



Vrije Universiteit Brussel

Faculteit Wetenschappen en Bio-ingenieurswetenschappen
Vakgroep Computerwetenschappen

Tangible Exploration and Querying of Cross-Media Information Spaces

Proefschrift ingediend met het oog op het behalen van de graad van
Master of Science in de Ingenieurswetenschappen: Computerwetenschappen

Antonios Kafazis

Promotor: Prof. Dr. Beat Signer
Begeleider: Dr. Bruno Dumas

AUGUSTUS 2014





Vrije Universiteit Brussel

Faculty of Science and Bio-Engineering Sciences
Department of Computer Science

Tangible Exploration and Querying of Cross-Media Information Spaces

Graduation thesis submitted in partial fulfillment of the requirements for
the degree of Master of Science in Applied Science and Engineering: Computer Science

Antonios Kafazis

Promoter: Prof. Dr. Beat Signer
Advisor: Dr. Bruno Dumas

AUGUST 2014



Abstract

Retrieval of digital information from databases is typically achieved through querying relational databases using Graphical User Interfaces (GUI). In contrast to GUIs, Tangible User Interfaces (TUI) are used to achieve interaction with digital systems using objects from the physical world. Researchers have tried to merge relational database querying with TUI, and give the users the chance to retrieve digital information through real-world tangible physical objects. This way, users are able to query databases using objects without the need to write queries in SQL (Structured Query Language) or a similar language.

In our work, we address this area of TUI and relational databases, designing and developing a Tangible Query Interface (TQI). We went through an extensive background work study and related work exploration and review. With this study a comparison between available TQI systems is done and decisions are extracted in what direction to follow in our work. ArtVis is a system of advanced visualisation with a basic TQI that was developed in the WISE lab in the past, mainly designed for a museum or exhibition context. We decided to apply our findings on a new TQI of this system. The extracted requirements helped us decide on the final functionality. A prototype was produced out of our design. We decided to develop a new TQI based on a tabletop interface using real-world objects with meaningful metaphors to their functionality. Complex queries matched to SQL queries can be formed in an individual or collaborative user environment. A broad range of functionality is covered and SQL operators are mapped to TQI. In this thesis we extensively describe and analyse the above context. Conclusions are extracted and paths to future work are carved that will either extend our work, or apply it in other contexts.

Acknowledgements

This thesis helped me gain experience towards research. I learned to properly conduct research, that will help me not only in Computer Science or Sciences in general, but in every aspect of my life. I would like to thank Prof. Dr. Beat Signer for letting me work on this thesis and sharing his knowledge and my advisor Dr. Bruno Dumas for his help and advice through the whole period of this work, also sharing his knowledge. I would also like to thank Sandra Trullemans, PhD student at the WISE lab, for her precious help with the building of the table of our interface. Finally, I would like to thank my family and my girlfriend for their support during this whole time.

Contents

Abstract	1
Acknowledgements	2
List of Figures	6
List of Tables	7
1 Introduction	8
1.1 Research Context	8
1.2 Problem Statement	9
1.3 Research Question	10
1.4 Approach	10
1.5 Outline	11
2 Background and Related Work	13
2.1 Background	13
2.1.1 Tangible User Interfaces	13
2.1.2 Information Visualisation	16
2.1.3 Tabletop Interfaces	18
2.2 Related Work	19
2.2.1 Tangible Query Interfaces	19
2.2.2 TeddIR	23
2.2.3 Interactive Venn diagram	24
2.2.4 Navigational Blocks	26
2.2.5 Stackables	28
2.2.6 The Search Wall	30
2.3 Comparison of Related Work	33
3 ArtVis	36
3.1 ArtVis System	37

3.1.1	Database	37
3.1.2	Application	37
3.2	ArtVis TUI	38
3.3	Tangible Querying	39
4	Requirements Analysis	40
4.1	Use Cases	40
4.1.1	ArtVis Compliant Use Case	40
4.1.2	Intersection	40
4.1.3	Union	41
4.1.4	Negation	41
4.1.5	Multiple Queries	41
4.1.6	Saving a Query	42
4.2	Requirements	43
5	Design	44
5.1	Scenario	44
5.2	General Design	46
5.3	Solution Description	48
5.3.1	Operand Physical Objects	48
5.3.2	Grouping Operands within Queries	48
5.3.3	Boolean Operators within a Query	48
5.3.4	Negation	49
5.3.5	Unifying Queries	49
5.3.6	Saving & Retrieving Queries	50
5.3.7	ArtVis Museum Console	51
5.4	Requirements Fulfillment	51
5.4.1	Functional Requirements	51
5.4.2	Non-Functional Requirements	52
6	Implementation	54
6.1	Architecture	54
6.2	ReactIVision Framework	55
6.3	Tabletop Interface	56
6.4	TQI Application	58
6.4.1	Key Elements	58
6.4.2	Message Protocol	59
6.4.3	Testing	60
6.5	ArtVis Changes	63
6.5.1	Prefuse	63
6.5.2	Querying the Data	63

