to their interests: music they knew previously (complete or partial information about – part of the melody, composer or title, for instance), and new music, depending on their knowledge.

- Understanding the connections among different pieces

- Having these tasks to be performed in an enjoyable and effective way

The implemented music browser takes into account the reduction of cognitive load always providing continuous transformations in the display. According to Robertson in Cone Trees [Rob91b], fig. 2.1, "animation shifts a user's task from cognitive to perceptual activity, freeing the cognitive processing capacity for application tasks". The time spent to reassimilate a new configuration is eliminated. When changing display configurations for example, the user, acting in a perceptual level doesn't spend energy figuring out what happened with the previous configuration and the relationship between that with the new one.

According to Robertson [Rob91a], this task shift from cognitive to perceptual activity is illustrated. In this work they demonstrate how animation saves several seconds that would have to be used for reassimilation between two different application states. In addition, not only the user performs the tasks in a more enjoyable way but also he or she understands the information structure more completely. In that work, in the presented framework, "many seconds and perhaps tens of seconds" would be taken to "reassimilate structural relationships after a tree transition without animation".

As disadvantages of animation, we highlight the implementation difficulty and the user distraction considering cognitive aspects (freeing the user from thinking).

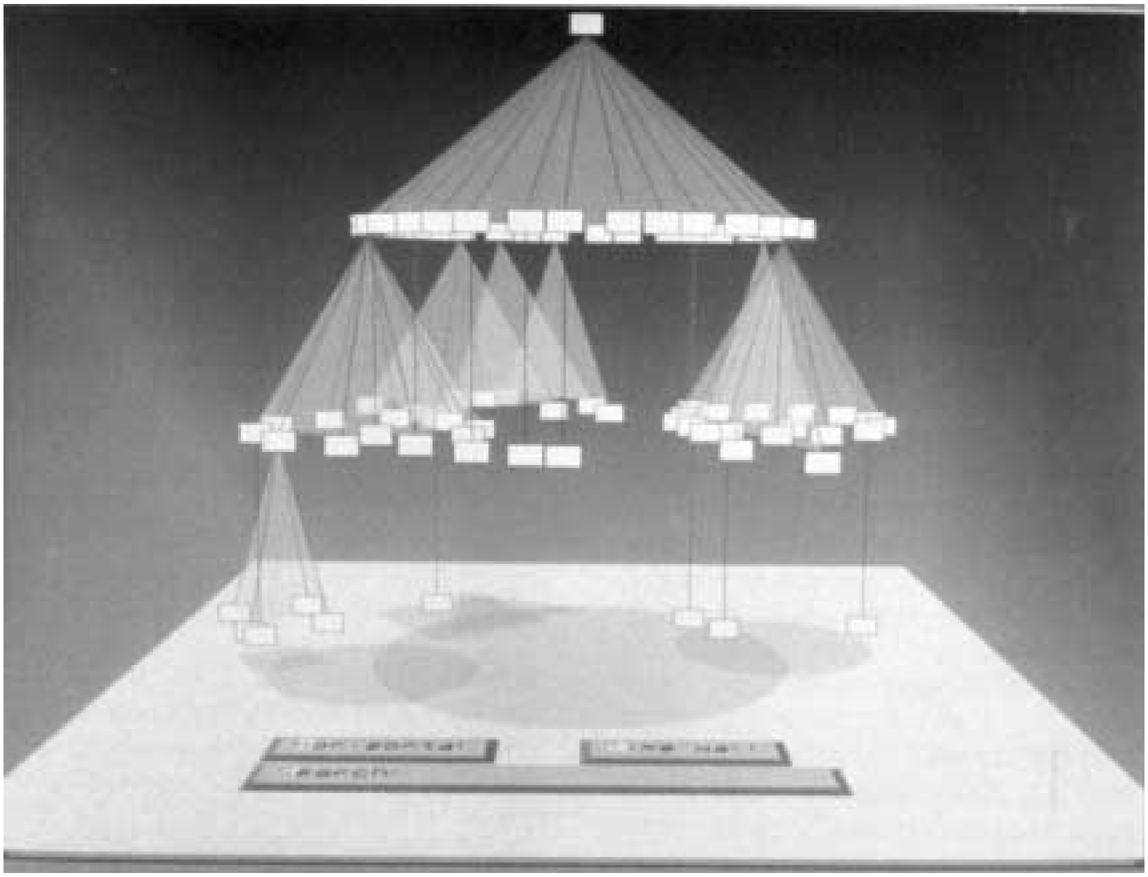

Figure 2.1 – Cone Trees (from [Rob91]). A hierarchical 4-level 3D structure is presented, as well as its 2D projection on a plane.

The relationship between the detail and context is also improved through animation, since we can make it even clearer where the detail came from or what it is related to. Methods for shrinking and expanding pieces of information representation like icons or any chosen symbol implement a mean for this goal.

Furthermore, some other interaction and visualisation techniques were applied in our music browser in order to investigate their values when applied to the mSpace. In the next sub sections, these techniques are described.

2.1 Interaction

An extensively investigated interaction technique, here applied to the music browser, is the use of dynamic queries, which avoid the use of simple queries that demands the previous knowledge of keywords. Semantic zooming and browsing by similarity have also been applied and are part of this section.

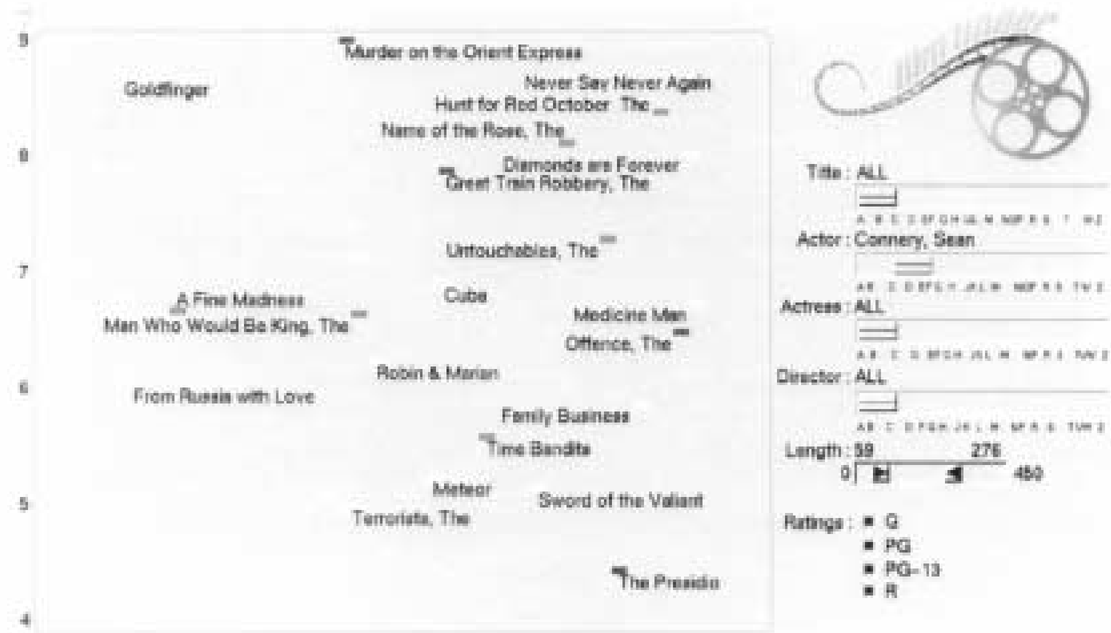

Figure 2.2 – The Film Finder. Using a combination of sliders, the user is able to create a query and retrieve the results spatially arranged (from [Ahl94]).

2.1.1 Dynamic Queries

Dynamic queries (fig. 2.2) are queries built incrementally and interactively by adjusting sliders and select buttons. Users may eventually observe animated results.

Ahlberg and Shneiderman [Ahl94] point the following principles of direct manipulation to the database environment:

- Visual representation of the world of action including both objects and actions

- Visual presentation of the results
- Rapid, incremental and reversible control of the query
- Selection by pointing, not typing
- Immediate and continuous feedback

As benefits, we can highlight the possibility of displaying more complex information, allowing the user to visualize and interpret results, explore and build relations. In addition, the time spent to learn how to use the system is reduced.

As disadvantages, we can mention the current necessity of rebuilding almost all existing software applications from scratch, since the existing software still doesn't predict the possibility of integrating dynamic queries, attending full Boolean capabilities.

The "output is input" principle eliminates the need of having distinct spaces on the screen for input and output, improving the efficacy of space usability. Sliders tight coupling are used in this sense, providing a description of the current station and, acting as new station generators or selector.

Kumar et ali. [Kum95] reported that a smaller, more manageable dataset is achieved after some iteration (iterative refinement or progressive querying of data sets). They presented a tree-browser visualization tool that consists of two trees in tightly-coupled views, one detailed and the other overview. Dynamic queries filter nodes at each level of the tree. Unselected nodes sub-trees are pruned out, which is a feature that they report very useful. Another interesting concept pointed by the authors as a future work is the semantics-based browsing, where the tree could be navigated in many different ways, like preorder, postorder and inorder, and tours of nodes marked either manually or by a query. These are important considerations when comparing pure GUI's with text tree based interfaces, since there is room for improving efficiency in these tree models.

2.1.2 Other Techniques

Prospecting information from large database systems is a requirement from many users.

Existing solutions are implemented in a number of formats: Hierarchical (Cone Trees [Rob91b]), Linear (The Perspective Wall [Mac91]), Positional (The Neighborhood Explorer [Spe01]).

In the hierarchical approach of the Cone Trees, nodes reveal a useful relation with their children and parent but sometimes a "multiple inherited" relation is needed as opposed to a single hierarchical relation.

According to the interaction perspective (see in the next section visualization considerations), a 2D hierarchical interface was chosen over a 3D interface, given a user works more effectively in a 2D computer environment requiring less computer skills. Sebrechts et ali. [Seb99] reported for their experiment that users with more experience demonstrated more facility to deal with 3D environments. Overall, 2D and text environments demonstrated to be more adequate for best performance. Time lag in system response caused by slow machines is longer in 3D environments and this lag affects considerably novices, according to Sebrechts, "causing some participants to 'get lost in the space.'" Cockburn and McKenzie [Coc01], reported that managing objects in a 3D metaphor is slightly slower that in a 2D one, but users reported the tasks to be more pleasant in a 3D environment.

Nonetheless, a very effective application should address the initial cognitive approach, with the starting point being the user's initial knowledge or will with little or non-existing information to start from, but quickly migrating to the perceptual field, when the user's action resulting feedback becomes concrete. Tweedie [Twe95] mentions these two approaches, one cognitive and another perceptual.

Another aspect is that data should always be visible, as opposed to many query-based applications where data is available only after being called. Spence [Spe01] points that a drawback to the dynamic queries [Wil92] is that data is only available when all object attributes satisfy all limits. Figure 2.3 shows the Neighborhood Explorer, illustrating the use of spatial arrangement of elements minimizing the use of queries.

A semantic zoomable interface completes the scene, allowing an overview + detail approach to take place: as we zoom in certain areas of the application canvas different levels of information are revealed. In “Pad”, [Per93] points out that zooming in different levels makes it easier for the user to organize informationally large workspaces (fig. 2.4). As a result of this semantic zooming, not all objects or pieces of information have always to be drawn, improving the application performance. In “Pad++” [Bed94], a zooming graphical interface properly implemented maintains a high frame rate interaction with very large databases. An interesting discussion about how limiting is the use of metaphors in contrast to the simple use of physics laws is presented. In this work, we kept that idea in mind, initially using metaphors, but trying not to be limited by them.

In order to facilitate the mSpace exploration, browsing by similarity also plays an important role, allowing users to easily group and visualize pieces of music that share common attributes. Rodden et ali [Rod01] evaluated users' behavior quantitative and qualitatively while they were trying to browse images arranged in two different ways: randomly and grouped by similarity. Users reported being more enjoyable and easier to find images arranged by similarity than placed randomly. In addition, it was easier to find images that complemented each other.

The experiments also compared arrangements image based versus caption based. They revealed a preference by arrangements caption based.

2.2 Visualization

As we stated in the introduction, we are investigating visualization techniques, despite the dominant non visual characteristic of the music domain. According to Sutcliffe et ali [Sut00], “The conclusions from the study are that while visual user interfaces for information searching might seem to be usable, they may not actually improve performance. Training and advisor facilities for elective search strategies need to be incorporated to enhance the effectiveness of visual user interfaces for information retrieval.”

Mereu and Kazman [Mer96] reported that audio helps improve a user's sense of depth perception. This may play a significant role when evaluating GUI with audio feedback.

In addition to these studies, we are investigating how these visual techniques apply to our domain.

Under the visualization perspective (see in the previous section interaction considerations), again, a question that immediately follows is whether or not to implement a 3D interface, as opposed to keep more conservative and use a traditional 2D interface. Despite the currently abundant 3D technology, the choice was the use of a 2D interface.

As an advantage of a 2D organization, a plane allows a matricial organization, providing at least two relations per element (axis x and y).

In addition, 2D environments facilitate proximity and shape relations that can be put together in order to link elements by one or more attributes.

Sebrechts [Seb99] reported for their experiment that users with more experience demonstrated more facility to deal with 3D environments. Overall, 2D and text environments demonstrated to be more adequate for best performance. Time lag in system response caused by slow machines is longer in 3D environments and this lag affects considerably novices, "causing some participants to 'get lost in the space.'"

Cockburn and McKenzie [Coc01] reported that managing objects in a 3D metaphor is slightly slower that in a 2D one, but users reported the tasks to be more pleasant in a 3D environment.

In the 2D world, Amento et ali. [Ame00] investigated an interface for web sites evaluation and organization, through the use of 2D spatially arranged icons (thumbnail images) and lists (fig. 2.5). The reported result was a more effective way of organization compared to conventional web browser (Yahoo™) bookmarks.

An interesting and convenient related piece of work is a comparison of the effectiveness between the spatial organization of information and common lists. There is not much research that compares these two modes. Risden et ali. [Ris00] compared the use of a 2D list with a 3D graph and reported the 3D graph to be more efficient when organizing elements. On the other hand, Cockburn and McKenzie [Coc01] reported quantitatively Cone Trees to be less effective in finding hierarchical pieces of information whereas considered qualitatively to allow a better way of visualizing the data.

An also interesting piece of work, designed to visualize large quantities of independently authored news stories, was implemented by Rennison [Ren94]. It makes use of 3+ dimensional spaces, semantic zooming, panning, animation and dynamic visual cues and presentation of information. The result of this work is a model of an interactive news information system. One problem addressed is "the lack of a general, or known, structure of the information available to the user". In our development we kept in mind this issue, providing to the users as much as possible means of context awareness. Another interesting aspect discussed is whether or not make use of filters for information retrieval given, in that case, the information is in the form of articles, being up to news editors or maybe intelligent autonomous agents to setup the filtering. In our case that could be something important to consider, for instance in the case of subjective ways of classifying classical music, like "happiness", "inspirational" etc.

