nou

TimeTunnel: Modeling and Interactive Information Visualization Using Three Dimensional Timelines

Ioannis Drossis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Knossou Av., P.O. Box 2208, Heraklion, GR-71409, Greece

Thesis Advisor: Prof. *Constantine Stephanidis*

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH).

UNIVERSITY OF CRETE

COMPUTER SCIENCE DEPARTMENT

TimeTunnel: Modeling and interactive information visualization using three dimensional timelines

Thesis submitted by

DROSSIS IOANNIS

in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Drossis Ioannis

Committee Approvals:

Supervisor _____
Constantine Stephanidis
Professor

Member _____
George Papagiannakis
Assistant Professor

Member _____
Dimitris Grammenos
Principal Researcher

Department Approval: _____
Angelos Bilas
Professor, Director of Graduate Studies

Heraklion, September 2012

To my parents,

Stelios and Vicky.

Abstract

Timelines are a widespread concept for temporal information representation which exhibit events over time. In general, the aim of timelines is to present information regarding specific temporal periods in combination with an overview of all occurring events. The content of timelines can vary from historical events to extensive data representation and is limited only by the chronological nature which is compulsory for any type of information. Timelines are employed in various environments and contexts, including exhibition spaces, museums and education, due to their ability to provide a clear picture of temporal information.

In this context, this thesis reports on the design, development and evaluation of TimeTunnel, a system allowing the modeling, storing, visualization and multimodal interaction with timelines and related multimedia information. TimeTunnel uses ontologies as a formal data model to represent knowledge, which are a broadly accepted approach to efficiently store and retrieve semantic information.

In addition to the typical 2D view supported by most available interactive timelines, which provides a temporal overview of events occurring during a time period, TimeTunnel also provides an alternate view of the same information that is more suitable for experiencing the time dimension as well as for 'first-hand' exploration of the available information. This view employs a time tunnel metaphor, i.e., a long corridor along which events are placed in chronological order, using distance to represent time. On the tunnel sides, next to each event, exists an extendable 'showcase' comprising various multimedia information objects (e.g., texts, images, videos, 3D models).

TimeTunnel supports multiple multimodal interaction techniques, even in combination, so as to offer natural interaction in a wide range of hardware set-ups employing various display and input devices. Apart from interaction techniques oriented to desktop-based interaction (e.g., mouse, touch screen), additional approaches were studied and implemented specifically targeted to navigation and manipulation in 3D virtual environments. In this regard, TimeTunnel supports full-body kinesthetic interaction. Firstly, the system is location-aware, as the user's position in space is tracked and handled as input. Moreover, users can interact with the system through gestures, using their bare hands to interact with items and their legs to navigate in virtual space. Furthermore, TimeTunnel integrates physical objects as supplementary means of tangible interaction (e.g. the Smart Box, a box with embedded sensors).

An evaluation study was conducted with 16 participants to assess TimeTunnel in terms of usability and user experience. The evaluation results were highly positive and promising, confirming the system's usability and encouraging further research in this area. Furthermore, they provide interesting insights towards selecting appropriate interaction techniques for interacting with timelines in 3D space.

Περίληψη

Οι χρονογραμμές (timelines) αποτελούν μια ευρέως διαδεδομένη μορφή αναπαράστασης χρονικής πληροφορίας η οποία παρουσιάζει γεγονότα με βάση το χρόνο. Ο σκοπός των χρονογραμμών είναι η παρουσίαση πληροφορίας σχετικά με χρονικές περιόδους σε συνδυασμό με την παροχή περίληψης όλων των γεγονότων. Το περιεχόμενο των χρονογραμμών ποικίλει και μπορεί να εκτείνεται από ιστορικά γεγονότα έως εκτενή δεδομένα με βάση το χρόνο, ενώ περιορίζεται μόνο από τη χρονική φύση της πληροφορίας, η οποία είναι απαραίτητη για οποιοδήποτε τύπο πληροφορίας.

Σε αυτό το πλαίσιο, η παρούσα εργασία αναλύει το σχεδιασμό, την υλοποίηση και την αξιολόγηση του TimeTunnel («Χρόνο-Σήραγγα»), ένα σύστημα που επιτρέπει την μοντελοποίηση, αποθήκευση, την απεικόνιση και την πολυτροπική αλληλεπίδραση με χρονογραμμές καθώς και με σχετική πολυμεσική πληροφορία. Το TimeTunnel χρησιμοποιεί οντολογίες για την αναπαράσταση της γνώσης με ένα τυπικό μοντέλο δεδομένων (formal data model), προσέγγιση η οποία είναι ευρέως αποδεκτή για την αποθήκευση και την ανάκτηση σημασιολογικής πληροφορίας.

Επιπλέον της τυπικής δισδιάστατης απεικόνισης που προσφέρεται από τα περισσότερα διαδραστικά συστήματα χρονογραμμών, η οποία προσφέρει τη χρονική περίληψη γεγονότων που έλαβαν χώρα σε μια χρονική περίοδο, το TimeTunnel προσφέρει μια διαφορετική άποψη της ίδιας πληροφορίας που είναι καταλληλότερη για την αίσθηση του χρόνου καθώς και για την εξερεύνηση της διαθέσιμης πληροφορίας «από πρώτο χέρι». Η άποψη αυτή χρησιμοποιεί την αλληγορία μιας σήραγγας, δηλαδή ενός διαδρόμου κατά μήκος του οποίου είναι τοποθετημένα τα γεγονότα με χρονολογική σειρά, χρησιμοποιώντας την απόσταση για να απεικονίσει το χρόνο. Στο πλαϊνό μέρος της σήραγγας, δίπλα σε κάθε γεγονός, υπάρχει μια «βιτρίνα» που μπορεί να επεκταθεί και περιέχει ποικίλα πολυμεσικά αντικείμενα, όπως κείμενα, εικόνες, βίντεο και τρισδιάστατα μοντέλα.

Το TimeTunnel είναι σχεδιασμένο ώστε να υποστηρίζει πολλαπλές πολυτροπικές τεχνικές αλληλεπίδρασης, οι οποίες μπορούν να χρησιμοποιηθούν ακόμα και συνδυαστικά, έτσι ώστε να προσφέρουν φυσική αλληλεπίδραση σε ένα ευρύ φάσμα εγκαταστάσεων, απεικονίσεων και συσκευών εισόδου. Εκτός από τους τρόπους αλληλεπίδρασης που βασίζονται σε επιτραπέζια συστήματα (όπως το ποντίκι και η οθόνη αφής), εναλλακτικές προσεγγίσεις μελετήθηκαν και υλοποιήθηκαν στοχεύοντας συγκεκριμένα στην πλοήγηση και το χειρισμό τρισδιάστατων περιβαλλόντων. Από αυτή την άποψη, το TimeTunnel υποστηρίζει την αλληλεπίδραση μέσω κίνησης, χρησιμοποιώντας όλο το σώμα. Πρώτον, το σύστημα γνωρίζει τη θέση του χρήστη στο χώρο και τη χρησιμοποιεί ως είσοδο. Επιπλέον, οι χρήστες μπορούν να κάνουν χειρονομίες για να αλληλεπιδράσουν με το σύστημα, χρησιμοποιώντας μόνο τα χέρια τους για την αλληλεπίδραση με επιμέρους στοιχεία και τα πόδια τους για την πλοήγηση στον εικονικό χώρο. Επιπλέον, το TimeTunnel χρησιμοποιεί αντικείμενα ως δευτερεύοντα μέσα αλληλεπίδρασης, όπως το «Έξυπνο Κουτί», στο οποίο είναι τοποθετημένοι αισθητήρες.

Μια διαδικασία αξιολόγησης με 16 συμμετέχοντες διεξήχθη για την εξαγωγή συμπερασμάτων σχετικά με τη χρηστικότητα του συστήματος και την εμπειρία χρήσης των συμμετεχόντων με αυτό. Τα αποτελέσματα της αξιολόγησης ήταν ιδιαίτερα θετικά και ελπιδοφόρα, επιβεβαιώνοντας τη χρηστικότητα του συστήματος και ενθαρρύνοντας την περαιτέρω έρευνα στο συγκεκριμένο πεδίο. Επιπλέον, παρουσιάζουν ενδιαφέρουσες πληροφορίες και απόψεις σχετικά με την επιλογή των καταλληλότερων τεχνικών αλληλεπίδρασης με χρονογραμμές που εκτείνονται σε τρεις διαστάσεις.

Acknowledgements

I would like to thank my supervisor Prof. Constantine Stephanidis for the support and encouragement throughout the conduction of this thesis.

Furthermore, I would like to thank my colleagues at the Human-Computer Interaction laboratory of ICS-FORTH, especially Dr. Dimitris Grammenos, Ilia Adami and Anthony Katzourakis for the graphics work.

The work reported in this thesis has been conducted in the context of a scholarship provided by the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH), including financial support and the use of the necessary equipment facilities.

Additionally, I would like to thank my friends Sterios, Maria, John, Effie, Nikolas, Panos and Nikos for their support throughout the conduction of this thesis. I would especially like to thank Chryssa who helped me in every phase of this work for her caring, support and patience. Last but not least, I would like to thank my parents, Stelios and Vicky, as well as Zozetta and Areti, who despite the distance between us stood by me at all times.

Table of Contents

Contents

List of Figures

List of Diagrams

1 Introduction

A timeline can be defined as a graphical representation of a period of time, on which important events are marked. A timeline can be regarded as a map which allows users to locate events in time. The use of timelines is not bound to specific domains, but allows the visualization of any type of content as long as a temporal representation is meaningful. Timelines comprise a familiar means of temporal events representation that may be incorporated in interactive systems in order to augment static information visualization by displaying additional detailed information on demand.

Graphical User Interfaces evolve from traditional WIMP (Window, Icon, Menu and Pointer) paradigm to a new generation of User Interfaces that rely on natural means of interaction of operand and operation specification [58]. Natural interfaces are subsumed under Ambient Intelligence domain, which aims to offer humans the ability to address their needs through the use of computers in a natural and anthropocentric manner. Ambient Intelligence environments are characterized by the ubiquitous and unobtrusive presence of electronic equipment in the users' environment that allow the system to sense users' actions and react accordingly in order to help them achieve their goals.

This thesis aims to create a system that has the ability to model, store, retrieve and visualize information through the use of interactive timelines. Moreover, the system should add up to timelines the interaction techniques that Ambient Intelligence mediates. TimeTunnel was developed towards achieving these objectives, allowing the visualization of content-independent temporal information inside the user's environment which can be manipulated through a variety of natural means of interaction. In particular, TimeTunnel:

- Uses a formal Data Model for information storage and retrieval
- Provides an overview of the events of the timeline
- Provides an alternative visualization of the events in a temporal sequence
- Allows the partitioning of timeline into time periods
- Offers events' categorization in concrete groups
- Allows the display of detailed event information on demand
- Can be manipulated using a variety of natural means of interaction
- Allows the combination of different interaction techniques, even concurrently
- Uses three dimensional space as a tool to enhance information visualization
- Develops techniques regarding three dimensional user interaction

The structure of this thesis is divided into the following Chapters:

Chapter 2 presents up-to-date related work regarding the different aspects of this thesis. The different approaches that exist in literature and are related to the characteristics of TimeTunnel are presented, analyzed and discussed. Each proposed technique is examined and its advantages and disadvantages are studied in order to reach conclusions regarding its efficiency and limitations of use.

Chapter 3 describes the research objectives of this thesis. The motivation concerning the rationale of the perception, the design and the implementation of TimeTunnel is analyzed, along with its innovative aspects.

Chapter 4 examines the requirements regarding this thesis. The process of requirements elicitation is described step-by-step, followed by the results concerning the user requirements of the final outcome.

Chapter 5 discusses the overview of TimeTunnel. Firstly, fundamental concepts are investigated and analyzed in order to allow the reader to perceive the concept of TimeTunnel's key terms. Moreover, the Data Modeling with ontologies is presented in respect to the aforementioned terms. Furthermore, the design of TimeTunnel is presented, along with the dilemmas that arose during the iterative design process. Finally, a Hierarchical Task Analysis is shown related to user actions.

Chapter 6 examines the natural interaction methods that are supported by TimeTunnel. Each interaction technique is thoroughly analyzed according to the available actions that the user can take in order to manipulate the system.

Chapter 7 describes the implementation details of TimeTunnel. The concrete parts are outlined according to their role inside the engine of the system.

Chapter 8 discusses the evaluation process and its results regarding the attractiveness, the unambiguity of information displayed and the usability of the system.

Chapter 9 presents the conclusions in respect to the initial objectives, taking into consideration the results of the evaluation process. Furthermore, future work is discussed concerning additions that can improve TimeTunnel and enhance user experience.

2 Background and Related Work

2.1 Ambient Intelligence

2.1.1 Ambient Intelligence Environments

Information technology has been continuously evolving since the introduction of computer systems, expanding the potential of technology assistance to people's everyday life. The integration of computers in the human environment becomes essential in order to offer users the ability to simplify their lives and aid the accomplishment of their goals. Ambient Intelligence is a research field which aims at computers disappearing in the background and making the coexistence of humans and technology anthropocentric.

The term ambient is defined as 'existing or present around you' in MacMillan Dictionary[2] and as 'relating to the immediate surroundings of something' in Oxford Dictionaries[3]. According to E. Aarts and J. Encarnacao Ambient Intelligence (AmI) refers to "electronic environments that are sensitive and responsive to the presence of people" [1]. Ambient Intelligence envisions the invisibility of technology in our natural surroundings, present when we need it, context-sensitive and autonomous [61]. AmI combines Ubiquitous Computing with intelligent User Interfaces in order to offer the users natural means of interaction.

Ubiquitous Computing is described by M. Weiser [62] as "making many computers available throughout the physical environment, while making them effectively invisible to the users". Ubiquitous computing includes the monitoring of the environment, making the system aware of the user's location and goals. The input of the system may be acquired through the use of sensors, cameras and any device that can extract information about the user's movement and interaction with the environment.

The user's communication with the system is transformed in such a way so that users have the impression of interacting with the environment rather than a personal computer or any electronic device. The user's actions to communicate with the system may vary and are similar to everyday movements, allowing the use of a variety of means of interaction that are interpreted through devices that are spread to the user's surroundings.

---

[2] MacMillan Dictionary Online, http://www.macmillandictionary.com/dictionary/british/ambient , last accessed on 10/09/2012.

[3] Oxford Dictionaries Online, http://oxforddictionaries.com/definition/english/ambient , last accessed on 10/09/2012.

AmI environments as far as human-computer interaction is concerned involve primarily two aspects: context awareness and natural interfaces [8]. Context awareness includes the use of emerging technologies to infer the location and the activities of the user, whereas natural interfaces refer to human-life communication capabilities as a way to employ direct human computer interaction.

2.1.2 Ambient Intelligence Interaction Techniques

Ambient Intelligence interaction is aimed to be human-centric, while the presence of computers should be as undetectable as possible. The use of Natural User Interfaces is considered to be of major importance so that users are able to interact with the system in an intuitive and instinctual manner. Several techniques appear in literature, each with their advantages and disadvantages, which were studied and analyzed for the implementation of this thesis. Ambient Intelligence allows the user to interact with several means often simultaneously, such as speech, body movements, gestures, eye and head tracking or even with physical objects. In that sense, a variety of interaction techniques exist and are refined so as to suit the needs for a wide set of diverse applications. A short review of state of the art categories follows, where benefits and drawbacks are presented for each of the categories of interaction techniques.

Multimodal interaction is a part of everyday human discourse: we speak, move, gesture and shift our gaze in an effective flow of communication. Jaimes et al. [20] define a multimodal system as a system that “responds to inputs in more than one modality or communication channel, such as speech, gesture, writing and others”. Multimodal user interfaces support interaction techniques which may be used sequentially or concurrently, and independently or combined synergistically. As a consequence, the selection of the combination of multimodal interaction should be done in a very careful, intuitive and natural manner because if not, the anthropocentric approach that ambient intelligence dictates will disappear. The users’ primary focus may turn from information exploration to an effort to try to understand the rationale of the interaction modes offered.

It is worth mentioning that breakthroughs in hardware setups influence trends as new opportunities arise by innovative features that may be introduced. For instance, recognition of objects and humans used to be a difficult, processor demanding and

expensive task. However, Microsoft's Kinect[4] included the employment of a laser sensor with a depth camera as a supplement of a normal RGB camera, providing a low cost – around 150 euro, accurate and stable input device. Yet, despite hardware evolution dependence, Ambient Intelligence aims to develop and provide all the necessary techniques for natural interaction in a variety of means as hardware evolution is considered a given.

2.1.2.1 Multimodal Input

According to Oviatt [37], « Multimodal interfaces process two or more combined user input modes (such as speech, pen, touch, manual gesture, gaze, and head and body movements) in a coordinated manner with multimedia system output. Carmigniani et. al. [7] provide a state-of-the-art extensive overview of modes of interaction as well as display modes in regard to Augmented Reality systems, which may be applied in a straightforward way to Ambient Intelligence environments. Papagiannakis et. al. [40] present a survey for the applications of Augmented Reality to ubiquitous computing, focusing primarily on the review of Mobile Augmented Reality Systems (MARS), presenting diverse approaches and their limitations. Furthermore, the authors link multimodal input to the accompanying interfaces, which are a new class of interfaces that aim to recognize naturally occurring forms of human language and behavior, and which incorporate one or more recognition-based technologies (e.g. speech, pen, vision) ». Multimodal systems enhance information visualization, replacing original WIMP (Windows, Icons, Menu and Pointers) interfaces with richer applications and accommodating users' perceptual capabilities.

- Voice Interaction

Voice interaction refers to the technique by which the user may interact with the system by voice commands. This type of interaction is easy to use and very intuitive to the user, but its implementation is impeded by several factors. First of all, the recognition of voice commands is nowadays not very precise and robust, especially as noise increases. Even if this problem is solved by hardware or specialized algorithms, the next issue presented is the understanding of natural language, which is ideally the way voice interaction should work. Each sentence can be paraphrased in many ways and as a consequence the difficulty of analyzing user's meaning becomes a very complex issue.

Due to the limitations of voice interaction, its application is implemented through the use of specified commands. These commands, if applied as the only means of interaction, require a specific and strict scenario with well-defined dialogues [48]. In every

[4] Microsoft Kinect, http://www.xbox.com/kinect/ , last accessed on 10/09/2012.

state of the dialogue, the user may interact with the system by apparently choosing a command from a range offered in order to reach his goal. This approach may be appropriate if no other means of interaction apply efficiently, for instance, if the user is visually impaired [24]. Finally, speech interfaces may become unpleasant and irritating for the user if there is any trouble in communication with the system [22], when an extensive set of questions is necessary to clarify the user's commands. As a result, voice interfaces are mainly used as a secondary means of interaction [47], either through verbal commands or through emotion detection. In this case, the user is able to trigger certain features of scenarios in a natural way, while not using his hands.

- Gesture-based Interaction

Gestures are a common approach which proves to be very intuitive to users and is widely used in literature [47, 64 and 34]. Gestures can be defined as a form of non-verbal communication in which visible body actions communicate particular messages. Regardless of the means of interaction, gestures may be expressed through mouse, body or multi-touch interaction systems. Gesture recognition remains an issue around which much work exists in literature and comprises a field of continuous evolution. Currently, gesture recognition is in a relatively mature state where its identification is robust and allows experts to develop applications that rely explicitly on it for user's communications with systems. Furthermore, gestures with hands can be used to augment systems and allow additional interactions [18] when combined with other means of interaction such as simple touch.

Gestures may not be limited to multi-touch and hands, but may be applied to feet as well. Sangsuriyachot et al. [46] propose using a foot platform able to recognize specific foot gestures which apply rotation to objects displayed on the surface to enhance user interaction in tabletop displays. Valkov et al. [57] use foot gestures to expand simple multi-touch interaction and boost navigation in dynamic and complex 3D Virtual Environments.

On the other hand, the use of gestures must not be regarded as a suitable solution to all issues, especially in the case of body gestures, as extensive and continuous body movements may end up tiring users. Another problem which arises when gestures are applied may be the precision and accuracy of interaction required by the user. For instance, spatial navigation in a 3d space using gestures may be more complex than other approaches, such as body tracking and exact mapping of user's movements.

- Body Movement

As Stivers and Sidnell [54] describe, a user's body movement can be supplementary but greatly helpful in the field of natural interaction. Body movement indicates the pose of a user's body as mentioned by Jaimes et al. [20], which can be tracked and applied for selective interaction with the environment. In this case, the system may interpret a specific body pose in order to enable interaction in a specific manner. Papadopoulos et al. [39] use defined body poses recognition in order to allow navigation in 3d environments. According to their approach, whenever a user poses in a certain way, manipulation of a camera in a virtual 3d environment begins.

- Context Awareness

Context aware (also mentioned as location-aware) systems track users and offer a wide range of information on demand. Context awareness may be used as a mean to aid user's navigation in real space. Klante et al. [25] use a Personal Digital Assistant (PDA) to aid visually impaired users' navigation, while Ghiani et al. [15] use location awareness in order to offer users relevant and context-dependent information.

Another form in which context awareness appears in literature is the gradual information visualization. More information is displayed when the user is further away from the system (providing an overview) and more details are offered when the user approaches the display. Grammenos et al. [17] use this approach to progressively visualize information according to the location of any user inside a room: each user can choose a topic by moving on one axis and explore different aspects of information by moving on the other axis of the room. Similar to this approach, Lee et al. [30] propose the use of context-awareness as a mean to offer augmented interaction with reality information visualization.

- Eye-tracking, Head Tracking and Gaze Detection

Eye-tracking and head tracking are types of interaction where the system monitors a user's eyes and head respectively in order to identify the direction pointed by the user. Due to the fact that users are not forced to act in any way apart from looking at what is of his interest, gaze detection may be considered as the most natural interface.

Eye-tracking interaction is a technique where users manipulate the system with their gaze. Jaimes et al. [20] group eye-tracking systems into wearable and non-wearable, depending on whether special equipment is worn by the user. Wearable systems are in

general more precise but concurrently require special equipment to be worn, while non-wearable are less accurate, often requiring calibration for each user yet unobtrusive.

Head tracking is widely used for Augmented and Virtual Reality systems, especially when virtual space is three dimensional. The main advantage of this technique in comparison to others is the fact that it offers information about the transformation of the user's head. Transformation includes the position of an item as well as its orientation, which is of significant importance in three dimensional spaces. As mentioned in [40], head tracking may be achieved also in combination with Head Mount Displays (HMDs), which additionally provide a wearable natural system display mode.

Head tracking is found in the vast majority of literature in cooperation with other means of interaction. Benko et al. [4] create an augmented space in a frame and use gestures to offer interaction with virtual objects, while head tracking is used to complement interaction through the manipulation of the camera in a virtual world. Another practice of head tracking is the increased accuracy of gesture recognition, firstly presented by Nickel et al. [34].

A significant issue as far as head tracking is concerned is the system's ability to understand whether the user is simply looking around and therefore head tracking should be ignored or the user's aim is to interact with the system.

- Interaction with physical (smart) objects

Smart artifacts can be defined as objects with embedded technology giving them the ability to describe possible interactions, widely used in smart environments. In many cases smart objects are everyday objects with the addition of sensors and feedback to the user. The size of the artifacts is not specific but varies from a mobile phone to furniture. Smart object interaction is placed among tangible interaction techniques.

Grammenos et al [16] developed a smart coffee table called BooTable offering multiple input and output technologies. The main objective of this type of furniture was to create everyday table that may be used for information visualization and entertainment purposes. Apart from the table as a whole, the authors proposed the idea of smart tablecloths, which were plain printed canvases, equipped with RFID (Radio Frequency Identification) cards. According to the suggested concept, tablecloths are used to launch applications upon their placement on the surface of the table. RFID tags are common in

literature for object identification, as they are cheap, robust and can be planted in most artifacts, such as mobile phones [50].

Another concept of smart objects is location-awareness. Molyneaux et al. [33] use accelerometers to detect the orientation of a book and augment real world by projecting on it. Lucia Terrenghi et al. [55] create a cube prototype working as a learning tool; information is presented by LCD displays on each of the six faces, using an accelerometer to define the currently seen face on which information should be presented.

Finally, smart object detection techniques may be used in wearable interfaces in order to enhance interaction techniques and open new horizons in human-computer interaction. As a conclusion, smart artifacts provide an intriguing mean of interaction, as it may become an important part of natural user interaction, mainly when used complementary to other interaction techniques.

2.1.2.2 Interaction Models in 2D and 3D Spaces

The majority of applications in Graphical User Interfaces extend in two dimensional spaces. In order to describe an object in a two dimensional space, its position is required (in X and Y axes) along with its rotation (usually in degrees). An object or a virtual camera which depicts the information to be displayed along the two dimensional board has only three degrees of freedom, as it may move only on X and Y axes and rotate around one axis. Therefore, moving in two directions and rotating only on one axis is enough to navigate through information in all available ways.

On the contrary, the definition of an object's transformation is more complex. In three dimensional spaces six degrees of freedom exist, three concerning its translation (X,Y and Z axes) along with three others regarding its orientation (yaw, pitch and roll). As a result, the object's transformation and navigation with a camera in three dimensional spaces demands more variables to be set, making interaction more complicated.

Technology nowadays tends to expand the traditional two dimensional spaces by adding the depth dimension. This trend appears to be effective, especially if the fact that any three dimensional world can be flattened in a two dimensional space is taken into consideration. Furthermore, computer hardware and up-to-date displays –such as 3D televisions - finally offer the ability to perceive depth in visualization. NVidia 3D Vision[5] was

[5] Nvidia 3D Vision, http://www.nvidia.com/object/3d-vision-main.html , last accessed on 10/09/2012.

mainly created for games visualization but its application in any three dimensional interface is straightforward and may be offered in any computer application supporting DirectX.

In order to simplify user interaction in three dimensional spaces Bowman et al. [6] propose two conceptual models: the natural approach, where a replication of real world is applied and the magic approach, where a user's actions are augmented in a magic way through common magical beliefs. Each approach has its own advantages and disadvantages; the natural approach is very self-explaining and natural to the user, however, it works in a limited scope, as a one-to-one representation is applied and limitations existing in real world are replicated in the virtual one. Furthermore, the selection of a virtual object positioned far away in the virtual world requires extensive navigation and interaction sequences. On the other hand, the use of the magic approach simplifies this scenario, as the metaphor of a magic wand is used to select through ray-casting an object in the three dimensional virtual world. Furthermore, concepts such as flying carpets may be used to facilitate the exploration of a virtual world in a more straightforward manner. The drawback in this approach is the fact that the user may not be familiar to the magic metaphors used.

Another concept widely adopted in literature [57, 6 and 10] is navigation distinction in travelling and way-finding. Way-finding refers to the cognitive process of path creation in order to accomplish a target such as finding information about a specific event in the case of a timeline. On the other hand, travelling has to do with the actual process of moving in a virtual space. Travelling may be further grouped into the motor movement in virtual space and maneuvering to place the viewport at the exact desired location. Both of these two groups require accuracy and often occur simultaneously, therefore traditional multimodal input methods such as multi-touch surfaces and simple two dimensional gestures are not sufficient on their own.

The adaptation of a model solely based on three dimensional interactions (such as gestures) has not proved to be successful due to the increased complexity of user required actions as well as the complex cognitive model accompanying these actions. As a result, the major guidelines found in literature regarding interaction in three dimensional spaces include a hybrid implementation of two dimensional and three dimensional interaction models.

2.2 Ontology based data modeling

A data model can be described as an abstract model which defines the organization, storage and retrieval of real information. A semantic data model, which is also referred to as conceptual data model, can be refined as a technique to define data within the context of its interrelationships with other data. The use of a semantic data model is essential in order to allow information to include the kind of relations between data elements.

Ontologies are the structural frameworks for organizing semantic information. The purpose of the use of ontologies is the ability to model information originating from different sources. According to [59], ontologies define the concepts and relationships used to describe and represent an area of knowledge. As pointed out in [56], any ontology consists of a vocabulary of terms that refer to the items of interest in a given domain and some specification of meaning for the terms.

The Resource Description Framework (RDF) [43] is a general-purpose language for presenting information, mainly in the Web. Its specification designates the usage of RDF for the description RDF vocabularies.

The Web Ontology Language (OWL) [60] is a family of knowledge representation languages that can be used by applications that need to process content of information, rather than simply presenting information to humans. OWL offers greater machine interpretability of Web content in comparison to XML, RDF and RDF Schema due to the provision of additional vocabulary along with a formal semantics.

Heml (Historical Event Markup and Linking Project) is introduced by [44] in order to represent knowledge regarding events. Heml is an XML-based description language extending RDF that aims to describe historical events by associating events with persons, locations and keywords. The components of Heml are sufficient for the definition of who, when and where something happened. However, its extension is necessary in order to support additional categorization of timeline semantic information, such as eras and events' partition to distinct groups.

Furthermore, another application of ontologies is the representation of historical and heritage knowledge. Doerr [11] proposes the use of CIDOC, a high-level ontology data model for the integration of cultural data representation. Doulaverakis et al. [12] adopt CIDOC and use ontologies as a tool for information retrieval and online search inside multimedia heritage collections.

2.3 Existing Timeline Approaches

The use of timelines is widely reported as a major means of chronological information visualization in existing bibliography. Several approaches exist in order to match requirements in a variety of research areas and contexts, in which temporal information needs to be presented in chronological order so that users may navigate through time.

The most prevailing purpose of the use of timelines is to provide an overview of events over a specified time. This characteristic is found in all timeline approaches, regardless of particular other objectives which may be present. Information about a specific event can be easily perceived by the human brain, but as information amounts and event count increases, the human brain fails to maintain the general context of both the event and its information. Furthermore, the perception of the sequence of events over time is often almost as important as information itself: when studying history, the evolution of a certain field may not only be informative, but also indicative of future direction. Therefore, the necessity arises to display the surrounding information that is relevant, while being able to explore event-specific information using highly interactive applications.

Human Computer Interaction offers users the ability to explore additional detailed information at will, according to the aspect of information the timeline focuses on. Due to the dynamic nature of technology, this overview can be extended to fit special needs that arise in various research fields.

In the following sections, timelines are classified in three concrete categories, according to the type of information they aim to present as well as the aspect of information they focus on. This categorization is not claimed to be complete; however, each category provides an indicative overview of the type of information that needs to be shown in a manner through which they can be easily conceived by the user. It is worth mentioning that each timeline approach may not fit to exactly one of the following categories, but may contain characteristics of multiple categories that create a suitable hybrid mode.

2.3.1 Timeline Visualization Techniques

Time is by definition examined on timelines, as it is the point of reference for any component displayed. In the vast majority of timeline approaches described in literature, time is extended along one axis and information is placed according to the time it takes place. A different opinion suggests time extent being placed in a spiral form [2], in order to provide a more compact overview of time. This idea is intriguing in the sense that the use of

spiral form offers the ability to use two axes to extend time into, instead of just one: if examined time is quite long, linear extent of time reduces usability. On the one hand, the user may understand the event distributions at a glance; yet on the other hand, time granularity is not so obvious and the exact time of an event is harder to perceive. The display limitations of the use of spirals include the limited space to show additional information as well as the angular orientation that makes placement of any information, such as text, harder to read.

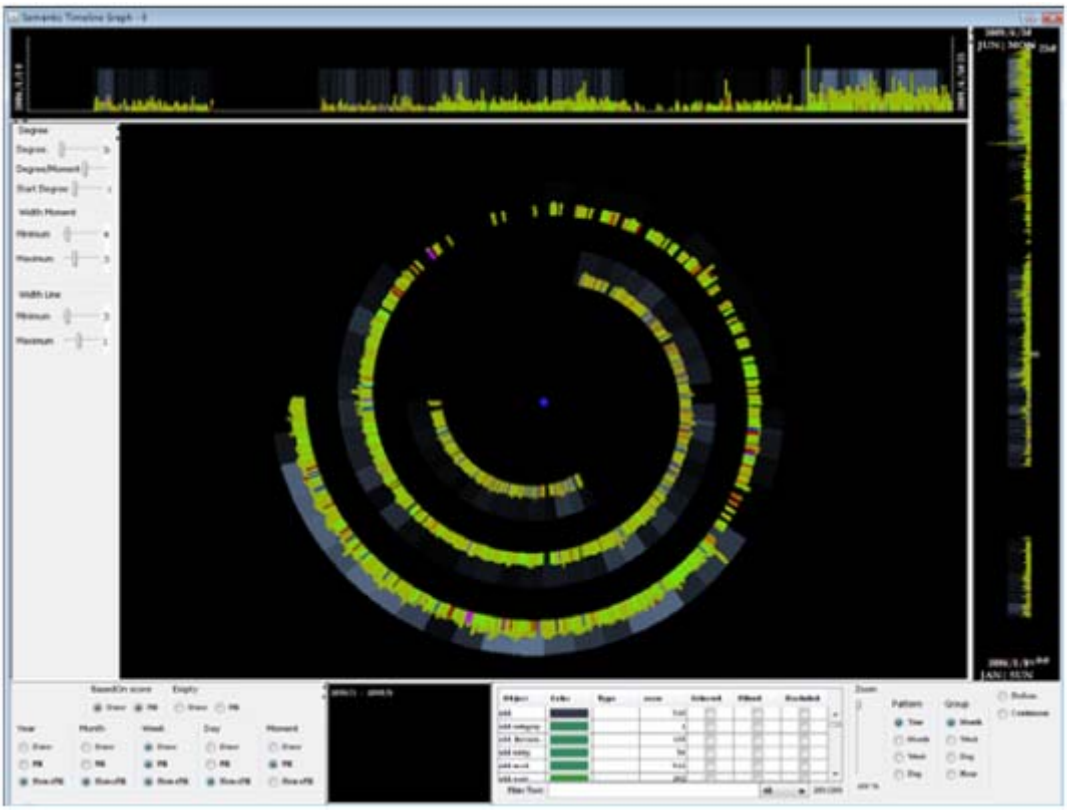

Figure 1 : Semantic Spiral Timeline Representation

One entirely different existing approach of extending traditional timelines is the inclusion of geospatial information. In this case, X and Y axes are used to define the location of an event on a map and time extends on Z-axis [23]. The authors' approach consists of three components, entities, locations and events which respectively designate people or things, geospatial data and facts. The navigation though time is accomplished by a scrollable time slider and components are linked according to their semantic meaning. The concept offers a complete depiction of the real world, as it answers to who, what, when and where it happened. The disadvantage of this approach is the complexity of the represented

information which makes conception of the overview of events along with their context more confounded. Kraak [26] adopts the aforementioned approach of time and space representation and enhances it by coloring focused time and displaying information regarding places when related to the visiting time. When events regarding a place do not take place in the currently visited time, information boxes are rendered empty.