6.5.3	Receiving the Query Groups	64
6.5.4	Query Processing	64
6.5.5	Saving and Retrieving Queries	66
7	Conclusions and future work	68
7.1	Future Work	68
7.2	Conclusions	69
7.2.1	Contributions	72
7.2.2	Query Operands	72
7.2.3	Query Operators	72
7.2.4	Saving and Retrieving Queries	73
	Bibliography	74

List of Figures

2.1	MetaDESK - A general overview	14
2.2	Tangible Query Interfaces - First prototype	20
2.3	Tangible Query Interfaces - Second prototype	21
2.4	TedDIR - The complete setup	23
2.5	Interactive Venn Diagram in use	25
2.6	Navigational Blocks in use and their facets	27
2.7	Stackables - The tokens	29
2.8	Stackables in use	30
2.9	The Search Wall in use by a youngster	32
3.1	The complete ArtVis for museums	37
5.1	Jigsaw puzzle concept	46
5.2	The architecture of the scenario	46
6.1	The architecture of the new ArtVis TQI	54
6.2	The ReacTIVision framework architecture	56
6.3	The projector and PS3Eye	58
6.4	The table in action	61
6.5	Under the table	61
6.6	What ReacTIVision "sees"	62
6.7	The result on the ArtVis canvas	62
6.8	The information visualisation reference model	63

List of Tables

2.1	Comparison of related work	35
4.1	Requirements for the new TQI	43
7.1	Comparison between related work and our systems	71

Chapter 1

Introduction

1.1 Research Context

For the last decades, interaction between users and digital information has been accomplished in the same way. The user types, points and clicks with a mouse, and lately, touches with her finger in order to access and query this information and be supplied with a digital visual representation as a result. This result can, for example, be text, graphs or images. People need to consciously become the system's users and adjust their whole mindset to interact with it.

Tangible (User) Interfaces (TUI) are systems where the interaction between the physical and the digital world is achieved through physical media that control the digital information. The physical part of the system also acts as representation of concepts and characteristics related to the digital information. The initial name used was Graspable User Interfaces [17], but it was replaced and became known as described above. One of the pioneers of the research context addressed is Hiroshi Ishii with the Tangible Bits [28], but the initial idea was captured by Durrell Bishop that created the Marble Answering Machine [46]. The motivation behind TUI relies on a particular Human-Computer Interaction point of view, where the interfaces between users and digital information are seamless.

TUIs have been part of and used in different ways and contexts. Tangible interfaces are used to augment paper, an area also known as interactive paper or paper-based tangibles. Another "sub-genre" of tangible interaction is identified in using self-describing building blocks, using blocks to identify structures and 3D models. A very active area of TUI is related to toys, both for educational (mainly) and entertainment purposes and aiming towards younger audiences. Among others, programming is taught as the learning aspect of these interfaces. Music interfaces,

mainly due to the introduction of the ReactIVision framework [32], discussed later, have become a fruitful application of the TUI. Industrial and interaction designers have also designed tangible objects that approach the concept of TUI. In many cases [18], tangible interaction is mixed with augmented and mixed reality. Tabletop interfaces are also an emerging application of said interface, having a practical application in most of the above fields and, usually, promoting collaboration between users.

Ullmer and Ishii [50] categorise TUI approaches and applications into three major categories. This is done without a goal to formalise this taxonomy, but to have a better understanding and characterise these systems as a leverage to research. TUI systems can be considered belonging to one or a combination of more of these categories: spatial systems, relational systems and constructive systems. Spatial systems involve spatial settings of the physical object tokens (physical objects used to reference digital information) in order to interpret and augment the system. Relational approaches connect the sequence, adjacency and logical relationships among the tokens to interpret digital data. Constructive ones, use assembling of elements analogous to well-known toy assembling blocks.