In addition, Chalmers [Cha96] has investigated and developed an algorithm to visualize high-dimensional data in low-dimensional space. His approach takes into account weighted relations among element attributes in a database and algorithmically converts these relations into forces, which, in the low-dimension space, push "similar" elements closer or pull less similar elements apart. His work has been applied to informational landscapes for document visualization.

2.3 A 3D Information Exploration Model

Waterworth and Chignell [Wat91] proposed a three dimensional information exploration model. The proposed dimensions are Structural Responsibility (navigational – user, mediated – application), Target Orientation (querying – definite, browsing – indefinite) and Interaction Method (descriptive – described by the user, referential – selected by the user). According to this model, a pure information retrieval application is query based and mediated whereas a hypermedia application is browse based and navigational. The implemented music browser is browsing based, both navigational and mediated, and referential.

Four prototypes were developed in order to design and evaluate an application that could allow the user to locate pieces of music effectively and in an enjoyable way, as well as capable of displaying interconnections between the musical elements.

In the next sections we will present design considerations and implementation details of this application.

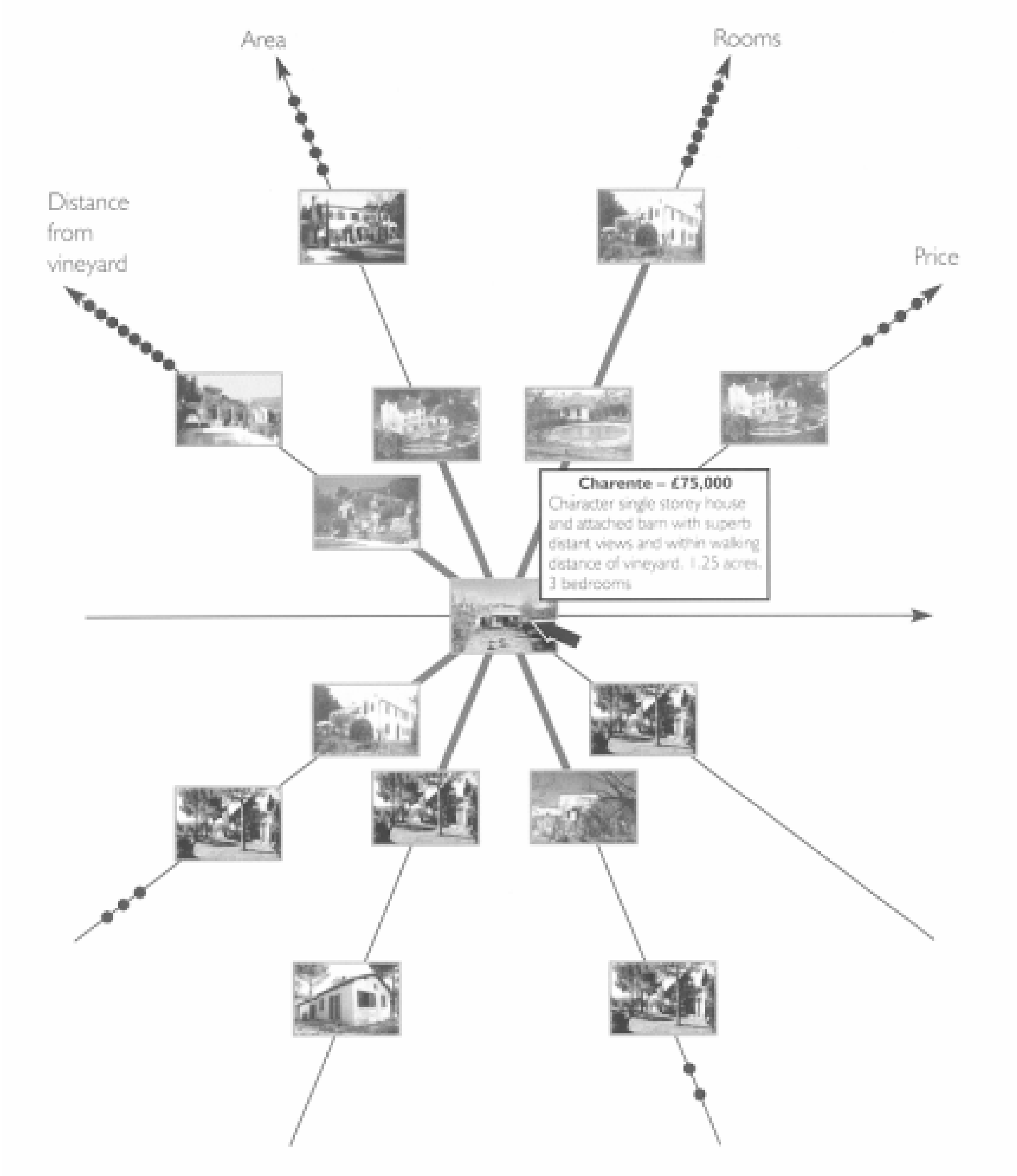

Figure 2.3 – The Neighborhood Explorer. Element positions are associated with radial scales, with the currently examined house at the center (from [Spe01]).

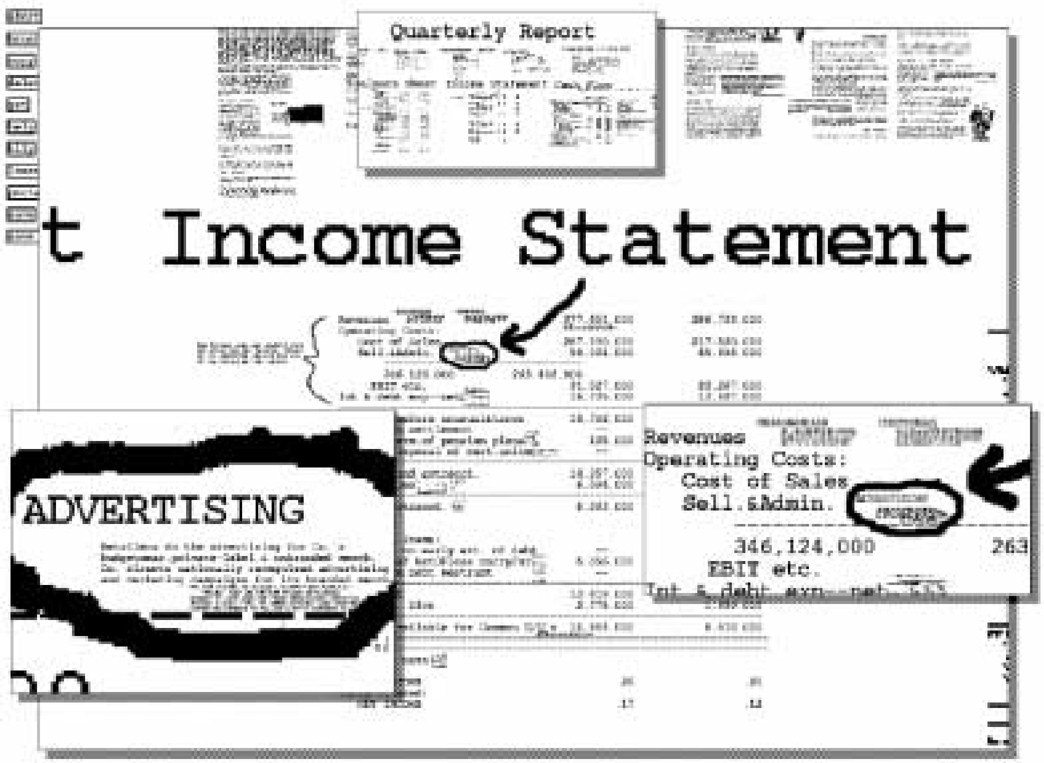

Figure 2.4 – Pad. Areas of interest can be zoomed in without loosing the context.

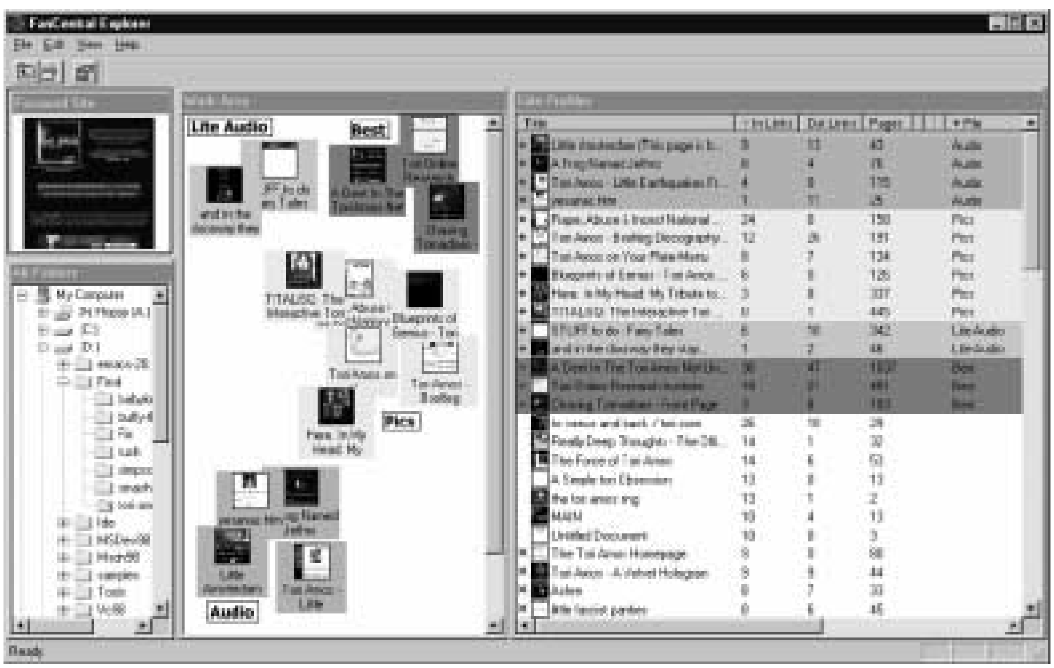

Figure 2.5 – TopicShop. Elements are spatially arranged as icons and displayed as a list (from [Ame00]).

3. Design Considerations

As mentioned at the end of the previous section, an application that could allow the user to locate pieces of music effectively and in an enjoyable way, as well as capable of displaying interconnections between the musical elements was designed and implemented.

During the design phase, the following constraints were kept in mind:

- The user is a 16 – 45 years old person, who knows how to read e-mail (at least once a week) and is interested in some genre of music.
- To display pieces of information, spatially arranged in a connected and meaningful way.
- To retrieve this information, avoiding the use of keywords and queries in an enjoyable and effective fashion.

The core design consideration, though, was to have a piece playing immediately so that we had an immediate audio feedback and we could investigate weather or not a visual feedback is important to the user.

We will present some discussion about these constraints.

3.1 Displaying information in a connected way

One of the objectives of this study is to analyze how an application can be useful to provide not only information to the user, but inform how these different pieces of information are connected so the user can have an easier way of understanding the domain lexicon than the way provided, say by a common hierarchical list, a very common way of displaying information in commercial applications.

3.2 Avoiding the use of keywords and queries

Since we are focusing on "naïve" users, in other words users who are not very familiar with the classical music domain, uncomfortable dealing with complex queries but interested in finding music, nonetheless, we want to free the users from dealing with terms that they are not familiar with. For example, when users are trying to locate composers they haven't heard about, it wouldn't make sense having them to type, say "Beethoven".

Considering [Spe01], as mentioned in section 2.1.2, we want the user to be able to find Beethoven without knowing this name, either randomly browsing and finding this composer by chance or as part of a context by similarity as we mentioned in the previous section. In other words, to find some piece of music sharing common attributes with an already familiar piece of music.

3.3 Design

As we stated before, we are interested in finding pieces of music and understanding the connections among them. Our literature review helps us consider some design aspects for a music browser:

- Sebrechts [Seb99], Cockburn and McKenzie [Coc01] and Amento [Ame00] indicate that a 2D interface is more adequate given its efficiency compared to a 3D interface.

- Rodden [Rod01] points that a system capable of pointing similarity among elements (pieces of music in this case) improves efficiency for locating new ones based on elements previously selected.

- Perlin and Fox [Per93] and Bederson and Hollan [Bed94] signalize the importance of zooming in different levels.

- Robertson [Rob91a] reinforces the use of animation, minimizing the cognitive transition load between two different configurations

In addition, the following question was kept in mind in order to achieve the best solution to accomplish our objectives: What happens first? In other words, should a random piece start playing first or it should be up to the user start whatever he/she feels like.

These aspects are going to be considered for the application design. In the section 4, more specific design considerations are pointed according to each prototype.

3.4 The Negative Hypothesis

- The user gets lost or disoriented while navigating in the mSpace domain. The application doesn't provide a clear map to the users, in order to make them be aware about where to find pieces of music and the interconnections among them. It must be clear to the user, after a couple of interactions (5min) with the application, what different areas in the output display mean, in terms of understanding each piece of information representation.
- It is difficult for the user to find pieces when using the application (it takes a long time). Not only the pieces of music must be statically well organized but it must be also intuitively possible to locate them, because the tool is easy to use and the meaning of different sectors is clear with respect to different attributes they hold.

4. Prototypes

Four prototypes were developed in order to design and evaluate an application that could allow the user to locate pieces of music effectively and in an enjoyable way, as well as capable of displaying interconnections between the musical elements.

The initial two prototypes were based on circles and lines. Circles were horizontally arranged representing stations. Each line was representing a set of stations sharing common dimensions.

The last two ones were based on a projection plane. The musical piece, represented by circles in an n-dimensional space, were projected on a 2D plane.

Details follow in the next sections.

4.1 The First Prototype

The first prototype was built initially using a low-fidelity (lo-fi) implementation. After going through a few cycles of user evaluation and design, the high-fidelity (hi-fi) prototype was coded in Java.

4.1.1 The Lo-fi prototype

4.1.1.1 Introduction

In order to define and perform preliminary tests, a low-fi prototype was implemented. As in "Prototyping for Tiny Fingers" [Ret94], a low-fi prototype is a paper and pencil implementation that allows a quick software implementation. According to Rettig [Ret94], "more complex and expensive stages are skipped and a feedback from the user can be quickly obtained and the model recursively corrected in a short period of time through a few cycles."

4.1.1.2 Implementation and Evaluation

Our objective was to develop an application that could:

- Suggest a piece to the user by automatically randomly start playing a piece
- Allow the music domain exploration by the user based on the suggested playing piece
- Allow the music domain exploration by the user starting from the whole domain (regardless the initial suggestion)

The chosen metaphor was the radio dial visually represented by aligned circles. Each circle represented a station, initially with a single piece. The stations are generally represented by numbers in a horizontal scale. The reason of this chosen metaphor is simple: the radio dial visually represents pieces and their relations among each other, and is a way of finding music. This comes across to our objectives of representing and finding pieces. In addition, users in general are very familiar locating stations in a radio.

The line containing pieces in our prototype would branch into other lines as the user wished, according to the basic idea of branching by similarity: given a piece with its attributes, the user should be able to select the attributes that he or she wanted to explore and then, by selecting one or more attributes (or better yet, its values) another line of pieces containing a subset of the previous line would be displayed.