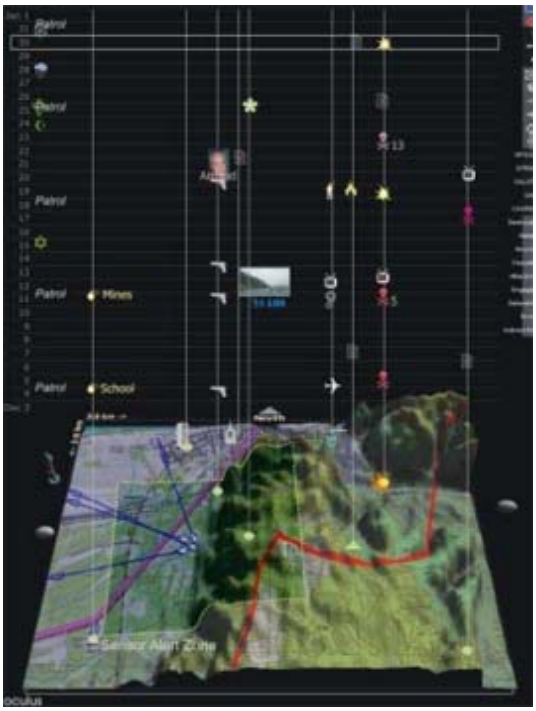

Figure 2: Geotime Visualization of geospatial information

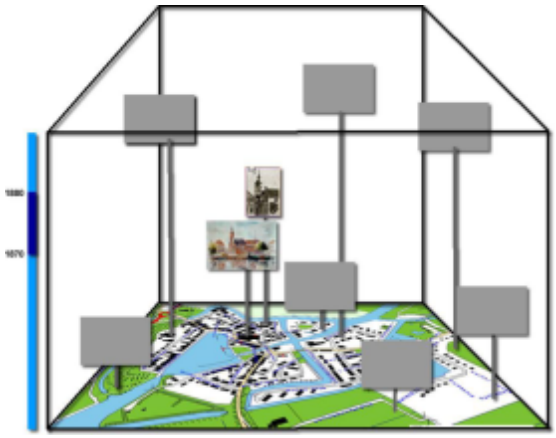

Figure 3: Visualization of time and space by Kraak

Information is denoted as events that take place over time and are categorized or interrelated with each other. Furthermore, periods of time are often proposed as an approach to time fragmentation, usually to define eras. Finally, literature contains many examples of implementation of faceted display in order to offer filtering and categorization of visualized events.

On the majority of timelines offering an overview display, at least two types of interactions are supported, panning and zooming. Panning refers to the user's ability to travel through time and examine parts of the timeline, while zooming refers to the level of temporal detail. Zooming in allows a time span to stretch to fit a larger area and therefore offers space which can be used to display additional information.

Faceted navigation is a common practice used for temporal data visualization in timelines [41, 65, 63 and 19]. It is implemented by the adoption of taxonomies to classify data in multiple ways and allows the application of filters to information. The practice of faceted display arises in all aforementioned timeline categories. The purposes of utilization of faceted display include but do not limit filtering, categorization, browsing and hierarchical display. Further ideas which involve faceted display are the exploration and comparison of events according to the specified facets. Due to the fact that facets describe information but are not directly part of it, their display in the interface provided to the user is distinct and embedded in a separate panel. However, changes in facets' selection are applied at real time to the main display. Faceted display is implemented in a variety of ways in literature:

- Hierarchical trees [65 and 63] are one of the dominant techniques to filter browsing information; users are able to refine their filtering to find the exact area of interest to navigate in.

- Separate quantitative controls [3] constitute another approach to modify the detail level of each facet.

- Separate toggle / filtering controls [29] allow hiding or displaying particular events in the timeline,

In this direction, filtering trees and histograms provide additional aspects and categorization of information [65]. In general, faceted display proves to be a very helpful and important attribute which a timeline must include in order to support a categorization of

events, in a meaningful to the end user manner. However, the display of the facets without making the total interface more complicated still remains a challenge. The adopted approach is the implementation of a panel through which the user may select the facet is interested in.

Krishnan et al. [27] propose a hierarchical activity-based information organization, which can be thought of as a grouping similar to geospatial categorization. According to the authors' concept, the main display of the timeline is not unique, but is split to different panels that depict the activity in which each event is placed. Apart from this fragmented display, timeline is displayed concurrently as a whole, containing all the events and providing an overview of all the facts which took place.

Another issue when presenting information, especially historical, is their being appealing as well as informative to the user. Expansion of traditional timelines may be accomplished through enhancing event information with multimedia elements such as images and text [23, 26, 65, 3, 28, 38 and 45]. In general, multimedia appears either beside an event or with a pop-up window, when an event is selected, containing detailed information.

2.3.2 Historical Events Representation

One major application of timelines is the visualization of historical events. Several case studies, such as the life of a musician, battles that occurred in a country, a person's activities or even the history of the earth could be integrated and visualized by a timeline. This category of timelines usually includes an overview of all events happening in the duration examined along with detailed information regarding specific events. The aim of this type of timelines is to provide the user with both the general image and concept and additional information for each event shown. The additional information varies according to the objectives, ranging from multimedia information to event metadata and relationships with other events. TimeSlice [65] stands an indicative timeline representing historical events. Furthermore, another approach is SIMILE Timeline[6], an open source framework for the creation of interactive timelines that are available through the World Wide Web. SIMILE Timeline is built on top of MIT's SIMILE Project [51], which aims at the enhancement of interoperability among digital assets, data models and services. Moreover, Apple's Time

[6] SIMILE Timeline, http://www.simile-widgets.org/timeline/ , last accessed on 10/09/2012.

Machine[7] is a widely known application that uses timelines to display the changes of the user's documents in accordance to the time that they occurred.

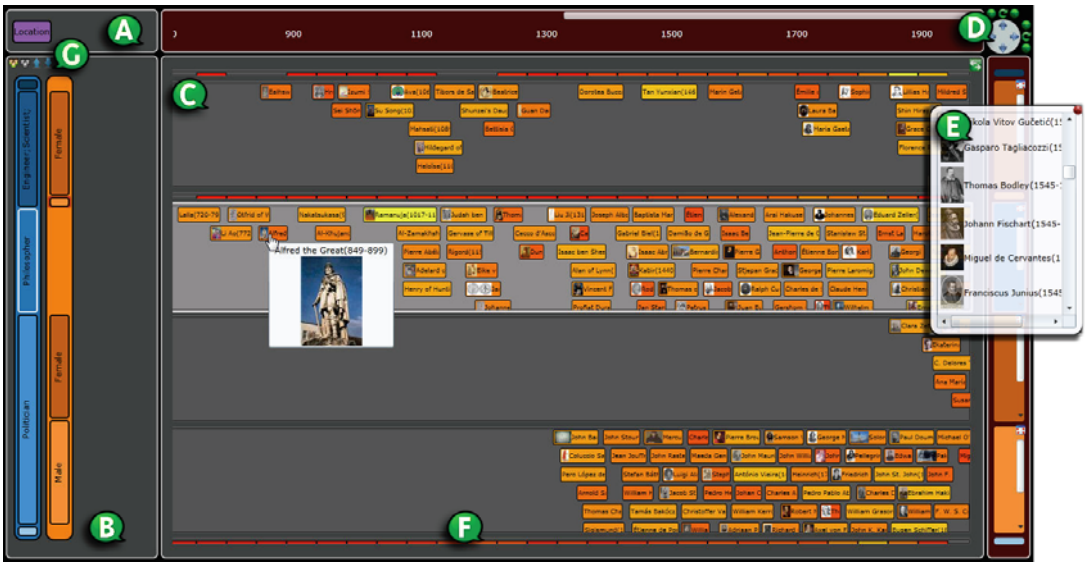

Figure 4: TimeSlice's main view

2.3.3 Timelines for Temporal Data Representation

Temporal information appears in numerous forms and fields. Usually, temporal data consist of several values of one or more variables that change over time and therefore timelines for temporal data representation may be thought of as enriched graph visualization techniques. Their aim is to allow users to analyze the distribution of the values presented and compare states over time. Furthermore, another characteristic of the timelines that are subsumed in this category is the presence of large set of data that need to be shown to the user. It is worth mentioning that the use of timelines is in some cases supplementary. For instance, [31] present a peak finding algorithm which generates events over time using twit counts and recognizing its patterns over time. A timeline is shown to show temporal dispersion of twits as well as the created events, which can be also regarded as an enriched graph display with labels denoting events. An interesting part is the combination of the aforementioned timeline with a geospatial display of the locations of twits on a world map that offers the ability to associate temporal information to places that are related.

[7] Apple Time Capsule and Time Machine, http://www.apple.com/timecapsule/ , last accessed on 31/09/2012.

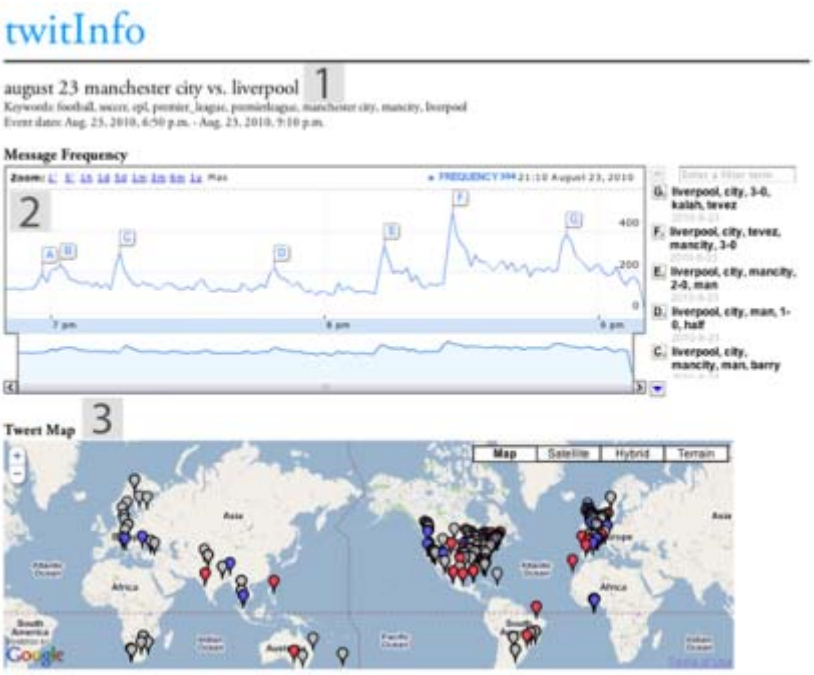

Figure 5: Temporal Data Representation in TwitInfo

An indicative example of this category is Event Line View by K. Matkovic et al. [32]. In this case, events are visualized using dots over time, offering a vague but very exact and indicative view of the count of events occurring over time. Another interesting idea presented is the use of techniques such as the Tag Cloud View, in which events count along a certain period defines the scale of the period's name display. These approaches prove to be very helpful for the perception of patterns over time, when information over time is quantitative. However, the drawback of this scenario is the inability to differentiate events from each other when needed.

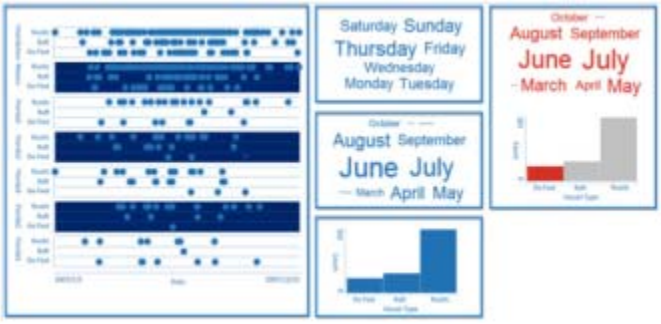

Figure 6: Event Line View

2.3.4 Semantic and Hierarchical Timelines

Information nowadays is not bound to strict cut off data, but is enhanced with additional information often mentioned as metadata. Metadata may be defined as "data about the meaning, content, organization or purpose of data" [49] and describe an item's content comprising supplementary information; for instance, a book's metadata may include its publication date, its author, its page count, its publisher etc. Therefore, events happening over time are extended to entities holding further details as well as relationships with other entities. Their relationships may have different form and meaning, dependent upon their definition of the used semantic data models. The need of visualizing relationships of events becomes imperative when taking into account the fact that the human brain has the ability and tendency to organize information and relate it with other facts. The combination of the temporal overview of events with the exploration of semantic relationships of events proves to be especially powerful, as it offers a thorough and exhausting presentation of a time period from different aspects.

Hierarchical timelines are a branch of semantic timelines whose relationships construct a tree-like structure. Tree-like structures are by definition graphs of nodes that have zero or more children and at most one parent node. The type of relationship is expressed by the semantic meaning given to it; for instance, a child node may be inferior in hierarchy to the parent or a more specialized and refined type in a generic category. Each event may be classified to one or more categories.

An indicative example of semantic timelines is presented by Jensen [21]. The author's aim to visualize complex semantic data is accomplished by the use of different types of arrows to connect events inside a timeline. The arrows implementing this idea are color-coded as well as shape coded (for instance dotted), while a legend describes the meaning of each type of arrow. The adoption of this approach, apart from it being simple and familiar to the majority of the users, has also the benefit of displaying direction in the relationships visualized. However, as the size of the information displayed or the inter-relations increase, the timeline becomes chaotic and impossible to understand, as arrows will be close to each other and possibly have intersections. Another issue may be the relation of events whose distance in the representation is considerable, as its linked events will not be clear. In order to avoid this, the author proposes and implements a mechanism of grouping several relative events to sub-timelines, which expand when users examine the details and collapse when user visits the overview of the timeline.

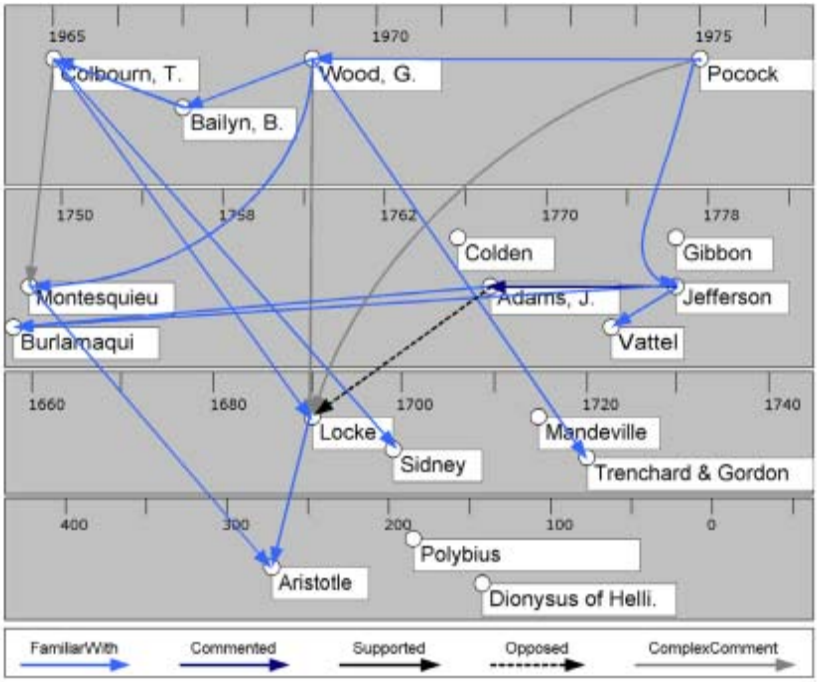

Figure 7: Semantic Timeliness' visualization of interrelated events

Semantic technologies facilitate the ability to categorize information in a domain-specific hierarchical schema by semantic time-dependent relations. Stab et al. [53] propose an interesting extension of traditional timelines to support relations that change over time. Their approach accomplishes this by implementing a faceted display in four different panels.

- Temporal navigation Panel, which is used as an overview visualization tool
- Main View, which is the main information display panel
- Hierarchical Categorization View, which is implemented by a faceted display and allows the user to filter the categories in the hierarchy to display
- Detail View, through which the user is able to access additional information he is interested in

Another implemented and interesting idea is that semantic relations are visualized only when one of the corresponding entities is selected. Hence, the display of all entities is clear and users are able to spot and apprehend the relationship easier, as the system interacts with their activities.

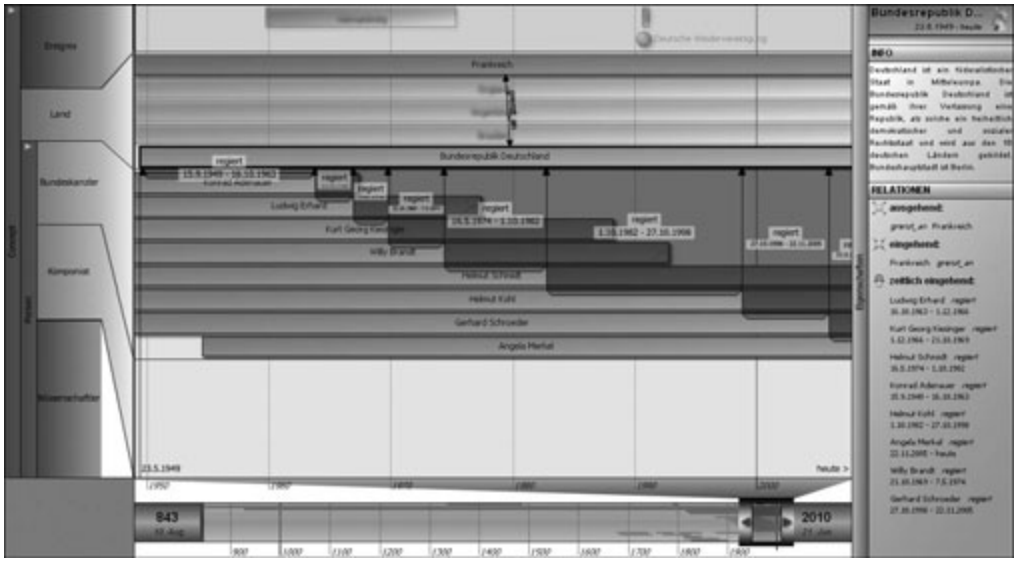

Figure 8: An indicative view of SemaTime's display

2.3.5 AmI Timelines

To the best of my knowledge literature does not contain any timeline facilitated and specifically designed to work in an Ambient Intelligence environment. Timeline incorporation inside an Ambient Intelligent environment offers users the ability to interact in a natural way. The proposed timeline engine generation is designed deliberately to be three dimensional in pursuance of extensive information visualization in a habitual manner, as the real world is also three dimensional.

Multimodal interaction is of essential importance especially when taking into consideration the fact that the proposed timeline extends in three dimensions, as manipulation of such systems becomes hard to handle conveniently using solely traditional input mechanisms.

It is worth mentioning that existing multimedia support in timelines is limited only to images and text. Nevertheless, information is not limited to textual representation enriched with images and graphs, but extends to the concept of multimedia. TimeTunnel supports multimedia visualization natively; incorporating multimedia includes support for audio, video and Speech Synthesis, allowing users to listen to the displayed text if needed. Apart from multimedia support, TimeTunnel facilitates rendering and handling of simple, animated and skinned three dimensional models.

3 Motivation and Rationale (Research Objectives)

The aim of this thesis is the creation of TimeTunnel, a system able to automatically create rich interactive timelines extending in three dimensional spaces which can be manipulated by natural means of interaction.

The fundamental characteristic of any timeline is the provision of a temporal overview of events. Several approaches exist in literature towards the creation of interactive timelines that augment the traditional overview of the events. The aim of TimeTunnel is to create an interactive timeline which offers two alternate ways to present the same information. Firstly, one view should depict the overview of the presented events in order to provide the general distribution of events in time. Secondly, another view should allow the sequential exploration of events in detail so as users are able to exhaustively examine the occurring events in temporal order.

Additionally, TimeTunnel should be able to express, store and retrieve event-related semantic information, and thus be able to retrieve knowledge from a formal data model. Additionally, TimeTunnel should be able to depict events' categorization, in order to group events according to their conceptual meaning in a non-restrictive way.

Furthermore, the context of the information visualized should not be limited; on the contrary, the timeline visualization should support any supplied content and duration, as a timeline's time detail may vary from seconds to centuries.

Another fundamental characteristic of TimeTunnel is the adoption of natural user interaction techniques. The user's movement and actions should be interpreted by the system in a meaningful way and, ultimately, allow users to easily handle the system. The interaction techniques applied may comprise multiple means of interaction that are used either independently or concurrently in combination. Any interaction technique applied should be fully functional and allow the users to manipulate all parts of the system. Furthermore, the system should adapt to display additional controls or hide redundant ones, according to the necessities of the interaction technique used.

Finally, another issue of great significance is the experimentation with three dimensional user interfaces and their interaction methods. Three dimensional virtual worlds, in comparison to two dimensional spaces, may be used to enhance the display of information by making use of the third dimension as well as the pitch and the yaw rotation. On the other hand, disorientation is an issue that may arise, especially in the case of users

not familiar with three dimensional worlds. Furthermore, the handling of three dimensional space increases interaction complexity. As a result, the means of interaction should be precise and rich enough to manage the virtual world, while remaining as simple as possible. TimeTunnel aims to create a virtual environment that exploits the functionalities of three dimensional worlds, prevents users' disorientation and is convenient to use in a natural way.

4 Requirements elicitation

This chapter illustrates the requirements for the creation of TimeTunnel as they were formed during the design process. Firstly, the process that was followed is described in detail, stating the different approaches that were adopted in order to achieve the optimal results. Secondly, the requirements are analyzed and grouped into two categories: the functional and the non-functional requirements.

4.1 Elicitation Process

A series of procedures were followed before the initiation of the design process in order to extract the requirements for the creation of TimeTunnel. Firstly, brainstorming sessions were conducted so as to develop the exact specifications of the characteristics of the system. Brainstorming sessions were the first step to produce ideas regarding the scope of a timeline application and create a vague overview of the scope of the system. Moreover, focus groups were scheduled in order to discuss in depth the detailed attributes that the system would have.

Interviews with end users took place so as to define the fundamental attributes that are necessary to offer the visualization of a timeline regardless of its context. The interviews also aimed at the definition of the properties that an end user would expect to see in a timeline, as well as the functionalities expected to be available. The age of the people interviewed ranged from 20 to 60 years old, which is similar to the age range that the system is addressed to. The results of the interviews were highly beneficial, revealing widely varied aspects that users imagined the system would have.

Once a rough image of the timeline was formed, mockups were designed to form an initial look of TimeTunnel. This approach was particularly helpful in order to bring weaknesses to the surface and allow the refinement of the design. An indicative example regarding the importance of the mockups was the replacement of the road with the tunnel,

a change of significant importance regarding the visualization of contextual information and the aid for the users' orientation.

Finally, throughout the development of the system constant evaluation took place regarding each new part of the system.

4.2 User Requirements

Functional requirements are related to the clarification of the functionalities that the system should support, whereas non-functional requirements refer to the system's behavior to support the user's goals.

The functional requirements that were stated in the process of requirement elicitation include the following:

- Provision of events' overview, as the fundamental characteristic of a timeline is the supply of the general image of incidents over a prolonged time.
- Detailed event exploration on demand, as users should be able to retrieve additional in depth information concerning the events of their interest, without interfering with the timeline's environment and perplexing the greater picture.
- Provide a representative view of an event at a glance, so as to offer an indicative aspect of any incident.
- Multimedia support to enhance the user's experience with rich information that may be easily perceived.
- Periods should be clear in order to discriminate different time frames and partition time according to general attributes that define eras.
- Categorization of events should be offered with the intention of allowing users to filter information and narrow down the range of the timeline to the scope of their interest.

As far as non-functional requirements are concerned, the necessities outlined may be divided into two groups: performance requirements and interface requirements. The interface requirements of TimeTunnel include:

- The ability to retrieve input from a wide set of sources that describe semantic information
- Interaction should be accomplished through the use of natural means of interaction
- Compatibility with a variety of means of interaction and support of multimodal interaction, using different interaction techniques
    - Separately
    - Concurrently in combination

The performance requirements refer to the characteristics of the system regarding the accomplishment of the user's objectives and include the following:

- The system should be self-explaining. Ideally, it should be easy to use for the first time without special training or at least after three minutes of training
- The interaction techniques should be intuitive and easy to memorize
- The system should be responsive to user actions
- Interaction should be precise
- The overall system setting should be stable and robust
- The system should run on a common personal computer without special hardware requirements

5 TimeTunnel Overview

5.1 Basic Concepts

In the following sections the central concepts of TimeTunnel are presented along with their integration into the system. Furthermore, the rationale of the decision taken regarding the adopted or rejected approaches is analyzed and explained thoroughly.

5.1.1 Timeline

The aim of timelines in general is to present detailed information regarding specific temporal periods in combination with an overview of all occurring events. Therefore, the technique used for information visualization includes two modes, each one serving a specific purpose, Classic2D and Tunnel Mode.

On the one hand, Classic2D Mode aims at providing information gradually to the user at the detail level they are interested in, ranging from providing an overview of everything present in the timeline to event-specific information of concern. The user shall be able to examine any event, but only on demand and if interested in it. The overview of the timeline is always shown in the background, displaying the context of each event in a consistent way.

On the other hand, Tunnel Mode of the TimeTunnel offers exhaustive exploration of all event information in a sequence. This mode of TimeTunnel aims at the comprehensive navigation of information focusing on the details of events and concealing their context.

The information displayed in both views of the timeline is the same, but presented in a way that fits requirements and interface needs better. Furthermore, the metaphors of interaction techniques used are similar, aiming at the information exploration itself and offering a consistent conceptual model applied which clarifies the available interaction mechanisms for the whole duration of the usage of TimeTunnel. In the following sections items and concepts that are adopted are explained in detail along with issues which arose during the design and implementation of TimeTunnel.

5.1.2 Events

An event is a central concept in a timeline which may be regarded as an object or an incident in time. Information concerning incidents over time is represented as events in a timeline arranged in chronological order.

TimeTunnel offers a variety of information types regarding events which may be split into two groups, event detailed information and event general information. Textual information is multilingual so that information is accessible to all users regardless of nationality. In order to achieve this, a wide character set is adopted which supports the vast majority of well-spread written languages.

The information regarding events may be further grouped into two generic categories: Event General Information and Event Detailed Information.

5.1.2.1 Event General Information

Event General Information is the part of information that is common for every event instance in TimeTunnel and consists of:

- **Event Date and Time**, supporting time detail from years to milliseconds: Microsoft's C# DateTime class is extended in order to support years BC enabling support for ancient years. Event occurrences may either be or considered as instant, for instance the birth of a person, or have duration in cases such as the construction of a monument. In order to fulfill this requirement and support events with duration, TimeTunnel separates event time to start and end times which in the case of instant events are equal.
- **Event Multilingual Title**, the descriptive name of the event. Because of the fact that title consists of a limited word count (usually ten words are sufficient) a three dimensional font is used in order to be visible from a wider view angle and to emphasize and underline the importance of this item.
- **Event Representative Item** which is a characteristic multimedia element regarding the accompanying event. Its aim is to provide a visual view of the event at a glance.

5.1.2.2 Event Detailed Information

Event Detailed Information includes any multimedia element that may be stored that reveals additional information and aspects that are related to an event, including:

- **Multilingual Text,** containing additional information about the event it belongs to. Three dimensional texts were firstly used to visualize textual information. However, evaluation with users showed that this approach was tiring as text length increased, particularly as text depth increased from totally flat to any increase of Z-axis scale. Another issue with three dimensional texts was the need for a container to be used as a background to clarify text, as items displayed on the background of the text would interfere with characters, especially if the background color was similar to the text. As a consequence, the approach of conventional two dimensional fonts was adopted, where text is rendered on an image quad in three dimensional space. Except from typical written multilingual representation, TimeTunnel offers multilingual voice synthesis on demand using Microsoft's Text To Speech technology.
- **Images**, which are the prevailing multimedia elements. The popular adage states that "a picture is worth a thousand words", reflecting the necessity of pictures to aid perception of incidents. Apart from description of items, photographic memory is common in human brains and has the ability to correlate concepts, facts and existing items with a visual impression. Image support is straightforward through the creation of quads rendering sprites in the three dimensional space; this implementation is enhanced in TimeTunnel through the use of a movable virtual zoom magnifier available on request.
- **Video playback,** which is embedded in TimeTunnel offering the most frequently used operations, play, pause and stop.
- **Three Dimensional Models,** which may be manipulated to offer detailed information regarding an item. Three dimensional models act as an extension to pictures and allow users to explore different aspects of items. The handling of the models is accomplished through rotation, offering stereoscopic view of any angle the user is interested in. Maneuver of models is done through multimodal interaction and therefore different techniques are supported simultaneously. The user may either virtually push appearing buttons for rotation in X and Y axes or virtually grab the object and rotate at will. Another implementation offered is the handling of a smart object that can transmit its orientation. In this case, the explored virtual item is rotated as a clone of the actual natural object.

5.1.3 Periods

Periods are time frames or time eras during which incidents share common characteristics or hold a distinct meaning. The use of periods aims to evoke an image and expectations of certain sets of characteristics and allows event labeling with respect to temporal categorization. For instance Stone, Copper, Bronze and Iron Ages may be an example of period, comprising an indicative example of eras of human evolution that offers someone the ability to imagine the greater picture of tools and materials used at that time.

Periods are usually sequential without any time breaks separating them. However cases may exist where a time frame between two periods does not constitute an era with specific characteristic or its visualization is of minor importance to the author of the timeline. TimeTunnel allows periods not to be continuous to fit cases where this is needed. Such an example is a timeline of major war battles in the twentieth century: the set of periods applied would include World War One and Two, the Korean War, the War of Vietnam and the Six Day War. These periods are not continuous and gap representation would result in empty spaces obstructing smooth navigation over time.

Furthermore, different time scaling may be needed in order to increase semantic weighting to specific periods, scale down time in extensive periods or expanding the extent of short periods. For instance, a timeline of ancient Greece would comprise both Bronze Age and Classical Greece. The forthcoming issue would be the wide extent of Bronze Age period, with sparsely placed events over 1500 years, while Classical Greece would include densely placed events over 200 years.

In order to solve such issues, TimeTunnel allows the scaling of periods by using an author-specified factor. This way event distribution over time can be smoother and over-crowding would be avoided. The adoption of this approach aids the visualization of dense eras in the timeline, where a large count of events can be spread in a broader area. An issue coming up when scaling time is the perception of time distribution over the timeline: in order to avoid this case labels are placed at the start and the end of each period, while time remains linear for the duration of each period. The labels are visualized above periods and if periods have no gaps in between, one of the labels' rendering is omitted to increase display clarity and information duplication.

Periods are color-coded in order to differentiate parts of the timeline at a glance. A sequence of colors is defined, chosen carefully in order to be distinct especially in

neighboring periods. The sequence consists of six colors in total which can be cycled indefinitely-the sixth period holds the same color as the first, the seventh of the second etc.

5.1.4 Categories

Categories are a field of major importance as far as semantic relationship grouping is concerned. Information representation evolves in modern systems through the use of metadata and semantic information accompanying specific facts. An incident which took place at a certain time is not a limited reference to an event but may include additional information concerning its context, its meaning, its consequences or even its relation to the reasons that lead to it.

TimeTunnel supports the separation of events into categories which act as groups and divide timeline events according to their conceptual meaning or their type. An indicative example may be in the case of a timeline of a director's movies, their categorization to their genre, for instance comedy, crime, thriller, drama etc. Each item may belong to several categories simultaneously, offering a model consistent to the real world. The combination of active categories rests to the user's selection to show only the events of interest and filter out irrelevant ones.

The first approach studied included the active categories to be mutually exclusive: whenever a category is selected, only the events belonging to the specific category are shown, whereas the rest of the events are hidden. Apart from all existing categories, an additional one is created that contains all the events that exist in the timeline.

An additional category visualization mechanism was developed with a different idea: the categories are not mutually exclusive and users may select as many categories as they want. When one or more categories are activated, all events belonging to any one of them are displayed. On the contrary, an event belonging only to the categories which are disabled will be hidden.

The first approach has the advantage of being clearer to users, as anyone may select the category of interest and explore the events belonging to the specific category, or show all events that exist in TimeTunnel. On the other hand, the approach does not allow a combination of the available categories to be applied, in order to exclude only specific categories on which the user is not interested in.

The second approach gives more freedom to the users to apply the filters they are interested in, by selecting the desired combination of categories. The major drawback of this approach is the fact that users who are not familiar to filtering combinations may not understand the classification of events, as filters are mainly adopted in the domains of sciences and engineering. Furthermore, the combination of categories makes unclear the specific categories that an event may belong to, as it may be visible to any of the active categories.

Although both methods have their own advantages and disadvantages, the first approach is preferred due to the fact of it being more suitable for the greater part of users that do not need to apply complex filtering but mainly to navigate through the category of interest.

5.1.5 Timeline Minimap

Minimaps are components which provide a wide view of the whole virtual world when examining parts of it. TimeTunnel does not show a minimap when navigating in Classic2D Mode, due to the fact that watching the overview of all events is straightforward, simply by moving backwards. On the contrary, the virtual tunnel aims at sequential event navigation and minimap provision is indispensable for the user to perceive the general picture of the currently examined time frame.

The minimap shown in the tunnel consists of three main parts:

- The **railway track**, which is placed along the roof of the tunnel.
- The **railway wagon**, containing a virtual screen with the overview of the timeline visualization in Classic2D Mode both at its front and at its back side, as user may navigate either forward or backward in time and look at the start or the end of the tunnel respectively.
- The **railway wagon frame**, depicting the equivalent display spot when exploring the same time at the Classic2D mode.

The minimap of the two dimensional timeline view is designed to be pre-rendered for performance issues, as the minimap contents are non-interactive. Therefore, the rendering of additional polygons would be a surplus to visualization requirements as long as pre-rendered displays can be created dynamically and be rendered with the same detail.

The automated creation of the minimap view is accomplished through the following simple but efficient mechanism: the railway wagon contains specific UV-coordinates for the texture being applied to the virtual screen that display the minimap, which can be changed at runtime. In order for the rendering of the whole timeline to be possible, a very wide display (the current implementation uses a 48:9 aspect ratio) is captured by moving the camera to the middle of the timeline on X axis and backwards on Z axis as much as needed so as to render its whole extent. Through the use of a very wide display viewport, the extent captured for the minimap is more than three times the normal display size in a resolution sufficient to cover the full display size. Cases such as event addition or period scaling at runtime require the dynamic substitution of the pre-rendered texture with an updated one, a process which TimeTunnel offers in an automated way.