Users apart from using TUI as a physical substitute of Graphic User Interfaces (GUI) can also make use of them to mine data. One can use tangible objects in her physical environment to ask questions and get digital feedback. The goal is to use physical objects representation as metaphors or use generic objects to retrieve knowledge from a digital system, helping non expert users (children, elder), or even experienced ones, to use it more naturally and / or efficiently.

Attempts have been made to find a happy medium to this context. Ullmer [52] introduced a TUI system for querying a database. Since then, there has been research done in order to investigate this area of TUIs. In these attempts, different approaches were investigated and physical tokens were used in different ways to achieve those goals.

1.2 Problem Statement

Limited research has been conducted for this interesting approach of TUI. Current approaches try to bridge the gap between the physical and digital world to retrieve information. Most of them try to make a connection between Boolean operators and operands, and the tokens used. Those are aiming to provide a fair mapping between Structured Query Languages (SQL) and TUI Systems, but in most cases do not provide an extensive relationship. The vast majority of research is oriented

towards collaborative use of these systems as well as usability.

In most cases, query formation using physical tokens is quite standardised. This happens as operators are usually neglected or constrained at the expense of the query's operands. A way to make use of more operators and flexibility on the user side to apply them will not only help users retrieve more accurate data, but will also lead to leveraging the use of querying TUI digital information systems. With a view to better investigate what is currently provided and what is actually missing, a centralised study should be conducted to help decide the steps that need to be taken. These steps should aim towards a more extensive relation between query languages as we know them and a tangible counterpart.

1.3 Research Question

Following the research context we will work upon and the problem statement we can summarise by formulating the following question:

- Question 1: How does the state of the art address tangible querying?
- Question 2: What are the links between tangible querying and InfoVis (Information Visualisation)?

1.4 Approach

In this thesis, an already existing digital system is used as the foundation for the research conducted. ArtVis [15] is a result of a research exploring visualisation techniques. The result was a system suitable to be used within the context of a museum using advanced visualisation techniques. Among other things, a simple TUI alternative to the main GUI is used to query the database and return certain results. The TUI of this system will be rebuilt in order to reflect the problem to be addressed, as stated in the previous section.

To begin with, we do an extensive research of the current approaches available for tangible querying of digital information. By analysing the advantages and drawbacks of the approaches, we are able to determine the gaps that need to be filled. Other approaches were followed before the final one described and discussed extensively on this thesis, but were overruled. The chosen approach is the following.

The approach followed takes into account the need for a more generic relational TUI system, but still keeping in mind that it should be suitable to be used within the context of a museum. Thus, there is an impact from the digital information database to be queried, the users it is addressed to and the environment it is intended to be used in. In our case the aim is to retrieve digital information through information visualisation, in a museum or exhibition environment, from different kinds of users varying in age and technology expertise.

A TUI able to be used and understood by a wide range of novice to expert users is conceived. The setup and structure aims to promote collaboration between users, among others, giving the chance to collectively retrieve knowledge from digital information. A table is chosen as the active area of the interface, offering the space needed for such an approach. Several physical objects can be placed on top of this area, allowing users to work both independently and together, in order to achieve a common goal, or even observe the outcome of each other's questions and ideas.

Several tangible objects can be used at the same time to form one, or more queries to be combined, and query the system. Groups of objects form a single query and all the groups/queries co-existing on the table at the same time form a complex query. In order to group objects, special symbolic tangible objects are created to designate the area of a group. All objects in the vicinity of this object are subject to be used in the same query. The query changes in real-time as the objects move and / or change groups, objects are added, etc.

1.5 Outline

Following the introduction a brief summary of the content of each chapter is provided.

In *Chapter 2* we provide a general background overview to our thesis and related work to our research. Afterwards, a comparison between the related work and our work is provided and analysed.

ArtVis is described in *Chapter 3* with its initial state. Also, the approach taken for its TQI is presented.

Chapter 4 provides the requirement analysis of our research through use cases. At the end of the chapter, we provide a table where we gathered all the requirements.

The design of our proposed system is described in *Chapter 5*. A scenario we investigated before our final approach is also discussed. Finally, a section is dedicated to the requirements' fulfillment.

In *Chapter 6* the architecture and implementation of our prototype is described. The changes done in ArtVis are described, too.