This prototype was tested using the protocol in Appendix A. Basically, users were asked to find a piece they liked, how would they create a new set of stations containing pieces sharing the same attributes as those in the original piece they selected originally, how would they create an entire new set of stations based on attributes they had in mind and how would they tag pieces they liked. A post questionnaire followed.

Users reported during the evaluation the following difficulties:

- "It is not clear where I should click to expand a line"
- "I am clicking here and I was expecting expanding a line but a combo box was showing up".

A very interesting learned issue from this evaluation was how clearly labeled every part of the interface should be, with respect to their functions (menus, combo boxes, and buttons).

According to the users' evaluation the metaphor adopted was adequate, clear to use.

The hi-fi prototype implementation took these observations into account.

4.1.2 Hi-Fi

The implementation of this prototype naturally followed the last stages of the lo-fi evaluation. This code-based implementation contained much clearer elements, since they were labeled according to their functions.

Figures 4.1 to 4.5 show a story board with different scenarios of use. The used metaphor was the radio with dial stations as explained in the previous section, 4.1.1.2.

Initially, the user can see a single line with a piece playing. A combo box is available for sorting by any of the dimensions (fig. 4.1). The double circle shows the selected station (the central one in the Vivaldi section).

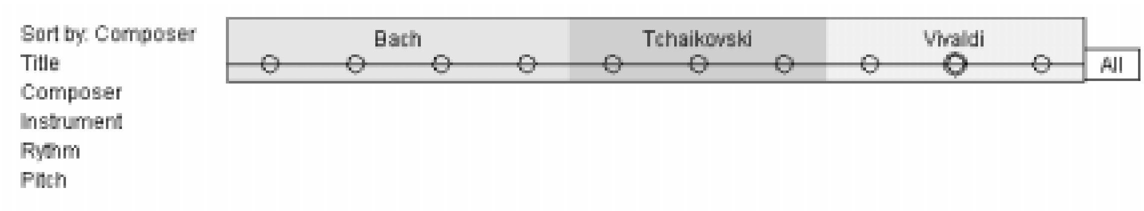

Figure 4.1 – Prototype 1 with the initial line of stations. A combo box allows sorting by any of the available dimensions.

If the user wants to create a new line of stations that share a common attribute with the selected station (and piece playing), a multiple selection box menu is displayed by clicking in the pre-selected station. The same attributes as those of the playing piece are made available for selection, so the user can create a new line based on a subset of those attributes. The number of stations (pieces) to be created in the new line is displayed besides the "GO" button (fig 4.2).

Figure 4.2 – Prototype 1. The user can select attributes related to the selected piece for branching and see how many pieces with those attributes are available.

By pressing the "GO" button, a new line is displayed with the stations sharing the attributes selected (fig. 4.3). Multiple lines are displayed to support context: the user has an overall view of station sets and a history of creation as well. The station that has originated the new line is linked to it. The grayed triangle indicates the all pieces from the original line were taken into account to create a new one, filtered by the selected station attributes.

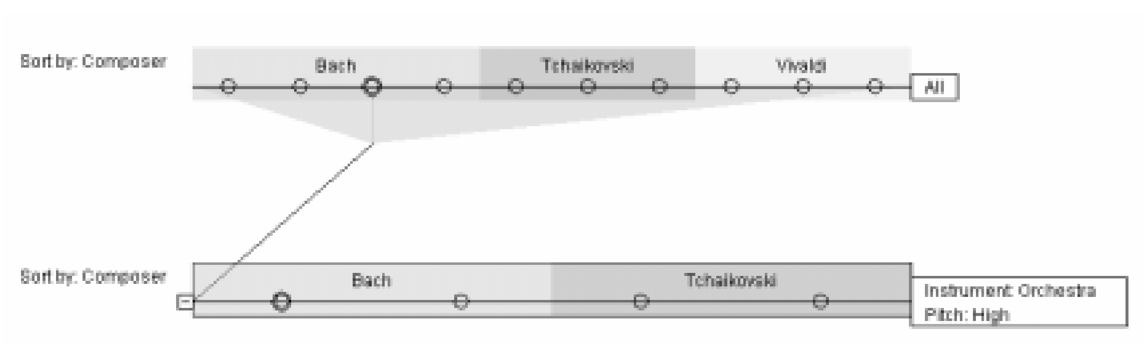

Figure 4.3 – Prototype 1 with a branched line. All pieces in the branched line share the same values ("Orchestra" and "High" for "instrument" and "Pitch" attributes, respectively) as the piece that originated the branch.

In a more general way, many lines can generate many new lines, through the same process described in the previous paragraphs (fig. 4.4). If there are too many lines to be displayed on the screen, a mechanism of expanding/shrinking lines has been implemented by clicking in the little boxes on the left part of the lines with a “minus” sign. Another solution would be a scrolling mechanism.

In order to allow the music domain exploration starting from the whole domain, we developed and implemented an interesting tool (fig. 4.5). It consists of vertical lines (one per dimension, e.g. Composer, Instrument, Rhythm and Pitch) with the respective dimension attributes. By selecting the attributes, users have an important immediate feedback showing how many stations would be available from that selections and how many stations would be available from alternative selections.

A design review of the prototype at this stage made it apparent that the application should be capable of representing clusters of pieces, since the database may contain hundred or even thousands of pieces and since the line based prototype contained a limited amount of space to represent pieces.

In order to address these clustering raised issues, a second prototype was developed. The prototype was adapted to be more tightly coupled with the metaphor of the radio dial.

4.2 The Second Prototype

The second prototype was then implemented, using the first prototype as a framework, without the need of a new lo-fi evaluation, since most of the UI issues were common for both approaches, very similar to each other.

Figure 4.6 shows the second hi-fi prototype. The stations contain multiple pieces and a play list window displays the titles of pieces contained in the selected station (indicated by a double circle).

The second prototype basically addresses the problem of clustering. It allows placing many pieces in a single circle (station). It was heavily based on the first prototype and it hasn’t been fully implemented in favor of the prototype discussed in section 4.4, because of problems pointed early, mentioned below.

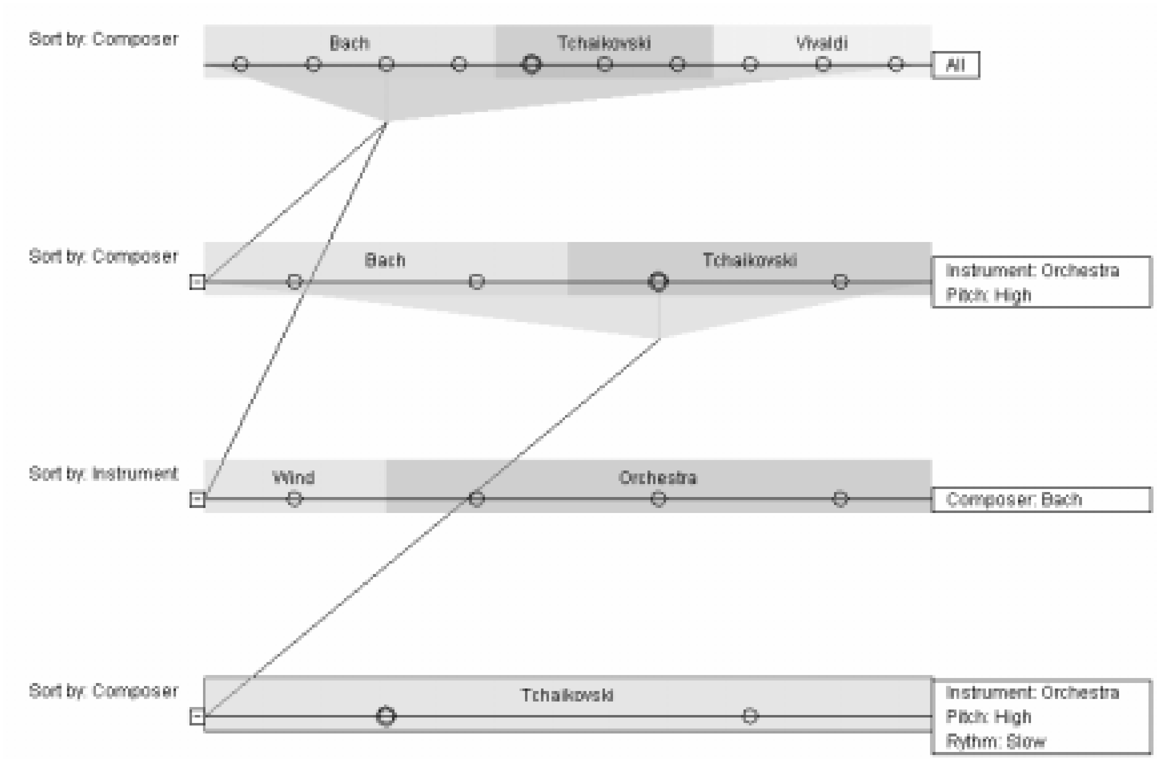

Figure 4.4 – Prototype 1 with nested branched lines.

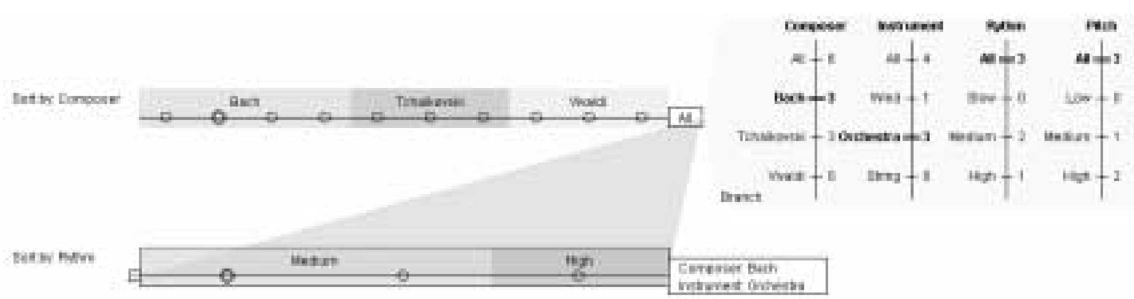

Figure 4.5 – Prototype 1. On the right side is a tool for dynamic queries with output number preview.

Figure 4.6 – Prototype 2, incorporating the idea of "Stations" (circles) containing clusters of pieces, and a playlist displaying the contents of the "Romantic" station.

By hovering the mouse over the circles, the user was able to get specific information about the cluster (station).

We identified a major potential problem with respect to creating new lines of stations by similar attributes (branching by similarity). The following questions were raised:

- How can we represent the branches from the stations in a connected way with the parent node (station)? Stations contain multiple pieces and the connection with them (within the station) may not be clear. Zooming could support this, but in this case we would start mixing metaphors – spatial representation (2D, 3D) of the station contents in contrast with the radio dial (linear) metaphor. This was a key metaphor raised issue.
- If the screen is not big enough, how should we represent a huge amount of branches? If we decide implementing a scrolling approach, how to keep all connections between children and parents if any of them are not visible? This was an important point with respect to keep the context working.

In order to collect more information about the way web applications were dealing with music browsing, as well as the mentioned raised issues on context and clustering representation, a pilot study was done.

4.3 Pilot Study on Web Music Browsing

A pilot study on web music browsing was done in order to investigate how the top most popular web applications were dealing with elements like focus+context, clustering representation, dynamic queries and music suggestions.

This investigation was not limited to classical music since the current applications are focused mostly on pop music and we wanted to evaluate how the user behavior was upon these web products in their full functionality. In other words, we didn't want to hide functionalities like advertisements or new releases banners.

Three Computer Science graduate students were asked to complete the following tasks:

- To find a piece of music that they like and they knew its existence previously
- To find a piece of music that they like and they didn't know its existence previously

And to answer the question:

- What did you learn with this process?

The subjects were allowed to use any available web-based tool. If the subject wasn't familiar with any application for music browsing purposes, Morpheus® (fig. 4.7), CDNow® (fig. 4.8) and Google® were suggested, since these were the most popular tools according to word of mouth. For the second task, the albums and/or artists mentioned were not previously known by the users.

4.3.1 Summary of the Experiment

The details of the experiment result are described in Appendix C.

As a result of this experiment, the first subject used 2 applications in 14 minutes; the second subject has visited 56 pages/links using 2 applications in 13 minutes and the third subject has visited 18 pages/links using 2 applications in 14 minutes.

The discussion follows below.

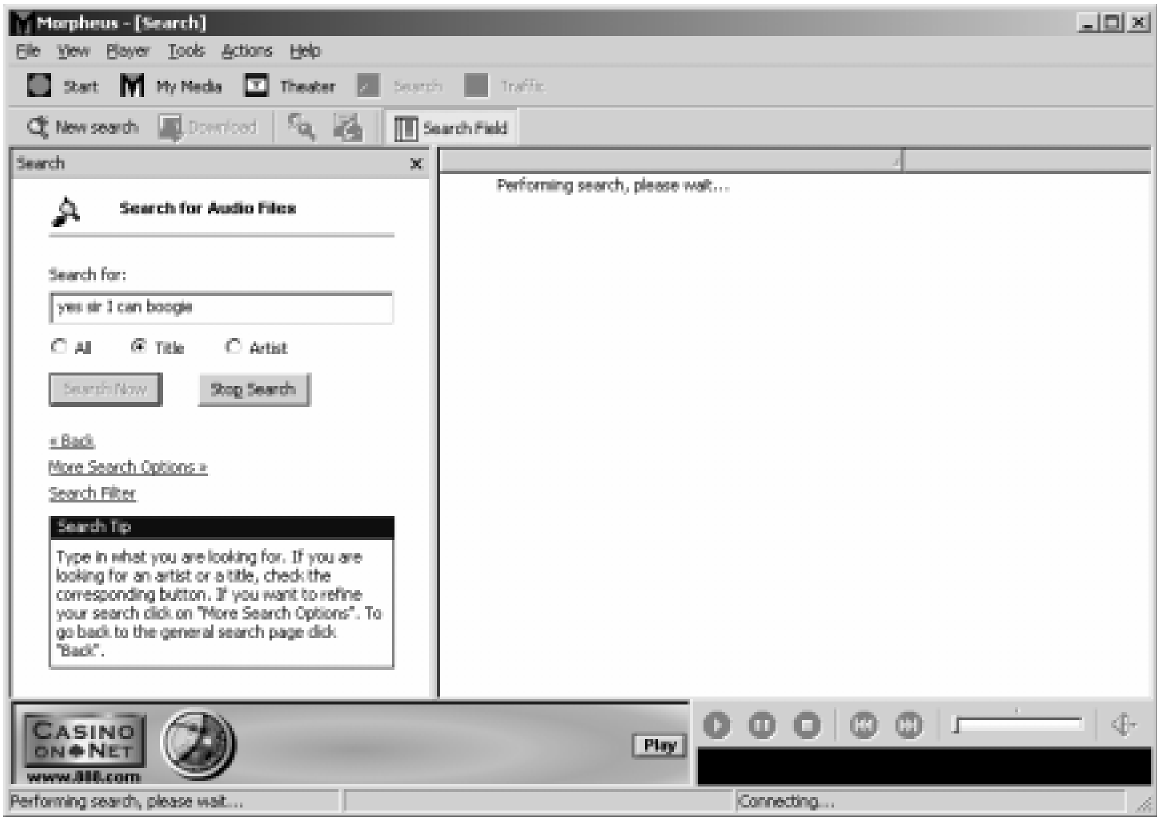

Figure 4.7 – Morpheus. A web-based tool for music browsing.