A significant issue in this case is the need for a display which remains the same at all the parts of the timeline, regardless of the camera's position at X axis (semantically equivalent to time frame examined). This, however, cannot be achieved by perspective cameras but only with orthographic projection. Perspective cameras render the virtual worlds with a certain field of view, usually between 30 and 50 degrees. On the other hand, orthographic cameras visualize objects with the camera moving almost infinitely backwards and an angle of almost 0 degrees: the lines of the projection are parallel to each other, resulting in the extinction of the sense of depth. As a result, timeline visualization is captured using orthographic display in order to completely flatten the rendered components.

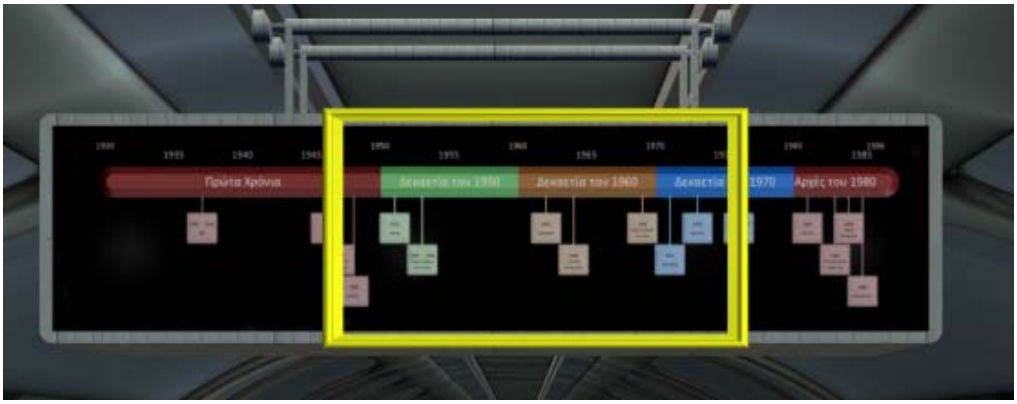

Figure 9: The railway wagon that hosts the minimap in the virtual tunnel.

5.2 Ontology Based Data Modeling

5.2.1 Data Representation

Each of the aforementioned fundamental concepts is stored and described by Web Ontology Language (OWL) [60], a common data model used for knowledge information representation in current up-to-date systems. OWL is built on top of a Resource Description Framework Schema (RDF)[43], endorsed by World Wide Web Consortium (W3C) and aims to process information mainly on the web.

TimeTunnel does not hold any limitations regarding the context of its use. Therefore, the need for support of any type of information arises as long as it is in a specific formal format. Web comprises an actually unlimited source of information continuously expanding; as a result, TimeTunnel's support of web information retrieval helps the system's instant inclusion of existing information as well as scalability in correspondence to web technology evolution.

Protégé [42] was used as a tool to design the ontology used and insert information in the existing case study, a brief overview of important events concerning computers. Protégé was selected as it is an open source ontology editor and knowledge base framework suitable for ontology definition and data insertion.

Entities are organized inside Protégé in classes containing properties. Each of the aforementioned fundamental concepts (the timeline, events, periods and categories) is represented in its own class designed to fit the needs of information storage and retrieval. The class inheritance hierarchy used is shown in the figure below.

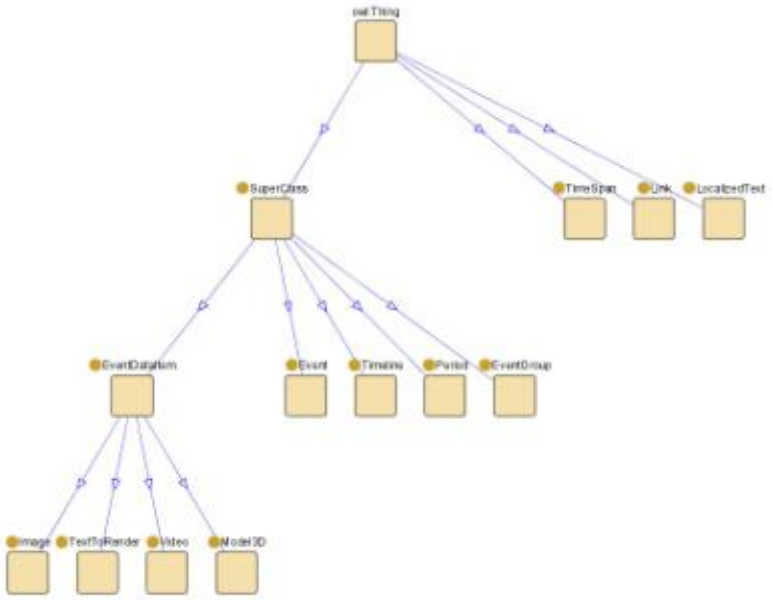

Figure 10: Class inheritance hierarchy in TimeTunnel's ontology

All the items contain a multilingual title due to the fact that they inherit from the class "SuperClass", as shown in the figure below.

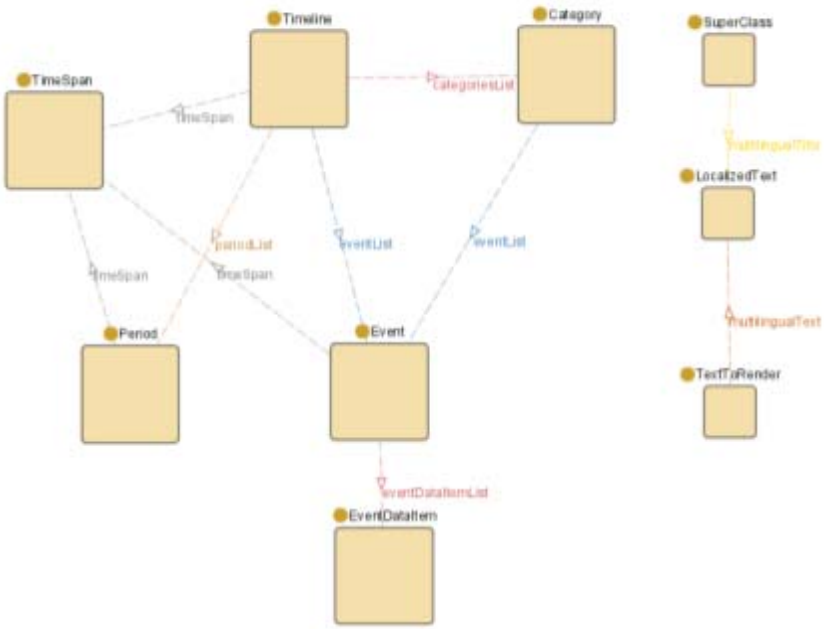

Figure 11: The overview of data architecture used for the description of TimeTunnel's information

- A timeline contains
    - Its chronological information (timeSpan).
    - The localized text with its multilingual description
    - The sum of all categories (categoriesList).
    - The sum of all events (eventsList).
    - The sum of all periods (periodsList).
- Each category contains
    - The localized text with its multilingual description
    - The sum of all events that belong to it.
- Each event contains
    - Its chronological information (timeSpan).
    - The localized text with its multilingual description
    - The sum of data that comprise its information (eventDataItemList).
- Each period contains
    - Its chronological information (timeSpan).
    - The localized text with its multilingual description
- Each TimeSpan includes
    - The item's start time
    - The item's end time

5.2.2 Querying and retrieving results

Data stored in the RDF ontology can be retrieved through the use of queries. The language used to express queries is SPARQL [52], a query language and data access protocol for the Semantic Web, which provides a wide support of statements.

- Conjunctions and disjunctions are supported in order to express complicated relations
- Required and Optional Values to get results that may either satisfy an obligatory or optional condition.
- Constraining Queries offer the ability to limit results within specified values and thresholds

RDF is built on the triple, a 3-tuple consisting of subject, predicate and object. Similarly, SPARQL is built on the triple pattern, which also consists of a subject, predicate and object.

The combination of SPARQL's available query statements results in the retrieval of the whole timeline information with a single query. TimeTunnel is able to execute SPARQL queries, retrieve the results and transform them to the engine's equivalent data, allowing instant dispatch of the visualization. This is achieved through a front-end parsing the results from generic queries and generating internally used XML information, which is later parsed by back-end reading information from the XML file. This way, the internal implementation is distinct from external information retrieval and may even be substituted by alternate front-ends for information input, offering scalability and extendibility to match any new data model to come in the future.

Furthermore, TimeTunnel allows the addition of information at runtime: apart from minor changes, such as the addition of a multimedia element, for instance, an image to an event, TimeTunnel allows the insertion of more complex items such as events, periods and categories. At the time that new information is added to TimeTunnel, the required actions are taken to recalculate all required information and adapt the visualization to fit the updated information. Finally, the ability to change information of elements is offered in the case of mistakes or updates at runtime.

5.3 System Design

TimeTunnel was created using an iterative design process during which constant evaluation took place. Although the system's general concepts remained constant, various dilemmas came up due either to limitation constraints or to more efficient techniques discovered along the implementation. Constant evaluation with both expert and novice users in a wide range of ages took place in order to arrive at the most suitable approach for every design issue.

In the following sections the design of TimeTunnel is presented not only by stating the final results, but also analyzing the advantages and disadvantages of the different routes examined and stating the rationale of the decisions taken regarding adopted and rejected ideas.

5.3.1 Visualization Common in Both Timeline Modes

This section describes the elements of TimeTunnel whose visualization remains identical in both Classic2D and Tunnel Mode. Consistency is a significantly important issue between the different representations of information that TimeTunnel offers. Therefore, as

long as an item's representation does not need to be altered so as to serve a specific purpose, it remains identical during the whole time.

5.3.1.1 Multimedia Elements

All the multimedia elements stay the same during both Tunnel and Classic2D Modes. The events' representation may alter their position and scale, but their appearance as well as the ways that users may interact with them remains constant.

The multimedia components that a user may explore include images, videos, text and three dimensional models. All of them, except for the three dimensional models, are placed inside a frame that adjusts to fit their dimensions and contains a spot on the bottom, where their description can be placed as a legend. Each component has exclusive ways of interaction, according to its unique characteristics and the way that users may extract information, easier or more detailed.

- **Images** may be zoomed in on the spot of the user's interest in order to find details of the picture. A cyclic quad appears on top of the image, which can be moved in any point of the image to allow the more comprehensive exploration of specific parts.
- **Videos** may be played, paused and stop on the user's demand.
- **Texts** may be read aloud, regardless of their language, using a speech synthesis system.
- **Three Dimensional Models** may be viewed in detail, simply by bringing them forward and rotating them in any direction at will.

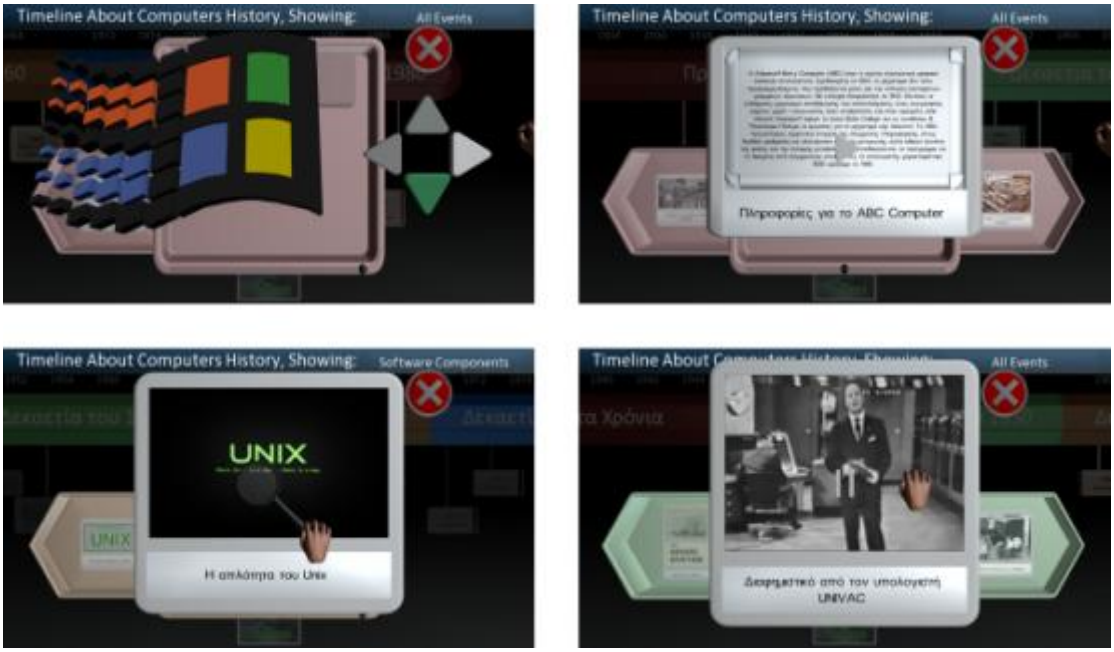

Figure 12: Multimedia elements exploration in TimeTunnel

Exploration of the additional features offered by TimeTunnel is achieved through virtual three-dimensional buttons, which appear when the user selects items to explore in detail. The buttons are designed to be suitable for the aim of the user's interaction: Images, videos and text contain toggle buttons which change their state when the user presses them. On the contrary, Three Dimensional Models' buttons for rotation on X and Y axes are continuous: while the user pushes them, the model rotates towards the corresponding direction. Furthermore, when an element is shown in detail, an exit button appears at the top-right part of the display, which the user may push to finish the detailed navigation.

5.3.1.2 Timeline Title and Categories

In both visualizations shown in TimeTunnel, the timeline's title is placed at the upper side of the display in order to inform users about the general subject that TimeTunnel presents. Moreover, the currently selected category is shown next to the timeline's title, hosted inside a distinct button. Upon pressing the button, a drop-down list appears containing all the available categories that the user may choose from. The fact that the categories appear next to the timeline's subject aids in the awareness of the role that categories have in TimeTunnel: categories allow the filtering of information and need to be presented in a way that users perceive this ability. Therefore, users are able to understand that any choice they make regarding the selected category will have a global impact on the

whole theme of the timeline. Furthermore, the adoption of this technique allows the encapsulation of a variable count of categories without occupying a large part of the display.

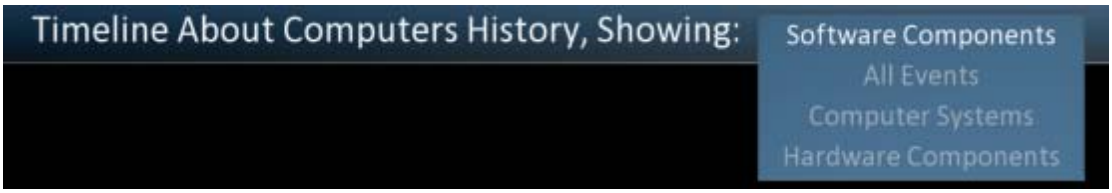

Figure 13: Timeline Title and Categories visualization

5.3.2 Classic2D Timeline Visualization

The central idea as far as Classic2D Mode is concerned is the provision of the overview of a wide range of information types and content. The basic concepts described earlier need to be incorporated in a visualization system offering an aggregation of well-organized and distinct information which the user may perceive as intuitively as possible.

The overview of the timeline events should be presented in a manner familiar to the user, easy to understand and convenient for navigation. In this direction, the Classic 2D Mode of TimeTunnel can be thought as a two dimensional grid where events are all placed. The one axis of the timeline (X) should represent time, in a way similar to the concept present in all timelines. The second axis (Y) is used for the placement of events occurring at close time intervals which would overlap if all events were placed in a row along one axis. In order to achieve this, an algorithm was developed which would resolve all events' bounding boxes collisions with each other; the algorithm displaces events whose bounding boxes collide in temporal order, moving events which occur at a later time below. Apart from collision detection avoidance, the algorithm's consistent acting offers another perception found in literature, the separation of time to macro and micro: macro time is visualized on X axis and micro time on Y axis. As a result, the user who explores the timeline is able to perceive the events' occurrence order at a glance, as their sequence is shown in a clear way – events which are moved below happen afterwards - and the exact moment they take place is shown inside the box they are visualized in.

Events are represented in Classic2D Mode as boxes linked with a vertical line to the time axis according to the exact occurrence time. The box contains the event's title in the center and its occurrence time is placed at the top part of the box. Furthermore, user may navigate through the event's multimedia elements using multimodal interaction techniques.

- **Instant events** in Classic 2D Mode own a vertical line to the time axis according to the exact occurrence time. Its occurrence time is placed in the middle of the top part of the box.
- **Events with duration** are represented in the same way as instant events in Classic 2D Mode, except for the fact that start time and end time are placed on the left and right side accordingly. Events are placed as if they occurred at their start time, but the end time is also displayed so that users may understand that they took place over the specified time. The initial approach of visualizing events with duration was different, as events were placed in the middle of their start and end times. The advantage of this approach was the more precise representation of time but had one significant drawback: the overview of all events became over-complicated and unclear, as lines extended in parallel to time axis. Particularly in the cases where events lasted long time, the vertical lines would be far away from events perplexing their exact time illustration.

The hidden state of events does not completely hide the events, but on the contrary displays them as non-interactive items existing only to provide contextual information in a manner meaningful for both the Tunnel and the Classic2D Mode.

5.3.2.1 Events

Event visualization in the Classic 2D mode is done inside a container. Initially this container was a two dimensional sprite rendered behind all event information, but was replaced by a three dimensional box-like container containing five solid faces, leaving the front face empty in order to make hosted items visible. This box better suits the encapsulation of all event components inside a discrete virtual storage item. Finally, the event container includes a dynamic area that acts as background for the event's title and time and is moved in respect to their positioning and scaling.

Events are displayed on a plane under the timeline's periods inside their containers. Events are able to transform from solid stand-alone components to highly interactive elements during information expansion. The user's exploration of event details in Classic2D Mode starts by selecting the desired item and transition to a more expanded view begins: the event container is brought in front of the user and detailed information is presented. The following sections describe in detail the available alternate views of an event's representation in Classic2D Mode.

- Date and Title View

As the camera of the virtual world moves away from the plane on which events and periods are placed, the screen space that each event occupies is reduced. As a result, the events' fundamental characteristics - their title and date they occur – are scaled up and become the only information visible. Event title is placed a little below the center, while date is placed on the upper part of the event container, as is customary about time visualization.

- Representative Item View

The default view of the event contains its date, title and representative item. The representative item is of major importance and is aimed at being indicative of the event, as it constitutes the visual cue that the user will map the event to. Date and time are scaled down a little and moved upwards, in order to make the necessary space for the representative item to fit in and be clearly visible.

Another approach that was studied and implemented, but finally abandoned, was the concept of showing the representative item scaled to fit the event container. This view of the event was designed to be triggered when the user is looking at the specific event. However, the drawback of this view was the title of the event being hidden whilst not offering additional information and was deemed as a surplus to requirements. Furthermore, the inclusion of multimodal interaction techniques offers a rich set of human-computer intercommunication, while the three dimensional space that the camera moves and rotates into, offers an environment of intuitive and easy to manipulate exploration. As a result, the user is able to navigate towards the event with ease, examine the representative item without further needed actions and explore additional information at will.

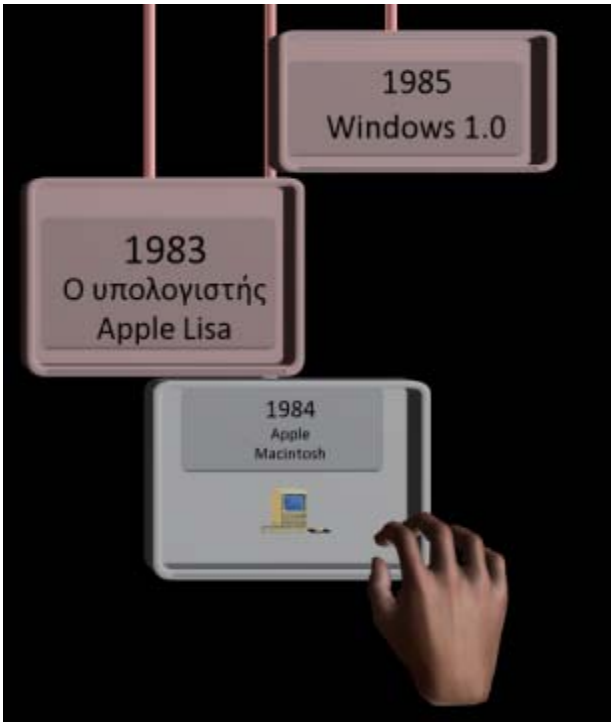

Figure 14: The Representative item (at the bottom) and the Date and Title (in the middle and at the top) Views.

➢ Detailed Information View

The event's content is rearranged to fulfill interaction requirements: event general information is no longer meant to be in focus, as the user has already acquired this information, but should remain visible in order to prevent the user's disorientation. The Detailed Information View of an event intends to exhibit all available information concerning each event. In order for this to be achieved, TimeTunnel shrinks and moves the event's date and title to the upper side of the container, thus creating space in which additional multimedia information can be displayed.

The initial idea of presenting information involved a grid-like layout structure able to present any type of information. The grid is highly customizable and allowed the dynamic optimal display for any count of distinct information, using the set of rules described below.

- One event extends over all available space
- Up to 4 events fit into a 2 by 2 matrix
- Up to 9 events fit into a 3 by 3 matrix
- For more than 9 events, an N by M matrix is created (with M, N >=3) of which only a 3 by 3 part is displayed. Through interaction, the user is able to explore additional items beyond the visible ones, simply by moving the visible part in one direction, up,

down, left or right. The matrix appears to be never-ending, as at the end of the total matrix, the items are looped (M+1 item is the first item, M+2 the second etc.).

This process works perfectly in the special case where the item count is equivalent to the power of a number. In the general case where the count of event is not square, the algorithm implementing the grid finds the nearest possible factors sufficient to fit all items in. For instance, if 11 items exist, a 4 by 3 matrix is created with 3 rows with four items and one row with three items aligned to the center. Considering the fact that the Classic2D Mode of TimeTunnel aims to offer the general concept of the distribution of events over time along with information, the item placed in the middle of the three by three matrix is brought a little forward in order to appear larger, uncovering details of the item which are otherwise obscured and indicating the fact that the item is currently seen by the user. Interaction with the items' grid is accomplished through panning over the grid: the user is able to visit all items by moving the grid in any direction (right, left, up, down or any combination of these directions).

This approach has the advantage of displaying a large set of informative items concurrently, while the centered item is evident because of it having been brought to the front. On the other hand, conjunction of information is apparent and leads to a complex view of the event's information.

Another idea which is common in literature apart from the grid approach involved the implementation of a virtual interactive carousel-like row of all components available for exhibition. In carousel style information display, items are placed in a row with the focused item in the center brought closer and the others further away at the sides. When the carousel is shown, the box containing all event information is transformed and buttons rise from the sides, allowing the user to navigate to items to the left or the right of the currently centralized item. Apart from box buttons the user may explore items using multimodal interaction techniques such as hand gestures.

The carousel-like row of multimedia elements has a very simple and natural conceptual model by displaying items in a row and scaling down items further away from the currently seen one. The major drawback of this design is the limited count of items displayed simultaneously, as well as the extensive space required.

Finally, another solution considered, implemented and finally adopted was the idea of a transformation of the event container to host additional elements, similar to a carousel with

only three visible components. In this case, the focused element is rendered in the middle of the event container, whereas the next and the previous elements are displayed on the right and left side respectively. This process is accomplished through an animation of the container at its sides and the creation of additional boxes that are able to contain components. After the boxes are created, selectable items are placed inside them. At the time that any item is selected, an animation similar to scrolling a carousel starts: the selected item is centered, the previously centered item is moved to the other side and a new item previously invisible appears, taking the selected item's place. In the case where no further multimedia component exists, the boxes collapse, indicating that the user has no other items to further explore (Figure 16**)**. Finally, the bottom of the container contains a scrollbar that indicates the count of the available items on the left and on the right of the current item.

The last idea of hosting additional components on the sides of the event container was preferred due to the fact that it offers the same conceptual model with the interaction inside the tunnel, as explained in [5.3.3.1], it makes use of a larger part of the display and is simple for the user to perceive. An effect similar to fog is applied to partly conceal the events at the background in order not to distract the user's attention.

Figure 15: The detailed view of the event in Classic2D Mode.

The focused multimedia element is placed in the center of the display, whereas the next and previous are placed on the side. The bar at the bottom of the container indicates the count of the available items on the left and on the right of the current item.

The focused multimedia item may be selected and explored further by selecting it. All the available interaction techniques of TimeTunnel are offered, according to the type of the item shown.

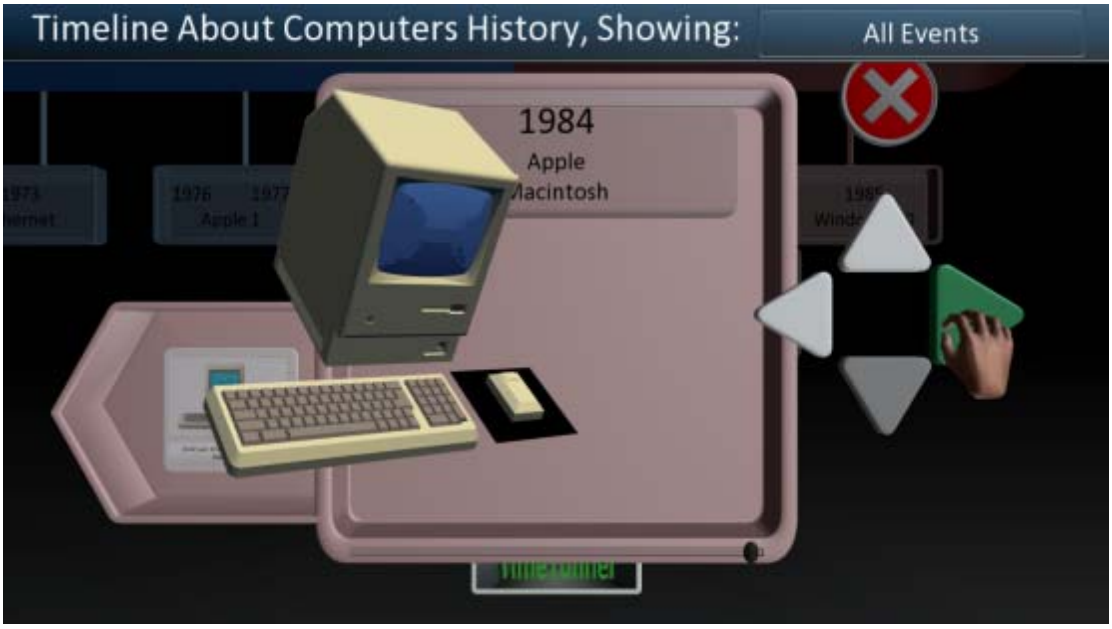

Figure 16: The focused item (a three dimensional model) is brought in front of the user.

The buttons shown allow its rotation on demand. Due to the fact that there is no event after the focused one, the box on the right of the container is closed and the sphere is on the right edge of the bar.

- Categorized Event View

Active categories define the set of events which are displayed normally as well as the “hidden” ones. The “hidden” state of events does not completely hide events for two main reasons: firstly, hidden events are actually part of contextual information regarding the rendered ones, and secondly, hiding or showing a large set of events would mean a significant rearrangement of the whole timeline that could make the user become confused. By contrast, the proposed approach makes all event components (title, date, container and representative item) partly transparent and grayed out.

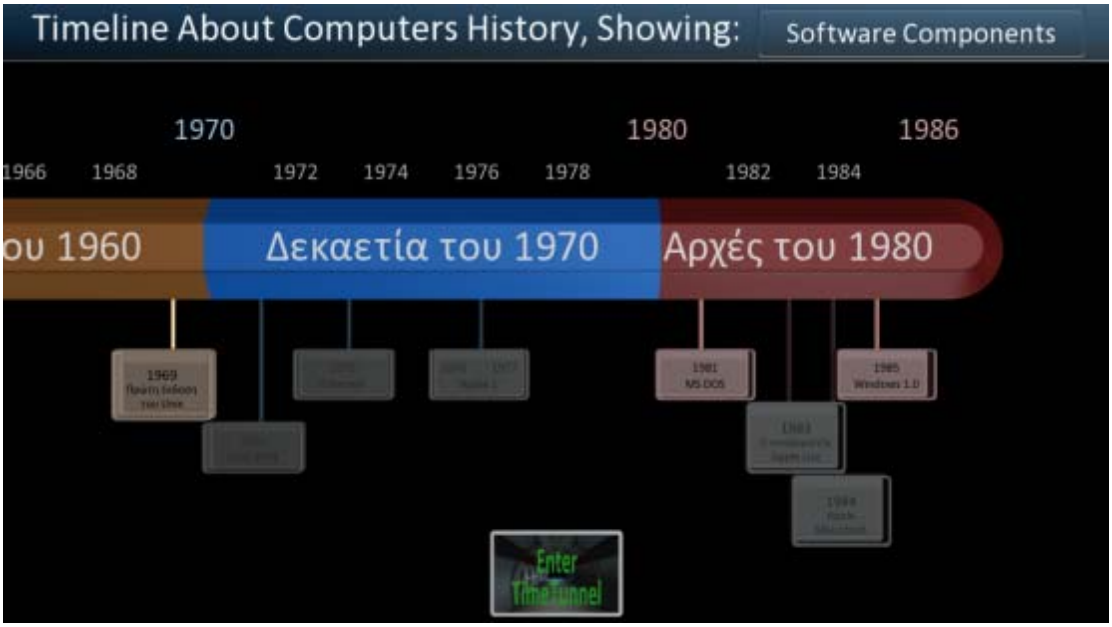

Figure 17: The three events that belong to the category "Software Components" are shown, whereas the rest of the events are grayed out and non-interactive.

5.3.2.2 Periods

The periods in Classic2D Mode of TimeTunnel extend over X axis in a row. The row is placed in the top side of the virtual world and act as an upper boundary, below which the user may explore available information.

- Period Exploration

Periods visualized in Classic2D Mode are not static objects, but interactive elements with which the user has the ability to interact. Upon selection, the camera of TimeTunnel moves to the center of the period extent along time axis. Furthermore, events belonging to the selected period are temporarily highlighted, emphasizing the incidents that are related to the specific period.

- Period Preview

In the case where the cursor hovering over an item metaphor is meaningful, TimeTunnel supports it as a means of interaction. Moving the cursor over a specific period conceptually means that the user is slightly interested in the element and the system reacts by highlighting the events that reside in the particular period. This option may not be available, if the interaction techniques used are reciprocal to the concept of hovering with a cursor. Such an indicative example of hovering unavailability is the incorporation of touch screens as the only mean of communication with the system.

5.3.2.3 Contextual Information

- Time Legends

Timelines arrange events over one axis according to the exact moment at which each one takes place. As users explore information, time extent needs to be perceived by the user in order to allow temporal event classification. Furthermore, time scaling inside periods makes the need of providing time legends imperative. In order to overcome this issue, the Classic2D Mode of TimeTunnel offers enhanced time legends which appear over periods which adapt to the level of detail the user is exploring timeline information.

The distinct values that the maximum detail level may be extend from seconds to years. An algorithm was created that allows the display of time detail on different levels that is described in the following paragraph applied to year's partition. However, the algorithm works respectively in seconds, minutes, hours, days and months.

The values of maximum detail that years may take are 1, 2, 5 and any of the aforementioned 3 values multiplied by a power of ten. The maximum level of detail the years have depends on each period's duration and is found by selecting the minimum interval at which the legend is clearly visible. The procedure to find the maximum detail level starts from months and continues increasing (1, 2, 5, 10, 20, 50, 100 etc.) until the minimum clearly visible interval is found. Apart from the maximum level of detail, the procedure continues using the same pattern to identify three additional sets of legend values. According to the distance that the camera has from the timeline (which depicts the distance at which the user looks at the timeline), the corresponding set of legends are shown, avoiding overcrowding the scene with redundant information. Furthermore, when not showing the maximum detail level, legends are scaled up in order to remain clearly visible, as they appear smaller due to increased distance from the camera.

The level of detail used for each period are only dependent to the period's duration, as time may not be linear throughout the whole timeline in the case that a period's extent is increased or decreased. For instance, consider two periods that have the same extent in timeline, one covering 1000 years and the other 100. In this case, if the legends of the first period are shown every 500, 200, 100 and 50 years, then the second period's legend sets would be shown every 50, 20, 10 and 5 years, according to the distance of the camera from the plane over which periods and events are shown.

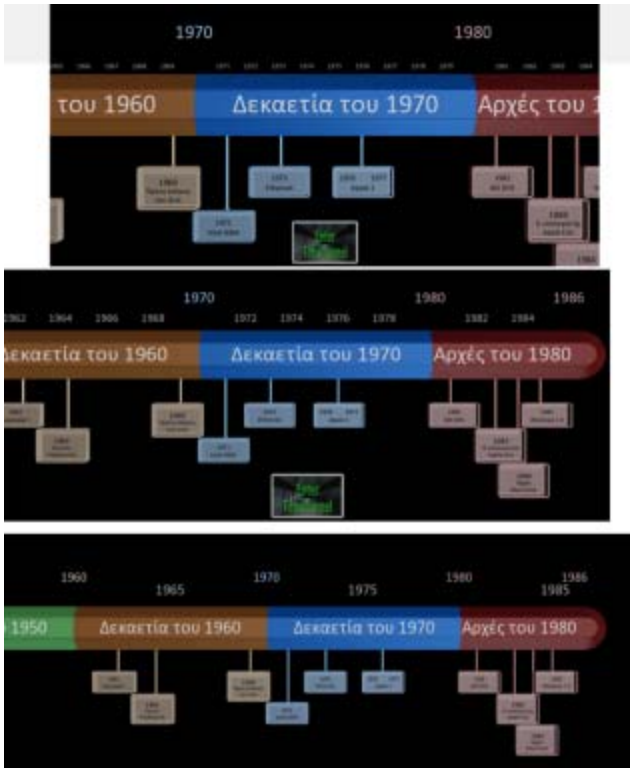

Figure 18: The different detail level that time legends have according to the distance at which user looks at the timeline

5.3.2.4 Transition to 3d mode

A button labeled "Enter TimeTunnel" with an image of the tunnel is placed at the bottom of the display, which the user may select to explore information inside a virtual tunnel. Upon selection, the camera in the three dimensional space moves to a certain predefined position on Z-axis, at which the display is almost identical to the display rendered on a part of the tunnel's minimap and an animating procedure begins to eventually match the minimap display.