Chapter 7 holds the conclusions and discusses possibilities for future work.

Chapter 2

Background and Related Work

In this chapter we will discuss the background of our work. We will focus on Tangible User interfaces, Information Visualisation, and Tabletop Interfaces, in order to provide some solid ground for our work. All these areas are key contents of our research and are tightly intertwined to the final product of our research. TUIs are the basis of of Tangible Query Interfaces (TQIs) as they represent different techniques, concepts, and frameworks towards integrating tangible interaction. We will focus on TQI in our related work but here we will discuss TUI in general. InfoVis is a key concept, too, as ArtVis main scope and functionality focusses there. Tabletop Interfaces are also of high importance because we decided to proceed by building our TQI system on top of one. ArtVis will be described in depth in Chapter 3.

Related work to this thesis is described and discussed afterwards. Existing approaches related to tangible query interfaces are presented in order to bring a clear view to the reader about what is already implemented. This chapter does not only cover state of the art methodologies and prototypes, but also the most important research papers related to the field. A section is provided for each one of them where a synopsis is provided underlining the most important elements. This chapter is fundamental for capturing what is currently provided in the field and where there is a gap.

2.1 Background

2.1.1 Tangible User Interfaces

The notion of TUI was first proposed by Ishii [28] built upon the work of Fitzmaurice [17]. TUI initially appeared as Graspable User Interface, which is now

obsolete as nowadays the former name is used. Fitzmaurice [17] introduced the concept of Graspable UI. Through physical artifacts, called "bricks", and physically handling them, electronic or virtual objects can be manipulated. This new paradigm was introduced as a mean of augmenting the GUI. Before the research around TUI was boosted, a conceptual TUI was designed by Durell Bishop as the Marble Answering Machine [46]. Urp [55] is a TUI for urban planning and architecture. It is an excellent example of an approach of a TUI which will benefit designers and urban planners in their work. An interesting study [42] was conducted about TUI in general, discussing approaches, techniques, frameworks, along with a more general discussion about TUI.

In [49] we have a first attempt to formalise mapping between TUI and GUI elements through working with the metaDESK prototype (Figure 2.1). A prototype application is built to test those and help the user work seamlessly between the physical and digital world. The mapping between GUI and TUI happens at different levels and aspects. The windows are instantiated as lenses, icons as symbolic tangible objects known as phicons, the menu uses a tray, handles are phandles, similar to Fitzmaurice [17], and control is instantiated as an instrument of similar capabilities. Phicons in relation to icons can be of different abstraction, generic, symbolic, model, and actuality. Another system that works as a framework is the Illuminating Light system [54] which is trying to connect light and physicality. Physically instantiated abstractions are augmented with visual feedback. This feedback is passed to the workspace where everything takes place by an overhead I/O Bulb, a conceptual variation of a high-resolution light bulb. An approach [38] provides a TUI where everyday objects are visually unaltered, while computer vision and an RFID reader determine the position and ID of the tangible object.



Figure 2.1: MetaDESK - A general overview

The concepts and research since Ishii [28] are summarised from the work of different researchers in [27], also discussing a model of TUI, and different properties and genres. TUI is considered to be still in its infancy by the time the paper was written in 2008. A framework for the interaction model of TUI is presented in Ullmer [50]. This model is a progression from the model-view-controller (MVC) model of GUI, to a similar one for TUI, called MCRpd. MVC is mainly placed within the digital layer of a system, with the input and output taking place in the physical layer through the controller and view part. Drawing from this model, the MCRpd (model-control-representation (physical and digital)) positions the model still within the digital layer, but all the other parts purely within the physical layer. Another system that influenced this framework is mediaBlocks [51] which is used, in contrast to Urp, to provide a physical interface to more abstract digital information by navigating media files.

The same research [50], identifies TUI properties and categorises them in three major categories: spatial, relational, and constructive systems, along with a combination of those categorised systems. The first category contains systems where the spatial configuration of the physical tokens matters. In relational systems, the sequence, adjacency, and logical relationships of the physical tokens reveal abstract interpretations. Mechanical connection between objects are a main characteristic of constructive systems.