4.3.2 Discussion

This pilot study allowed us to conclude that a number of pages were visited (at least 17) in order to locate pieces. That indicates that the information required is not clearly placed in the application. Sometimes, not even the application itself is enough; the need of keyword knowledge to satisfy the query built is an obstacle for the users to find something they are not familiar with.

In addition, the average time users took to use the application to perform the tasks (about 14 minutes) was severely disturbed by the download time, causing psychological discomfort. As a possible solution, either new web technologies should take place or the mechanism used for the users to access the music servers should be improved, maybe adopting restrictive measures for the servers.

Context was something very badly implemented. Users complained about not having access to it. Lists were missing.

This pilot study was very important for us to carry on with the third prototype, addressing the issues of clustering representation and context visualization.

Figure 4.8 – CDNow. Another web-based tool for music browsing.

4.4 The Third Prototype

After a clustering representation analysis and design during a few mSpace group meetings, another different approach was upon discussion, in order to fulfill the following ideas:

- To obtain an application capable of being implemented in different environments including a non-visual one, in order to apply in situations where displays are not convenient, for example, while the users are driving a car or performing any activity that requires their visual attention. The goal here was to develop a concept that could be designed also for an application in a pure auditive (as output) and motor sensitive (as input) environment.
- To position the pieces in a spatial distribution, which would be conflictive with the station dial buttons. Clusters should be freely represented as entities immersed in the space, and not lined up.
- To have each point connected to audio playback

4.4.1 The Bi-Dimensional Scaled Space Approach

The just exposed ideas about our design are addressed by this prototype. Spence [Spe01] reinforces this important idea of position and Rodden [Rod01] addresses the problem of similarity by spatial position. The bi-dimensional orthogonal with vertical and horizontal selectable scales implements a spatial arrangement of elements and is shown in figure 4.9. Using the combo boxes the user can select the attributes in each scale of the 2D orthogonal plane. It is interesting to observe that when changing the scale attribute, an animation takes place showing the element position changes (figs. 4.9 to 4.11). Details about its constituents follow below.

Output canvas: This window displays the pieces and its relative positions among each other. This canvas is also "panable", i.e. by dragging the mouse in its area the user can translate the position of its contents, for the case when not all pieces are visible.

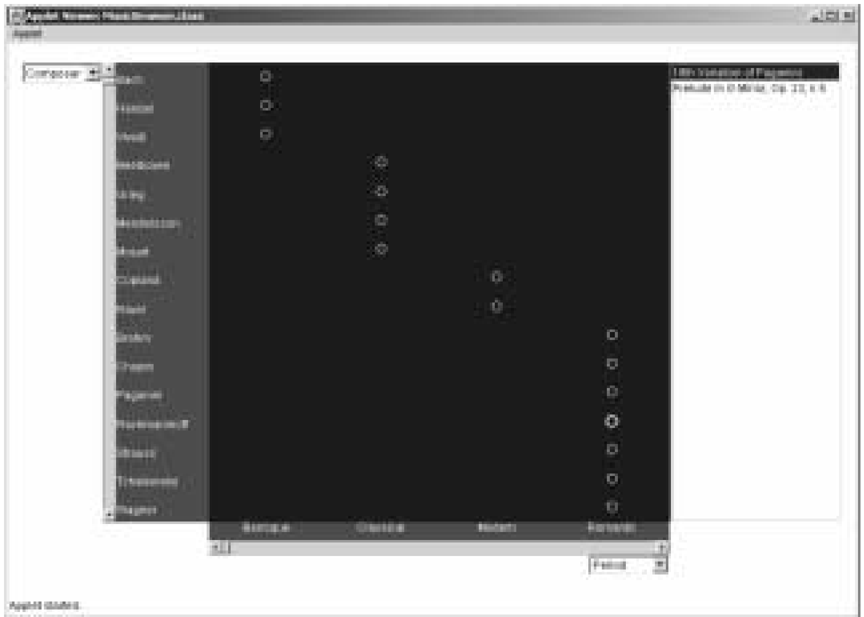

Figure 4.9 – Prototype 3. Period x Composer.

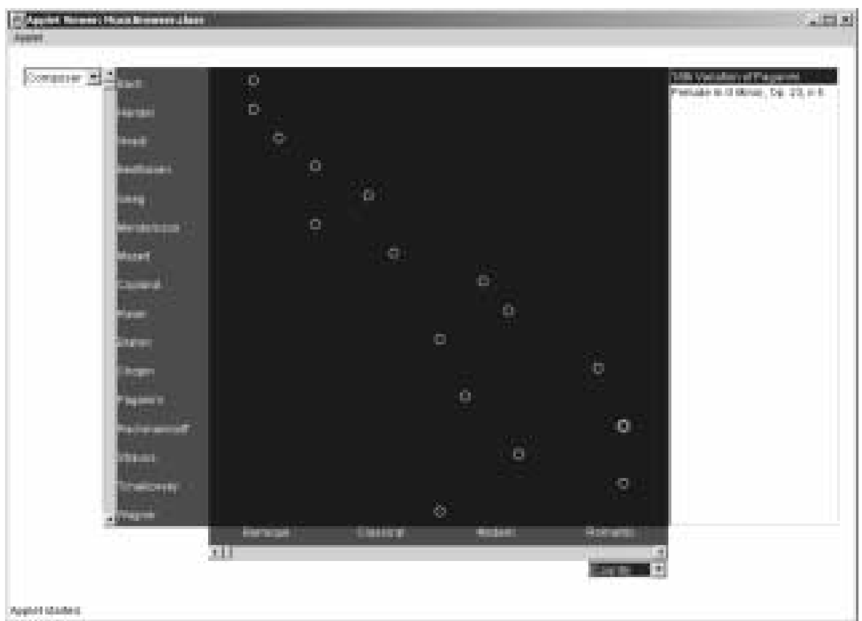

Figure 4.10 – Prototype 3. One animation frame showing Period x Composer transitioning to Country x Composer.

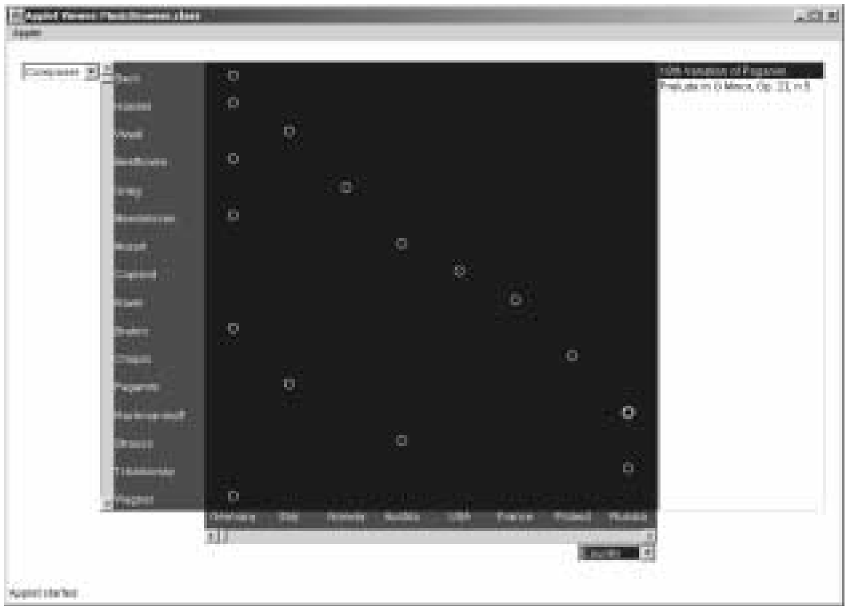

Figure 4.11 – Prototype 3. Country x Composer.

Scale canvases: One horizontal and another vertical contains the mutable scales for the x and y axis. A scroll bar is responsible for zooming, horizontal and vertical individually. A system of combo boxes allows changing dynamically the attributes in the scales. An interesting animated feedback in the output canvas shows the migration from one axis attribute selection to another.

Play list: When a piece or a cluster of piece is selected by simply clicking with the mouse in the circle that represents it, all piece titles associated to that chosen position are displayed in the list.

This approach deals well with the clustering problem, since it is possible to represent many elements (pieces) in the same position. The animation effect showing the element position changes between two successive combo box selections plays an important role in lowering the cognitive load. The limitation of this idea is the lack of hierarchy. Having a hierarchical representation is good for understanding the context. The next approach addresses the hierarchical representation. A new tool for dealing with multiple dimensions is introduced.

4.5 Fourth Prototype – The N-Dimensional Projection Approach

Manually shifting two values of multiple dimensions did not necessarily provide enough context with respect to the other dimensions. For example, information about composer while the axes were "Period" and "Country" wasn't possible.

In addition, Robertson [Rob91b] points out hierarchical concepts, missing in the previous prototype. In order to achieve more distinction among pieces belonging to the same cluster and also represent hierarchies, the idea of implementing virtually any number of dimensions by simply rotating an n-dimensional space and projecting it in a 2D plane came about.

The challenge was then how to manipulate this n-dimensional plane without requiring the user to have extraordinary skills to visualize in more than 2 dimensions and at the same time obtaining a user friendly tool.

The solution was to consider an n-dimensional space plane projected onto a 2D plane. Since our major goal was to have a flexible hierarchical output regardless of its physical meaning, the decision was to consider a non-orthogonal n-dimensional coordinate system. Therefore, any phase – module pair in the 2D plane would correspond to at least one axis real position in the n-dimensional space. A controller, the Phaser tool, would deal only with the projection, disregarding what is actually generating the projection.

4.5.1 The Linear Approach

One output that would still mimic the reality of a geometric projection is the Linear Approach.

Considering a line segment of unit sized in an n-dimensional space, its projection in a two dimensional projection plane can be generically represented by the diagram in figure 4.12.

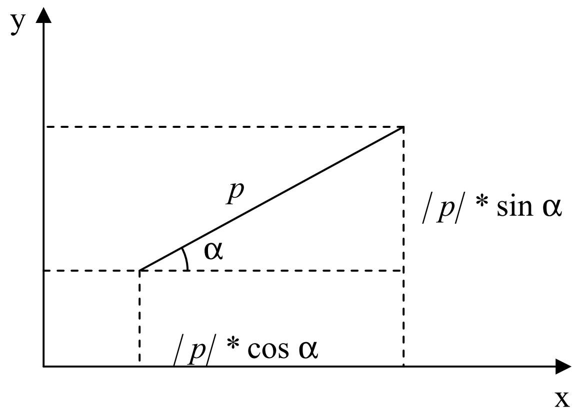

Figure 4.12 – A unit segment projection in the x – y plane. The segment '*p*' you see in the figure is already the projection of a segment from the n-dimensional space.

Assuming that a non-orthogonal n-dimensional system is formed by line segments (axis) connected by their vertices in a common origin, this system can be projected in a two dimensional projection plane and generically represented by the diagram in figure 4.13.

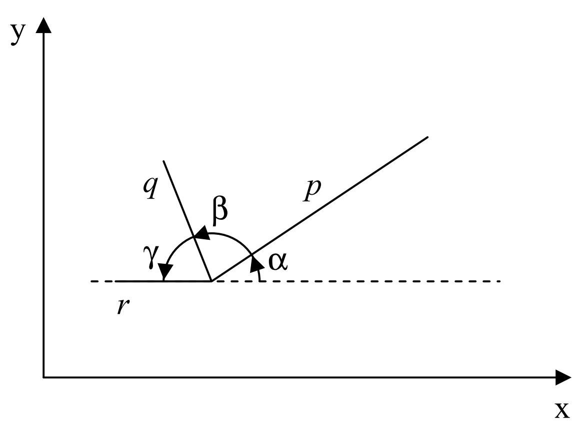

Figure 4.13 – Three segment '*p*', '*q*' and '*r*' projections from the n-dimensional space and their respective phases.

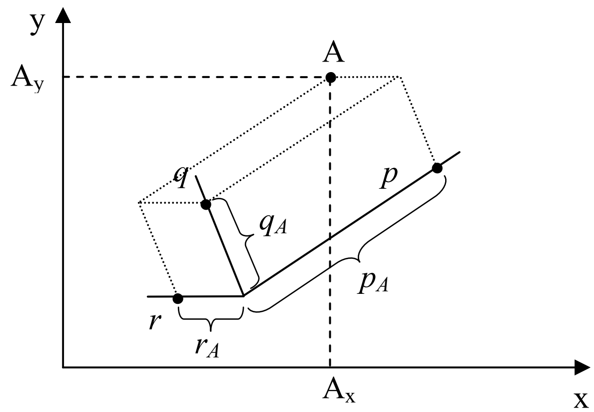

Figure 4.14 – A point 'A' and its coordinates in the coordinated axis '*p*', '*q*' and '*r*'.

Finally, given each point in this system has a coordinate (value) for each attribute (axis), its coordinates in the x – y projection plane are:

$$x = k * \sum_{i=0}^{n} Value_i * Module_i * \cos(Phase_i)$$

$$y = k * \sum_{i=0}^{n} Value_i * Module_i * \sin(Phase_i)$$

Where:

- ***x*** and ***y*** are the coordinated axis in the projection plan
- ***Value$_i$*** is the value of the point with respect to attribute *i*
- ***Module$_i$*** is the module of the projected attribute *i* normalised [0, 1] in the x – y plan, set on the Phaser tool by the user
- ***Phase$_i$*** is the phase α of the projected attribute *i* in the x – y plan
- ***n*** is the number of attributes
- ***k*** is a convenient arbitrary constant for screen positioning

Figure 4.15 – Prototype 4, linear version.

Figure 4.16 – Prototype 4, linear version, with another phase displacement for the attributes.

This is called the linear approach since the element positions are proportional to their values in the respective attribute scale and projected axis modules. No extra mathematical transformation is implemented here. The implementation of this approach is shown in figures 4.15 and 4.16.

In figure 4.15, the user positioned the spike for “Country” vertically down and the spike for “Period” horizontally to the right. As a result, the pieces are projected on the output canvas classified vertically by “Country” and horizontally by Period. Similarly, in figure 4.16, the spikes still have approximately an angle of 90° but inclined about 45°. As a result of this Phaser configuration, the pieces are projected in orthogonal lines rotated by about 45°.

4.5.2 The Orbital Approach

If mimicking the reality of a geometrical projection is not important, we can make use of a more sophisticated mathematical transformation and obtain a useful way of displaying our output.

In the orbital approach, the data is projected in a way that the length of the spike in the Phaser tool becomes the radius of the orbit and the phase of the spike is the phase in the orbit. Therefore, longer spikes correspond to larger orbits.