The transition from Classic2D to Tunnel Mode is achieved through interpolating the timeline display at the time of change to the corresponding display in the minimap. The issue to be dealt was the fact that perspective display should match orthogonal display: TimeTunnel uses a perspective camera with a field of view of 40 degrees, whereas the minimap is captured with orthographic projection. In order to achieve results similar to orthographic projection and match the display of items on the minimap, the camera's field of view is narrowed down significantly to 5 degrees for the capture of the timeline minimap, rendering all objects at the very wide minimap almost flat. As a result, the minimap view remains almost the same from the beginning to the end of the timeline extent as if it was rendered with orthogonal projection allowing the transition to be smooth.

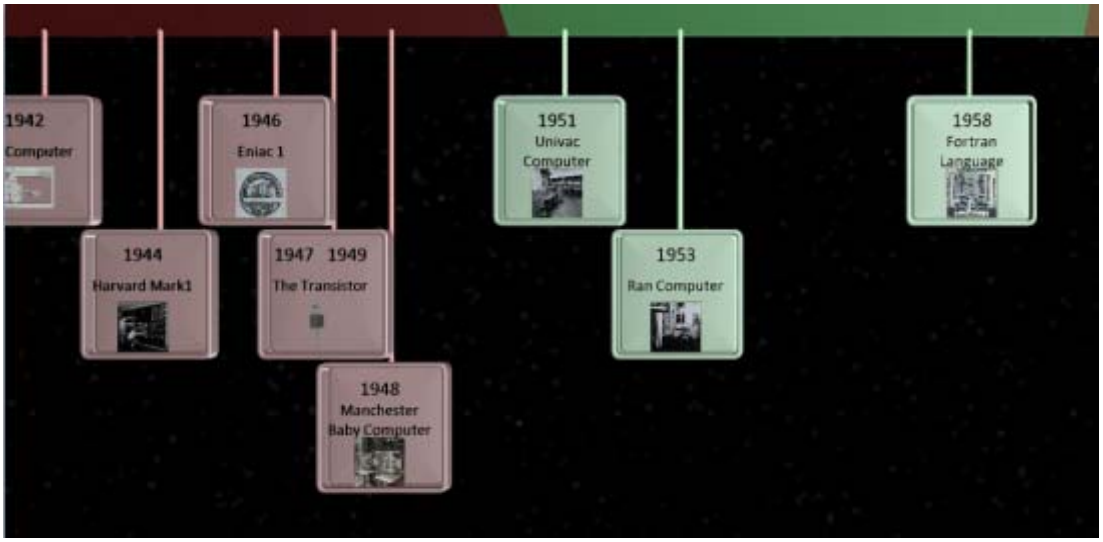

Figure 19: View with a field of view of 45 degrees

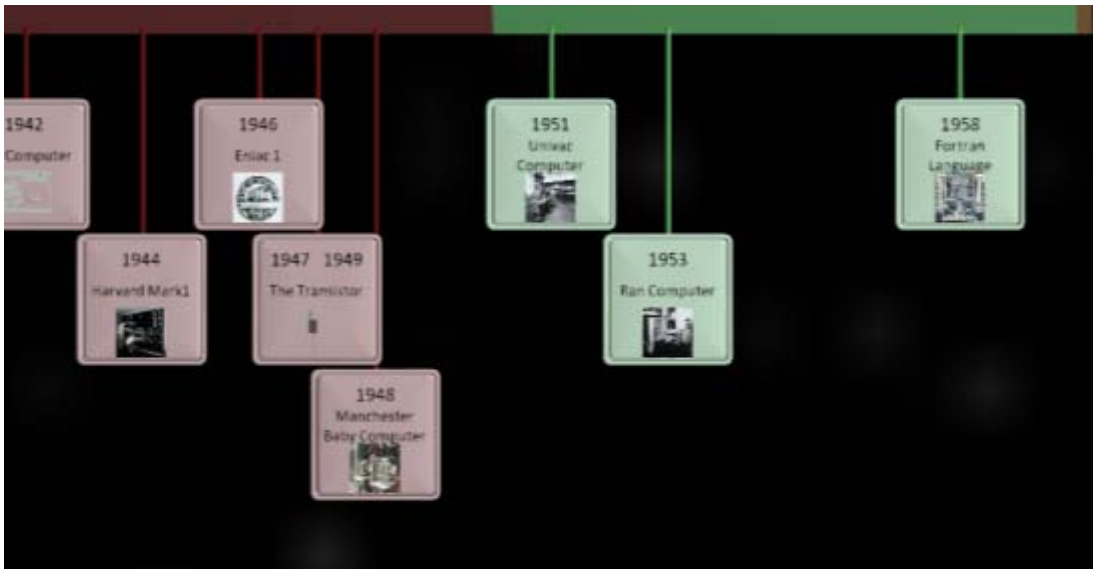

Figure 20: View with a field of view of 5 degrees

While the view of the container of “Fortran Language” (on the right) remains almost the same in both images, the view of the container of “Harvard Mark 1” changes significantly.

Because of the fact that narrowing the field of view is equivalent to zooming in, and rendered objects appear larger, an algorithm was developed which moves the camera backwards according to the current field of view angle, in order to render exactly the same plane at the depth of the timeline visualization regardless of the field of view.

This algorithm is enhanced with an interpolation mechanism, which narrows the field of view and moves camera backwards respectively as needed, resulting in a visual animation of flattening the three dimensional view to a two dimensional image. The process of this interpolation is equal to moving away from a target while zooming in as much as necessary for the target to have the same size. At the end of the interpolation, when the field of view results in the rendering the timeline similarly to minimap's capture, the railway wagon starts fading in, from transparent to opaque, placed exactly at the position needed to result in the same outcome as the current display. The minimal differences because of the transition of the camera on the X-axis are veiled by the gradual appearance of the minimap, creating the illusion of freezing the display.

At the time that the railway wagon is completely opaque, both the camera and the railway wagon are instantly placed on the Tunnel Mode's position equivalent to the current time examined in Classic2D Mode while all objects rendered in the Tunnel Mode are placed accordingly in order to create the new scene. Afterwards, the camera of the virtual world starts moving away from the minimap, following the conceptual model of stepping out of a box: the camera's field of view increases while moving backwards and downwards, until it reaches its final position in Tunnel Mode and re-enables user's interaction with the timeline in its new mode.

5.3.3 Timeline Visualization as a walking path of a Virtual Tunnel

TimeTunnel aims to provide information visualization in different modes and formats, providing alternative ways to depict information. The fundamental aim of a timeline is to provide an overview of events over time. However, another common procedure when exploring historical information is detailed and often exhaustive information exploration in a chronological sequence. As Classic2D display offers just the general picture of the periods along with occurring events, the need arises to support another more detailed view where one event is shown at a time.

This task is accomplished through the second supported mode of TimeTunnel, the **Tunnel Mode**. Sequential navigation may be of great benefit so as to iterate through all events visualized, arranged in chronological order. On the other hand, exploration in a virtual space is not limited to strictly defined movements but involve travelling in a free mode and looking around. The first approach to match this requirement was creating a virtual road and place events on the sides. However, navigation in three dimensional virtual spaces often causes user disorientation, especially to novice users who are not familiar with

virtual environments and six degrees of freedom. As a result, the need of visual cues for the definition of axes arose for the aid of perception of orientation in the virtual three dimensional spaces: navigation in three dimensional spaces requires the limitation of movement to a subset of virtual space, as it is infinite otherwise. A tunnel is by definition an underground passageway completely enclosed, except for opening and ending. Tunnels' extents are by definition long and relatively narrow making them a suitable model for a timeline implantation. Furthermore, the walls of the tunnel have a dual advantage: they offer constraint of navigation in three dimensional spaces to a well-defined part as well as a mean of presenting contextual information, such as lightning and item implantation on the sides.

Therefore, the Tunnel Mode of the timeline has a different concept representing temporal information: it focuses primarily to event information representation, while transforming secondary information to be absorbed by the environment. Instead of using boxes placed inside a virtual two dimensional grid, events are placed on the floor, along the extent of the virtual tunnel. Periods are located on the side corners of the tunnel roof in order to be present for as long as user navigate inside the scope of one.

The basic concepts of the TimeTunnel remain the same, but are transformed in a way suitable to fit the needs of detailed event information demonstration. The most significant change in comparison to Classic2D Mode of TimeTunnel is the fact that time distribution is altered: instead of being linear inside each period, time "freezes" inside tunnel in a region around the event, equal to its tunnel container slice. Without the adoption of this technique, each event would hold an extent of the tunnel around it and regardless of the dimensions, after a certain point, the events' tunnel slice containers would overlap each other. The modification is applied in order to support even the worst case scenario of timeline visualization, where large counts of events occur in a short time, allowing the placement of infinite events sequentially. As a result, all events can be clearly and distinctly visualized, regardless of the level of time detail, which may vary from years to hours and seconds. Furthermore, the distance between events remains constant and linear, depicting the interval that intervenes between their occurrences.

The Virtual Tunnel's components are analyzed in detail in the following sections, explaining requirement demands and design problems alongside proposed solutions.

5.3.3.1 Tunnel Events

The virtual tunnel is actually a sum of tunnel parts, which may be thought as slices that are placed in a row. Each slice of the tunnel may either belong to an event or be just a part of the tunnel generated to connect the sequence of event slices.

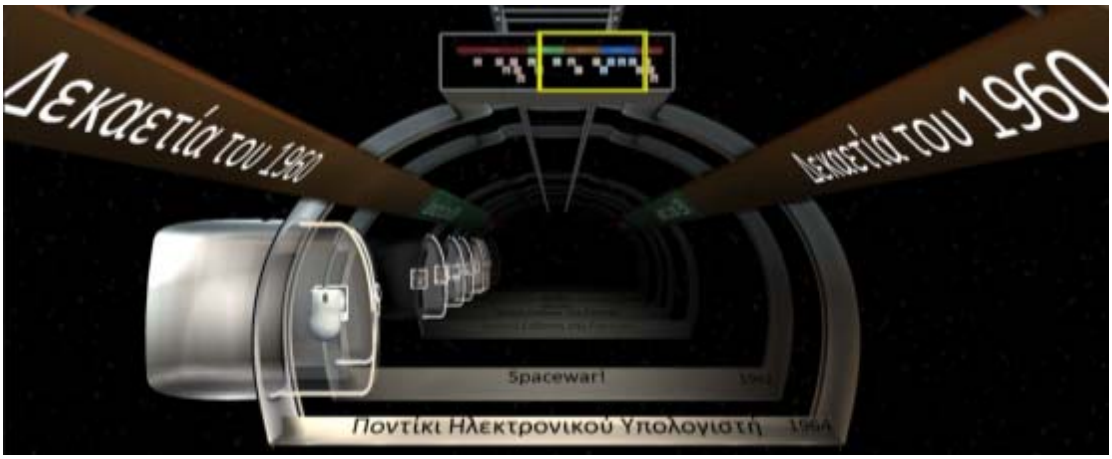

Figure 21: The view of the tunnel without the tunnel gaps placed between the events' slices.

Each event is placed along the virtual tunnel according to its time of occurrence. Event information is placed inside the event slice, with its title displayed in the middle of the floor and its exact time on the side at all times. Furthermore, a small cave with a showcase on its front is placed on the one side of the walls of the tunnel. The cave provides space for the implantation of each event's representative item, whereas the showcase operates as an indicator of interaction as it draws the user's attention. The showcase is a three dimensional model virtually made of glass and extended by a handle which the user may virtually drag out in order to explore the event's additional information.

The events shown in the tunnel contain three main modes:

1. The **default** event state, where the only visible information about it is its title, its time/date and its representative item inside the showcase.

2. The **preview** event state, which is similar to the default state, but its showcase glass along with its representative item is brought forward, illustrating the ability of the user to interact with it. Furthermore, TimeTunnel lightens the showcase, by placing a point light just outside it and the floor on which the event title is placed is lowered.

3. The **expanded** event state, where users may explore additional information related to the event. The showcase glass expands outwards and becomes solid (by reducing

its transparency and becoming opaque), while the multimedia information is shown to the user.

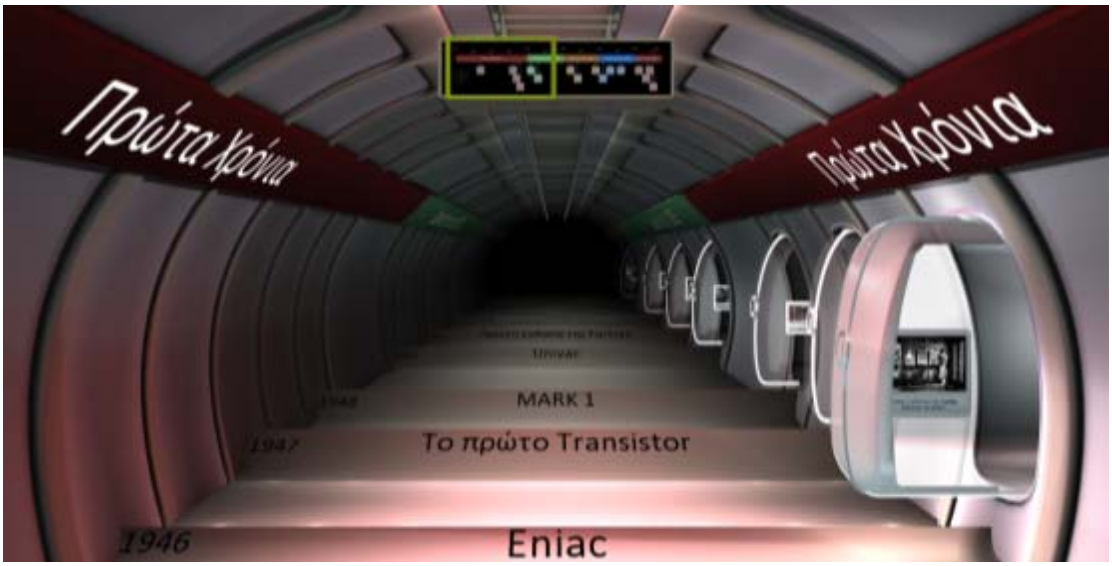

Figure 22: The events' Preview and Default Views.

In Figure 22, "Eniac" is the only event in the preview state (its showcase glass is brought forward and its title is lowered), while all the other events are in the Default state.

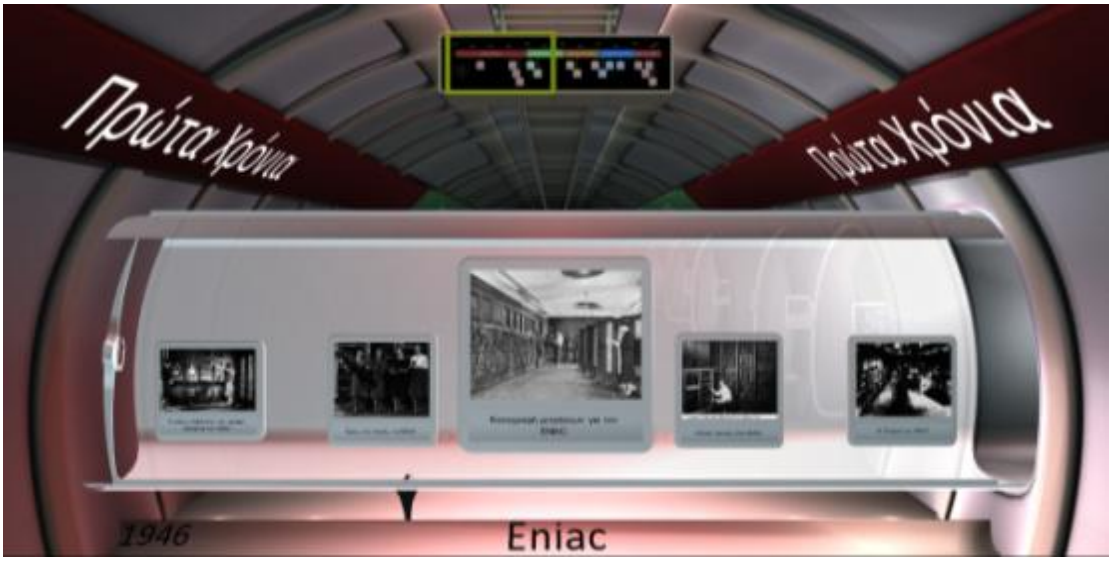

Figure 23: The Expanded View of Tunnel Mode

Events are all in the default state, except the one which is currently in front of the camera. That event switches to the preview state and becomes active for interaction: user may pull the virtual handle placed at the outer side of the glass in order to explore additional information.

The events that have duration are represented as two distinct events, one at the event's beginning and one at its end. The event at the beginning is fully interactive and identical to the representation of the instant ones. On the contrary, the second event's title and time stamp is grayed out to indicate that it is an event's end and behaves differently. The Tunnel Expanded View of the end of the event is altered, not showing the event's detailed information, but informing the user that it is an event's end and displaying an informative text and a button which the user may press to visit the event's start.

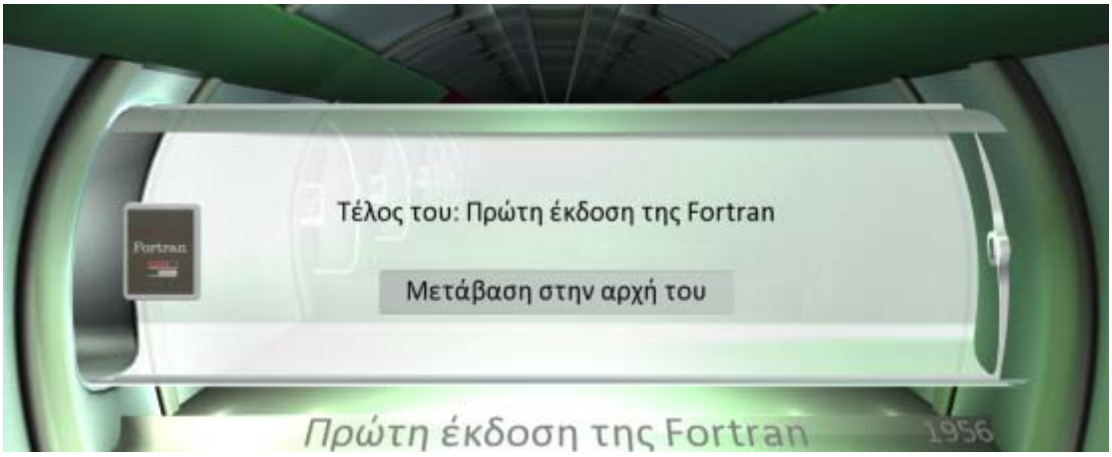

Figure 24: The Expanded View in Tunnel of the end of an event with duration

- Categorized Event View

The concept of not completely hiding the events that do not belong to the currently selected category embraced in Classic2D Mode is adopted in the Tunnel Mode of the TimeTunnel, but expressed in a different manner. The events that do not belong to the currently selected category completely collapse on the axis that the virtual tunnel extends. However, their presence is still clear, due to the fact that the tunnel's union with each event's collapsed slice is specially designed so as to be different in comparison to the rest of the tunnel's walls, allowing the user to identify the presence of an event at that spot.

Figure 25: All categories visible in Tunnel Mode

In Figure 25 and Figure 26 the same part of the tunnel is shown, displaying all the events of the timeline above but only the software components below. Two events have collapsed (the first and the third one) in comparison to the image above, but the seams where the events have collapsed are still visible, indicating a “hidden” event.

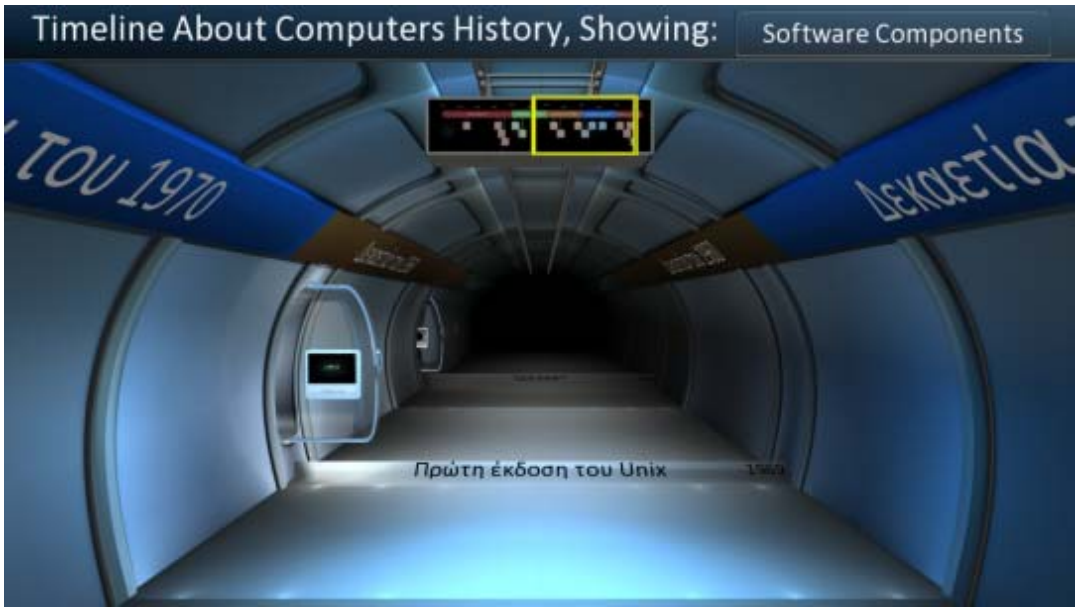

Figure 26: One category selected in Tunnel Mode

5.3.3.2 Periods

Periods are planted in a row over specifically designed gaps on both sides of the tunnel roof. The color-coding of the periods remains the same, offering a consistent to Classic2D abstract model of discrimination of time to periods, enhanced with lightning effects to endure the classification of evens to periods.

Each period's title is placed along its extent and moves according to the railway wagon. When the railway wagon's position is inside the dimensions of the period, the title of the period is highlighted and placed in front of the user, at the wagon's depth. On the other hand, the other periods' titles move as close to the user as possible while being in front of the periods they belong to.

5.3.3.3 Contextual Information

- Tunnel Lightning

Tunnel's roof and side walls constitute a virtual area where contextual information may be planted. The lightning of the TimeTunnel has been studied to be sufficient to make objects rendered inside tunnel clear enough and not too bright, tiring the user's eyes.

Changes in lightning can be used to make the perception of environmental changes clearly and unobtrusively, provided that the component's display remains clear. Following the metaphor of period color-coding, navigation in Tunnel Mode is enhanced with a point light placed in front of the camera serving a dual purpose: firstly, the light denotes the currently explored period in an ambient way and secondly it lightens up the area in front of the user, increasing the brightness of objects in front of the camera and making objects further away appear darker.

- Railway Wagon

The railway wagon's position holds the role of stating the current time frame examined in the Tunnel View. It is placed in front of the camera in a distance far enough for the user to be able to watch the walls of the tunnel, but near enough for the items seen to be clear.

Furthermore, the railway wagon frame moves along the minimap in regard to the current time frame, depicting the equivalent position in the Classic2D Mode of TimeTunnel. This way, the user's navigation is aided and disorientation is avoided, while the functionality

to go back to the Classic2D Mode is accessible at any time in an intuitive manner as the minimap's view is a very wide display of the timeline's overview.

5.3.3.4 Interactive areas of interest

- Event Titles

The virtual camera in the three dimensional tunnel may look either forward, towards the future compared to the currently examined time frame, or backwards, towards the past. As long as an event is not viewed in detail (expanded view), the event's title which is placed on the floor of the tunnel is selectable: upon selection, the virtual camera moves towards the event, travelling in virtual time to the event of the user's interest.

Furthermore, the Tunnel Mode of TimeTunnel allows the user to move along the virtual tunnel in a free mode. Apart from the showcases, event titles which are placed at the side of a triangle prism located at the tunnel floor act as indicators of an event available for interaction. When the user passes in front of an event, the prism animates downwards and the event title lies down, indicating that the current event can be thoroughly examined through the showcase at the side.

- Event Showcase

Once passing next to an event, its showcase is brought forward automatically. Concurrently, it becomes active for interaction: the virtual handle placed at the outer side of the showcase's glass may be pulled outside, offering a detailed view of all information concerning an event.

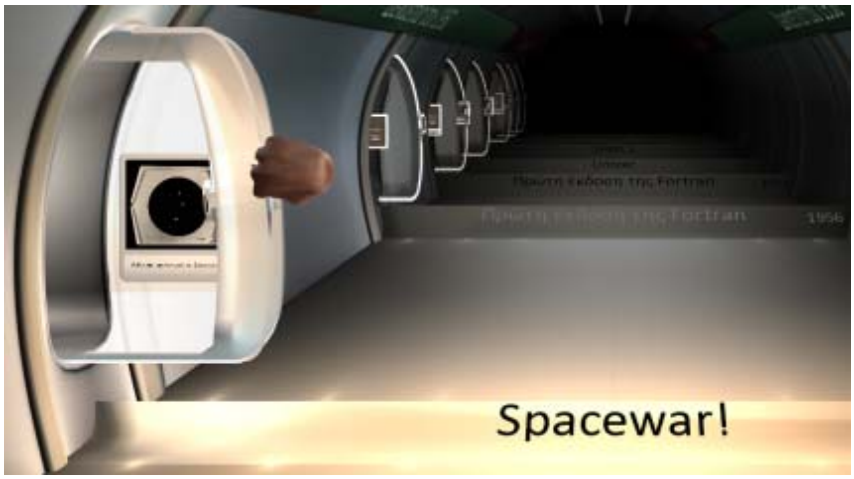

Figure 27: Event Showcase Handle dragging

Upon dragging the handle, the event switches to the detailed state and the showcase expands above the floor and towards the opposite wall while its material gradually becomes opaque, so that it hides the tunnel on the background and emphasis is given to the items displayed inside. The color of the glass is white, but due to the lightning system of TimeTunnel contains a portion of the color of the current period, indicating its belonging period in an ambient and unobtrusive way. The three dimensional hand used as a cursor rotates and grabs the glass handle, indicating that the user may drag it outwards.

The problem that came forth was the need for the faces of the model to be totally transparent in front of the camera and almost opaque at the back, either when looking forward or backward. The approach used to solve the issue was inverting the normals of the model of the glass for the parts of the showcase needed.

- Exploration of Event Information

The event's multimedia elements are rendered in a row inside the solid glass. The row is similar to the Classic2D representation, with the item placed in the center of the glass considered to be the one in focus. However, due to the fact that more space is available to display multimedia elements, the count of elements displayed increases to two on each side. When the glass expands towards the opposite wall, event components also extend in a row. TimeTunnel keeps the order of elements the same, after each interaction. As a result, the user may continue event information exploration from the item viewed before in Tunnel or in Classic2D Mode. The user is able to either navigate to the next element if existing or bring the focused one near and explore supplementary details. Each multimedia component may be further explored in a different way, according to its type.

5.3.3.5 Transition to Classic2D Mode

TimeTunnel offers the ability to change from Tunnel Mode to the Classic2D display by selecting the minimap item encapsulated in the railway wagon. Upon selection, the camera starts moving towards the time frame examined in the minimap, while at the same time the camera's field of view starts narrowing down until the whole display is covered by the currently visiting part of the minimap. Once the display is entirely covered with the minimap, all events are rearranged for the Classic2D Mode and the invert animation of changing from Classic2D to Tunnel Mode begins: the camera with a small field of view is at the beginning far away from the timeline view, with the minimap placed on its front. The minimap fades away (becoming totally transparent), while the field of view increases (and

the camera is respectively moved forward) creating an illusion in which events come to life and items' depth is restored.

5.4 Task Analysis

The following sections describe the tasks that can be accomplished using TimeTunnel. The Hierarchical Task Analysis (HTA) of the available tasks and their sequence is grouped according to the view of the timeline, Classic2D or Tunnel. The execution of each task is not necessarily sequential, as, in general, users have the ability to combine different functionalities in the order they wish.

Due to the large extent of the diagram, some tasks are further analyzed in separate diagrams in order for them to be clear. These tasks are underlined in the overviews of the hierarchical task analysis.

5.4.1 Classic2D Mode

The tasks that Classic2D Mode of TimeTunnel supports are described in the following diagrams:

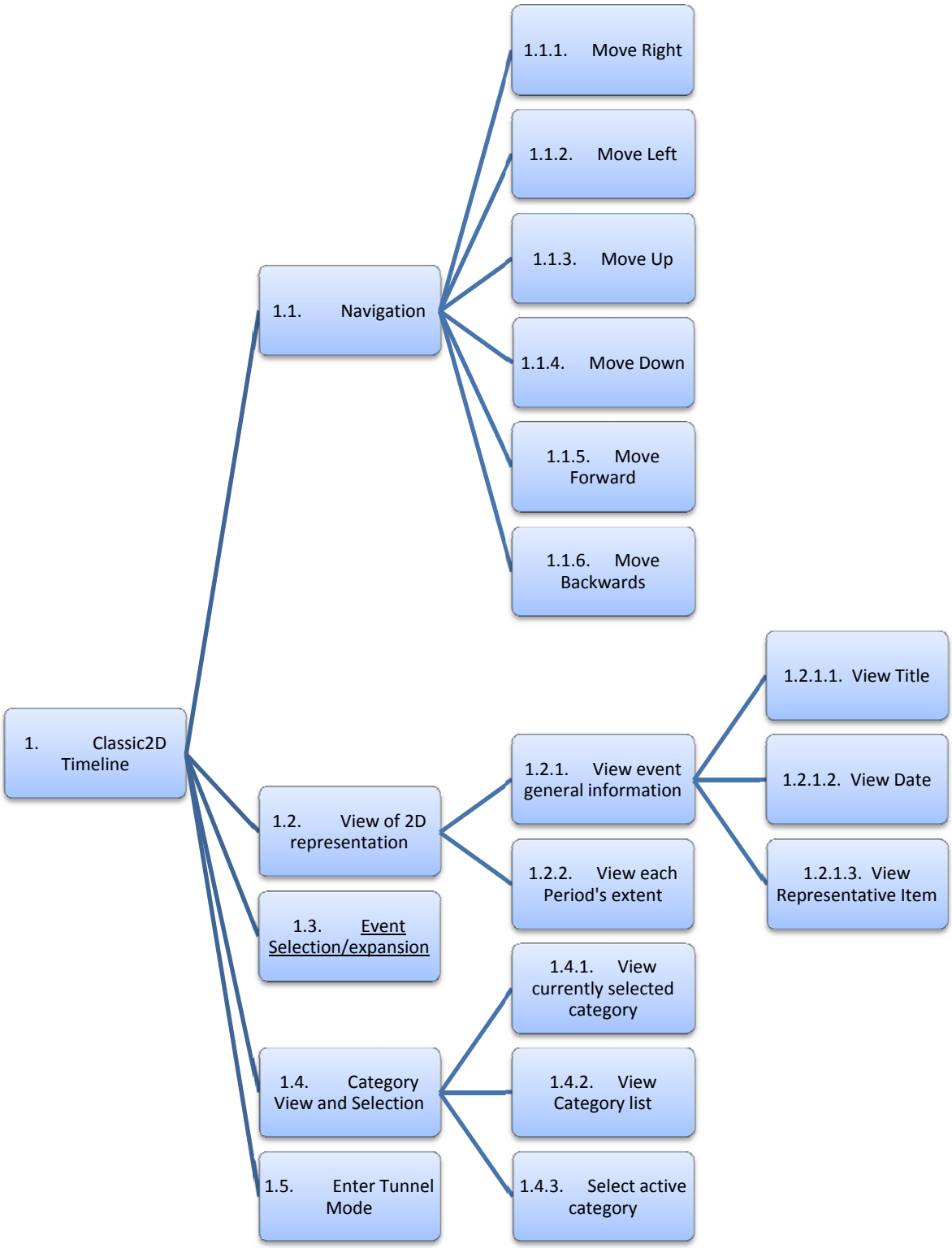

Diagram 1: The Hierarchical Task Analysis (HTA) in Classic2D Mode

Classic2D Diagram Plan

Plan 1: Execute any of 1.1 to 1.5.

Plan 1.1: Execute any of 1.1.1 to 1.1.6.

Plan 1.2: Execute 1.2.1 or 1.2.2.

Plan 1.2.1: Execute any of 1.2.1.1 to 1.2.1.1.

Plan 1.4: Execute tasks 1.4.1 to 1.4.3 in a sequence

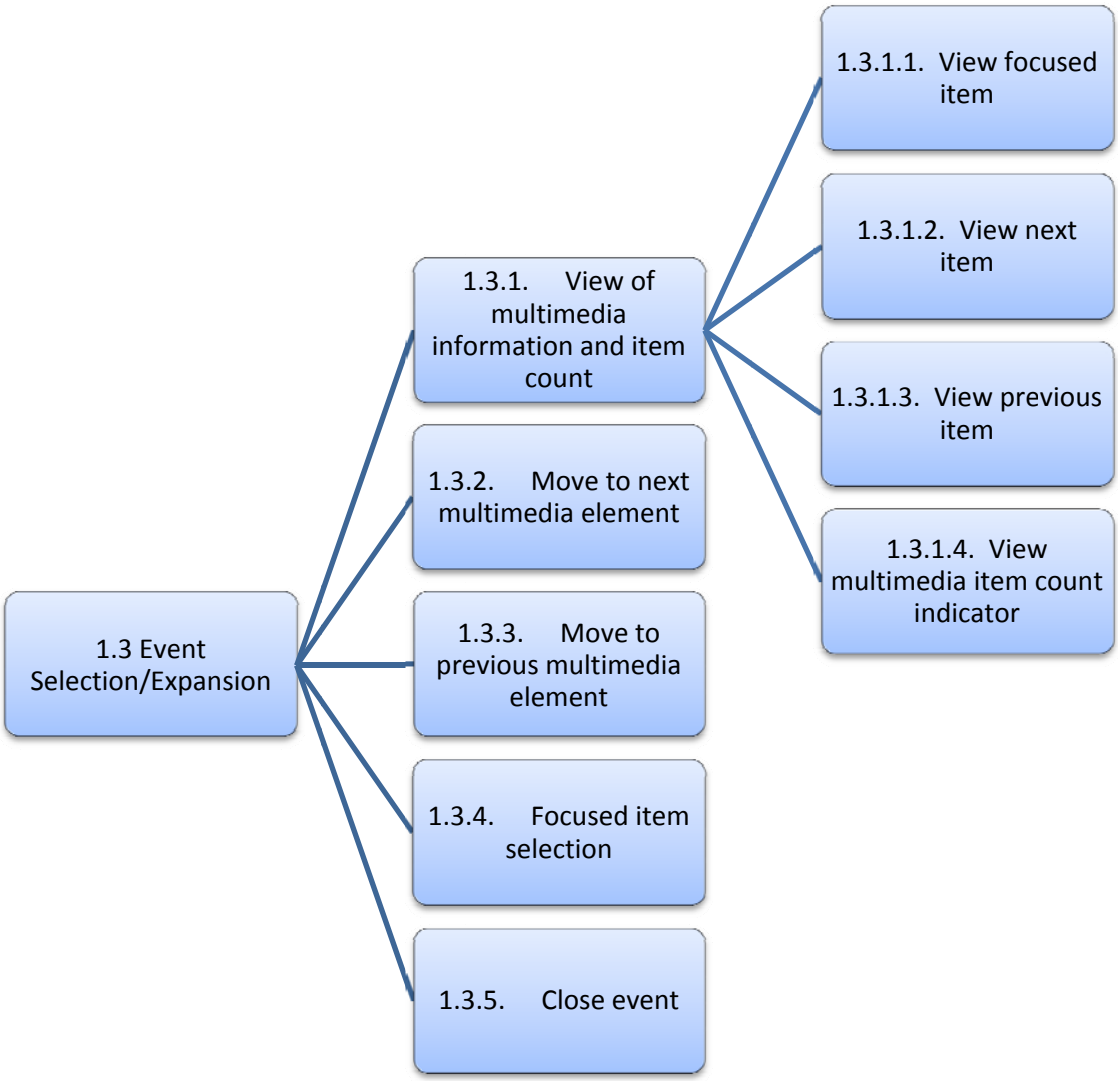

Diagram 2: HTA of Event Selection/Expansion in Classic2D Mode

Plan 1.3: Execute any of 1.3.1 to 1.3.5.