An example of spatial systems is the already discussed Urp. Sensetable [39], is a tabletop interface where physical objects are detected wirelessly, as opposed to computer vision. Attaching dials and modifiers on these objects, modifies their state. This system can be used in business modelling, along with urban planning, interactive visual and audio art. MagicPad [10] is a spatially aware TUI using also augmented reality in a CAVE-like system. Through a pad and an infrared pen, one may manipulate and interact with visualisations within a 3D environment. A relational system that we will analyse extensively later, due to the very important relation to our work and research field of TQI, is [52]. Also mediaBlocks discussed above belongs here. Constructive systems are jigsaw puzzle TUIs like [5], [58] or learning TUI for computer programming like [25], where children learn computer programming concepts in a museum environment, through tangible objects.

On top of the broader generic categories, application domains can also be found. According to [50], Information storage, retrieval, and manipulation (a) is one of the largest classes of TUI. Information visualisation (b) is another one. Modelling and simulation (c), like Urp, is another, and systems management, configuration, and control (d), to manipulate and control complex systems. Finally,

TUIs can be used for education, entertainment, and programming (e). Our system that is a product of this thesis and is extensively presented and discussed later, can be considered as a mixture of (a), (b), and (e).

The TAC paradigm [6],[44] is a conceptual framework to represent TUIs. The TAC paradigm is based on the Token+Constraint classification, something that is formalised in [53]. Rules and terminology were introduced to provide a formal design of TUIs and left for the future the application to software architecture design patterns specifically for TUIs, toolkits. The Token+Constraint was already used in different approaches but the technique was identified in [52] and as already said analysed extensively in [53]. The paper of Shaer [43] introduces TUIML (TUI Modelling Language), a high level User Interface Description Language (UIDL) targeting to provide a common ground for creating TUIs for developers of different disciplines. Conventional TUI prototyping goes through slow cycles of prototyping hardware and testing. The virtual TUI prototype (vTUI) offers faster cycles, saving time and cost, due to it being virtual. A framework for TUIs is proposed in [26] on connecting the physical and the social, providing concepts and perspectives for the social aspects and user experience of TUI. The framework is structured around four themes; the tangible manipulation, spatial interaction, embodied facilitation, and expressive representation.

Education is a very important field for TUI. Electronic Braille Blocks [29] is teaching through a TUI, the Braille letters to young blind children. In [25], that we already mentioned, computer programming is taught to children in a museum, while [24] gives an alternative approach for a school environment. AstroGrasp [1] uses a tabletop TUI to teach its users through collaboration, basic astronomy concepts of eclipse and seasons. Linguabytes [22] is a study where a play-and-learning system was designed for toddlers with multiple disabilities to help them in the development of language through a storybook reading. A discussion and reviews of some TUIs for assisting children learning takes place in [59]. The study concludes that TUI is of a great potential given enough technology support, leverages exploration, expression, and collaboration in relation to other interfaces, and, in general, benefits children's learning.

2.1.2 Information Visualisation

Information visualisation (InfoVis), based on [9] is an interactive visual representation of abstract data aiming to amplify human cognition, through a computer-based digital environment. This data can be of any type, from numerical values to text and location. Amplifying the human cognition means that one can extract knowledge about the data and recognise relationships between the data, leading

to conclusions. In this context, InfoVis is trying to provide tools to handle information overload. The assistance that is provided with the information glut [45] is related to actually turn the data into use, extract knowledge, employ the data in decision making processes, etc. In general, assist in avoiding getting overwhelmed by the huge amount of data available. InfoVis may be considered ideal for exploration, especially when the goal is vague, and using raw data is either hard or impossible to handle by humans.

One of the first examples of InfoVis, widely used across papers and courses, is M. Minard's static visualisation of Napoleon's March on Moscow in 1812, drawn in 1869 [16]. Another fine example, which actually led to discovering the source of an epidemic, is the map of London in 1854 marked by the position of cholera victims [16]. In the same paper [16], the authors consider InfoVis from different points of view, and explain the values of it through arguments, an important research for strengthening the use of InfoVis. A classification of InfoVis and visual data mining techniques is proposed in [34] based on data type to be visualised (one-, two-, or multidimensional table data), the visualisation technique (e.g. charts, scatter plots, etc), and the interaction and distortion technique (e.g. zooming, filtering, etc). The classification allows any visualisation technique to be used along with any interaction technique and any data type. Interactivity, according to [14], will handle these tasks with an added value: highlighting and focus, accessing extra info (drill down and hyperlinks), overview and context (zooming and fisheye), same representation –changing parameters, same data– changing representation, and linking representations (temporal fusion).