For the orbital approach, the following transformations were used:

$$x = k * \sum_{i=0}^{n} Module_i * \cos(Phase_i + 2\pi * Value_i)$$

$$y = k * \sum_{i=0}^{n} Module_i * \sin(Phase_i + 2\pi * Value_i)$$

Where the elements from the previous section were simply rearranged but their meanings remain the same. In this approach, phase α of the projected attribute *i* in the x – y plan is added to the value of the point with respect to attribute *i,* which is constrained in the interval [0, 1], given the axis projections are normalized, becoming a contribution value in the interval [0, 2π].

As a result, the projection phase controls the phase in the x – y plane and the projection module controls the "orbital" radius.

The advantage of this approach is a better use of space, since, without making further analysis, all directions around a semantic point are used.

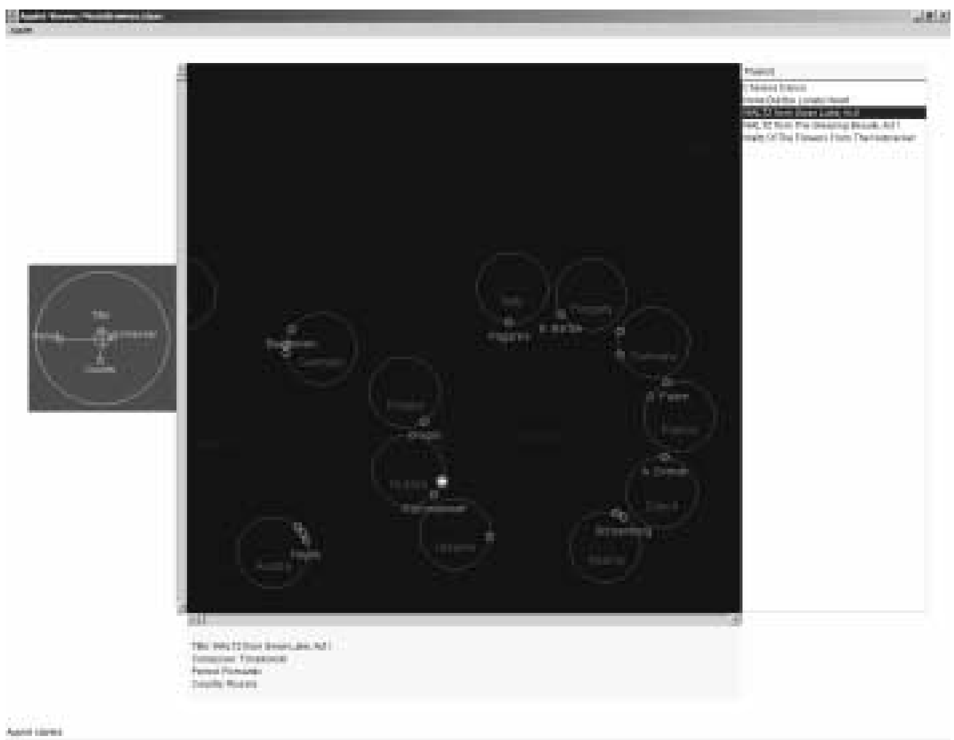

Figure 4.17 – Prototype 4, orbital version.

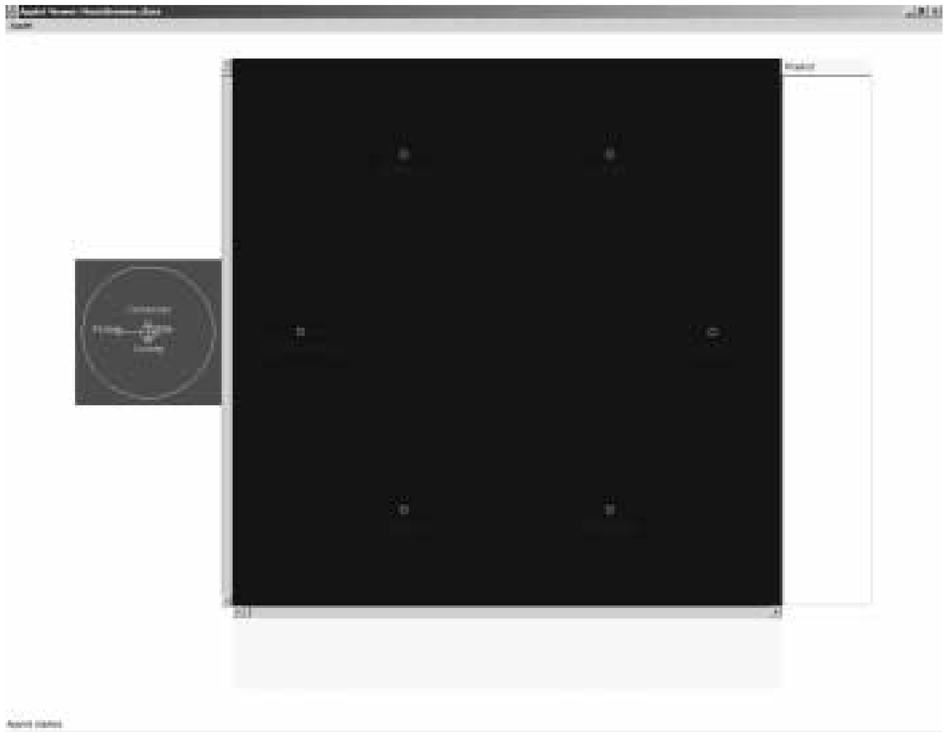

Figure 4.18 – Prototype 4, orbital version, only with the attribute “Period” greater than 0.

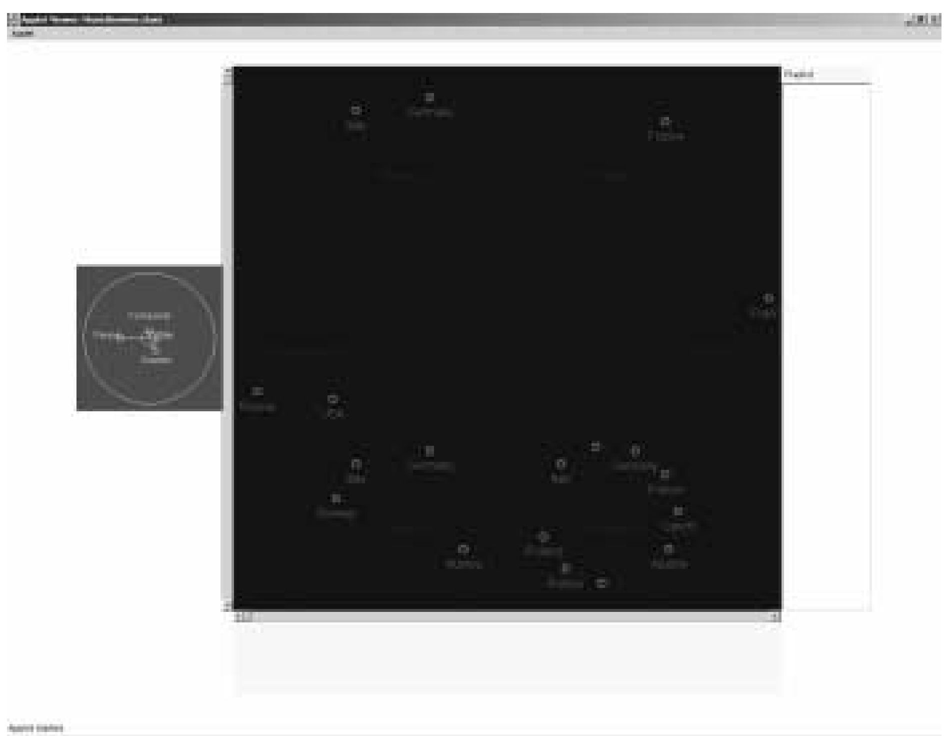

Figure 4.19 – Prototype 4, orbital version, with the attributes “Period” and “Country” greater than 0. “Period” is greater than “Country”.

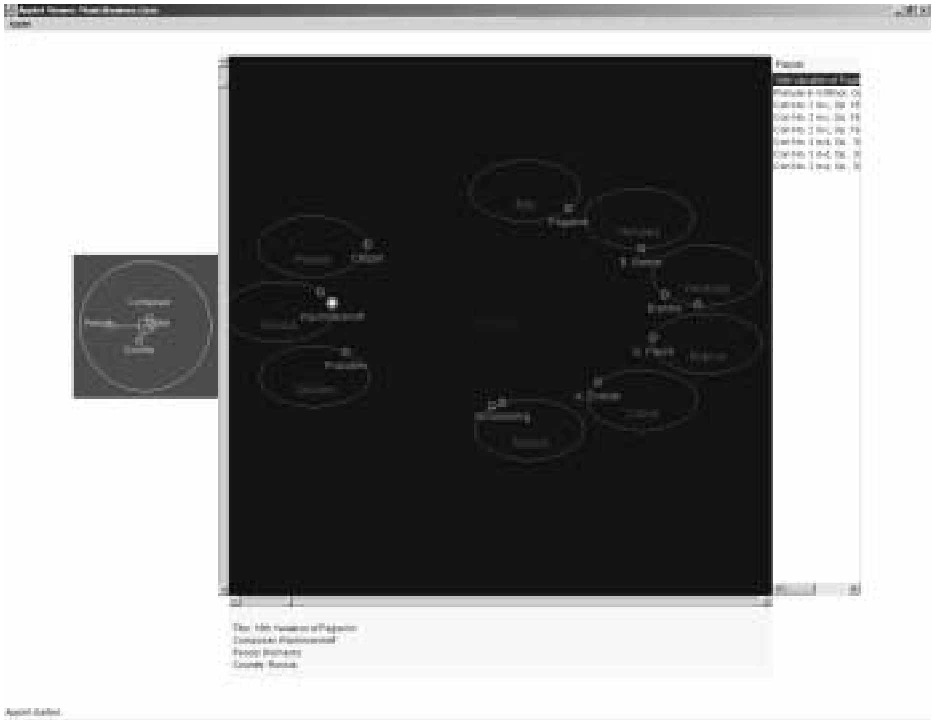

Figure 4.20 – Prototype 4, orbital version, showing the use of independent horizontal and vertical zooming, creating the elliptical shape in the orbit.

4.5.3 The Implementation

Figures 4.17 to 4.20 show the implementation as well as some scenarios of use.

In figure 4.17 we can observe a hierarchy set by the Phaser tool spikes in the order Period → Country → Composer. As a result, the largest orbit corresponds to Period and the smallest to Composer.

In figure 4.18, only the dimension “Period” is greater than 0 and, therefore, we have only one orbit showing the attributes available for “Period”.

Similarly we see in figure 4.19 two dimensions greater than 0, “Period” and “Country” (“Period” greater than “Country”), therefore two orbits, one big, containing clusters of periods, and smaller ones containing countries, one for each period displayed.

Finally, in figure 4.20, we can observe the effect of the zooming, controlled by the horizontal and vertical scroll bars. Note that now the user has set an elliptical shape for the orbits due to different values for the horizontal and vertical zoom.

The interface elements are:

Output canvas: This window displays the pieces and its relative positions among each other. This canvas is also “panable”, i.e. by dragging the mouse in its area the user can translate the position of its contents.

Zooming scroll bars: Responsible for a semantic zooming factor, horizontal and vertical individually. The term semantically, again, is used with restriction, since a true semantic zoom would change the actual amount of information being displayed.

Play list: As in the previous version, when a piece or a cluster of piece is selected, all piece titles associated to that chosen position are displayed in the list.

The Phaser Tool: Described below.

4.5.4 The Phaser Tool

The Phaser tool (fig. 4.21) consists of two concentric circles, limiting the user action and the attribute percentage values from 0% (inner circle) to 100% (outer circle), and legs coming from the center of the circle determine, by their size, the projection value and by their angle (phase), the phase of the affected attribute displayed and combined with other attributes.

Depending on the chosen projection (orbital, linear), the size and phase generated by the phaser tool will manifest in a different way.

After having a design and implementation concluded, we needed to have a reference to evaluate against. The next section describes the chosen approach for this comparison.

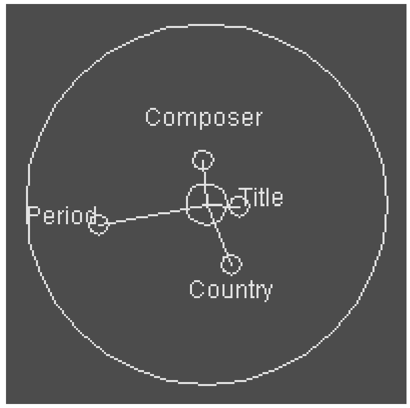

Figure 4.21 – The Phaser Tool

4.6 The Tree-Based Approach

In order to have a reference for the evaluation of the previous prototype, given [Ris00] and [Coc01] a tree-based interface was developed (fig. 4.21). The objective was to investigate this tree-based model as a reference particularly I the case with a non visual (audio) feedback component. Its interface elements are:

Tree canvas: This window displays the pieces in a hierarchal organization, in a tree.

Play list: As in the previous version, when a piece or a cluster of piece is selected, all piece titles associated to that chosen position are displayed in the list.

Phaser Tool: In order to minimize the differences and difficulties in manipulating this tool in the previous prototype, the same Phaser tool was implemented to operate the tree. The only difference is that only the legs length is considered for this approach, not considering the legs angles (phases).