Plan 1.3.1: Execute any of 1.3.1.1 to 1.3.1.4

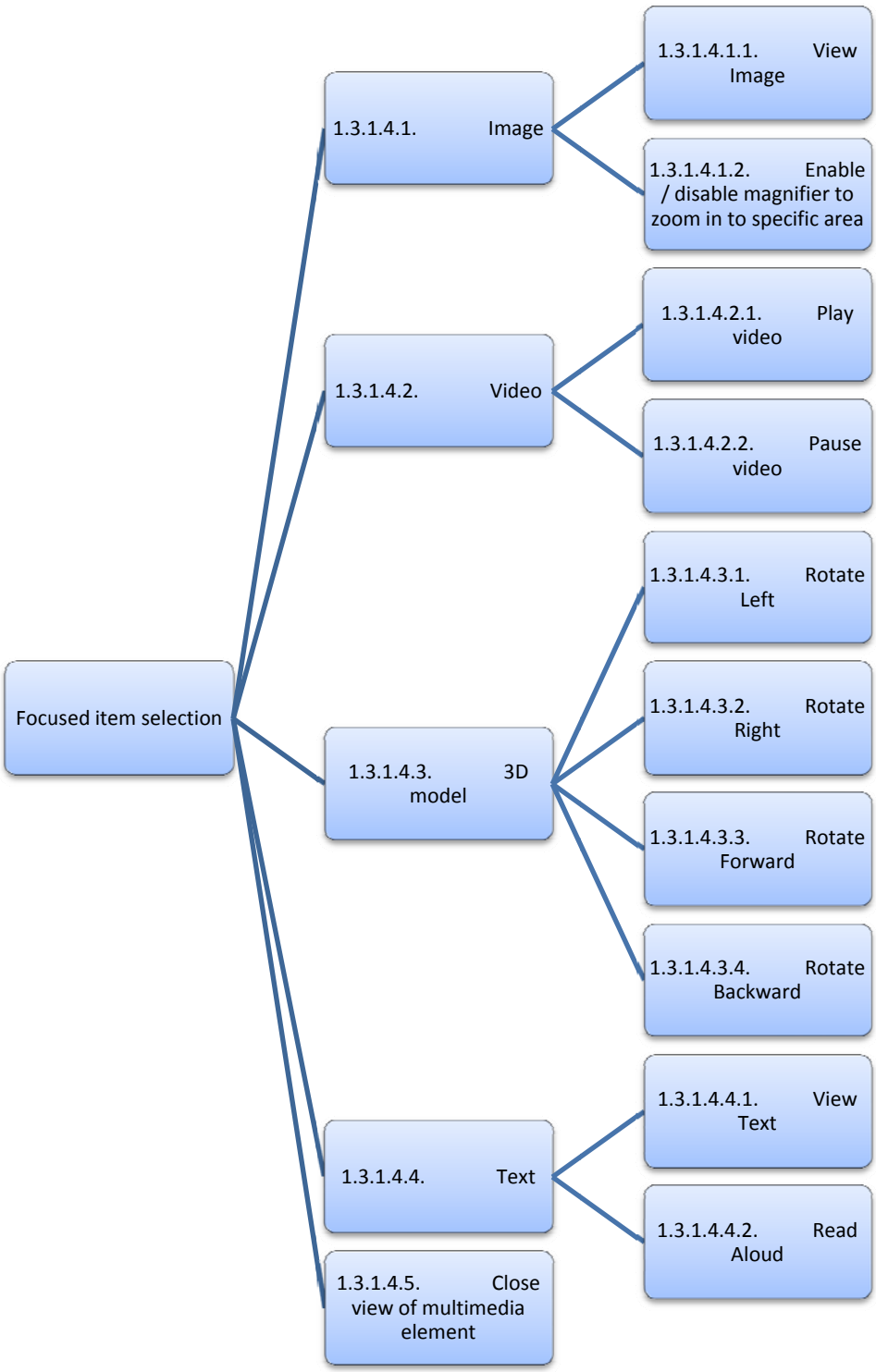

Diagram 3: HTA of Focused Item Selection in Classic2D Mode

Plan 1.3.4: Execute any of 13.4.1 to 1.3.4.5

Plan 1.3.4.1: Execute any of 1.3.4.1.1 to 1.3.4.1.2

Plan 1.3.4.1: Execute any of 1.3.4.2.1 to 1.3.4.2.2

Plan 1.3.4.1: Execute any of 1.3.4.3.1 to 1.3.4.3.4

Plan 1.3.4.1: Execute any of 1.3.4.4.1 to 1.3.4.4.2

5.4.2 Tunnel Mode

The tasks that Classic2D Mode of TimeTunnel supports are described in the following diagrams:

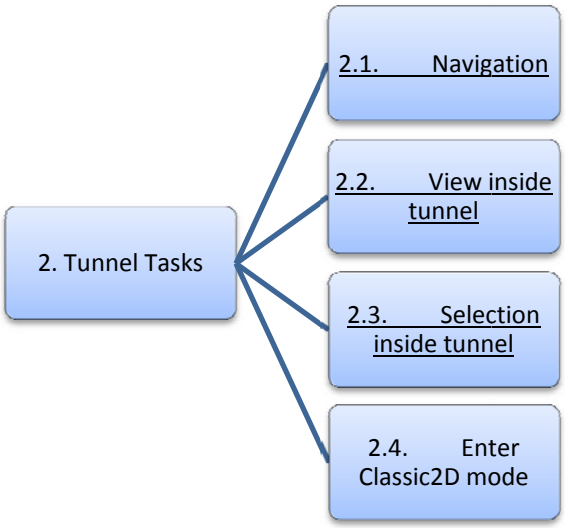

Diagram 4: The Hierarchical Task Analysis (HTA) in Classic2D Mode

Plan 2: Execute any of 2.1 to 2.4

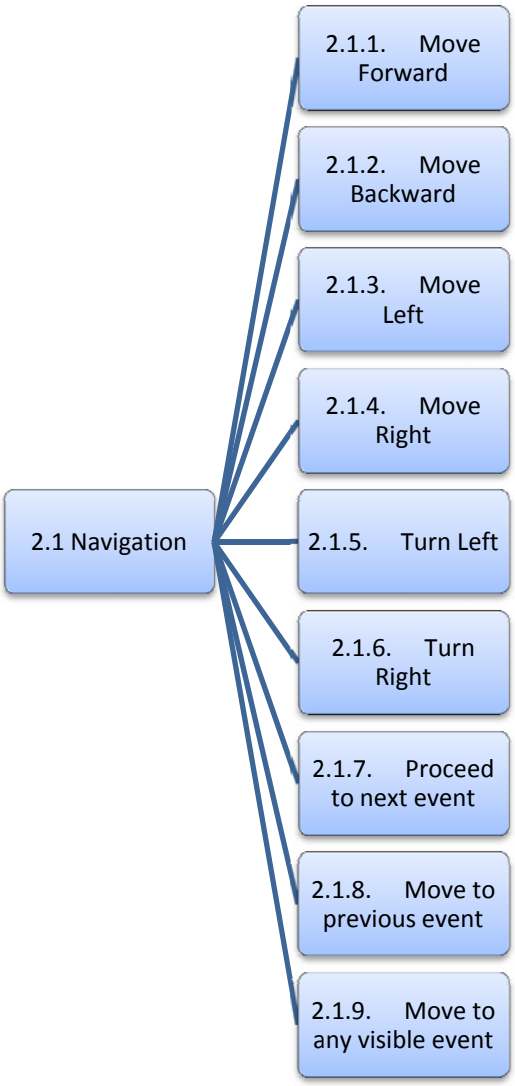

Diagram 5: HTA of navigation inside the virtual tunnel

Plan 2.1: Execute any of 2.1.1 to 2.1.9

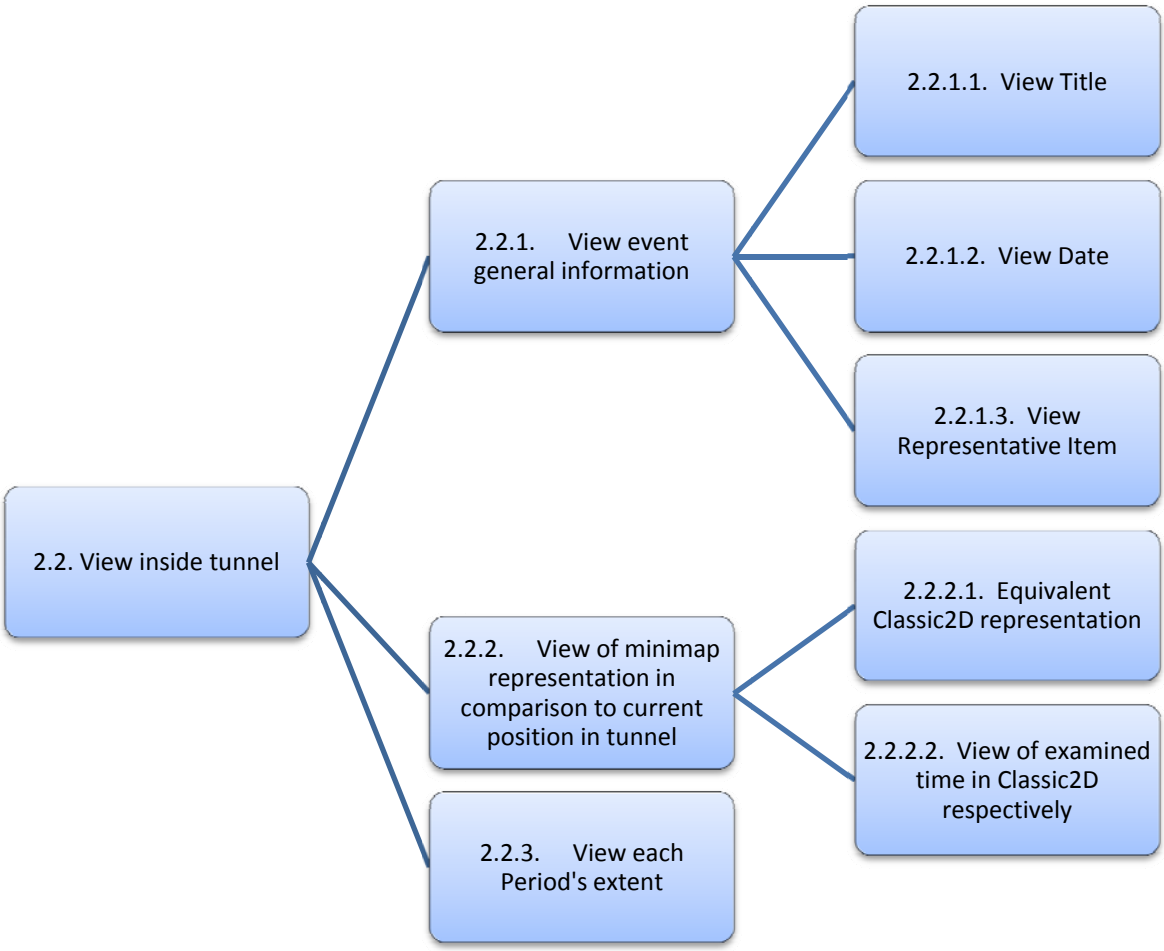

Diagram 6: HTA of the available view tasks inside the virtual tunnel

Plan 2.2: Execute any of 2.2.1 to 2.2.3

Plan 2.2.1: Execute any of 2.2.1.1 to 2.2.1.3

Plan 2.2.2: Execute any of 2.2.2.1 to 2.2.2.2

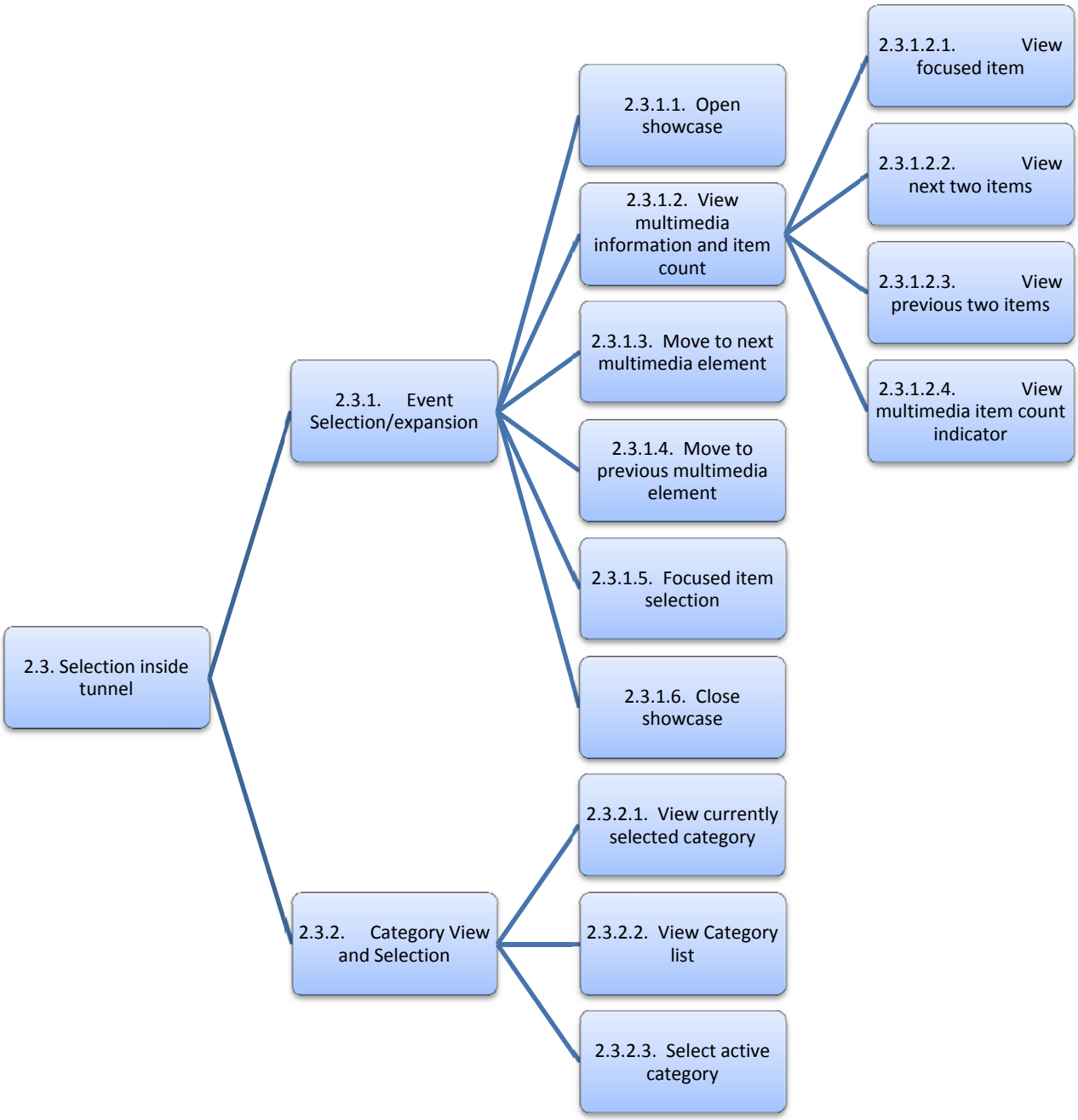

Diagram 7: HTA of tasks regarding selection inside the virtual tunnel

Plan 2.3: Execute any of 2.3.1 to 2.3.2

Plan 2.3.1: Execute 2.3.1.1 first and then execute any of 2.3.1.2 to 2.3.1.6

Plan 2.3.2: Execute 2.3.2.1 or 2.3.2.2->2.3.2.3

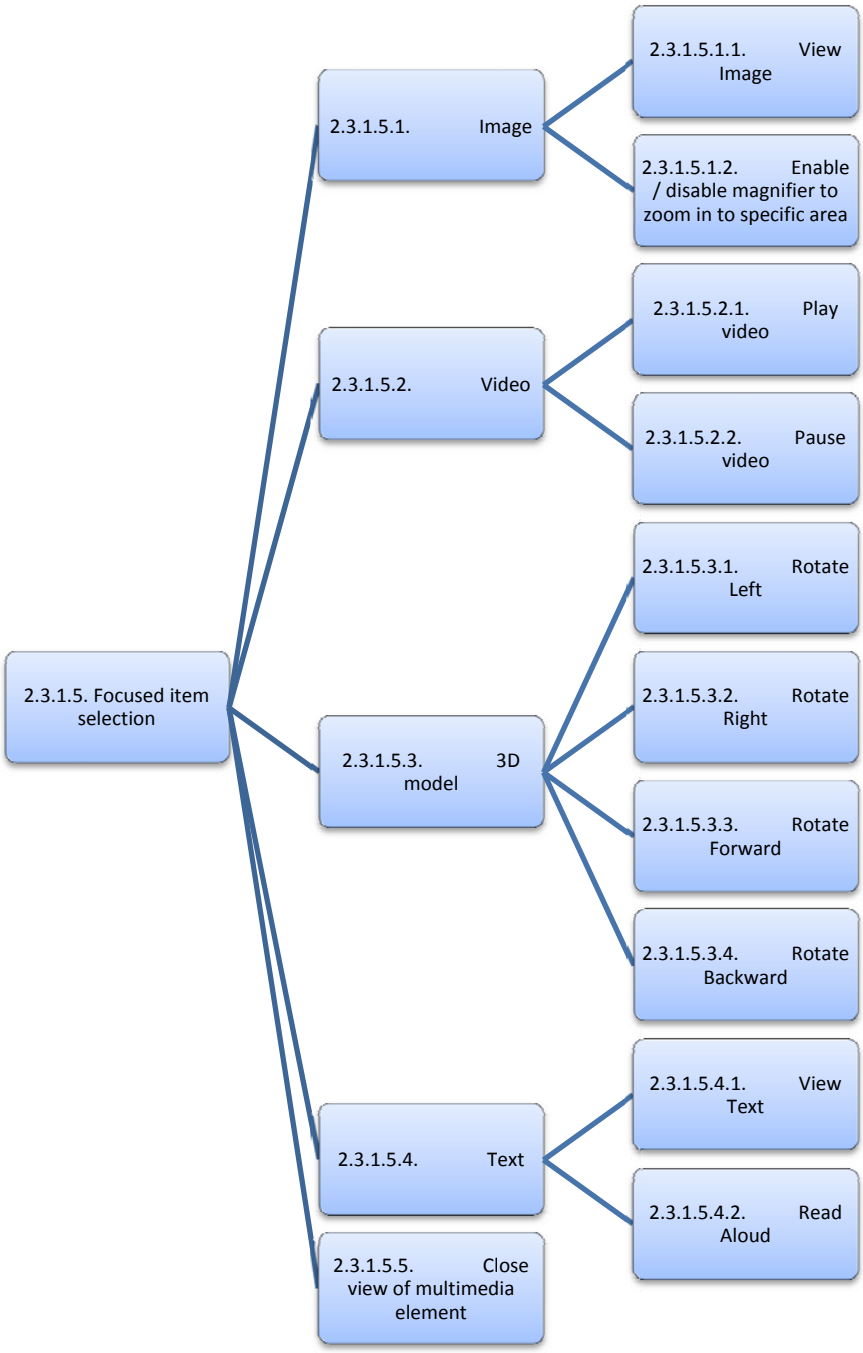

Diagram 8: Focused Item Selection HTA

Plan 1.3.4: Execute any of 2.3.1.5.1 to 2.3.1.5.5

Plan 2.3.1.5.1: Execute any of 2.3.1.5.1.1 to 2.3.1.5.1.2

Plan 2.3.1.5.2: Execute any of 2.3.1.5.2.1 to 2.3.1.5.2.2

Plan 2.3.1.5.3: Execute any of 2.3.1.5.3.1 to 2.3.1.5.3.4

Plan 2.3.1.5.4: Execute any of 2.3.1.5.4.1 to 2.3.1.5.4.2

6 Multimodal Interaction

A variety of interaction techniques were studied in order to conclude to the most suitable ones for the handling of TimeTunnel in a natural way. A major aim of TimeTunnel was the creation of an affordable system that may be set up using an ordinary budget which is context-aware and offers means of interaction that are applied in everyday human communication.

Voice interaction was rejected as an input mechanism due to the limitations that arise from its use. Firstly, the inevitable usage of dialogues is not very helpful for navigation in space and therefore the only approach that would be meaningful in the scope of timeline visualization would be the adoption of voice input as a secondary mean of interaction. Secondly, the input interpretation by the system is not yet robust enough, especially in cases where the user's voice is not clear and loud or noise exists in the background.

On the other hand, the tracking of users with cameras has become moderately stable and has reached a point that their skeletons may be tracked with sufficient precision. The skeleton data that may be generated by the use of Microsoft's Kinect allow the experimentation with body movement as well as gesture recognition using hands and legs. Apart from user tracking, a more traditional approach regarding the usage of touch screens was studied and developed, where the user may navigate in TimeTunnel using simply touch input on a screen. Finally, for the creation of TimeTunnel the application of smart items was considered as a tangible interaction technique.

Moreover, the interaction techniques used for the manipulation of the system should be robust and tolerant to possible user behavior that does not match the exact system specifications: the system should be able to prevent reacting in such a way that may be unexpected by the user. The recognition of a left swipe gesture with the usage of hands, which may be tracked by both hands is a case in point. In many cases, users may firstly move their hands to the right, in order to be able to move their hand to the left, which may be considered as a right swipe gesture by the system. As a result, users will be surprised that although they were trying to take an action with a gesture, the system understood the exact opposite. For the specific problem mentioned, TimeTunnel adopts the concept of the user being able only to make a left swipe gesture with the right hand and a right swipe with the left hand.

TimeTunnel adopts a cursor-like metaphor to provide feedback to the user regarding the actions that take place using his or her hands. The use of a cursor is imperative especially in the remote handling of intangible interfaces, where the user has no clear perception of how his or her movement is handled by the system. The magic approach (the user's actions result in the interaction with the system using common magical beliefs such as ray-casting with a magic wand) is preferred to the natural approach (where the real environment is replicated in the virtual one) for the handling of the system, due to its large extent.

The items existing in TimeTunnel share common characteristics as far as their handling is concerned. Firstly, all the interactive elements may be selected in a way that depends on the interaction technique used. Secondly, other items may be selected and dragged in a direction that is meaningful to the environment and the user.

6.1 Touch screens

Touch screens are a widely used input mechanism that may be used as a single but complete input method. The user's selection point on screen is projected to the virtual world of TimeTunnel and allows the selection and the dragging of items. These concepts are implemented in TimeTunnel and allow the manipulation of the system in a natural way.

Items which may be selected include but not limit to the event's representation in Classic2D, the virtual buttons, the multimedia elements and the categories' representation. These components of the timeline may be brought near the user by touching on them.

Furthermore, items may be dragged towards the direction the user chooses. Such an example is the glass of the events inside the virtual tunnel, which may be dragged outwards if it collapses or inwards in the case it expands.

Items may be also controlled in a combination of ways, such as multimedia elements. The multimedia elements are placed in a row and their selection will result in them being brought in front of the user. Additionally, the items may also be dragged left or right, allowing the interaction of the user with the system in a natural way.

Finally, users may navigate in the virtual world by dragging points of the screen. In the case of Classic2D representation, the user is able to move around on X and Y axes (left/right and up/down) by touching a point on the screen in any direction. Handling of movement on Z axis (backward/forward) is accomplished by the use of a scrollbar, allowing the view of the timeline at the desired distance. In the case of Tunnel Mode, the user may drag the tunnel's

floor in any direction and the viewport of the user will move in that direction. The selection of whether the user looks towards the future or the past of the timeline is accomplished by dragging the tunnel's wall backward, which will rotate the camera by 180 degrees.

6.2 Natural and full-body interaction

Apart from the implementation of TimeTunnel, an individual service, named Kinect Skeleton Tracking Service, which tracks the users' movements in space, was implemented. Microsoft's Kinect was the only hardware component used and the service is built on top of OpenNI [35] modules in C# programming language.

Kinect Skeleton Tracking Service keeps track of the skeletons of all the users within range and informs TimeTunnel only about the nearest one, due to the fact that TimeTunnel is currently single-user. The information that is available contains the user's vital body parts, such as head, torso, hands, legs, feet etc. The information is transmitted to TimeTunnel and the system translates the users' actions with regard to the current context of use.

The following means of interaction were designed and implemented in order to fit the needs of rich interaction techniques in the demanding field of three dimensional environment interaction. The primary concern in natural user interfaces is the movements and the actions of the users to be similar to the ones they use every day. As a result, the metaphors used include pointing with hands, moving hands in a specific direction, virtually pushing and pulling, moving in space and using walk-like poses. Finally, the set of interaction techniques are chosen so as to be as less tiring as possible, keeping in mind the fact that extensive employment of gestures for instance may make the system exhaustive to use.

6.2.1 User Localization and Skeleton Tracking

Kinect Skeleton Tracking Service recognizes all users within Kinect's field of view and specified boundaries which are previously defined. Providing that a user is tracked, the service notifies TimeTunnel about the information described in the following sections.

6.2.1.1 The User's position

The position of the user's torso is sent as long as the user is tracked. The position is then handled by TimeTunnel according to its current state:

- In Classic2D Mode, the camera moves forward and backward in response to the user's distance from the display, allowing the user to see the timeline at the level of detail he or she is interested in. Furthermore, while the user moves from the center of the display to the sides, the camera moves towards the corresponding direction.

- In Tunnel Mode, the distance of the user from the display is not taken in consideration. However, the user's location on X axis – left or right – affects the camera's position in the tunnel, which moves according to the user's current position. This approach creates the illusion of the display extending real world in the display.

6.2.1.2 The User moving his or her Hand

The user may extend his or her hand towards the display in order to interact with the system. The movement of the hand is tracked and while the user has his hand raised, Kinect Skeleton Tracking Service sends the relative to each shoulder position of the hand projected in screen space as well as which hand is raised. The point of the hand is used by TimeTunnel to move a virtual emulated hand cursor in the screen space, while the identification of which hand is used allows the precise representation of the hand used. Due to minor instabilities of the hands' detection precision, TimeTunnel keeps a short queue of the incoming points and uses its average. The queue size is short enough so as not to create irritating latency to the user's actions, but sufficient to keep the cursor steady.

6.2.1.3 Head Tracking

The tracking of the user's head is implemented but is not used in the final release of TimeTunnel, due to reduced accuracy. The head orientation is tracked and sent TimeTunnel, which rotates the camera's orientation, creating a virtual world that the user may look into. However, minor offsets often occur in current implementation, resulting in the camera being constantly moving and eventually unusable. However, TimeTunnel supports head tracking treatment and its incorporation can be done on the fly, when head tracking becomes robust and exact.

6.2.1.4 Leg Gestures

Another form of interaction is gestures with the legs, which are poses that users may use in order to navigate in Tunnel Mode of TimeTunnel. The gestures available include the placement of any of the user's legs forward or backward, matching the metaphor of stepping forward or backward, in order to move in the direction the users want. Moreover, users may move their legs to the side, which causes the camera to move towards the direction the leg points at.

6.2.1.5 The User Turning to the Side and Raising Hand at the side

The user raising hand at the side is a gesture that is tracked when the users rotate their bodies and raise their hand on the side at the same depth with their torso. This gesture is used by TimeTunnel to rotate the camera's view, left or right accordingly. The rotation of the camera in Classic2D is temporary and aims to offer a short view of events in the past and the future. On the other hand, the orientation change in Tunnel Mode is permanent and allows the view of the virtual tunnel at the desired angle, either towards the future or towards the past.

6.2.1.6 Hand Gestures

Kinect Skeleton Tracking Service implements a gesture generator that is able to recognize movement of points in three dimensions. The generator is used to track hand movement of the user's hands separately as well as connected together.

- **Single Hand Gestures** include movement of the hands in any direction, up, down, forward, backward, left and right. These gestures are used in both the Classic2D Mode and in Tunnel mode to navigate through the multimedia elements shown.
- **Both Hands Gestures** which are identical to single hand gestures but occur when the user moves both of his or her hands simultaneously. The directions which are used are backward and forward and have an effect of pulling or pushing respectively any item currently in use. The gesture is used when focusing on multimedia elements, which me pulled near or pushed away and for the end of event examination in Classic2D Mode. Furthermore, the recognition of a gesture forward or backward when exploring the virtual tunnel results in the sequential movement of the camera to the previous or next event respectively.

6.2.1.7 *Summary of User Tracking Interaction Methods*

- Classic2D Mode

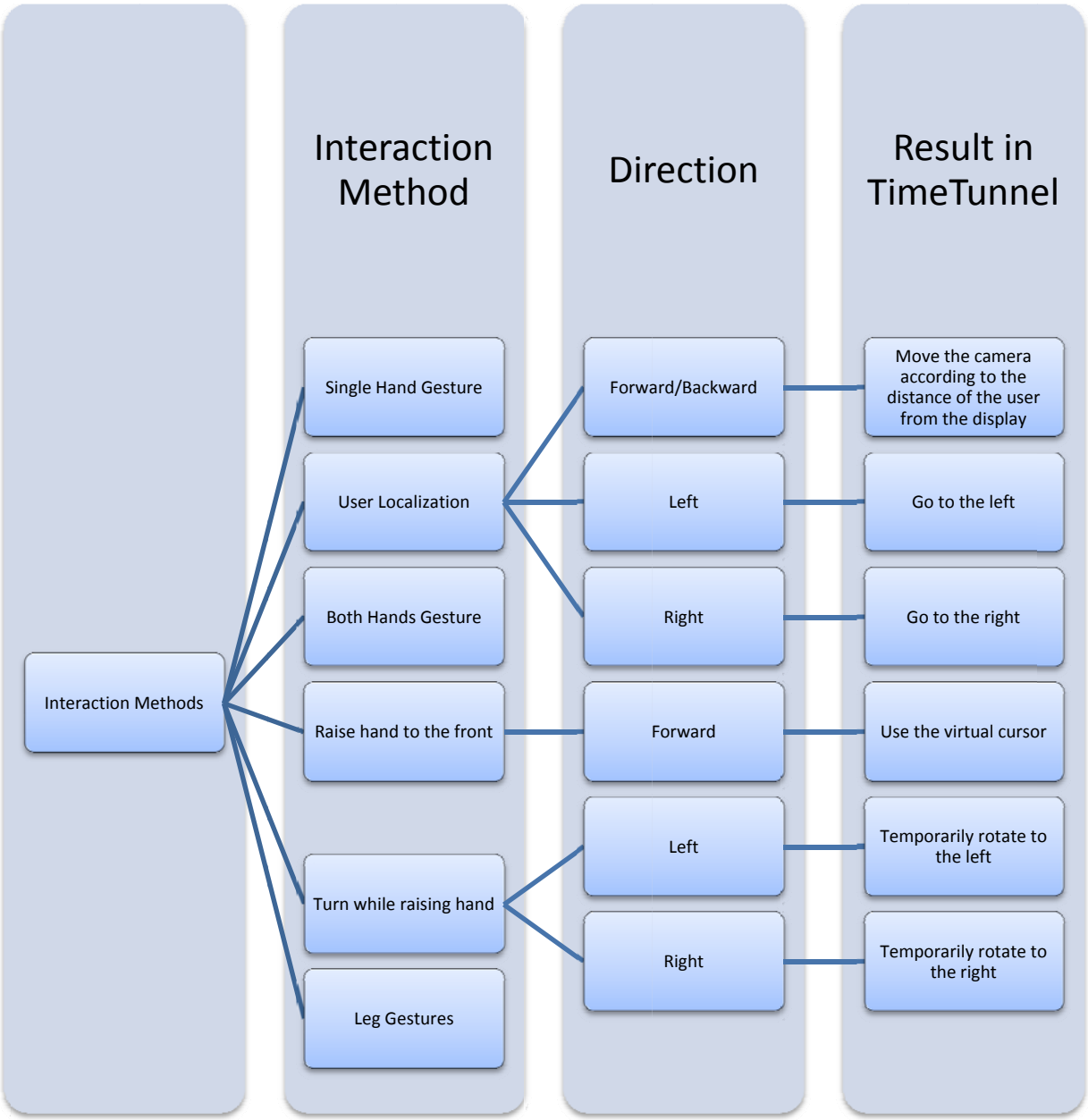

Diagram 9: Summary of Classic2D Interaction Methods

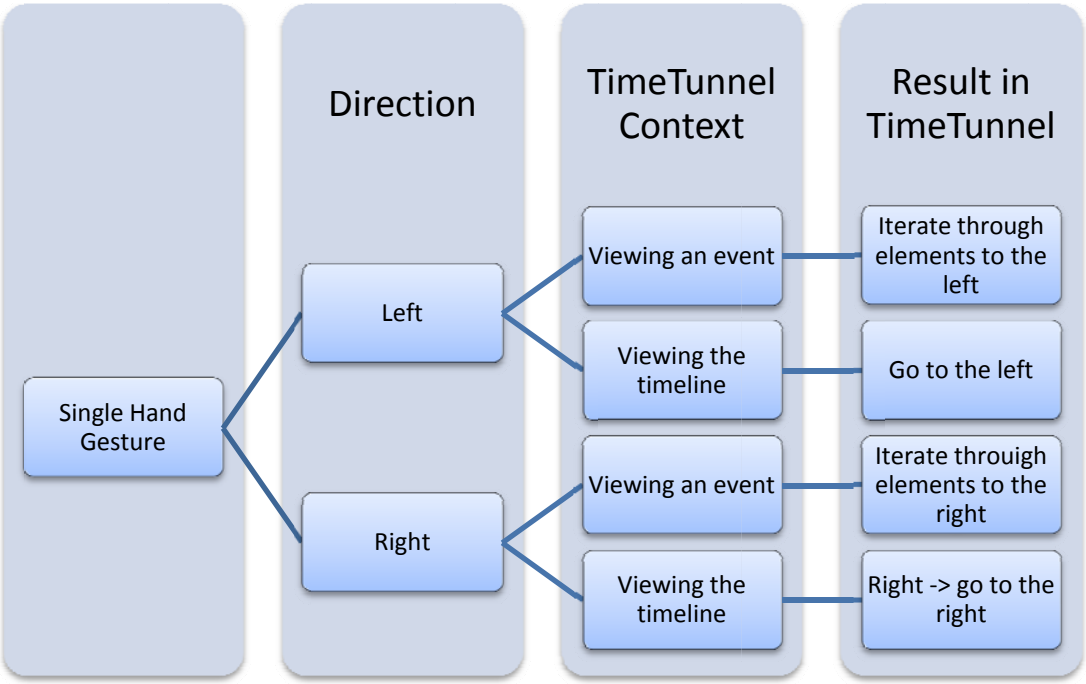

Diagram 10: Single Hand Gestures in Classic2D Mode

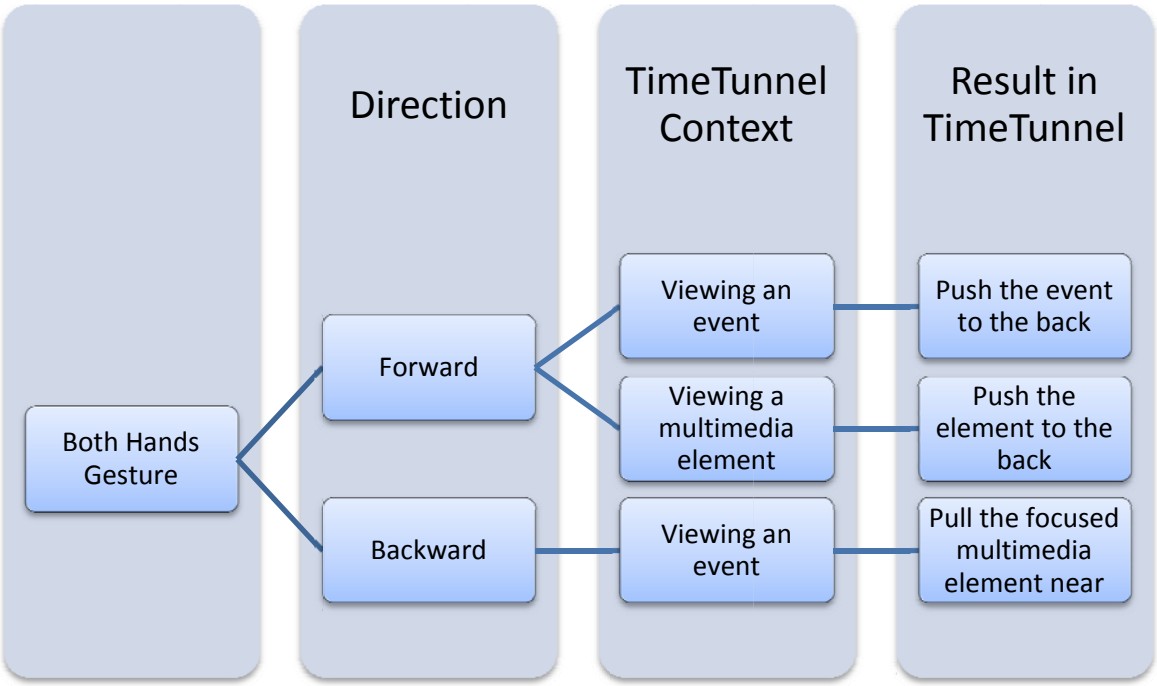

Diagram 11: Both Hands Gestures in Classic2D Mode

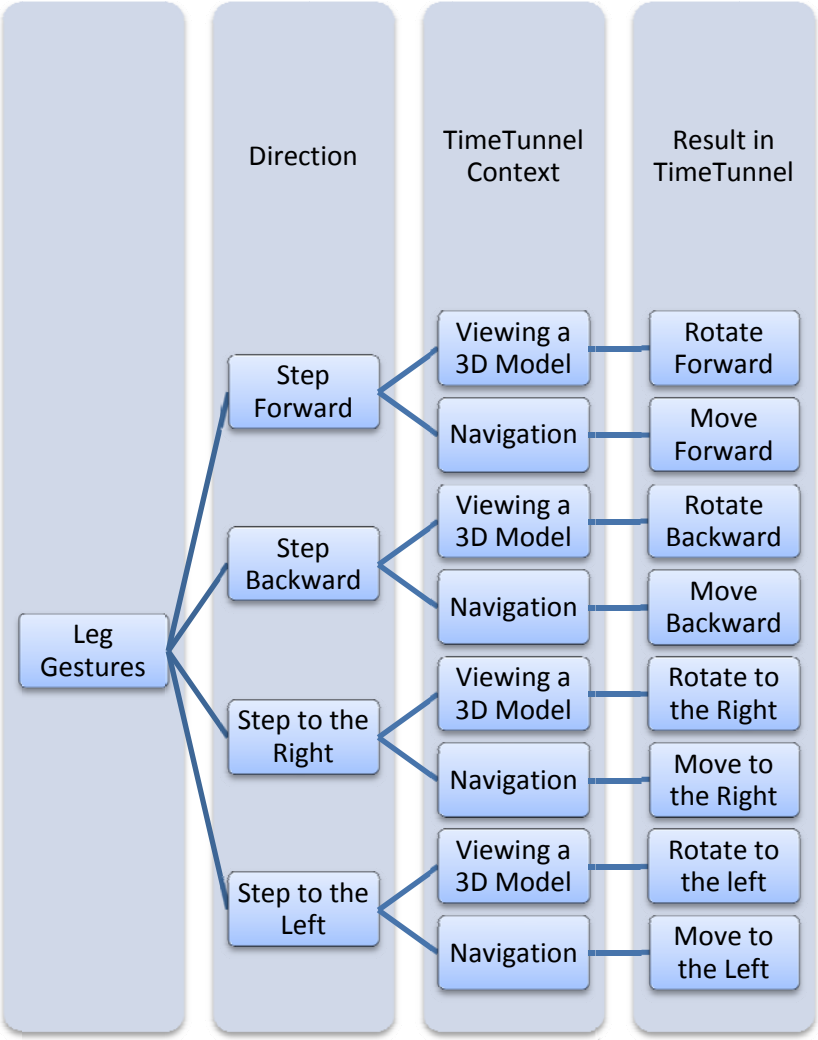

Diagram 12: Leg Gestures in Classic2D Mode

➢ Tunnel Mode

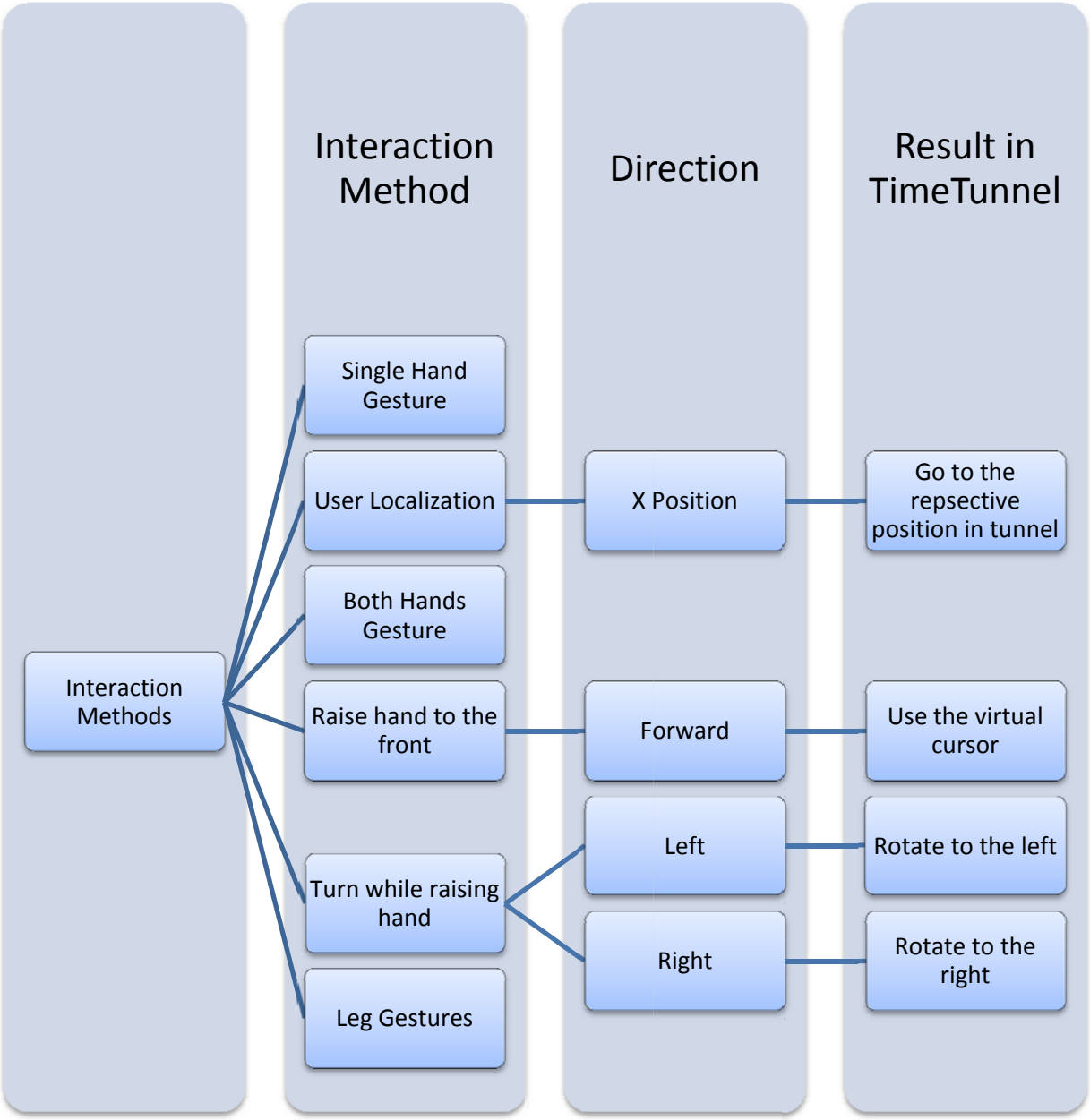

Diagram 13: Summary of Tunnel Interaction Methods

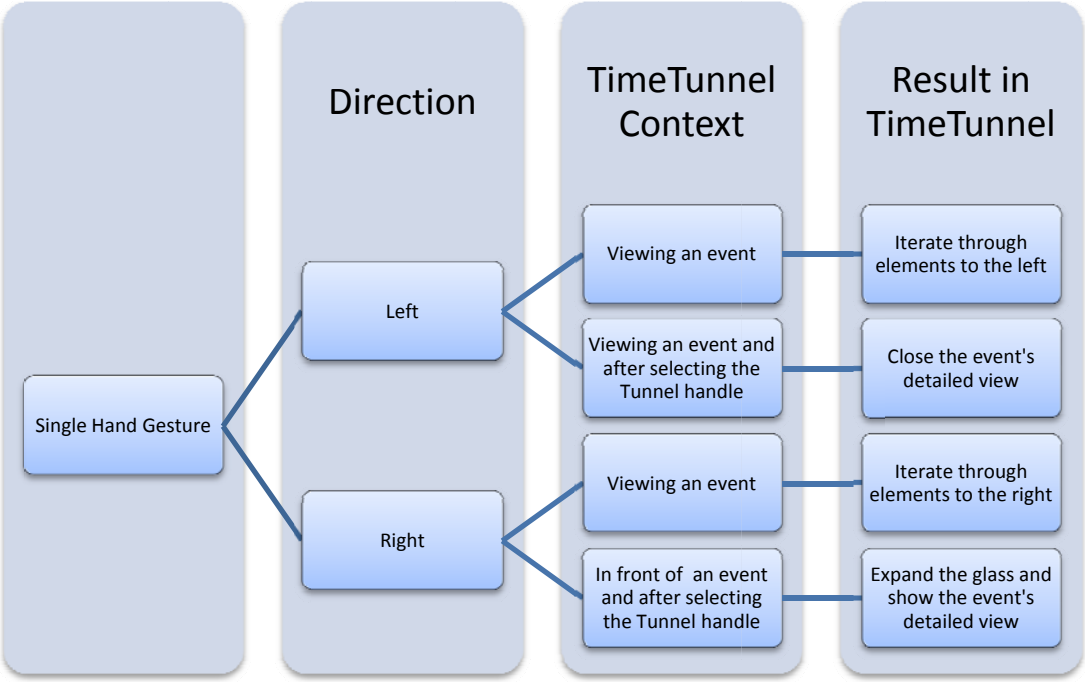

Diagram 14: Single Hand Gestures in Tunnel Mode

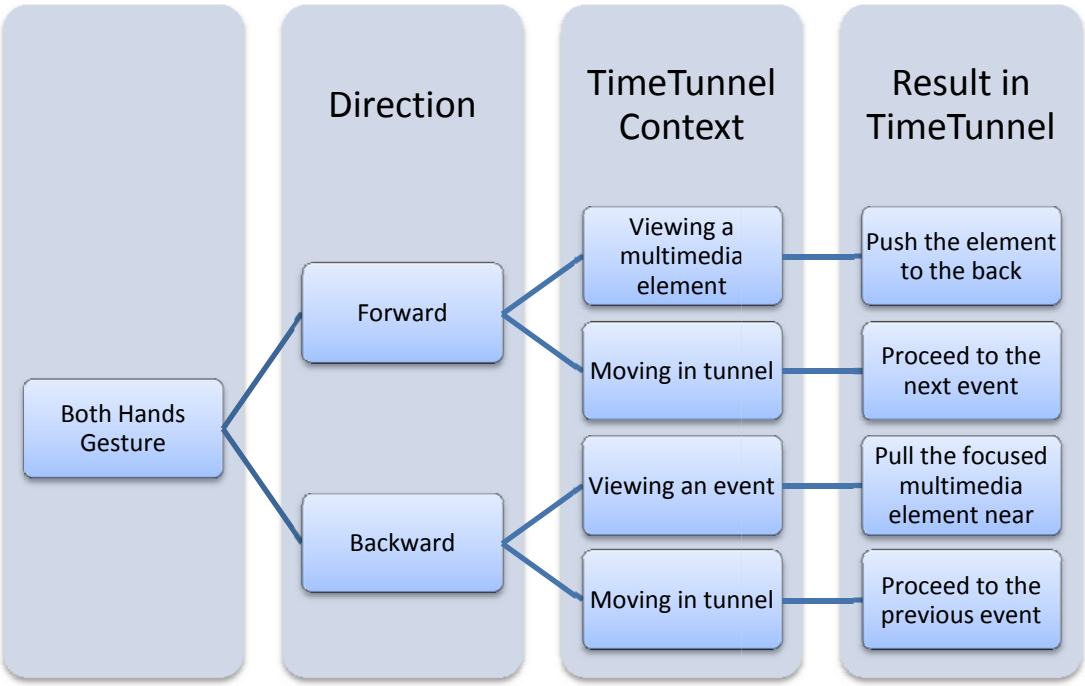

Diagram 15: Both Hands Gestures in Tunnel Mode

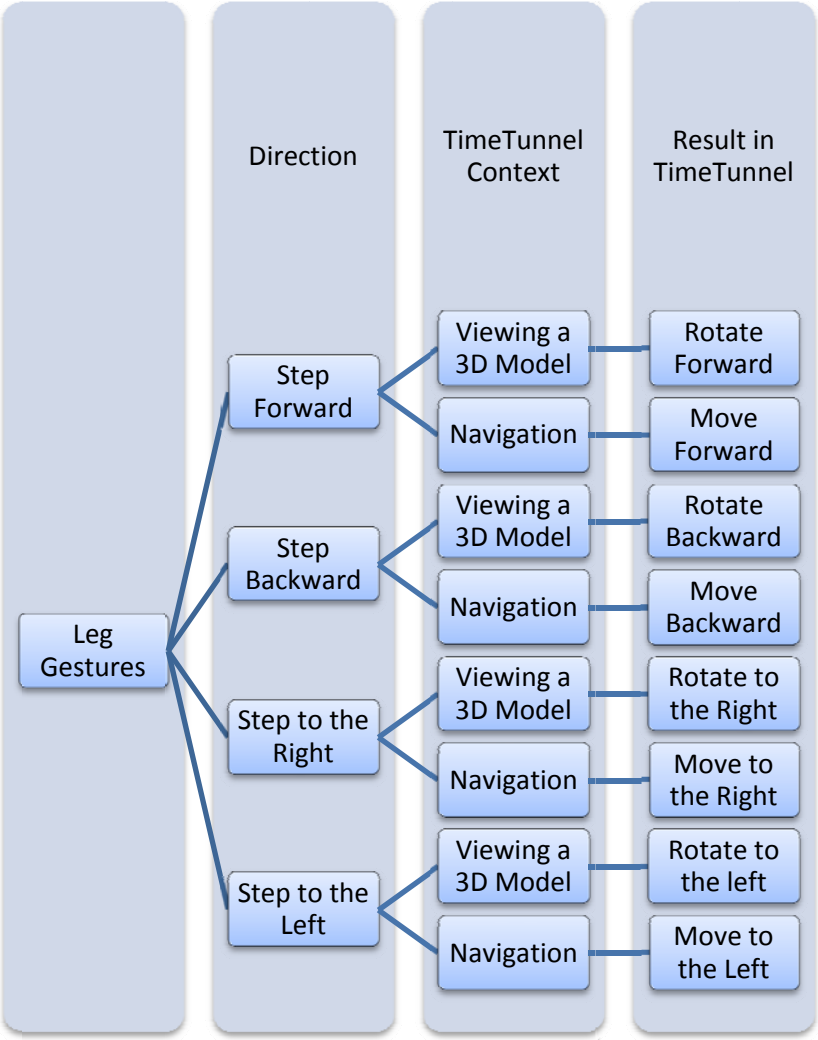

Diagram 16: Leg Gestures in Tunnel Mode

6.3 Interaction with physical objects

Another field of multimodal interaction is the application of tangible means of interaction using smart objects. In order to experiment with such items, a box was created, equipped with a 3-axes accelerometer[8] and a magnetometer[9] created by Sparkfun. The box is a wireless device that sends data using a Zigbee[10] wireless transmitter, the receiver of which is plugged in a USB port.

A service was developed, named Smart Box Listener Service, which reads the data from the COM port of the computer and creates an orientation matrix. This matrix is sent to TimeTunnel, which can currently use it in three possible manners:

- Firstly, if the user is exploring a virtual three dimensional model, it may behave in the exact same way as the real one that the user holds in his or her hands.
- Secondly, the orientation matrix can be used to rotate the camera of the virtual world in the direction the user is interested in.
- Thirdly, the orientation matrix can be used to move the camera of the virtual world forward, backward, left and right.

6.4 Communication of TimeTunnel with Sensors

TimeTunnel has been developed and is exhibited in a building dedicated to Ambient Intelligence within the Institute of Computer Science (ICS) of the Foundation for Research and Technology (FORTH). The communication of TimeTunnel with applications used for input from the environment, such as Kinect Skeleton Tracking Service and Smart Box Listener Service is supported by a middleware layer based on CORBA, developed for the intercommunication of applications and services [14]. The middleware layer offers tools and libraries that allow the creation of services using Application Programming Interfaces (APIs) in any of the supported programming languages, C++, .NET languages, Python and Java. The services running using the middleware may be spread across the network and programmers do not interfere with network connection establishment. On the contrary, developers focus only on the functionality that each service should offer, implementing the desired interfaces

[8] SparkFun Electronics, 3-D axis accelerometer, http://www.sparkfun.com/products/9045 , last accessed on 10/09/2012.
[9] SparkFun Electronics, 3-D axis magnetometer, http://www.sparkfun.com/products/10703 , last accessed on 10/09/2012.
[10] Zigbee specification for high level communication protocols, http://www.zigbee.org/ , last accessed on 10/09/2012.

as a service and creating wrapper classes that can handle incoming data for any client that receives them.

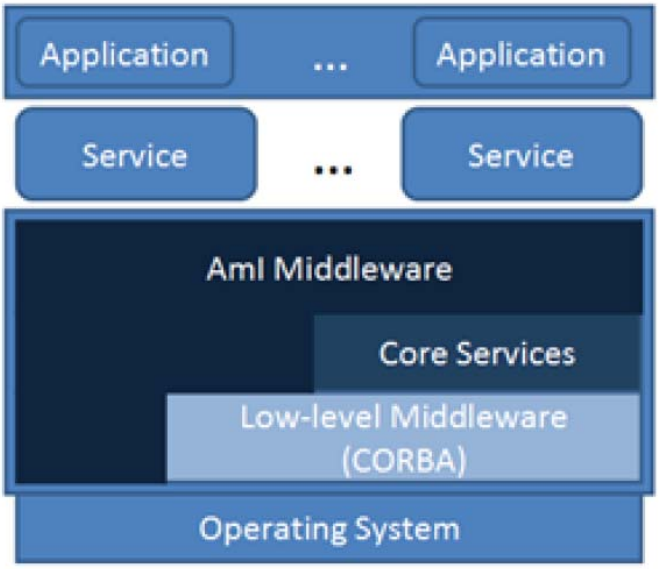

Figure 28: The architecture of AmI Middleware

7 TimeTunnel Implementation

TimeTunnel is written in C# programming language on Microsoft's XNA Platform[11], which is a runtime environment built over .NET Framework 2.0 that facilitates video game development and management. XNA was chosen as a freeware framework which may run on personal computers, Xbox and mobile phones running Microsoft Windows Mobile operating systems. Finally, Ox Game Engine[12] is used as a base to incrementally add additional components needed in order to cover needs that arose along the implementation.

Furthermore, XNA framework was extended to support sophisticated graphic effects through the implementation of shaders written in High Level Shading Language (HLSL)[13], which allows the implementation of custom three-dimensional effects through Microsoft's DirectX[14].

[11] Microsoft's XNA Game Engine, http://create.msdn.com/en-US/ , last accessed on 10/09/2012.
[12] OX Game Engine, http://oxgameengine.codeplex.com/ , last accessed on 10/09/2012.
[13] High Level Shader Language (HLSL), http://msdn.microsoft.com/en-us/library/windows/desktop/bb509561%28v=vs.85%29.aspx , last accessed on 10/09/2012.
[14] Microsoft's DirectX, http://msdn.microsoft.com/en-us/directx/ , last accessed on 10/09/2012.

7.1 Information Visualization

The components described in the following sections constitute general types of information which may be used in a variety of ways, according to the purpose of their use. TimeTunnel is assembled by two main parts that handle different aspects of an element:

- **The Component Part**, which contains the information needed for each element and implements all the functionalities needed to handle it at a higher level. Each component in TimeTunnel:
    - Contains values for
        - Transformation (translation, rotation and scaling)
        - Transparency
        - Visibility
        - Bounding Box
        - Draw priority definition
    - May be part of a hierarchy
    - Know how to render itself

- **The Script Part** for each component, which defines the behavior of each element and controls the component associated to.

Scripting architecture is selected aiming at each element's distinct implementation from its behavior. This way, code reusability is achieved and system's extendibility is assured. All components and scripts are registered to TimeTunnel at startup or on demand and the engine of TimeTunnel is responsible for updating and drawing them.

TimeTunnel implements a hierarchical mechanism in order to allow parent-child relationship. Components which are children of other components use a local transformation with regard to their parents. This way, TimeTunnel aims to be able to depict hierarchical relationships to the elements' transformation in the virtual world. Indicative examples of case where hierarchical relationships are applied are periods, which depend upon a virtual period holder, and events which contain elements such as images that are dependent to them.

7.1.1 Non-Animated Three Dimensional Models

Three dimensional models form an integral part in any three dimensional application as they stand the primary mean of displaying information to a user. They consist of a set of vertices and indices that describe their shape and effects that control their display.

TimeTunnel implements three dimensional models through the component called StandardModel, offering the following additional functionalities:

- Loading of new models
- Automatic bounding box generation
- Selection of draw mode, which may be either rendering as part of the virtual world and be affected by scene lightning or rendering with the default material without being affected by lightning changes.
- Explicit painting parts of it with the specified color along with an interpolation mechanism for the transition from the current to the desired color.
- Use of multiple textures that may be smoothly changed at runtime.

The practice of multiple textures is adopted so as to allow transition of model visualization across different pre-rendered textures. This mechanism is applied inside the virtual tunnel, in order to achieve more sophisticated and realistic lightning without affecting performance significantly.

TimeTunnel scripts that are StandardModels include the tunnel walls, the railway wagon and its frame inside the virtual tunnel, the lines that connect events to periods in Classic2D Mode and models that are shown as part of event information.

7.1.2 Animated Three Dimensional Skinned Models

The need for animation is apparent in three dimensional models and is implemented through the use of skinning: models are extended to support bones and joints that influence each vertex according to a specified weight.

TimeTunnel supports animations using XNAnimation[15], which is a skeletal animation library for XNA. However, XNAnimation – as most animation libraries - supports the playback only of static, predefined and pre-calculated animations, generated in 3D computer graphics software. On the contrary, TimeTunnel required some of the models used to be transformed most dynamically: for instance, in Tunnel Mode, tunnel walls and period models need to

[15] XNAnimation Library, http://xnanimation.codeplex.com/ , last accessed on 10/09/2012.

expand by dynamic values, in order to fit to the placement of each event and time respectively. Therefore, TimeTunnel contains a highly dynamic manual animation manager which offers the interpolated playback of animations, either instantly or by using a queue. As a result, custom animations can be added at runtime by code instead of requiring changes to the models in a modeling tool such as AutoDesk's Softimage XSI or 3D Studio Max. The implementation allows the interpolation to be at a variable time, ranging from instant animation to very slow, and may refer to the models' bones translation, rotation or scaling.

The scripts that use manual animations include the event's containers in Classic2D Mode, the showcase glass, the virtual buttons, the periods and the parts of the tunnel where no event is placed.

The scripts that utilize the default pre-rendered animations include the one for the virtual hand that depicts a three dimensional cursor and any three dimensional models that may be included as parts of event information.

7.1.3 Images

Images are implemented through components containing dynamic quads that render the picture to be displayed. Image components may visualize images of any aspect ratio, either in its native dimensions or through stretching to fit defined ones. Furthermore, a magnifying circle is implemented which may be used to explore image details at any position inside the picture's bounds. Apart from the normal image rendering, TimeTunnel offers the rendering of images using a god ray's effect as well as a virtual magnifier that may be used for detailed picture examination.

Image scripts are used in order to visualize event-related information. Each item is placed inside a container that expands exactly to fit the image width/height ratio and accompanies the picture with a descriptive legend at its bottom.

7.1.4 Videos

Video is supported through a custom component that hosts the playback of video in a quad fit in a predefined container. This component is an extension of a simple image quad, with the addition of video texture and the variation of the video's texture being rendered with its default material and lightning in order not to be affected by the lightning of the scene.

Video scripts are used as informational elements of the events which appear as virtual interactive screens that may start, pause, resume and stop the playback at any time. This is achieved through their correlation to the three dimensional buttons used for video manipulation.

7.1.5 Three Dimensional Text

TimeTunnel supports three dimensional texts that appear as common three dimensional models. The meshes are generated by the Nuclex Framework library[16] for each required text as Text3D components. Upon creation, meshes are handled as three dimensional models and may consequently be scaled, rotated and translated in any mode necessary. Furthermore, Text3D components may be stretched to fit their scale on X axis, Y axis or to the maximum scale to fit both X and Y axis, limiting themselves to their scale.

Scripts that manipulate three dimensional text components include the titles of events, periods and categories. Three dimensional texts were chosen to present titles for elements in TimeTunnel as they imply importance regarding the item displayed and titles do not require a specific background. On the other hand, they were not chosen as a means of presenting textual event information due to the fact that they are tiring for long texts, especially without the presence of a background as well as requiring increased computer resources.

Text3D Components are limited to expand on a single line; as a result, textual representation would have to either be scaled down to fit constraints in x axis or expand into as many lines as necessary. This functionality is offered by Multiline3DTexts that are able to split text at runtime into many lines and fit any given dimensions on demand. An example of Multiline 3D Text are the event titles which in Classic2D Mode may extend to more than one line so as to fit in the event's container, whereas in Tunnel Mode are placed in a line on the tunnel's floor.

7.1.6 Two Dimensional Text

Two dimensional texts are used to render extensive textual information on a plane. It is implemented by rendering text over a sprite (two dimensional image texture) using

[16] Nuclex Framework Library, http://nuclexframework.codeplex.com/ , last accessed on 10/09/2012.

XNA's Spritebatch[17]. Text is rendered once at its initialization and afterwards is treated as an ordinary image.

The only script that handles two dimensional text components is the one responsible for event detailed textual information. Additionally, the script is able to communicate with TimeTunnel's speech synthesis system and read the content aloud, when the user presses the related button.

7.1.7 Lights

TimeTunnel implements a variety of different lights that are combined to produce the final lightning of the scene. The used lights consist of:

- Ambient Lights, which is a light with fixed intensity that has no source and affects all items on the scene.
- Directional Lights, which is a source of light that illuminates all items with a certain direction
- Point Lights, which is a light that is placed at a certain position and spreads light outwards in all directions

Ambient and directional lights are used to produce the general lightning of the scene. On the contrary, point lights are used only when exploring the virtual tunnel and act as indicators of items that the user should notice. Specifically, the point lights fade in and out in front of the showcase of the event that the user approaches. The Lightning System Manager handles the scene lightning and is responsible for managing the way objects appear at both states of the timeline, Tunnel and Classic2D Mode.

Moreover, an additional point light with large range is used, placed below the railway wagon. This point light illustrates the current period examined by using its color, affecting all items shown and the tunnel itself.

7.1.8 Scene Components

Scene components are elements which are invisible, yet hold transformation and are able to be included in hierarchies. Such components hold primarily a manipulator and grouping role by their accompanying scripts. The scripts that are scene components include the Event Script, the periods' holder and the timeline script.

[17] Microsoft's XNA Spritebatch, http://msdn.microsoft.com/en-us/library/microsoft.xna.framework.graphics.spritebatch.aspx , last accessed on 10/09/2012.

7.2 Core Engine Implementation Details

TimeTunnel architecture consists of two main parts, scripts and services. The scripts are individual components which may have many instances, whereas services are unique items that offer a specific functionality.

7.2.1 TimeTunnel Services

Services are singleton classes which either offer functionality that may be accessed by the scripts at runtime or propagate information which is handled by the scripts. The singleton pattern has been chosen due to the fact that services are unique and therefore their instantiation should be limited to one object at a time.

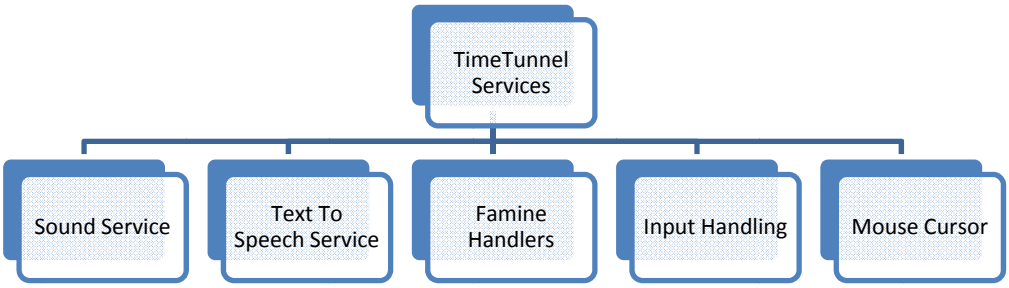

Figure 29: The services available inside TimeTunnel

7.2.1.1 Sound Service

TimeTunnel supports the playback of sounds and ambient music during users' exploration through a service available at runtime. Ambient lengthy sounds play at the background of the scene at a lower volume; on the contrary, other short sounds are used to provide feedback to the user and enhance the visual representation with audio.

7.2.1.2 Voice Synthesis Service

In addition to sound playback, TimeTunnel offers the synthesis of voice in any language. Speech synthesis is achieved through the integration of Microsoft's Text To

Speech SDK 5.1[18]. Speech synthesis is used in TimeTunnel by textual event information after user's demand.

7.2.1.3 Input Handling Manager Script

The Input Handling Manager is responsible for being informed about the system's input and acts as a mediator between the raw system inputs and the scripts. At the time of its initialization Input Handling Script defines its interaction means and informs any script that is interested. TimeTunnel currently supports two main types of primary interaction, touch input and user tracking with the use of Microsoft Kinect; yet, its design allows the addition of different techniques that may be developed in the future on the fly. It is deliberately implemented as an individual front-end for system to distinguish the system's input from the action that the system takes to react to the users' choices and actions. Furthermore, the Input Handling Script supports the forwarding of input from specific items or sensors: such an example is the Smart Box, which is a box with planted 3-axis accelerometer and magnetometers and transmits its orientation while it moves.

The Input Handling Manager contains two main mechanisms to inform the scripts about new input: firstly, it provides hooks for information propagation regarding input that may apply to a variety of items, such as a press of a button or the use of Smart Box. Secondly, the Input Handling Script contains structures that include all items that may be manipulated through ray-casting user's input to the virtual world. This part applies to embedding input from mouse, touch surfaces or virtual mice, such as the one used for interaction with user tracking.