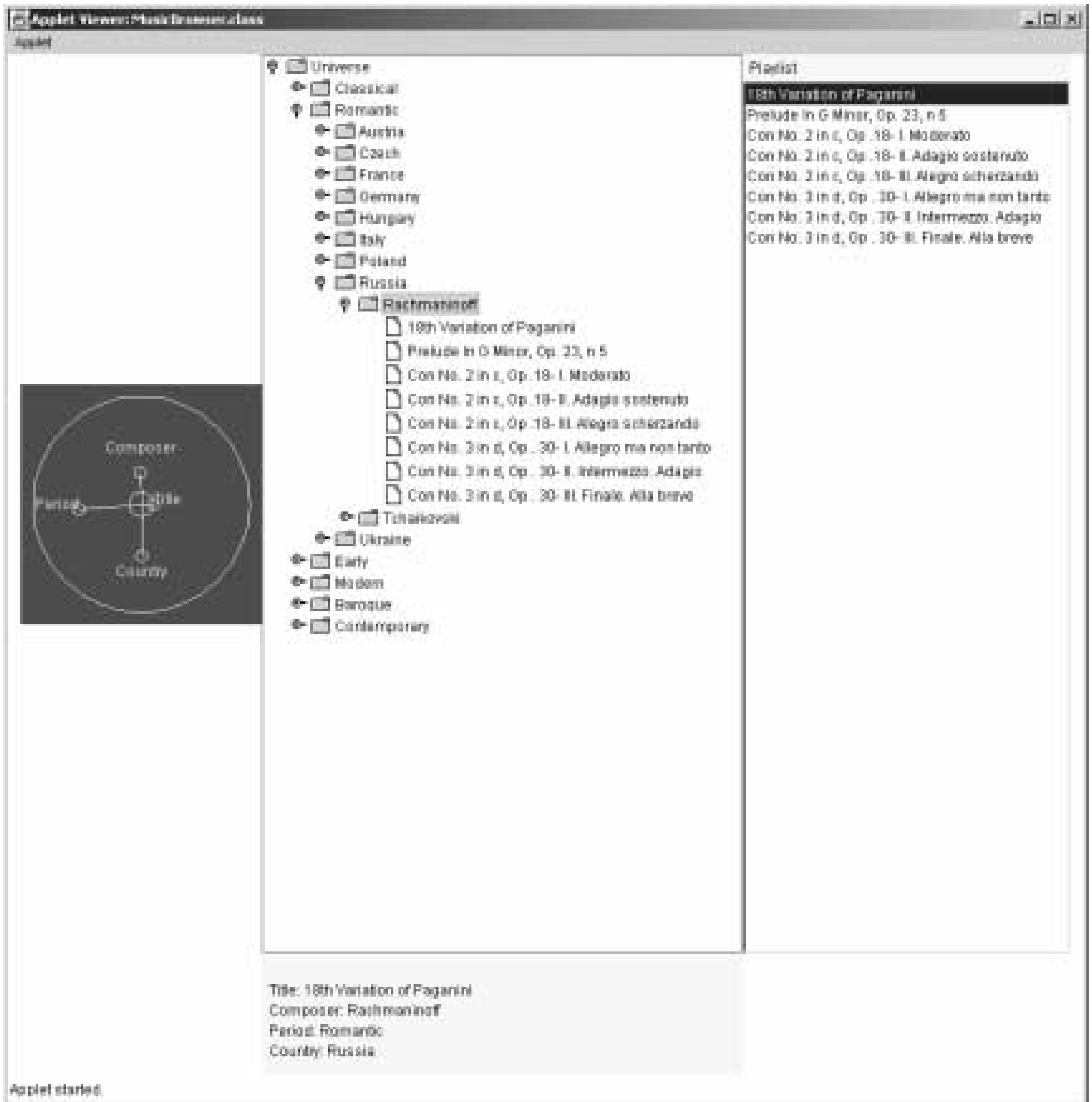

Figure 4.22 – Prototype 4, tree-based.

5. Implementation Details

The prototypes were coded using JDK1.2.2, given the multi-platform feature of Java. The IDE used was IBM VisualAge for Java 3.5.3.

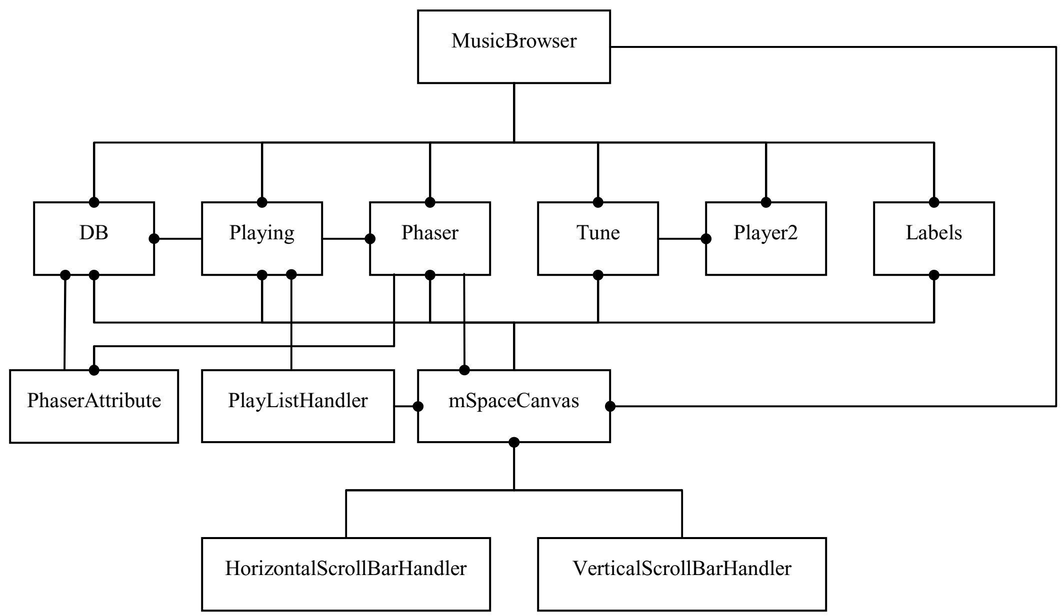

Figure 5.1 – Class diagram. Lines with circle terminations indicate a "has" relation.

Figure 5.1 shows the class diagram for the implementation. The following components can be identified:

- **MusicBrowser** (extends Applet implements KeyListener, MouseListener, MouseMotionListener, WindowListener): Main class. It has seven other classes: DB, Playing, Phaser, Tune, Player2, Labels and mSpaceCanvas.
- **DB**: Responsible for the implementation of a database interface with high level routines to retrieve records.
- **Playing** (extends java.awt.Canvas): Canvas class responsible for displaying music information. It has the classes DB and Phaser.

- **Phaser** (java.awt.Canvas implements java.awt.event.MouseListener, java.awt.event.MouseMotionListener): Another Canvas class responsible for implementing the Phaser tool. It has two other classes: PhaserAttribute and mSpaceCanvas.
- **Tune**: Responsible for storing piece properties. It has a Player2 class.
- **Player2**: Class that deals with mp3 decoding. Adapted from Ben Fransen and Donn Morrison.
- **Labels**: Responsible for drawing textual labels and orbits.
- **PhaserAttribute**: Auxiliary class that stores the Phaser tool state with respect to its attributes.
- **PlayListHandler** (implements java.awt.event.ItemListener): Implements the play list. It has the classes mSpaceCanvas and Playing.
- **mSpaceCanvas** (extends java.awt.Canvas implements java.awt.event.MouseListener, java.awt.event.MouseMotionListener, java.awt.event.KeyListener): Canvas class responsible for the main output window and mouse events. It has seven other classes: DB, Playing, Phaser, Tune, Player2 and Labels.
- **HorizontalScrollBarHandler** and **VerticalScrollBarHandler** (implements java.awt.event.AdjustmentListener): Classes that implement the scroll bar functionalities. They have an mSpaceCanvas instance.

6. The Experiment

An experiment was realized in order to evaluate the effectiveness of finding music with the proposed implemented music browser. The experiment also intended to evaluate how pleasant was the use of the proposed tool. In both cases, under the condition that sound was present as a feedback element, we wanted also to observe how important the GUI is to the user. In order to perform this evaluation, two interfaces were used: one orbital (graphic), another tree (text) based.

6.1 The Subjects

Sixteen subjects participated in this experiment, 8 females and 8 males, with ages ranging from 19 to 31 years old. 3 Computer Science undergraduate students, 12 Computer Science graduate students and 1 non Computer Science graduated.

These subjects reflected the average web music consumer according to simple observation.

6.2 The Design and Procedure

Each subject performed both techniques. Technique order was counterbalanced using a Latin Square research design, considering **order** and **gender.**

Subjects were shown each interface (orbital or tree based). A quick demonstration (2 min) on how the tools work was presented by the evaluator. The task was verbally read to them. There was no practice time, since the time for adaptability was also desired to be taken into account.

The tasks for each interface were to find 5 pieces of classical music the subjects were familiar with and they like. The next task was to find 1 single piece of classical music they were not familiar with and they happen to like. The total time given for them on each interface for all two tasks was 10 minutes. The tasks were verbally prompted and users were asked to press a key on the keyboard to acknowledge the finding.

A post questionnaire was administered right after the experiment and the results were analyzed. They are discussed in the next session.

6.3 Qualitative Results

The questions in the following subsections were done on a Likert Scale; therefore all values for average and standard deviation belong to the interval [0, 5].

6.3.1 Question 1: Which version you prefer (tree based or orbital based)?

Assigning 1 to Orbital and -1 to the Tree version, the result obtained was 0.13 for average and 1.03 for STD.

Order was not significant (F = 0.46; p = 0.5034) but sex somehow was (F = 4.15; p = 0.0527), giving a mean of 0.5000 for males (in favor of orbital) and - 0.2500 for females (in favor of trees). These results were not enough to assign different t groupings in the t test performed.

6.3.2 Question 2: Do you like the way the graphics look?

None of the independent variables significantly contributed for differences in the means between the two interfaces. Figure 6.1 shows the average and STD for each interface.

Interface	Average	STD
Orbital	3.50	1.03
Tree	3.75	0.77
Difference	-0.25	1.29

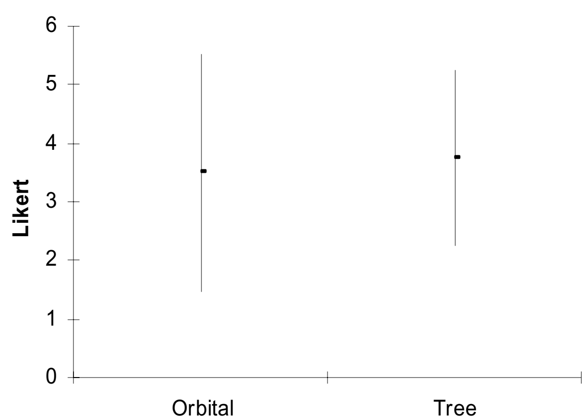

Figure 6.1 – Effect of each interface on the Likert scale for question 2 (95% confidence interval).

6.3.3 Question 3: Do you think this application is simple to use?

None of the independent variables significantly contributed for differences in the means between the two interfaces.

Sex was the most significant factor evaluating the two interfaces (F = 3.09; p = 0.0917), but still not enough to consider. Figure 6.2 shows the average and STD for each interface.

Interface	Average	STD
Orbital	3.63	1.15
Tree	4.38	0.81
Difference	-0.75	1.18

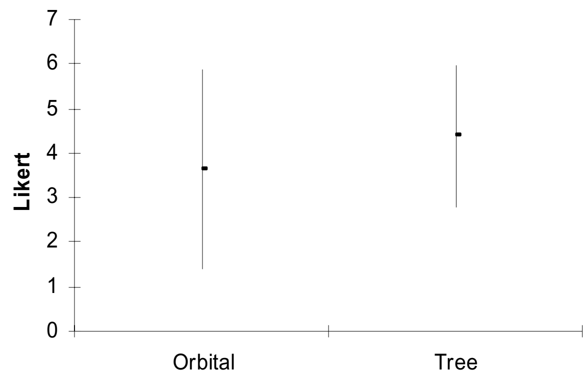

Figure 6.2 – Effect of each interface on the Likert scale for question 3 (95% confidence interval).

6.3.4 Question 4: Would you consider using this application when made available?

None of the independent variables significantly contributed for differences in the means between the two interfaces.

Interestingly, sex again was the most significant factor (F = 5.45; p = 0.0282), but now in favor of the Orbital approach. Figure 6.3 shows the average and STD for each interface.

Interface	Average	STD
Orbital	3.69	1.40
Tree	3.94	0.85
Difference	-0.25	1.53

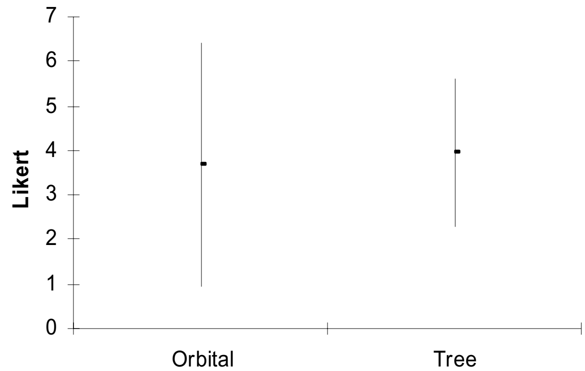

Figure 6.3 – Effect of each interface on the Likert scale for question 4 (95% confidence interval).

6.3.5 Question 5: Was this application enjoyable to use?

Sex was significant to create two different groups according to the t test. For the differences in a range [-5, 5], male mean: 0.8750; female mean: -0.5000.

On average, subjects felt the Orbital application more enjoyable to use.

These results show that there is a significant tendency ($F = 5.90$; $p = 0.0230$) of females to feel more enjoyable a tree based application.

Figure 6.4 shows the average and STD for each interface

Interface	Average	STD
Orbital	4.06	1.12
Tree	3.88	0.89
Difference	0.19	1.60

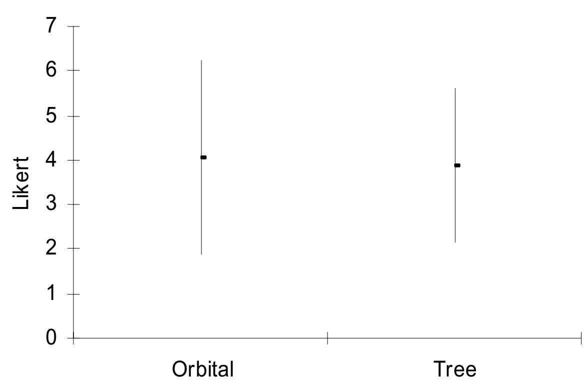

Figure 6.4 – Effect of each interface on the Likert scale for question 5 (95% confidence interval).

6.3.6 Question 6: Did you feel disoriented during the experiment?

Females, not significantly, felt it more disorienting than males.

Subjects in general reported significantly the Orbital tool to be more disorienting (F = 6.58; p = 0.0170).

Figure 6.5 shows the average and STD for each interface.

Interface	Average	STD
Orbital	3.94	1.12
Tree	4.75	0.45
Difference	-0.81	1.33

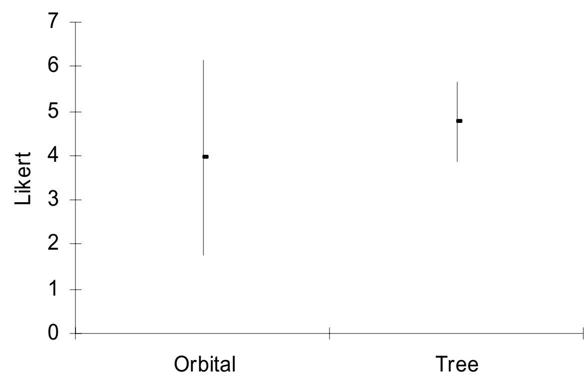

Figure 6.5 – Effect of each interface on the Likert scale for question 6 (95% confidence interval).

6.3.7 Summary

According to our results, overall users reported the orbital – spatially arranged information to be more enjoyable but more disorienting than the tree-based approach. Female subjects considered significantly more enjoyable the tree-based tool.

6.4 Quantitative Results

The result on efficiency was measured according to the time taken to locate pieces.

None of the independent variables significantly contributed for differences in the means between the two interfaces.

Figure 6.6 shows the average and STD for each interface.

Interface	Average (s)	STD
Orbital	2.34	2.14
Tree	1.82	1.70

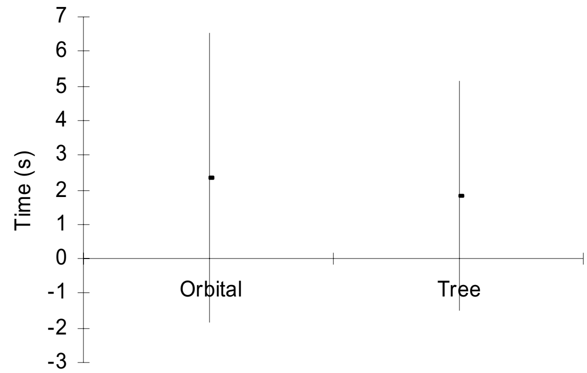

Figure 6.6 – Effect of each interface on time spent (95% confidence interval).

6.5 Observations

Some observations during the execution of the tasks were made during the evaluation and they are presented below:

- Only one or two dimensions were considered at a time in the orbital version. The next step performed by the user after expanding these one or two dimensions was using the play list. This was not true for the tree list, even when an expansion was not necessary. For example when the final dimension was "Title", the play list would display this information.
- In the orbital version, users got confused when the orbits were overlapping. Manipulating and controlling the Phaser tool and the subsequent output was an issue, especially when many orbit levels were set in the hierarchy.
- The Phaser tool was more often used than panning and zooming.
- The use of the orbital version was very captivating for many users.

6.6 Discussion

The orbital version was somehow discouraging the users to expand the hierarchy in more than two levels. The "escape" alternative was the use of the play list "as soon as possible”. This was possibly caused by the initial inexperience on manipulating the tool and understanding clearly how the orbits work.

The Phaser tool is at least captivating, since users were spending more time on it than adjusting the output canvas panning and zooming. It seems that there (in the Phaser tool) is where more effort for improvement must be spent.

The fact that females demonstrated different preferences for versions compared to the male subjects must be analyzed with caution. The cause may be either physiological, intrinsic to a male/female physiological difference or just a socio-cultural way in which males and females are exposed to different experiences.

In addition, some mechanisms should be implemented in order to avoid orbits to overlap, as a great source of confusion to the users.

The user ability to manipulate the Orientation of the information space has demonstrated to be a powerful resource for an application not only to provide focus + context in fixed hierarchies, but to allow the whole hierarchy to be dynamic, *orientable* according to the user convenience.

7. Future Work

Some automatisms could be introduced and considerably improve the usability of the orbital version, such as double clicking to expand one level in the hierarchy and an automatic orbit overlapping detection.

In order to make the output smoother, since it is a magnification of the movements performed in the Phaser tool, a system for animation with frame interpolation could be implemented.

In addition, in order to combine the two enhancements mentioned in the previous two paragraphs, once orbits reach each other an animation for the transition between the two different dominant attributes could take place.

Another aspect for a future implementation is how the clusters should be displayed. At this point, their positions are simply proportional to their positions in the database. An alternative would be spreading them evenly in their orbits, at a cost of loosing their semantic position. For example, in the orbit "Composer", Beethoven is always at the same angle in any orbit at the same time.

In order to verify weather or not the two tested interfaces were being influenced by the audio feedback, another set of pure visual experiments could take place, since, according to our results they seem to be more or less equivalent in preference and efficiency.

Finally an investigation of a pure non-visual application maybe even with bi-manual input could take place in order to develop an application using movement detectors and loudspeakers only.

8. Conclusions

Our literature review (Section 2), described a few studies comparing 2D applications to 3D applications. We found that a more complex 2D application, with more cognitive load (orbits, compared to lists) may already be disorienting to the user. Therefore, recommend future work to focus still on 2D applications. Another aspect mentioned in that section was the use of animation to low the cognitive load and improve the use of perception. We not only implemented animation, but real time animation feedback, which was reported very interesting and pleasant to the users. In addition, the attempt of suppressing queries from the application was positive. We ran experiments with naïve users (as defined in section 3) in classical music and the users were able to locate pieces of music without knowledge of composers or titles, which highlights the strength of our technique.

There is little research that scientifically compares spatially arranged to listed information. We ran an experiment that compares the efficiency and degree of pleasure between a GUI and a hierarchical text tree based interface (section 5). According to our results, overall, users reported the orbital – spatially arranged information to be more enjoyable but more disorienting than the tree-based approach. Female subjects considered significantly more enjoyable the tree-based tool.

Our Phaser tool revealed to be a captivating, interesting and effective way of manipulation an n-dimensional space. Since we are note concerned with spatial projection accuracy but only with the qualitative aspects of it, simplified projection formulas, non orthogonal space and even mathematical transformations (linear, orbital, spiral, logarithmical) can be used, enhancing the flexibility of the information manipulation.

As a final thought, this work has contributed to the scientific community by introducing a new concept: Orientation. The user, exploring high-dimensional information spaces, manipulates the Orientation of the information space, in contrast to simple Focus + Context tools, where the Orientation is fixed, in order to *orient* the data to privilege a desired hierarchy.

9. Bibliography

[Ahl94] Ahlberg C., Shneiderman B., “Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays”. Proceedings of the CHI '94 conference companion on Human factors in computing systems, p. 222. April, 1994.

[Ame00] Amento, B. et ali. “TopicShop: enhanced support for evaluating and organizing collections of Web sites”. Proceedings of the 13th annual ACM symposium on User interface software and technology, pp. 201-9. November, 2000.

[Bed94] Bederson and Hollan. “Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics”. Proceedings of the 7th annual ACM symposium on User interface software and technology, pp 17-26. November, 1994.

[Bur96] Burkhart, B. “Using Animation to Aid Process Flow Visualization”. Proceedings of the CHI '96 conference companion on Human factors in computing systems, pp. 21-22. April, 1996.

[Car96] Card, S. et ali. “Readings in Information Visualization: Using Vision to Think”. Academic Press, p. 7. 1999.

[Cha96] Chalmers, M. “A Linear Iteration Time Layout Algorithm for Visualizing High-Dimensional Data”. Proceedings of the conference on Visualization '96, pp. 127-132. October, 1996.

[Coc01a] Cockburn, A., McKenzie, B. “3D or not 3D? Evaluating the Effect of the Third dimension in a Document Management System”. Proceedings of the SIGCHI conference on Human factors in computing systems, pp 434-41. March, 2001.

[Coc01b] Cockburn, A. & McKenzie, B., “An evaluation of cone trees,” *In People and Computers X*V. *Proceedings of the 2000 British Computer Society Conference on Human-Computer Interactio*n, University of Sunderland, pp. 4-8. September, 2000.

[Fur95] Furnas, G. and Bederson, B. “Space-Scale Diagrams: Understanding Multi-Scale Interfaces”. Conference proceedings on Human factors in computing systems, pp. 234-241. May, 1995.

[Kum95] Kumar, H. et ali. “Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning”. University of Maryland, Dept. of Computer Science Technical Report, CS-TR-3474, 1995.

[Mac91] Mackinlay, J., G. Robertson, et al. “The Perspective Wall: Detail and Context Smoothly Integrated.” Human factors in computing systems conference proceedings on Reaching through technology, pp. 173-6. March, 1991.

[Mer96] Mereu, S., Kazman, R. “Audio enhanced 3D interfaces for visually impaired users”, Conference proceedings on Human factors in computing systems, pp 72-8. April, 1996.

[Per93] Perlin, K. and Fox, D. “Pad - An Alternative Approach to the Computer Interface”. Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp. 57-64. September, 1993.

[Ren94] Rennison, E. “Galaxy of News”. Proceedings of the 7th annual ACM symposium on User interface software and technology, pp. 3-12. November, 1994.

[Ret94] Rettig M., "Prototyping for Tiny Fingers", Communications of the ACM, Volume 37, Number 4, pages 21-27. April 1994.

[Ris00] Risden, K. et ali. “An Initial Examination of Ease of Use for 2D and 3D Information Visualizations of Web Content”. *Int. J. Human-Computer Studies, 53,* 695-714. 2000.

[Rob91a] Robertson, G. et ali. “Information visualization using 3D interactive animation”. Human factors in computing systems conference proceedings on Reaching through technology, pp. 461-2. March, 1991.

[Rob91b] Robertson, G., et ali. “Cone trees: animated 3D visualizations of hierarchical information”. Human factors in computing systems conference proceedings on Reaching through technology. pp. 189-94. March, 1991.

[Rod01] Rodden, K., et ali. “Does Organisation by Similarity Assist Image Browsing?” Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 190-7. March, 2001.

[Seb99] Sebrechts, M. et ali. “Visualisation of Search Results: A Comparative Evaluation of Text, 2D and 3D Interfaces”. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp 3-10 August, 1999.

[Spe01] Spence, R. “Information Visualisation”. ACM Press., p. 77. 2001.

[Sut00] Sutcliffe, A. et ali. “Evaluating the effectiveness of visual user interfaces for information retrieval”. International Journal of Human Computer Studies, Special Issue on Empirical evaluation of information visualizations, Vol. 53, No. 5, November, 2000

[Twe95] Tweedie, L. “InteractiveVisualisation Artifacts: How can abstractions inform design?” People and Computers X, Proceedings of HCI ’95, pp. 247-265. 1995.

[Wat91] Waterworth, J. and Chignell, M. “A Model for Information Exploration”. Hypermedia, 3(1):35-58, 1991.

[Wil92] Williamson, C. and Shneiderman, B. “The Dynamic Homefinder: evaluating dynamic queries in a real estate information exploration system”, Proceedings of the Fifteenth Annual International ACM SIGIR conference on Research and development in information retrieval, pp. 338-346. June, 1992.

APENDIX A – Protocol for Lo-Fi Evaluation

Music lines LoFi Evaluation Protocol

A.1 Preparation

Thank you for your participation in this evaluation!

We are interested in testing the first version of a low-fidelity prototype of a system that allows users to navigate, explore, discover and play pieces from a set of classical music.

By building such a system we hope to assist users on the discovery/learning about a field they are not familiar with by means of exploration, association and information filtering.

The system's screen is divided into the following sections:

- The **piece's information browser** window, where the information regarding the piece being played
- The **player** window, which provides basic controls to pause/resume/rewind/play the piece currently selected/being played
- The **music lines** window, a canvas that contains horizontal lines showing pieces

The system works with a database of classical pieces, which are initially displayed in the "All Tunes" line. From there users can:

- Play pieces
- Tag/annotate pieces
- View information about the piece being played
- Filter songs according to a similar attribute (composer, period, rhythm, instrument)
- Filter songs according to an arbitrary choice of attribute (composer, period, rhythm, instrument)

A.2 Evaluation

All your personal information will be maintained strictly confidential

For analysis purposes we will tape this session, if you do not agree, please tell us so

You can withdraw from the evaluation at any time you'd like

Imagine that the interface appears on a generic screen and that you have at your disposal a pointing device to select/activate elements on the screen and a keyboard to type text when necessary.

When you start the system the initial screen shows you in the main line all the pieces available in the database and start playing a piece (selected @ random) from that line

We will ask you to perform some of the representatives tasks, please try to "think aloud" as you interact with the interface.

A.2.1 Task 0

You wish to find a piece that you like. What would you first try with the interface presented?

A.2.2 Task 1

You are listening to a piece on the "main" line that you happen to like. Now you want generate a new line that contains pieces with the same composer, so you can explore/listen to that set. What would you do?

A.2.3 Task 2

You want to listen to pieces with a slow rhythm and string in them. How would you do that using the interface?

A.2.4 Task 3

As you performed tasks 1 & 2 you found pieces you would like to go listen to again. try to repeat what you did, but now before going to the next song, tag the ones you like. How would you do it?

A.3 Remarks

- Does the interface remind you of any other devices or interfaces you've used? Which ones?
- Did you find the interface easy to use? Did you find the interface easy to use to learn?

APENDIX B – Post Questionnaire

Name: ____

Age: _______

Education: ____

Gender: F M

Please answer objectively these questions.

1. **What was the <u>best</u> part of the tool in each modality?**

2. **What was the <u>weakest</u> aspect of the tool?**

3. **Are there any features you would like to add?**

4. **What context could you imagine using the tool in?**

5. **What other observations do you have about using this tool?**

6. **In its current form, is it a tool you would like to have available?**

7. **Which version you prefer (tree based or orbital based)?**

8. **Did you learn anything about the domain that you were not aware of?**

9. **Do you like the way the graphics look?**

Orbital

Not At All Very Much

1 2 3 4 5

Tree

Not At All Very Much

1 2 3 4 5

10. Do you think this application is simple to use?

Orbital

Very Hard Very Simple

1 2 3 4 5

Tree

Very Hard Very Simple

1 2 3 4 5

11. Would you consider using this application when made available?

Orbital

Not At All Definitely

1 2 3 4 5

Tree

Not At All Definitely

1 2 3 4 5

12. Was this application enjoyable to use?

Orbital

Not At All Very Much

1 2 3 4 5

Tree

Not At All Very Much

1 2 3 4 5

13. Did you feel disoriented during the experiment?

Orbital

Completely lost Aware about the location

1 2 3 4 5

Tree

Completely lost Aware about the location

1 2 3 4 5

9.1 APENDIX C – Results for the Pilot Study on Web Music Browsing

1st Subject (14:00)

1st Task (3:30)

Using Morpheus, he searched for Placebo, in the Artist field and then for the title “Every You, Every Me”. (2:00).

The subject had to spend sometime to remember about another piece of music. He found the title “Dr. Strangelove”. (1:30)

2nd Task (10:30)

The limiting factor for this task was the excessive time to download songs.

Still using Morpheus, the subject started downloading “So Danco Samba”.

The subject tried then browsing another category (“House”), and finally decided going for another application (CDNow). According to his comment, it is not possible to see an artist list or an album list using Morpheus.

In CDNow it is possible to listen to the first 20 seconds of music, which is not enough for a piece of music evaluation, sometimes, according to the subject.

2nd Subject (13:40)

1st Task (6:40)

The subject searched for “tindersticks”, clicked on the first album and clicked on the track, using Morpheus. (2:25)

In a second attempt, tried to find something familiar, but not he was not certain about album. The artist name was Cesaria Evora. There were about five attempts to locate what he was looking for before giving up (4:15).

Another attempt was made again, without success. (?)

<u>2nd Task</u> (7:00)

A set of attempts was made searching for the keywords Electronic and Dance. The artist Sven Vath was located, but nothing was considered from the first album. (3:24)

After other searches, the albums Essentials, Ambient and Zero 7 were located. (1:46)

Finally, the album Moheeba was found (1:50)

In total, 18 pages/links were visited while using Morpheus and 38 pages/links were visited while using CDNow.

As general comments, CDNow is more organized, but listening is not always possible. Good music was located in the album “Moheeba”. Having access to the whole context is also missing in this tool. Customization is needed as well. Morpheus is more query based and downloading time using this tool is very slow.

3rd Subject (14:00)

<u>1st Task</u> (2:00)

Initially, 4 pages were navigated using CDNow. (2:00)

<u>2nd Task</u> (12:00)

Jazz was selected but the subject wasn’t happy with the clustering (no clustering). The next search was for the artist “Sten Getz”.

Google was then chosen in order to find out a song name he had a memory fragment about. The result of this process was the song title “Hymn of the Orient”. Using the feature “Related Artists” on the “Massive Attack” group, another album was located. (12:00)

17 pages/links were visited.

The subject learned the definition of “Trip-Hop” and related performances of previously known songs.