Ray-casting in three dimensional spaces is ideally accomplished by triangle perfect collision detection of all items with the input ray; however, such an approach is too expensive in computational resources and therefore unattainable in large scale worlds with a hefty count of models, especially if they contain plenty of polygons. Therefore, an approach where each component includes a bounding box which exactly fits to its dimensions is put into practice. The issue that arises has to do primarily with items that are not solid on every side – such as the container hosting the event representation in Classic2D mode and contain items inside them. As a result, TimeTunnel implements a hybrid priority- and depth-based sorting of all items that may be ray-casted. All the components are sorted according to their priority, based on which the ray-casting takes place. In the case where

[18] Microsoft's Speech Software Development Kit, http://www.microsoft.com/en-us/download/details.aspx?id=10121 , last accessed on 10/09/2012.

multiple items have the same priority, depth sorting is used in regard to their distance from the camera – items which are nearer are informed earlier.

Scripts that are given input return a flag whether they have used the input hook or not. This way the items may consume an input and inform Input Handling Manager not to select other items with lesser priority. Furthermore, the maximum priority of selectable items may be defined in order to make items not selectable and optimize performance, in cases where specific items do not need to be interactive.

The script is able to inform interested scripts with input for touch, keyboard, mouse and other devices for multimodal interaction such as Microsoft Kinect and smart objects. Microsoft's C# events do not offer a formal order of execution (the order is the same as they are registered, but this is an implementation detail, not reliable for future usage) and are therefore extended with custom delegates to support priority for callbacks of all items registered, as long as no element consumes the input.

The following table shows the priorities for input handling through ray-casting with an emulated mouse.

Priorities of components that are informed about input changes
Navigation Script
Category Buttons
Animated Buttons
Focused Event Informational Component
Event Informational Component
Focused Event
Periods
Tunnel Glass
Events' Titles
Event's Containers in Classic2D
Railway Wagon (Minimap)
Camera Dragging (only for mice and touch)

All items that register to listen for ray-cast handling implement an interface which provides access to information such as their availability for input, their priority, bounding boxes and callbacks for high-level events.

Mouse input (actual or virtual) is sent to all selectable items as mouse information (position on X and Y axes, buttons states) on demand, along with a ray projected to the virtual scene. However, Input Handling Manager offers high level events for:

- **Mouse Enter**, when mouse virtual ray intersects the item's bounding box and previously did not.
- **Mouse Over**, while mouse virtual ray intersects an item's bounding box
- **Mouse Leave**, if mouse virtual ray stops intersecting an item's bounding box
- **Mouse Click**, if mouse virtual ray intersects the item's bounding box and user presses one mouse button.
- **Mouse Selection,** if mouse virtual ray intersects an item for certain duration.

Apart from traditional mouse, TimeTunnel offers the emulation of the mouse metaphor with different means of interaction. Such an example is input from Kinect, where two different approaches are implemented and supported:

- Selection with **virtual clicks** (by pointing forward with the hand).
- Selection after a **timeout** threshold of mouse ray intersecting an object.

Furthermore, for the needs of touch implementation where only touch is applicable, additional events are created that extend the fundamental ones and include:

- **Touch Leave**, when the user stops touching the surface.
- **Item Dragging**, for items that may be moved in the direction indicated by the user.
- **Camera Dragging**, in the case that the user grasps the tunnel floor or the Classic2D background in order to move in space.

7.2.1.4 Famine Integration

Famine services are integrated with singleton classes that listen for changes of the sensors' input, beginning at TimeTunnel startup. The implementations of all service handlers share the characteristic of listening to the updates of the service's state and inform TimeTunnel about high level events that the application components are interested in. Any element that is interested in such an event may become a subscriber and will be informed on any update.

Famine service handlers do not interfere with TimeTunnel's logic and act as a frontend for communication with the user's environment, simply propagating messages to all the components that are involved with any status update. At the current state of TimeTunnel, Input Handling Manager is the only component that receives new information about input from Famine services.

7.2.1.5 Three Dimensional Mouse Cursor

The cursor that shows the place where the user is pointing is of great importance, as it provides visual feedback to the user to aid interaction and help the accomplishment of interaction. The traditional mouse cursor is replaced in TimeTunnel by an animated three dimensional model of a pointing hand. The model is placed at small depth in front of the camera and moves in respect to real or emulated mouse movements. The approach of a three dimensional came up due to the need for more realistic display – especially with the use of Nvidia's 3D Vision, where the difference between a two dimensional image like the mouse cursor and a three dimensional model is apparent- but also because of the need for animation according to user interaction.

Apart from depicting the mouse movements, the three dimensional hand has the ability to move to defined positions and play animations. Such a case is to grab items, illustrating actions the user may take: when the user moves the virtual hand over items which may be dragged, the hand moves to the place of interest, "grasps" the item and rotates, denoting the gesture that the user should do to accomplish the task, such as the glass handle of events' showcases. Finally, the virtual hand may rotate accordingly to depict the left or right hand being tracked and used to interact if the system is aware of that information, for instance when the Kinect sensor is used.

7.2.2 TimeTunnel Scripts' Architecture

The scripts implemented in TimeTunnel refer to items that may have multiple instances. The principal script is the one handling the state of the timeline as a whole and informing individual elements about any occurring changes. The figure below shows the overview of the script hierarchy used which is described thoroughly in the following sections.

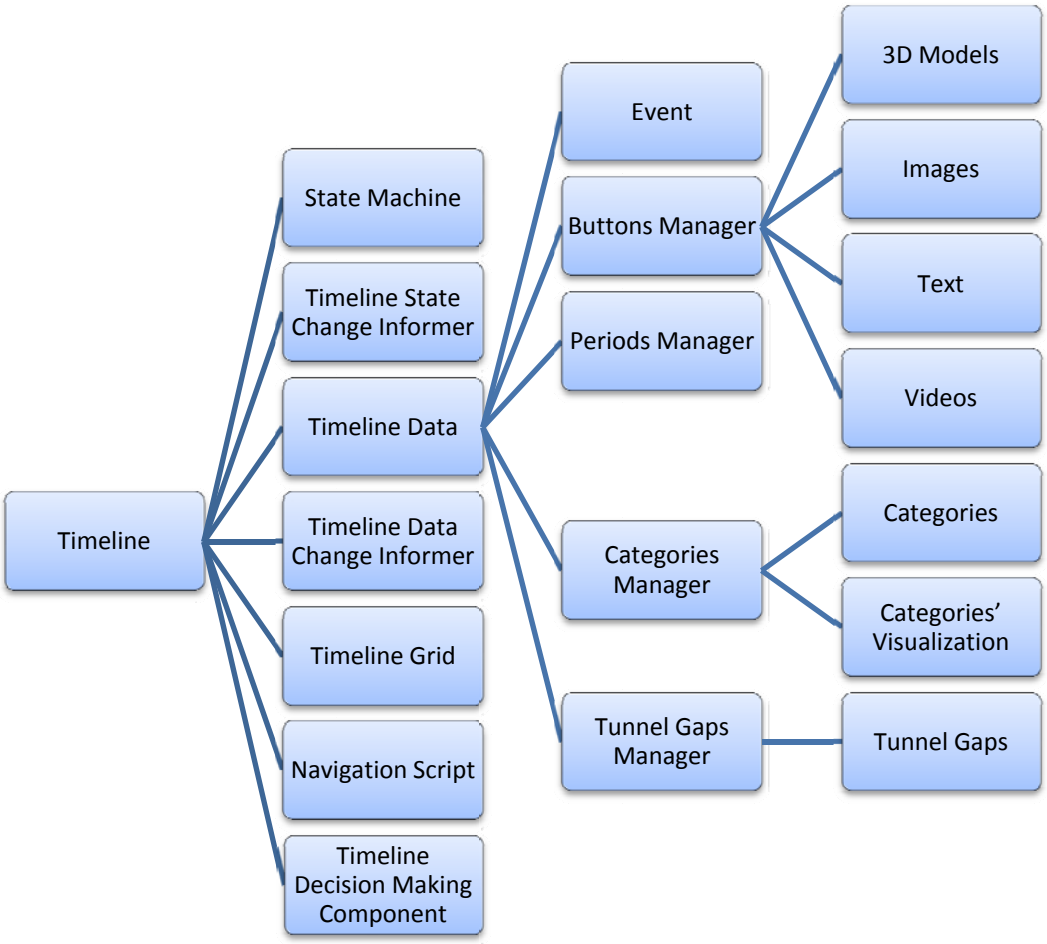

Figure 30: The scripts' architecture in TimeTunnel

7.2.2.1 Timeline

➢ Timeline Script State Machine

TimeTunnel is designed to be extendable and allow support to multiple timelines in the future, in order to be able to:

- Host timelines inside other timelines that may show in more detail specific eras. Such an example may be a timeline of a movie creation in the case of a timeline of the life of a movie director.
- Include additional timelines with a different context. Such an example may be the timeline of ancient Egypt in the case of the timeline of ancient Greece at the time observed.

The TimelineScript class holds the current state of the timeline, which may be:

- **Classic2D**, when the user is able to view the two dimensional view of the timeline. At the time that the user presses the button to enter the tunnel mode, the timeline's state is changed.
- **Started Changing from Classic2D to Tunnel**, where the two dimensional view is shown and the animation of switching to Tunnel Mode begins. This state finishes when the railway wagon becomes totally opaque. The main characteristics of this state are the following:
    - Any actions by the user are temporarily undone as the timeline switches to the default view (events visited retain their normal positions and all categories are activated)
    - The camera's field of view starts narrowing and concurrently moves backwards
    - The Railway wagon fades in
- **Finished Changing from Classic2D to Tunnel**, where the camera steps out of the minimap inside the railway wagon inside the virtual tunnel. This state ends when camera reaches the pre-calculated position according to the time examined. The actions taken at this state are:
    - Classic2D Components are instantly hidden
    - Tunnel components are instantly positioned
    - Common components in Classic2D and Tunnel mode are transformed to fit Tunnel Mode
    - The camera's field of view is increased
    - Fogging is applied
    - Event tunnel containers collapse and tunnel gaps expand to fill the gaps, according to the set of active and inactive categories.
- **Tunnel Mode**, where the user may explore inside the virtual tunnel. When the railway wagon is selected, the transition to Classic2D mode begins.
- **Started Changing from Tunnel to Classic2D**, which is characterized by the camera stepping inside the railway wagon's minimap, finishing when the whole display is covered by the minimap. At that time:
    - Fogging is disabled
    - The camera's field of view is narrowed.
    - The camera moves in from of the railway wagon frame

- **Finished Changing from Tunnel to Classic2D**, where the part of the minimap covering the screen is transformed to the Classic2D display. The changes taking place are the following:
    - Tunnel Components are instantly hidden
    - Classic2D elements are placed as in Classic2D Mode instantly
    - Common components in Classic2D and Tunnel mode are transformed to fit Tunnel Mode
    - The minimap covering the screen fades away
    - Events that are hidden start disappearing
    - Classic2D visualization is uncovered.

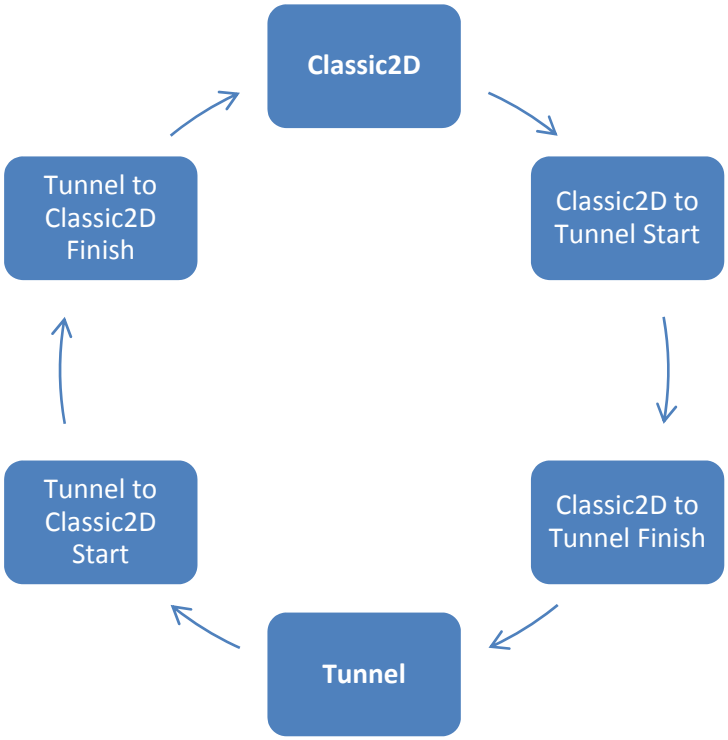

Figure 31: The sequence of TimeTunnel's states switch

After any change in the timeline's state, Timeline State Change Informer takes charge to notify all interested elements according to their priority. The mechanism used to implement event raising is similar to the one used for input handling [7.2.1.3], with the difference that all items are informed about the state change and may not "consume" the information. The priorities have been selected according to dependencies between each component's role in order to ensure implementation optimization and robustness.

Priorities of components that are informed when timeline's state changes
Categories Holder
Tunnel
Railway
Railway Wagon
Periods Holder
Period Script
Timeline Grid
Event Grid
Timeline Decision Making Component
Tunnel Gap Holder
Timeline Navigation Script
Timeline

- Timeline Decision Making Script

The Timeline Decision Making Script is responsible for keeping track of all the timeline's Event Scripts and their states. Its role is to make decision regarding parts of TimeTunnel that need to be refreshed when changes occur.

During navigation in Classic2D Mode, the Timeline Decision Making takes care of setting the other events' style to their default state (Date and Title Only), when an event is selected.

During exploration in the virtual tunnel, the script updates the state of the events as the camera moves in space. Apart from this, it also updates the lightning of the events' showcases by moving the point lights to the events in front of the user.

- Timeline Navigation Script

Another fundamental part of the timeline is the Timeline Navigation Script, which handles all camera-related variables inside the timeline. The primary role of the script is to handle camera movement and orientation, ensuring the smooth interpolation among places. Apart from camera transformations, the script manipulates the field of view used to render the virtual scene.

Furthermore the Timeline Navigation Script visualizes the transitions from Classic2D to Tunnel Mode and vice versa. It is informed by the Timeline State Machine about timeline mode changes and is the script that informs the Timeline State Machine that is ready to proceed to the next state, in all transition states:

- Classic2D to Tunnel Start
- Classic2D to Tunnel End
- Tunnel to Classic2D Start
- Tunnel to Classic2D End

During all these states, the camera is updated to produce the visual results described in [5.3].

7.2.2.2 Event Script

Each event is responsible to present itself according to its state, which is defined by the camera's position in the three dimensional space and the actions the user takes to interact with the system.

In addition to holding its state, the Event Script contains the multimedia elements that are shown as information as individual scripts that are set as its children, as well as event-related visualization scripts that are automatically created by TimeTunnel. All these items are stored in Event Data Holder. The items that are stored in Event Data Holder are controlled by another element, which is responsible for their arrangement, called Event Grid Controller. Upon any change of the state of the event, Event Grid Controller is informed to update the arrangement of the items it contains.

Finally, the Event Script is responsible for keeping track of the multimedia elements shown in TimeTunnel at any time. An index of the currently show item is internally held, which may change at runtime due to the user's interaction with the system. Upon any change, the Event Grid Script takes charge of rearranging the affected elements.

- Event Data Holder

The Event Data Holder simply contains the event's information and generates the scripts required of the event's visualization that include Classic2D as well as Tunnel visualization-specific elements. The created items are held internally and provide accessors which are used by Event Grid Script to arrange them.

As far as Classic2D Mode representation is concerned, the Event Script creates an **Event Container 2D Script**, which controls the box that hosts the event data that are displayed. It provides a high level Application Programming Interface (API) that the Event Script uses to control its visibility, dimensions and animations. Moreover, it is registered within the Input Handling Script as a selectable item while at Classic2D Mode. Upon any input change, it notifies the Event Script that is accountable to react.

The view of the event inside the virtual tunnel requires an increased count of scripts that produce the final view of the event in Tunnel Mode, including:

- The **Tunnel Event Slice Script**, which is an item that is able to expand or collapse, according to whether the event belongs to the active categories or not. Upon collapsing, the slice informs the following elements to collapse as well.

- The **Showcase Glass Script** is the item that controls the event's glass in Tunnel Mode. It provides an API to Event Script that controls its state (default transparent, preview transparent but brought a little to the front and expanded opaque). In addition, it registers itself to Input Handling Script and provides the correct bounding box that may be selected, as the handle that it may be dragged from moves according to its state.

- The **Tunnel Event Floor Script**, which is shown as a part of the tunnel. Its role is to show an animation when the user focuses on the specific event.

- The **Tunnel Scrollbar Script**, which is an item displayed to depict the currently shown items index in relation to the total count.

➢ Event Grid Controller

In order to separate the event's state from the arrangement of all event-related information, each event creates an Event Grid Controller that is in charge of arranging the items belonging to the event according to its current state. Moreover, it is in charge of visualizing the ones that should be shown, according to Event Script's state and multimedia index.

At all times, the Event Grid Controller arranges the following items:

- Event Container 2D Script
- Tunnel Related Components

    - Tunnel Event Slice
    - Tunnel Showcase Glass
    - Tunnel Event Floor
    - Tunnel Scrollbar
- Event Title
- Event Date
- Event Representative Item
- Event Multimedia elements
    - Images
    - Videos
    - Text
    - Three Dimensional Models

The event's title is an example of the manipulation of items that the script handles. Apart from updating its transformation, the event's title is shown in one line when TimeTunnel is in Tunnel Mode, but in the case of Classic2D Mode, the title is wrapped and extended to many lines if it does not fit in the event container using the default font size. In a similar manner, all elements are placed according to the specifications of the system design [5.3].

Multimedia elements are arranged regardless of their type. Event Grid Controller updates the scale, the position and the orientation of the affected elements. If the display mode is Classic2D, the event container is animated to depict the effect of the user. Furthermore, the script is responsible for informing the multimedia item that it is accessible for additional exploration and the item becomes selectable, as described in the forthcoming section.

➢ Event Multimedia Elements

Each multimedia item implemented in TimeTunnel is in charge of rendering and presenting itself when selected.

Apart from their visualization, each multimedia element controls the accompanying buttons that are necessary for interaction. The buttons' choice and usage depends on the input method used and is stored internally. For instance, if a three dimensional model is shown to the user, four buttons are also displayed representing the directions the item may be rotated towards, if user tracking through a camera is used. On the other hand, if a smart

object that knows its orientation is used, the presentation adapts to hide the buttons as they are no longer needed.

- Events with Duration

In the special case where an event has duration, an additional event script is created (Event End Script) that is only valid for the Tunnel Mode of TimeTunnel, as described in [5.3.3.1]. The new script is registered to all the items that a normal event is registered, such as the Categories Holder, the Period Holder and the Timeline Data Holder. The Event End Script creates clones for the parts of the event that need to be presented in the Tunnel Mode, the event's title, representative item and the tunnel visualization-specific components. The Event End Script contains a modified version of Event Grid Controller that arranges its components according to the specifications of TimeTunnel.

7.2.2.3 Period Script

Each Period Script contains its visualization along with additional information describing itself, including its color, its name and the time legends showing time details according to the user's position. Furthermore, each Period Script is responsible to distribute time to space both at Classic2D and at Tunnel Mode.

As far as Classic2D Mode is concerned, each period's extent is calculated by a field of Period Script called Virtual Years, which is calculated by multiplying the actual time duration that the period covers by the factor that defines the stretch that is applied. Periods' visualization consists of an animated skinned model spreading in X axis, in parallel to time extent. The bar contains round edges that are shown if previous periods' end time and next periods' start time not match current period's start and end time respectively. Finally, each period's title is placed in the middle of its extent.

In Tunnel Mode of TimeTunnel, time is "frozen" around events placed during the extent of a period. Therefore, the total extent is increased by event count inside period multiplied by the extent of each event's tunnel container slice. As a result and given the fact that each Period Script is aware of the events it contains, the equivalent place of a timestamp may be calculated.

Periods Visualization in Tunnel Mode is accomplished through two symmetrical models placed at the specially designed area of the virtual tunnel's walls, whose extent is calculated similarly to Classic2D Mode. Periods' titles position, whose positions are described in detail in [5.3.3.2], is calculated by moving them towards the position of the

railway wagon along the axis the periods extend, while also constraining them to stay inside the periods' bounding box.

The color of each period remains constant throughout the execution of TimeTunnel as a point of reference for the user. Finally, Period Script is responsible to provide to the Periods Manager the local position (the position relative to the middle of the period) of each moment that belongs to it.

- Period's Legends Holder

Classic2D Mode of the timeline includes the visualization of time legends above each period, which is handled by each period's Legend Holder. The mechanism described in detail at [5.3.2.3] is implemented by creating a class that divides time in frames according to the period's duration. The Legends Holder contains a constant value which is a threshold defining the minimum legend scale at which the characters are visualized clearly.

The mechanism implemented supports detail up to months, but may easily be expanded to represent more detailed time, such as days, hours, minutes and seconds. The Legends Holder starts calculating the dimensions of the legends in order to fit all legends in the space available within the periods' bounding box. At the time that the first value which is larger than the minimum scale is found, that level is considered as the maximum detail level and the pattern is continued to create three additional sets. Values of each level of detail are stored and all distinct timestamps are merged to exclude duplicates.

At runtime, camera's distance from the timeline is divided in four areas, each associated with a set of values. As a result, each Legend Holder knows which of the four levels of detail to show and acts accordingly.

7.2.2.4 Timeline Data

The Timeline Data class includes the scripts that are needed to manipulate timeline behavior. Its role is to hold the information, but not take any decisions regarding the timeline's state. On the contrary, it acts as a mediator that provides access to any interested party. It consists of the following parts, which are analyzed in detail in the following sections.

- Events Manager
- Period Manager
- Categories Manager
- Buttons Manager

- Tunnel Gaps Manager

➢ Period Manager

The Period Manager is in charge of holding and arranging all the periods of TimeTunnel. The Period Manager treats time frames which do not belong to any period as new virtual periods with the default time scale, one. However, their visualization is altered, as they appear smaller than actual periods and are colored gray. Virtual periods do not interfere with the visualization of existing periods, but are arranged in a similar way.

The Period Manager actually holds the distribution of time in space, as it is the component responsible for taking every period's virtual time and arrange them all, taking total timeline duration into consideration. Each period is placed according to the previous one, disregarding whether it is an existing or a virtual one. After all periods are sorted, the Period Manager may provide the actual positioning of any moment in time, by getting the local position of the moment inside the –existing or virtual- period it belongs to and adding the position of the Period Script.

➢ Events Manager

The Events Manager contains all the events that exist in TimeTunnel. Events are created as timeline script's children and Events Manager resolves and stores their references in order to be accessible at runtime.

Rather than simply storing events, Events Manager handles them according to whether they are instant or have duration. At the time that an event is resolved, it is registered in the following structures:

- Events List, containing all the events that are represented in Classic2D Mode
- Events Merged For Tunnel, containing all the events that exist in Tunnel Mode.

Furthermore, if an event has duration, its implementation as Event End is registered in the Events Merged for Tunnel structure, due to the fact that in Tunnel Mode it is visible as an independent entity.

➢ Categories Manager

The Categories Manager contains the information regarding events' categorization. It is responsible for both categories' storage and visualization handling and its role is to provide access to the set of all visible events both for Tunnel and for Classic2D Mode at any time.

At startup, each category is created along with its visualization and the events that belong to it are registered in a list. The visualization of the category includes its title as a three dimensional text which may be selected by Input Handling Manager. An additional category is created that includes all the events that TimeTunnel contains and is initially set to be the active one.

The Categories Manager takes care of holding and displaying the currently active category visualization and displays the additional ones, when the user selects the active one. Additionally, when a new category is selected, it notifies the Timeline Script about the newly selected category in order to update the representation of the events.

- Buttons Manager

The Buttons Manager is the class responsible for creating the virtual three dimensional buttons used for interaction with event multimedia elements. It contains different sets of buttons each suitable for the interaction mechanisms needed for each multimedia type:

- Video Buttons, which consist of a play and a pause button.
- Image Buttons, which consist of a start and a stop magnifier button.
- Text Buttons, which consist of a start and a pause reading text out loud button.
- 3D Model Buttons, which consist of four buttons to rotate the model left, right, up and down.

Video, image and text buttons are placed in the middle of the screen and are toggle buttons: after one is pressed, it fades away to give its place to the second one at the exact same position. On the other hand, 3D model buttons are continuous buttons that keep rotating the current three dimensional model while they are pressed. They are all visible while the user is exploring the model, placed at the right side of the display. Apart from item-specific buttons, the Buttons Manager always shows alongside visible button sets a return button which the user may press to finish exploring the item.

The visible set of buttons may be any of the above or none otherwise. When an item is selected and brought in front of the user to interact with, the Button Manager is informed to show the corresponding buttons set. Once buttons are visible, the Buttons Manager raises events that any key is pressed and the interested parties act accordingly, interpreting the meaning of each button.

- Tunnel Gaps Manager

The space along the tunnel that intervenes between events' slices, which is described in [5.3.3.1], is filled by the Tunnel Gaps Manager. The Tunnel Gaps Manager is initialized at the time that all events are loaded, by getting Events Merged for Tunnel from Timeline Data and creating N+1 slices, where N is the count of Events Merged for Tunnel. The N-1 slices are placed between the events and the remaining two extend towards the start and the end of the virtual tunnel.

At the time that TimeTunnel switches to Tunnel Mode, the Tunnel Gaps Manager recalculates the extents of all slices between neighboring events so as to stretch to fit the space exactly. This is accomplished through the exhaustive iteration over events, which are sorted by time of occurrence and therefore are located in a row. The same process takes place when a new event is added to the timeline or when the Categories Manager alters the currently active set of events.

7.2.2.5 Timeline Grid Controller

The Timeline Grid Controller is the script responsible for placing the events of the timeline according to the time they occur at, whenever the need arises:

- TimeTunnel changes from Classic2D to Tunnel Mode and vice versa
- New events are added
- A Period's scale changes

In line with the current state of the timeline, the Timeline Grid Controller arranges the events either inside the tunnel or along the plane that events should be located. The script resolves the global position for all existing events from the Periods Manager and places them accordingly. This procedure is sufficient for the creation of the virtual tunnel, but needs to be refined for the Classic2D visualization, as the events' containers may interfere with each other if their occurrence time is close.

The discrimination of time to macro and micro is used to resolve Classic2D events' placement: events that occur a short interval after others and whose containers would overlap, are moved below by the Timeline Grid Controller. Therefore, the script firstly arranges all the events at the same height and afterwards moves them below if the need arises.

7.3 Ontology data Manipulation

7.3.1 Query Engine - Translation of ontology query results

The Sparql query executed to retrieve the timeline information from the ontology returns the results in custom wrappers, each designed to fit the corresponding entity's needs. For the execution of a query a Sparql Engine is created, along with a Result Sink that is able to host all the results to corresponding wrappers. The architecture of the wrappers is equivalent to the ontology's architecture and therefore, a Timeline Wrapper is returned as a result to any component that wants to fetch all timeline-related information.

7.3.2 Creation of an XNA compatible format

The transformation of the timeline wrapper to the TimeTunnel format is done by a standalone application, the Ontology Translator. The Ontology Translator uses the aforementioned Query Engine and creates an XML document that contains both the dynamic timeline-related information and the following constant elements of TimeTunnel:

- Lightning System
    - Ambient Light
    - Directional Lights
- Tunnel Minimap Components
    - Railway
    - Railway Wagon
    - Railway Wagon Frame
- Skybox (used as a three-dimensional background)

After XML's generation, TimeTunnel captures automatically the minimap of the two-dimensional representation and loads it at runtime. The minimap is stored for the subsequent executions of TimeTunnel and is generated again only if the Classic2D view is altered.

8 Evaluation

8.1 Set-up and Participants

The evaluation sessions took place inside FORTH's AmI Facilities in two different rooms, one for the system's evaluation using the touch screen and one for the assessment of user tracking through Microsoft's Kinect. A camera was placed in each of the rooms in a position suitable to capture the user's interaction with the system.

The setup of the first room, where touch interaction and Smart Box were evaluated, was rather simple. A Dell SX2210T touch screen[19] was placed on a desk and the Smart Box was at its side.

The second room's setup was rather different: a Dell S500 short throw projector[20] was used as a display method that projected on a sheet placed on the wall. A Microsoft Kinect was placed right above the projector's screen so as to allow user tracking inside the room. The projector model offers refresh rate of 120Hz and therefore allows the application of Nvidia's 3D Vision technology as long as the user wears Nvidia's 3D Glasses.

Figure 32: TimeTunnel's set-up in the second room

[19] Dell SX2210T touch screen http://www.dell.com/ed/business/p/dell-sx2210t/pd, last accessed on 10/9/2012.
[20] Dell S500 short throw projector http://www.dell.com/us/en/highered/peripherals/projector-dell-s500/pd.aspx?refid=projector-dell-s500 , last accessed on 10/9/2012.

8.2 The Evaluation Process

The goal of the evaluation process was to assess TimeTunnel in regard to the four usability principles outlined by [5]: usefulness, effectiveness, learnability and likeability. A total of 16 volunteers participated in the evaluation, 7 females and 9 males. The age of the participants varied from 20 to 40 years old. Twelve of the users (75%) had intermediate or high computer expertise whereas the other participants had limited expertise. Even though the majority of the users were familiar with computers and touch screen systems, they did not have great familiarity with hand gesturing as a mode of interaction with a system.

Since Time Tunnel supports multiple interaction modes, it was deemed necessary to divide the evaluation session into two separate segments, one for each of the primary modes. More specifically, the first evaluation segment included the interaction using a touch screen [6.1] along with the Smart Box [6.3], while the second one integrated the manipulation of the system remotely, through the use of user tracking [6.2.1]. Moreover, the usage of 3D Vision was evaluated separately in order to measure its effect in TimeTunnel's presentation.

In order to achieve objective results and avoid users being biased towards either interaction method, users were split into two even groups. The first group started the evaluation process using the touch screen, whereas the second one began interacting with TimeTunnel remotely. The evaluation process continued with the assessment of the other interaction method respectively, remote interaction for the first group and touch screen for the second one.

The evaluation process started with the users being informed about the goals of the evaluation process. Moreover, a consent form that describes the evaluation procedure and declares that the whole procedure will be taped was given to the users to be signed, stating that any personal information would remain anonymous and strictly confidential. After the short introduction to the concept of the evaluation, the functionality that TimeTunnel aims to support was explained to each user. The evaluation process of the remote interaction using user tracking additionally included the explanation of the available gesture sets that TimeTunnel supports. Similar series of tasks were assigned to the participants in order to measure the usability of the system in both segments of the evaluation process. The minor differences aimed at assessing the individual characteristics of each of the interaction method.

In addition to the tasks, the participants filled in questionnaires in order to assess the opinion of the users and retrieve qualitative results in a formal way. However, apart from the questionnaires, all participants were encouraged to express their thoughts throughout the evaluation process, which were written down using notes and further processed through the examination of the recordings.

The evaluation process was designed to cover the needs of this thesis; however, further assessment of the system would be interesting in order to reach supplementary conclusions. An additional evaluation could involve only non-expert users to measure the users' reactions in the case of TimeTunnel being set up in an exhibition space like a museum. Furthermore, another interesting approach could include asking the participants to evaluate without having been given any type of guidance in order to assess how close the system's interaction design is to the users' perception of natural interaction.

8.3 Results

8.3.1 Overall System Design

Overall, the users found that the two different modes of content presentation were complimentary to one another. The majority of the users preferred using the 3D Tunnel representation in order to explore specific events, while they preferred using the 2D mode to obtain a chronological overview of the events. This discrimination of information display was one of the fundamental goals of this thesis as far as design is concerned and the comments from the evaluation participants indicate that this goal is accomplished.

The Classic2D representation was regarded as sufficient and clear. The application of periods' color coding to the events was easily comprehensible. Furthermore, the "hidden" (grayed out) state of the events due to categorization was preferred to disappearing completely. The hosts at the side of the event container were indicative of the existence of additional items, as the users understood whether any multimedia element was available or not.

The immersive design of the tunnel was met with enthusiasm and excitement. The showcases with the events' representative items were very clear and natural to the users, who immediately understood the ability to pull the glass outwards through the animation of the hand applied as a cursor.

The elements displayed at the timeline were considered in general as intuitive and self-explaining. However, the placement and the graphic representation of the content categories were not evident to the users, and, as a result, many of them failed to acknowledge their existence. Another issue was the visualization of some type of event detailed information: some users were unsure regarding the presence of videos and three dimensional models. The video elements had a figure of a tape that implied the existence of a video; yet, users believed that the existence of a play button would aid in the perception of a video item. Furthermore, some of the users proposed the addition of controls for volume control and seeking forward or backward. As far as the three dimensional models are concerned, users proposed the addition of alternate concepts that could aid the recognition of the item as three dimensional. The concepts could be used in combination and are the following:

- A three dimensional container placed beneath the three dimensional model
- The three dimensional model to be initially rotated in order to depict the sense of depth
- The addition of a three dimensional thumbnail with the label '3D' that could potentially rotate.

Interesting ideas were expressed for the enrichment of TimeTunnel throughout the evaluation process. One of these ideas was that in Classic2D, when selecting a period, the camera would not only move in front of it, but would also zoom in so that the period's extent covers the display dimensions. Moreover, another enhancement proposed by some of the users was the addition of numbering to the events' multimedia elements, as they would prefer the count of the elements to be explicitly shown in a numerical value. Another suggestion concerned the close button of elements to be placed exactly on the top-right side of each element and not to a fixed point of the display.

8.3.2 Touch Results

The following table shows the tasks that were assigned to each of the participants for the first segment of the evaluation process. Tasks 1-5 are the same with the ones in the second segment of evaluation and aim at the assessment of the system's usability. The set of the tasks covers all the common processes that the system is designed to offer and involve the understanding of the information displayed and the accomplishment of usual goals that a user could have. The rest of the tasks differentiate from the tasks assigned to the second segment in order to assess interaction with the Smart Box.