With respect to CDNow, complete listings would be important to reinforce memory fragments.

APENDIX C – ANOVA Tables

The independent variables considered for the experiment were *sex* (F, M), *order* (TO – OT) and *interface* (O, T). O: Orbital, T: Tree.

The variables *preference*, *graphics*, *simple*, *consider*, *enjoyable* and *disoriented* were the dependent variables, corresponding, respectively to the questions 7, 9-13. The dependent variable *time* represents the time the user spent to find the first piece. The duplicated variables with a *D* appended are the represent the difference between the value for Orbital and Tree. Therefore, negative values represent a greater value for the Tree than for the Orbital (table C.1).

Table C.1 – Base data for the ANOVA tables

Independent Variables			Dependent Variables												
Sex	order	interface	preference	graphics	graphicsD	simple	simpleD	consider	considerD	enjoyable	enjoyableD	disoriented	disorientedD	time(min)
M	TO	O	T	3	-1	4	-1	4	0	4	1	3	-2	2.70
F	TO	O	T	2	-2	3	-2	3	-1	3	-2	3	-2	1.52
F	TO	O	T	4	-1	5	0	4	-1	4	-1	5	0	1.37
M	TO	O	O	3	0	4	-1	4	0	5	1	5	0	8.15
F	OT	O	T	3	-1	1	-3	2	-2	4	0	2	-3	
M	OT	O	O	4	2	3	-2	4	1	5	2	5	1	0.70
M	OT	O	O	4	1	4	1	5	2	5	2	4	0	1.48
F	OT	O	T	1	-3	2	-3	1	-4	1	-4	2	-3	3.08
F	OT	O	O	5	1	4	0	5	1	5	1	5	1	2.08
M	OT	O	O	4	0	5	0	5	1	4	0	5	0	1.78
F	TO	O	T	3	0	3	-1	3	-1	3	0	3	-2	2.73
M	TO	O	O	4	1	3	0	3	1	4	2	3	-2	1.60
F	TO	O	O	5	1	4	0	5	1	5	1	4	0	0.75
M	TO	O	O	4	-1	5	0	5	0	5	0	5	0	2.25
F	OT	O	O	4	0	5	0	5	0	5	1	5	0	2.58
M	OT	O	T	3	-1	3	0	1	-2	3	-1	4	-1	2.63
M	TO	T	T	4	-1	5	-1	4	0	3	1	5	-2	
F	TO	T	T	4	-2	5	-2	4	-1	5	-2	5	-2	1.33
F	TO	T	T	5	-1	5	0	5	-1	5	-1	5	0	0.72
M	TO	T	O	3	0	5	-1	4	0	4	1	5	0	6.98
F	OT	T	T	4	-1	4	-3	4	-2	4	0	5	-3	
M	OT	T	O	2	2	5	-2	3	1	3	2	4	1	1.08
M	OT	T	O	3	1	3	1	3	2	3	2	4	0	1.63
F	OT	T	T	4	-3	5	-3	5	-4	5	-4	5	-3	1.37
F	OT	T	O	4	1	4	0	4	1	4	1	4	1	1.45
M	OT	T	O	4	0	5	0	4	1	4	0	5	0	0.90
F	TO	T	T	3	0	4	-1	4	-1	3	0	5	-2	0.50
M	TO	T	O	3	1	3	0	2	1	2	2	5	-2	3.93
F	TO	T	O	4	1	4	0	4	1	4	1	4	0	3.08
M	TO	T	O	5	-1	5	0	5	0	5	0	5	0	2.67
F	OT	T	O	4	0	5	0	5	0	4	1	5	0	1.70
M	OT	T	T	4	-1	3	0	3	-2	4	-1	5	-1	

Dependent Variable: preference

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	5.50000000	0.78571429	0.73	0.6522
Error	24	26.00000000	1.08333333		
Corrected Total	31	31.50000000			

R-Square	Coeff Var	Root MSE	preference Mean
0.174603	832.6664	1.040833	0.125000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	4.50000000	4.50000000	4.15	0.0527
order	1	0.50000000	0.50000000	0.46	0.5034
sex*order	1	0.50000000	0.50000000	0.46	0.5034
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	4.50000000	4.50000000	4.15	0.0527
order	1	0.50000000	0.50000000	0.46	0.5034
sex*order	1	0.50000000	0.50000000	0.46	0.5034
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Dependent Variable: graphics

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	2.50000000	0.35714286	0.37	0.9092
Error	24	23.00000000	0.95833333		
Corrected Total	31	25.50000000			

R-Square	Coeff Var	Root MSE	graphics Mean
0.098039	27.00538	0.978945	3.625000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	0.12500000	0.12500000	0.13	0.7211
order	1	0.12500000	0.12500000	0.13	0.7211
sex*order	1	0.00000000	0.00000000	0.00	1.0000
interface	1	0.50000000	0.50000000	0.52	0.4771
sex*interface	1	1.12500000	1.12500000	1.17	0.2894
order*interface	1	0.12500000	0.12500000	0.13	0.7211
sex*order*interface	1	0.50000000	0.50000000	0.52	0.4771

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	0.12500000	0.12500000	0.13	0.7211
order	1	0.12500000	0.12500000	0.13	0.7211
sex*order	1	0.00000000	0.00000000	0.00	1.0000
interface	1	0.50000000	0.50000000	0.52	0.4771
sex*interface	1	1.12500000	1.12500000	1.17	0.2894
order*interface	1	0.12500000	0.12500000	0.13	0.7211
sex*order*interface	1	0.50000000	0.50000000	0.52	0.4771

Dependent Variable: graphicsD

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	7.00000000	1.00000000	0.56	0.7819
Error	24	43.00000000	1.79166667		
Corrected Total	31	50.00000000			

R-Square	Coeff Var	Root MSE	graphicsD Mean
0.140000	-535.4126	1.338532	-0.250000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	4.50000000	4.50000000	2.51	0.1261
order	1	0.50000000	0.50000000	0.28	0.6022
sex*order	1	2.00000000	2.00000000	1.12	0.3012
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	4.50000000	4.50000000	2.51	0.1261
order	1	0.50000000	0.50000000	0.28	0.6022
sex*order	1	2.00000000	2.00000000	1.12	0.3012
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Dependent Variable: simple

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	7.50000000	1.07142857	0.97	0.4746
Error	24	26.50000000	1.10416667		
Corrected Total	31	34.00000000			

R-Square	Coeff Var	Root MSE	simple Mean
0.220588	26.26983	1.050793	4.000000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	0.12500000	0.12500000	0.11	0.7394
order	1	1.12500000	1.12500000	1.02	0.3229
sex*order	1	0.00000000	0.00000000	0.00	1.0000
interface	1	4.50000000	4.50000000	4.08	0.0548
sex*interface	1	1.12500000	1.12500000	1.02	0.3229
order*interface	1	0.12500000	0.12500000	0.11	0.7394
sex*order*interface	1	0.50000000	0.50000000	0.45	0.5074

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	0.12500000	0.12500000	0.11	0.7394
order	1	1.12500000	1.12500000	1.02	0.3229
sex*order	1	0.00000000	0.00000000	0.00	1.0000
interface	1	4.50000000	4.50000000	4.08	0.0548
sex*interface	1	1.12500000	1.12500000	1.02	0.3229
order*interface	1	0.12500000	0.12500000	0.11	0.7394
sex*order*interface	1	0.50000000	0.50000000	0.45	0.5074

Dependent Variable: simpleD

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	7.00000000	1.00000000	0.69	0.6829
Error	24	35.00000000	1.45833333		
Corrected Total	31	42.00000000			

R-Square	Coeff Var	Root MSE	simpleD Mean
0.166667	-161.0153	1.207615	-0.750000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	4.50000000	4.50000000	3.09	0.0917
order	1	0.50000000	0.50000000	0.34	0.5637
sex*order	1	2.00000000	2.00000000	1.37	0.2531
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	4.50000000	4.50000000	3.09	0.0917
order	1	0.50000000	0.50000000	0.34	0.5637
sex*order	1	2.00000000	2.00000000	1.37	0.2531
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Dependent Variable: consider

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	5.37500000	0.76785714	0.52	0.8112
Error	24	35.50000000	1.47916667		
Corrected Total	31	40.87500000			

R-Square	Coeff Var	Root MSE	consider Mean
0.131498	31.90059	1.216210	3.812500

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	0.50000000	0.50000000	0.34	0.5664
order	1	0.50000000	0.50000000	0.34	0.5664
sex*order	1	0.12500000	0.12500000	0.08	0.7738
interface	1	0.50000000	0.50000000	0.34	0.5664
sex*interface	1	3.12500000	3.12500000	2.11	0.1590
order*interface	1	0.12500000	0.12500000	0.08	0.7738
sex*order*interface	1	0.50000000	0.50000000	0.34	0.5664

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	0.50000000	0.50000000	0.34	0.5664
order	1	0.50000000	0.50000000	0.34	0.5664
sex*order	1	0.12500000	0.12500000	0.08	0.7738
interface	1	0.50000000	0.50000000	0.34	0.5664
sex*interface	1	3.12500000	3.12500000	2.11	0.1590
order*interface	1	0.12500000	0.12500000	0.08	0.7738
sex*order*interface	1	0.50000000	0.50000000	0.34	0.5664

Dependent Variable: considerD

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	15.00000000	2.14285714	0.94	0.4982
Error	24	55.00000000	2.29166667		
Corrected Total	31	70.00000000			

R-Square	Coeff Var	Root MSE	considerD Mean
0.214286	-605.5301	1.513825	-0.250000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	12.50000000	12.50000000	5.45	0.0282
order	1	0.50000000	0.50000000	0.22	0.6446
sex*order	1	2.00000000	2.00000000	0.87	0.3595
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	12.50000000	12.50000000	5.45	0.0282
order	1	0.50000000	0.50000000	0.22	0.6446
sex*order	1	2.00000000	2.00000000	0.87	0.3595
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Dependent Variable: enjoyable

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	4.21875000	0.60267857	0.54	0.7951
Error	24	26.75000000	1.11458333		
Corrected Total	31	30.96875000			

R-Square	Coeff Var	Root MSE	enjoyable Mean
0.136226	26.60128	1.055738	3.968750

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	0.03125000	0.03125000	0.03	0.8684
order	1	0.03125000	0.03125000	0.03	0.8684
sex*order	1	0.03125000	0.03125000	0.03	0.8684
interface	1	0.28125000	0.28125000	0.25	0.6200
sex*interface	1	3.78125000	3.78125000	3.39	0.0779
order*interface	1	0.03125000	0.03125000	0.03	0.8684
sex*order*interface	1	0.03125000	0.03125000	0.03	0.8684

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	0.03125000	0.03125000	0.03	0.8684
order	1	0.03125000	0.03125000	0.03	0.8684
sex*order	1	0.03125000	0.03125000	0.03	0.8684
interface	1	0.28125000	0.28125000	0.25	0.6200
sex*interface	1	3.78125000	3.78125000	3.39	0.0779
order*interface	1	0.03125000	0.03125000	0.03	0.8684
sex*order*interface	1	0.03125000	0.03125000	0.03	0.8684

Dependent Variable: enjoyableD

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	15.37500000	2.19642857	0.86	0.5530
Error	24	61.50000000	2.56250000		
Corrected Total	31	76.87500000			

R-Square	Coeff Var	Root MSE	enjoyableD Mean
0.200000	853.7499	1.600781	0.187500

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	15.12500000	15.12500000	5.90	0.0230
order	1	0.12500000	0.12500000	0.05	0.8271
sex*order	1	0.12500000	0.12500000	0.05	0.8271
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	15.12500000	15.12500000	5.90	0.0230
order	1	0.12500000	0.12500000	0.05	0.8271
sex*order	1	0.12500000	0.12500000	0.05	0.8271
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Dependent Variable: disoriented

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	7.96875000	1.13839286	1.42	0.2435
Error	24	19.25000000	0.80208333		
Corrected Total	31	27.21875000			

R-Square	Coeff Var	Root MSE	disoriented Mean
0.292767	20.61792	0.895591	4.343750

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	0.78125000	0.78125000	0.97	0.3335
order	1	0.03125000	0.03125000	0.04	0.8452
sex*order	1	0.03125000	0.03125000	0.04	0.8452
interface	1	5.28125000	5.28125000	6.58	0.0170
sex*interface	1	0.78125000	0.78125000	0.97	0.3335
order*interface	1	0.28125000	0.28125000	0.35	0.5593
sex*order*interface	1	0.78125000	0.78125000	0.97	0.3335

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	0.78125000	0.78125000	0.97	0.3335
order	1	0.03125000	0.03125000	0.04	0.8452
sex*order	1	0.03125000	0.03125000	0.04	0.8452
interface	1	5.28125000	5.28125000	6.58	0.0170
sex*interface	1	0.78125000	0.78125000	0.97	0.3335
order*interface	1	0.28125000	0.28125000	0.35	0.5593
sex*order*interface	1	0.78125000	0.78125000	0.97	0.3335

Dependent Variable: disorientedD

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	7.37500000	1.05357143	0.56	0.7838
Error	24	45.50000000	1.89583333		
Corrected Total	31	52.87500000			

R-Square	Coeff Var	Root MSE	disorientedD Mean
0.139480	-169.4637	1.376893	-0.812500

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	3.12500000	3.12500000	1.65	0.2114
order	1	1.12500000	1.12500000	0.59	0.4486
sex*order	1	3.12500000	3.12500000	1.65	0.2114
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	3.12500000	3.12500000	1.65	0.2114
order	1	1.12500000	1.12500000	0.59	0.4486
sex*order	1	3.12500000	3.12500000	1.65	0.2114
interface	1	0.00000000	0.00000000	0.00	1.0000
sex*interface	1	0.00000000	0.00000000	0.00	1.0000
order*interface	1	0.00000000	0.00000000	0.00	1.0000
sex*order*interface	1	0.00000000	0.00000000	0.00	1.0000

Dependent Variable: time

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	33.55012738	4.79287534	2.02	0.1030
Error	20	47.46065833	2.37303292		
Corrected Total	27	81.01078571			

R-Square	Coeff Var	Root MSE	time Mean
0.414144	68.74884	1.540465	2.240714

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	7.22172857	7.22172857	3.04	0.0964
order	1	7.26591696	7.26591696	3.06	0.0955
sex*order	1	16.96053601	16.96053601	7.15	0.0146
interface	1	0.28000000	0.28000000	0.12	0.7348
sex*interface	1	0.93622857	0.93622857	0.39	0.5370
order*interface	1	0.81913125	0.81913125	0.35	0.5634
sex*order*interface	1	0.06658601	0.06658601	0.03	0.8687

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	4.29083601	4.29083601	1.81	0.1938
order	1	7.26591696	7.26591696	3.06	0.0955
sex*order	1	16.96053601	16.96053601	7.15	0.0146
interface	1	0.42643125	0.42643125	0.18	0.6762
sex*interface	1	0.98908601	0.98908601	0.42	0.5259
order*interface	1	0.81913125	0.81913125	0.35	0.5634
sex*order*interface	1	0.06658601	0.06658601	0.03	0.8687