1. Navigate inside the Classic2D view of TimeTunnel and answer the following questions: 1.1 How many decades does the timeline cover? 1.2 How many events are there in the 1960s? 1.3 Which event categories are shown in the timeline and what others do exist?
2. Focus in the 1980s and find information regarding Apple Lisa. What types of information are shown?
3. Select the photograph of Apple Lisa displaying a man on its monitor. What is his name?
4. Find a video element and watch it.
5. Find a three dimensional model and rotate it.
6. Use the Smart Box to rotate the three dimensional model.
7. Use the Smart Box to navigate in Classic2D Mode.
8. Enter the Tunnel Mode.
9. Navigate inside the virtual tunnel using the Smart Box.
10. Navigate inside the virtual tunnel and explore information regarding the event of the first computer mouse.
11. Go back to Classic2D Mode

The first segment of the evaluation process indicated that the touch interaction was very satisfying to the users, as they felt comfortable and familiar with the manipulation of the system through the touch screen. The users easily understood the way that they could interact with the system, both by selecting items through simple touches and by applying the necessary gestures. Such examples are the navigation inside the virtual tunnel by dragging the floor, dragging the events' plane during the Classic2D Mode and iterating through the multimedia elements while they are displayed.

Apart from the broad acceptance of touch as a means of interaction, evaluation illustrated further enhancements that TimeTunnel could offer. Firstly, multi-touch is a very common feature, therefore Time Tunnel should support it. Secondly, the gestures for exploring multimedia elements were by some users considered to be applicable regardless of whether they began from some element. Furthermore, the events were expected to close when examined in detail and the user clicked outside the expanded showcase glass. Finally, specific components of the timeline could become interactive to aid the system's usability. These components are:

- The zoom slider for the Classic2D Mode, which should be able to be dragged, as it currently can only be used by pressing the increase and the decrease buttons.
- The events' multimedia element scrollbar, which should be allowed to be dragged left and right to iterate through the available elements.
- The minimap frame inside the virtual tunnel, which upon dragging would travel the user inside the virtual tunnel to the frame's corresponding position in Classic2D.

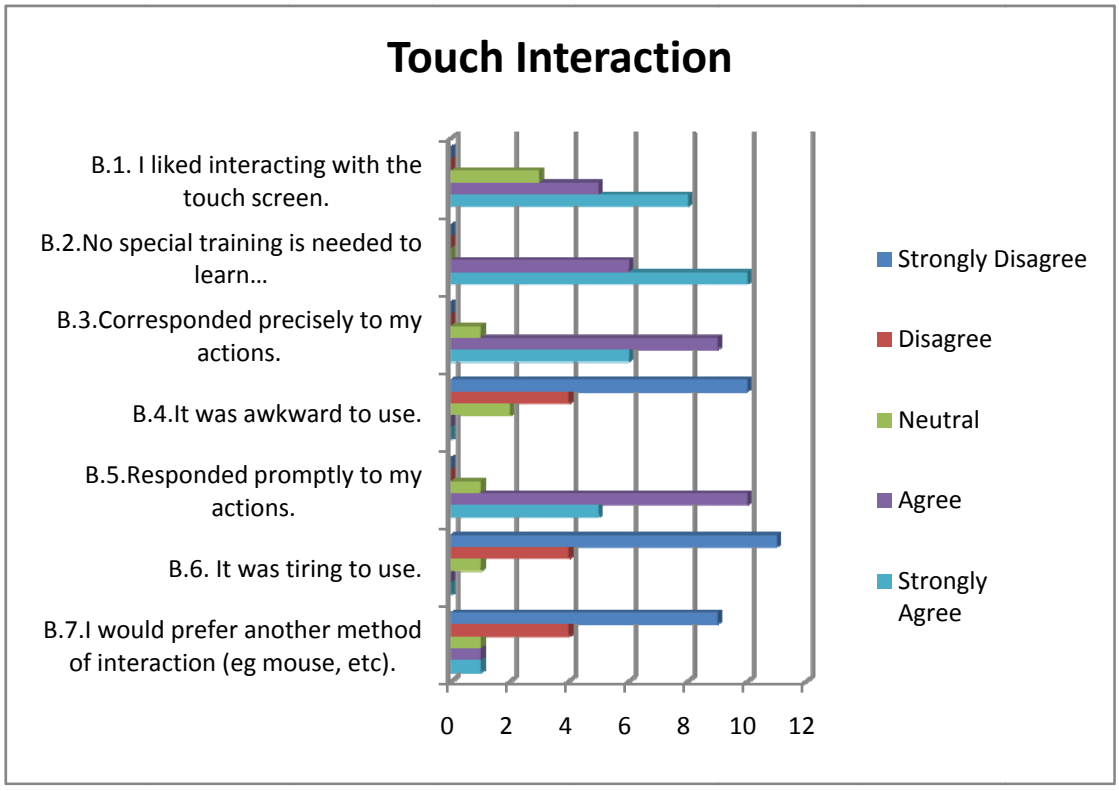

Diagram 17: Questionnaire results for Touch Interaction

Despite the broad acceptance of touch interaction, the interaction with the Smart Box received mixed reactions, as half of the participants were divided concerning its usefulness and its application as a secondary means of interaction.

The supporters of the Smart Box argued in favor of its use, as they found it useful for the manipulation of the three dimensional models and the navigation in the three dimensional space. Furthermore, a significant remark pertained to the exploration of extensive data and the necessity of continuous navigation, either forward/backward inside the virtual tunnel or right/left in Classic2D Mode.

The participants who considered the Smart Box to be unnecessary mainly argued that they could not accomplish their goals with the box alone. Furthermore, another addition included the manipulation of the three dimensional model in a one to one representation of the virtual model in correspondence to the Smart Box.

The mixed reactions that the users had regarding the employment of the Smart Box to supplement the interaction with TimeTunnel are pointed out in the diagram shown below. The users' likeability of the Smart Box's usage is questioned, despite the fact that it was in general responsive, precise and neither tiring nor awkward.

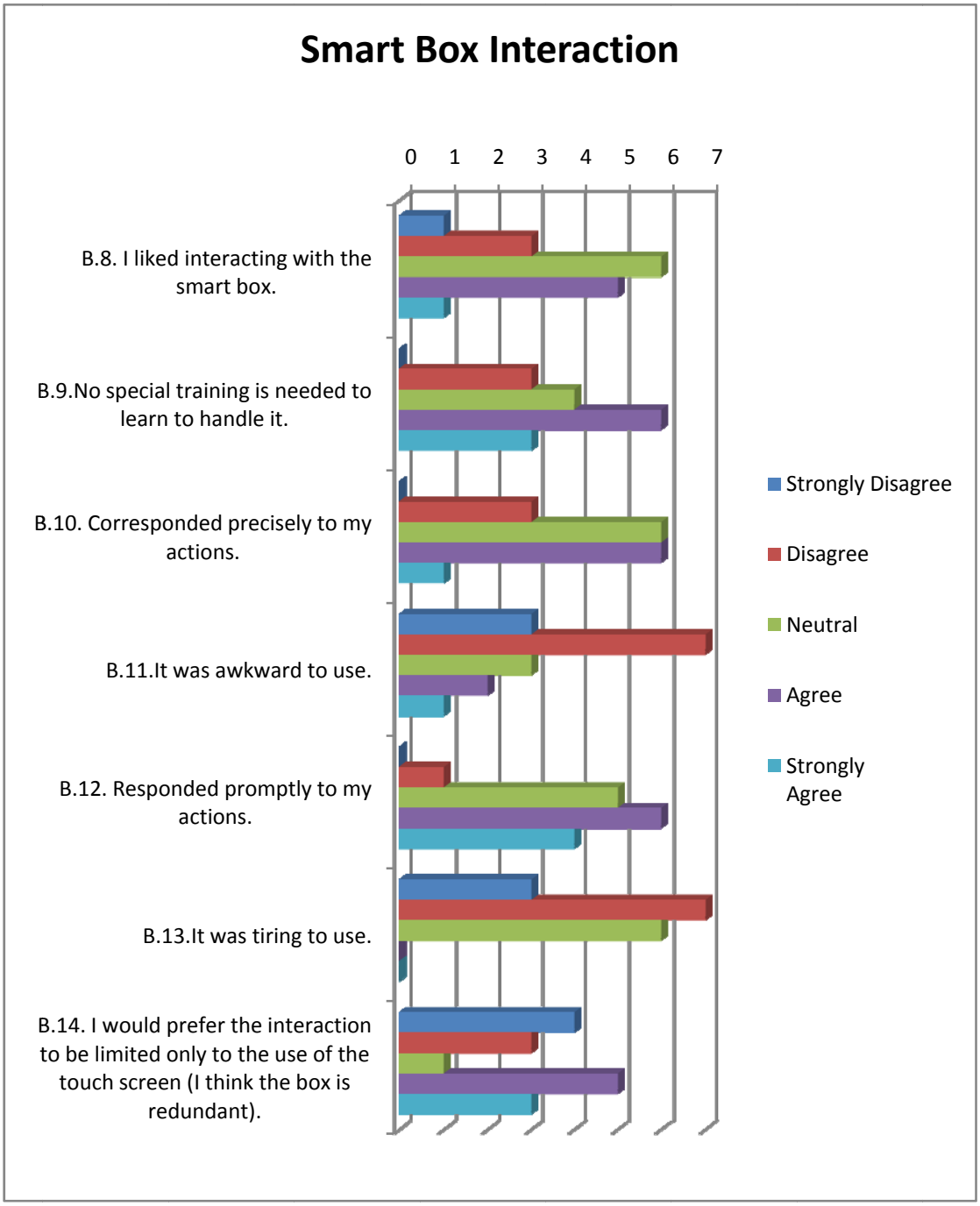

Diagram 18: Questionnaire results for the Smart Box

8.3.3 Kinect Results

The second segment of the evaluation process focused on the interaction with the system using body movement. A short explanatory introduction preceded the evaluation process in order to outline the available interaction modes that the system supports. The introduction involved all available techniques, including the user's localization, the virtual cursor, the hand gestures and the leg gestures. Despite the relatively large count of different modes of interaction, the users in general memorized the techniques successfully and did not require any further assistance throughout the system interaction.

The following table shows the tasks assigned to the participants. Tasks 1-5 are the same with the ones for touch interaction, whereas the other ones aim to evaluate the use of leg gesturing for the interaction with the system.

1. Navigate inside the Classic2D view of TimeTunnel and answer the following questions: 1.1 How many decades does the timeline cover? 1.2 How many events are there in the 1960s? 1.3 Which event categories are shown in the timeline and what others do exist?
2. Focus in the 1980s and find information regarding Apple Lisa. What types of information are shown?
3. Select the photograph of Apple Lisa displaying a man on its monitor. What is his name?
4. Find a video element and watch it.
5. Find a three dimensional model and rotate it.
6. Enter the Tunnel Mode.
7. Navigate inside the virtual tunnel and explore information regarding the event of the first computer mouse.
8. Use your legs to navigate inside the tunnel.
9. Wear 3d vision glasses and explore the tunnel.
10. Go back to Classic2D Mode.
11. Navigate in Classic2D using your legs.

The enthusiasm of the users was apparent during the use of the TimeTunnel, as only one user out of 16 stated his preference to interact with the system using more traditional modes. The users were captivated by the remote handling of the system achieved without the use of any wearable component. The moves of the hands were regarded as very natural, as indicated by the comments during the evaluation and the completed questionnaires. All

the users found the gestures representative of their functions and the only request made by a small percentage of the users was that the gestures should become shorter. The rationale of this request was due to the factor of fatigue that could emerge after prolonged interaction with the system.

The manipulation of the virtual hand (cursor) came easily to the users. A common remark involved the stability of the cursor, which the users requested to be more precise in order to be able to select items which are small in size. Furthermore, some of the users would prefer the duration of the hand being over items to be shorter so as to select them.

The hand gestures were embraced without difficulty and the users largely preferred them to the use of the virtual hand. Moreover, the system reacted as expected as far as hand gesturing is concerned, avoiding the false interpretation of the users' movements.

An issue that was raised and that was common among the vast majority of the users was the fact that single-hand gestures were deliberately ignored when the cursor was visible. This design decision was taken so as to avoid unexpected system behavior, as the users could simply move their hand in one direction to select an element. After explaining this limitation to the users, they felt comfortable with the system interaction and managed successfully to apply gestures with respect to their goal. Additionally, about half of the users (9/16) expected the expanded events inside the tunnel to collapse when making the forward gesture in accordance with the events' manipulation in Classic2D Mode.

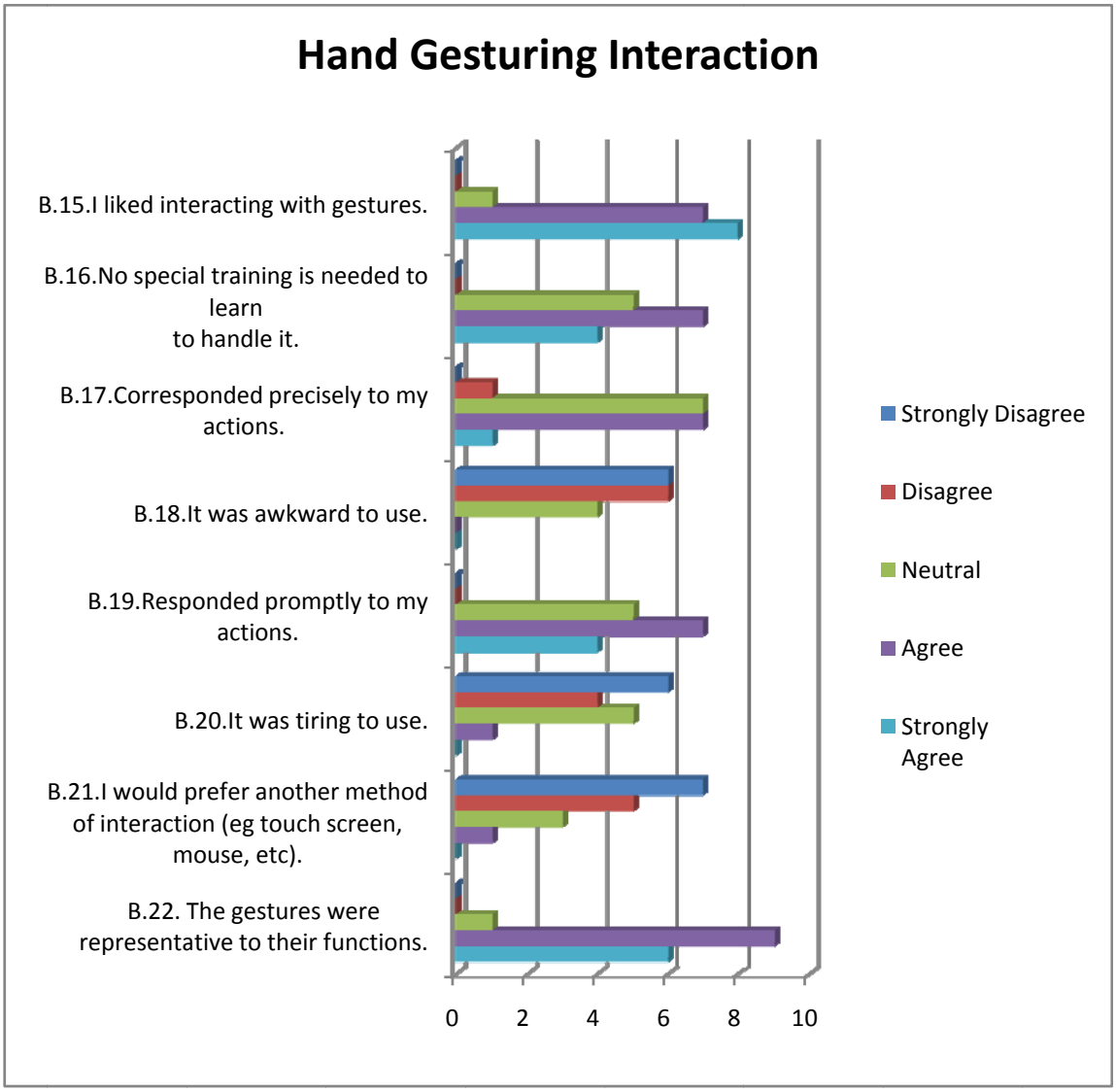

Diagram 19: Questionnaire results for Hand Gesturing

Leg gesturing was almost unanimously accepted, as only one user had trouble with navigation using his legs (Kinect failed to successfully recognize the exact placement of his legs due to his pair of trousers). The idea of stepping in any direction in order to travel in space and interacting with the system elements using their hands served the initial goal of discriminating the navigation from the interaction. This observation is more evident in the non-expert users, who supported the leg gestures even more than the expert users, as they felt more comfortable with handling the system naturally but in a strictly defined manner.

The participants did not have any problem understanding the conceptual model of moving in the space as they found it to be efficient, tireless and fascinating. This is clearly illustrated in the graph below which contains the answers from the users' questionnaires.

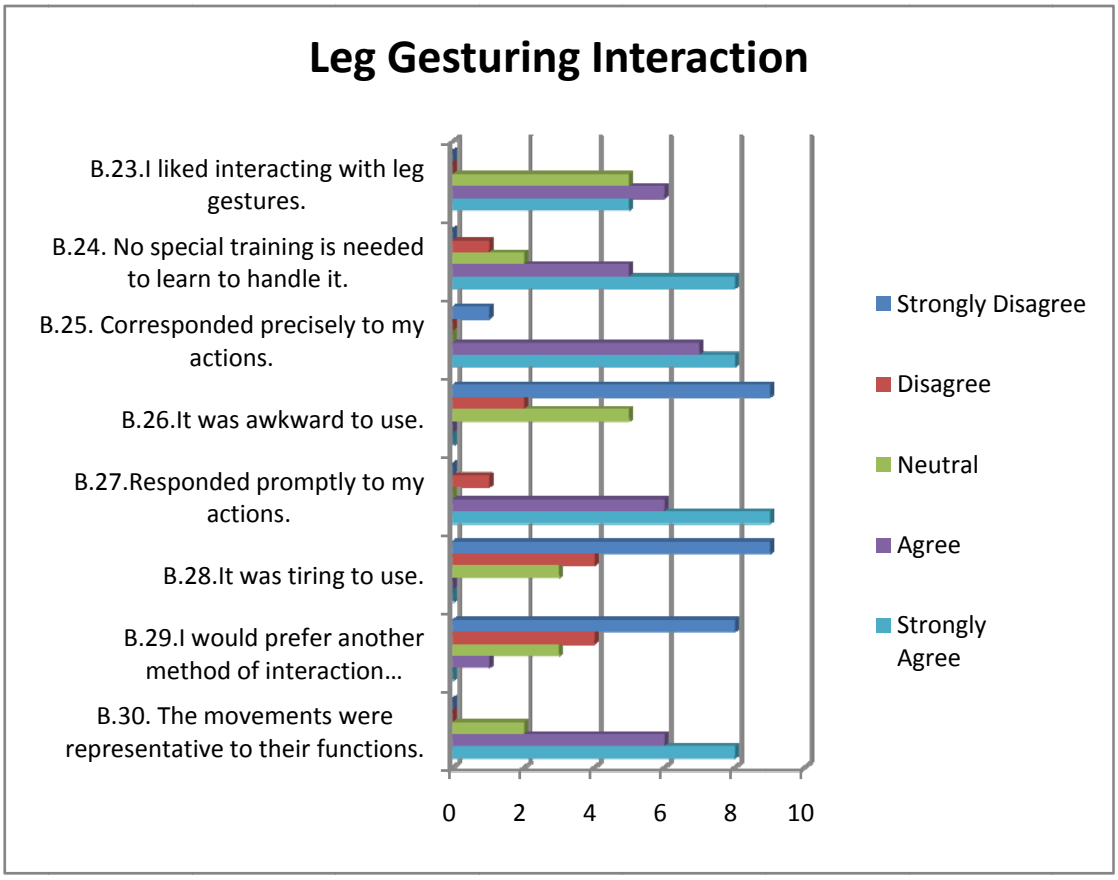

Diagram 20: Questionnaire results for Leg Gesturing

The known limitations regarding Kinect's usage include leather clothes which cannot be tracked accurately and long dresses which hide the users' legs and therefore cause the leg gestures to malfunction. However, apart from these two cases user tracking works precisely regardless of the user's outfit.

8.3.4 3D Vision Results

After the evaluation process regarding interaction by user tracking with Kinect was finished, the participants were given NVidia's 3D Vision Glasses to assess the effects of stereoscopic information presentation.

The users were overall in favor of the stereoscopic view, as 13 participants state that 3D Vision adds up value to the system's representation, while only two disagree and would prefer the normal view in a total of 16 users. Another interesting outcome of great importance is the fact that none of the participants expressed that the usage of 3d glasses was tiring or awkward, as such an issue would be a major drawback in the future. Users did not find that stereoscopic view had any direct effect, positive or negative, on their performance (this fact is depicted in the question B.35 at the graph below, where the answers of the participants spread over all possible answers); however, it improved the overall user experience and entertainment value. Some of the users stated that they felt intrigued to further explore the timeline when using the glasses.

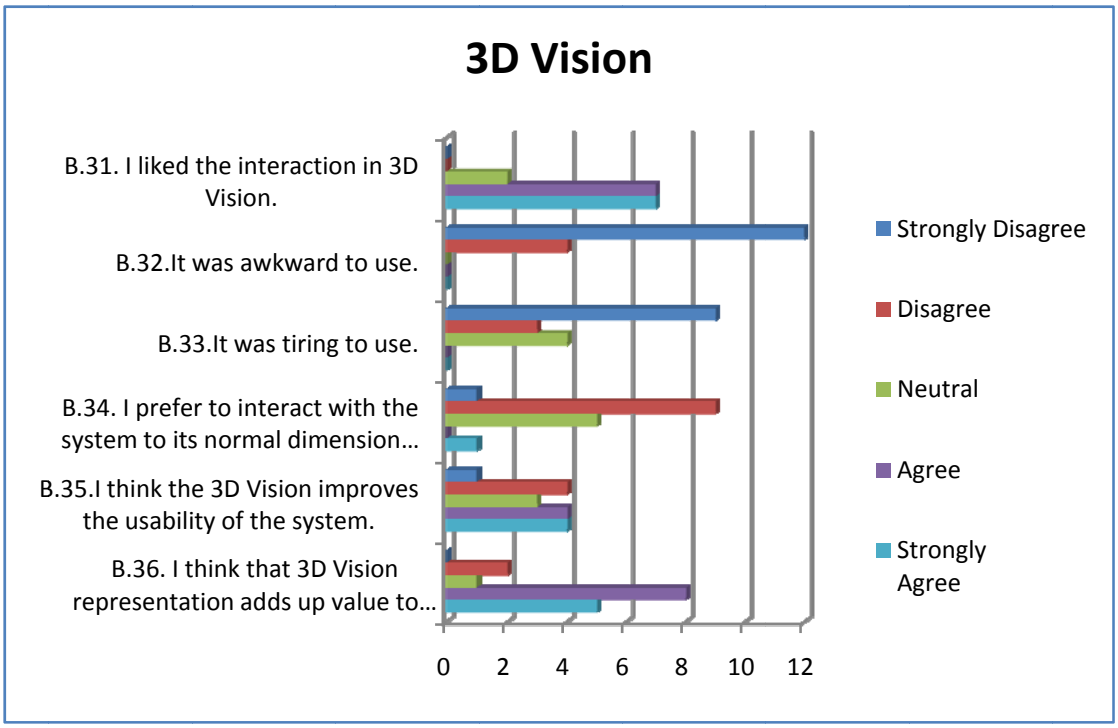

Diagram 21: Questionnaire results for 3D Vision

9 Conclusions and Future Work

Ambient Intelligence is an emerging field of research that promises to enhance traditional user interfaces and augment information visualization to be manipulated in smart environments, where users can interact in a natural manner.

The TimeTunnel reported in this thesis has been designed and developed to offer automatic creation of interactive timelines which can be manipulated using natural interaction modes. The system's design is intended to support a wide variety of contexts and contents: the Classic2D Mode fits any content and even though the current design of the tunnel may be regarded as futuristic, this impression may be instantly altered with the replacement of the three dimensional models used or even with the substitution of their textures with other ones that would better suit the content. However, these changes are totally independent of programming and do not require any deviations from the current implementation.

The system's knowledge representation is designed to support semantic information through the usage of ontologies, which are a widespread and up-to-date formal Data Model, rather than simply visualize resources. An extension of describing relationships between events could be the employment of links between them. The adoption of such an approach could lead to the direct expression of dependencies and specific relationships among events, but would also result in the complication of the graph of events. Such an extension requires extensive study and is out of the scope of this thesis but could be examined in future extensions of TimeTunnel.

Furthermore, the system's input manipulation is developed to be device-independent, through the existence of a layer that interprets input from the environment to actions that take place in the virtual three dimensional world. Therefore, any new interaction modes can be simply supported and aid the system's extendibility.

Another major issue that this thesis experiments towards its solution is the provision of natural interaction modes to handle three dimensional worlds. As shown from the system's evaluation results, TimeTunnel successfully achieves the natural interaction in a variety of means. Further experimentation is planned based on the remarks of the evaluation process in order to enhance user experience and support additional features that were visualized by the participants.

Improvements in the system's design are also planned based on the outcomes of the evaluation. The categories' representation can be improved so as to illustrate their interactivity and allow the users to perceive their meaning. This can be achieved through the replacement of the current approach by a new one, where the categories will be more discrete and will provide the sense of depth in order to depict their difference from the timeline's title.

After the successful adoption of the concept of gestures by the participants during the evaluation process, additional ideas exist in order to improve the existing interaction mode. Such an indicative example may be in the case of leg gesturing the notion of different travelling speeds so as to cover the needs of a part of the users who requested it: in addition to stepping at any direction with one leg in order to move, the practice of standing with both legs could be interpreted as a natural way to move with increased speed to the respective direction.

As far as natural interaction is concerned, the precise head tracking could benefit the systems' augmentation and increase the sense of natural interaction, as the camera in the virtual space could follow the user's head orientation.

Apart from improvements on existing aspects of TimeTunnel, several ideas exist concerning the extension of this thesis. Firstly, the need for multi-user support is of major importance. The concept of multiple users interacting with the system could be either standalone with each user viewing the information of his or her interest or cooperative, with the users sharing information with each other. Secondly, the provision of additional timelines is another addition that could be useful. The extra timelines could serve different aspects and serve different functionalities. The added timelines can:

- Be nested inside a greater timeline to describe a specific era in greater detail
- Be shown concurrently on the side to display other contexts, such as concurrent information in a different location and other types of content in the same time era.

Moreover, TimeTunnel's input could be adapted to provide up-to-date information regarding a subject through information retrieval of online sources, such as Wikipedia: this way, the timeline can be extended to support real time information.

10 References

1. Aarts E. and Encarnacao J. (2008). True Visions : the emergence of Ambient Intelligence

2. Aguilar D. A. G., Theron R. and Penalvo F.G. (2009). Semantic Spiral Timelines Used as Support for e-Learning

3. Andre P, Wilson M. L., Russell A., Smith D., Owens A. and Schraefel M.C. (2007). Continuum: Designing Timelines for Hierarchies, Relationships and Scale

4. Benko H. and Wilson A. (2009). DepthTouch: Using Depth-Sensing Camera to Enable Freehand Interactions On and Above the Interactive Surface

5. P. Booth (1989). An Introduction to Human-Computer Interaction

6. Bowman D. A., Kruijff, Laviola J.J and Poupyrev (2001).An introduction to 3D User Interface Design
7. Carmigiani J., Furht B.,Anisetti M., Ceravolo P., Damiani E. and Ivkovic M. (2011). Augmented reality technologies, systems and applications.

8. Cooca D. J., Augustob J. and Jakkulaa V.R. (2009). Ambient Intelligence: Technologies, applications and opportunities

9. Coutaz J., Nigay L. (1993). The MSM Framework: A Design Space for Multi-Sensori-Motor Systems

10. Darken R. and Peterson B. (2002).Spatial Orientation, Wayfinding and Representation

11. M. Doerr (2003). The CIDOC Conceptual Reference Module An Ontological Approach to Semantic Interoperability of Metadata

12. C. Doulaverakis, Y. Kompatsiaris and M. G. Strintzis (2005). Ontology-Based Access to Multimedia Cultural Heritage Collections - The REACH Project
13. M. H. Fogtmann, J. Fritsch and K. J. Kortbek (2008). Kinesthetic Interaction - Revealing the Bodily Potential in Interaction Design

14. Y. Georgalis, D. Grammenos and C. Stephanidis (2009). Middleware for Ambient Intelligence Environments: Reviewing Requirements and Communication Technologies

15. Ghiani G., Paterno F., Santoro C. and Spano L. D. (2009). UbiCicero: A location-aware, multi-device museum guide

16. Grammenos D., Georgalis Y., Kazepis N., Drossis G. and Ftylitakis N. (2010). The BooTable experience: iterative design and prototyping of an alternative interactive tabletop

17. Grammenos D., Zabulis X., Michel D., Sarmis Th., Tzavanidis K., Argyros A. and Stephanidis C. (2011). Design and Development of Four Prototype Interactive Edutainment Exhibits for Museums

18. Hilliges O., Izadi Sh., Wilson A., Hodges St., Mendoza A. G. and Butz A. (2009). Interactions in the air: adding further depth to interactive tabletop

19. Hoeber O. and Gorner J. (2009). BrowseLine: 2D Timeline Visualization of Web Browsing Histories

20. Jaimes A., Sebe N. (2006).Multimodal human–computer interaction: A survey

21. Jensen M. (2003). Visualizing Complex Semantic Timelines

22. Kamm C. (1995). User interfaces for voice applications

23. Kapler Th. and Wright W. (2005). GeoTime Information Visualization

24. Keefer R., Narayanan S., Bourbakis N. (2010).Voice Commands for a Mobile Reading Device for the Visually Impaired

25. Klante P., Koresche J. and Boll S. (2004) AccesSights – A Multimodal Location-Aware Mobile Tourist Information System

26. Kraak M. J. (2005). Timelines, temporal resolution, temporal zoom and time geography

27. Krishnan A. and Jones S. (2005). TimeSpace: activity-based temporal visualization of personal information spaces

28. Kullberg R. L. (1995). Dynamic Timelines Visualizing Historical Information in Three Dimensions

29. Kumar V., Furuta R. and Allen R.B. (1996). Interactive interfaces for knowledge-rich domains

30. Lee J. Y., Rhee G.W., Kim H., Lee K.W., Suh Y. H. and Kim k. (2006). Convergence of Context-Awareness and Augmented Reality for Ubiquitous Services and Immersive Interactions

31. Marcus A., Bernstein M. S., Badar O., Karger D. R., Madden S. and Miller R.C. (2011). TwitInfo: Aggregating and Visualizing Microblogs for Event Exploration

32. Matkovic K., Lez A., Gracanin D., Ammer A. and Purgathofer W. (2010). Event Line View: Interactive Visual Analysis of Irregular Time-Dependent Data

33. Molyneaux D. and Gellersen H. (2009). Projected interfaces: enabling serendipitous

interaction with smart tangible objects

34. Nickel K. and Stiefelhagen R. (2003). Pointing Gesture Recognition based on 3D-Tracking of Face, Hands and Head Orientation

35. Open Natural Interaction (OpenNI), http://openni.org/Documentation/ProgrammerGuide.html

36. J. van Ossenbruggen, L. Hardman (2005). Semantic timeline interfaces for annotated multimedia assets

37. Oviatt S.L. (2003). Advances in Robust Multimodal Interface Design

38. Pandya A., Mulye A. and Teoh S. T. (2009). Enhancing Online Timeline Visualizations with Events and Images

39. Papadopoulos Ch., Sugarman D. and Kaufmant A. (2012). NuNav3D: A Touch-less, Body-driven Interface for 3D Navigation

40. Papagiannakis G., Singh G. and Magnenat-Thalmann N. (2009). A survey of mobile and wireless technologies for augmented reality systems

41. Plaisant C., Milash Br., Rose A., Widoff S. and Shneiderman B. (1996). LifeLines Visualizing Personal Histories

42. Protégé Ontology Editor and Knowledge-base Framework Overview, http://protege.stanford.edu/overview/

43. Resource Description Framework Overview, http:/www.w3.org/RDF/

44. B. G. Robertson (2006). Visualizing an Historical Semantic Web with Heml

45. Roes I., Stash N., Wang Y. and aAroyo L. (2009). A Personalized Walk through the Museum: The CHIP Interactive Tour Guide

46. Sangsuriyachot N., Mi H. and Sugimoto M. (2011). Novel Interaction Techniques by Combining Hand and Foot Gestures on Tabletop Environments

47. Serrano M., Nihay L., Lawson J-Y.,Ramsay A., Murray-Smith R. and Denef S. (2008). The OpenInterface Framework: A tool for multimodal interaction

48. Shriver S., Toth A., Zhu X, Rudnieky A. and Rosenfeld R. (2001).A unified design for human-machine voice interaction

49. Siegel M. (1991). A metadata approach to resolving semantic conflicts

50. Siegemund F., Floerkemeier and Vogt H. (2005). The value of handhelds in smart environments

51. SIMILE Project, http://simile.mit.edu/wiki/Papers

52. SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/

53. Stab S., Nazemi K and Fellner D. W. (2010). SemaTime - Timeline Visualization of Time-Dependent Relations and Semantics

54. Stivers T. and Sidnell J. (2005). Introduction: Multimodal interaction

55. Terrenghi L., Kranz M., Holleis P. and Schmidt A. (2005). A cube to learn: a tangible user interface for the design of a learning appliance

56. M. Uschold, M. Gruninger (2004). Ontologies and semantics for seamless connectivity

57. Valkov D., Steinicke F., Bruder B. and Hinrichs K. (2010). Traveling in 3D Virtual Environments with foot gestures and a multitouch enabled World In Miniature

58. A. Van Dam (1997). Post-WIMP User Interfaces

59. W3C Semantic Web Frequently Asked Questions. W3C Semantic Web FAQ, http://www.w3.org/2001/sw/SW-FAQ

60. Web Ontology Language Overview, http://www.w3.org/TR/owl-features/

61. Weber w., Rabaey J. and Aarts E. (2005). Ambient Intelligence

62. Weiser M. (1993). Some Computer Science issues in Ubiquitous Computing

63. Wongsuphasawat Kr., Gomez J.A.G, Plaisant C., Wang T.D., Shneiderman B. and Maimom M. T. (2011).LifeFlow: Visualizing an Overview of Event Sequences

64. Yoo B., Han J-J.,Choi Ch., Yi K., Suh S., Partk D. and Kim Ch. (2010). 3D User Interface Combining Gaze and Hand Gestures for Large-Scale Display

65. Zhao J., Drucker S. M., Fisher D. and Brinkman D. (2012). TimeSlice: Interactive Faceted Browsing of Timeline Data