ArtVis [15] offers a number of InfoVis techniques. It combines panels with different visualisations, such as semantic zoom, details on demand, stacked area chart, fisheye view, the latter is introduced in [19]. In [52] the author combines TUI with InfoVis to produce Tangible Query Interfaces. Tangible View [48] is a spatially aware TUI to augment visualisation parameter handling, providing patterns to use in common visualisation tasks. An interesting approach for InfoVis in a museum (or exhibition) environment is EMDialog [23], an installation for an art exhibition about the life of an artist. Two interlinked text visualisations are presented about an artist, with the dimensions about context and time. Users are curious and have the desire to further explore the digital data, because of the presentation and interaction techniques, the visualisation's content should be accessible. Collaborative use of the system is of benefit to the social aspect of a museum visit, and the exploration, in any style, either short-term or long-term should be rewarded. Toolkits for the development of visualisations exist, where Prefuse [21] is a product of conducted research, where the toolkit was evaluated as usable and effective by usability studies and usage surveys on developers.

2.1.3 Tabletop Interfaces

Tabletop interfaces are gesture interaction devices which have as their main feature that they have the form of a flat horizontal surface, like a table, with some specific features. The input of the interface is continuous as the user touches the surface, and it can be multi-finger and multi-user, thus meaning a large number of input events to manage and interpret, and location specific, because the users need to touch the surface of the table. Apart from touching which is not always a requisite, those surfaces may also recognise physical objects, thus making them important for some TUIs.

The tables may support different technologies in the following main categories. An example of a marker-based table is ReactIVision [32] which will be discussed further in a following chapter, as it is a core type of tabletop interface for our research. DiamondTouch [13] is using capacitive coupling where users are identified by sitting on chairs transmitting a low voltage current also allowing objects not to obstruct the table's functionality, but has a high cost. Frustrated Total Internal Reflection (FTIR) tables are introduced in [20], using a simple technique to build an inexpensive and scalable high-resolution multi-touch table with rear projection. The cost and simplicity to build them allows more developers and designers to be involved in the creation of tabletop interfaces, and can be coupled with marker-based technologies. Microsoft PixelSense¹, initially known as Microsoft Surface, is an optical glass tabletop. It is based on a Samsung table of full HD 40 inches display, a high-end table without the need of a camera to record the active area, being also really thin. Except for direct interaction through multi-touch input, physical objects recognition is available through tagging them. The table is highly responsive, but as a negative point is expensive.

Territoriality in a collaborative tabletop interface is understood better due to the research conducted by Scott [41]. Users that collaborate use three types of tabletop territories to assist in their interaction: personal, group, and storage. The first area is where independent activities take place, while the second one is reserved for group tasks. The third type is used to store task resources (such as tools, unused items, reference material) and non-task items (such as food and drinks). Tabletop interfaces provide enough room for all these territories to exist, and also allow more than one user to sit or stand around and share the interface.

Digital Desk [57] is one of the pioneers of tabletop interfaces where real paper is augmented with digital information from a projector displaying from above the table. Gestures are identified through a camera, and pen and paper are digitised.

¹<http://www.microsoft.com/en-us/pixelsense/default.aspx>

Physical objects are used on the tabletop interfaces of [17] and [55] where virtual objects are manipulated through the use of physical ones. Multimedia storytelling is achieved through manipulating tangible objects on a tabletop interface in [37]; collaboration is also a main element. Lumino [4] provides a table where the objects position on it can not only be recognised horizontally, where the object's 2D marker touches the surface, but also vertically, through recognising multiple markers. To achieve that, glass fiber bundles are used for the blocks to reflect the light of the table to some height above the surface without the need of each point to touch the surface. DataTiles [40] combines GUI and TUI through tagged transparent tiles. The tangible tiles are augmented with dynamic graphical information when they are placed on the surface of the tabletop interface. In a similar way, Tangible Tiles [56] use transparent physical plexiglas tiles to manipulate visual data (e.g. erase, magnify, move images).

2.2 Related Work

In this section we analyse extensively each of the related work to our thesis. This part is a product of the research training of this thesis.

2.2.1 Tangible Query Interfaces

The fundamental research on tangible query interfaces is [52], a paper written in 2003. By that time most TUIs developed were using a one-to-one mapping between the physical objects used and the corresponding digital elements. This raised a problem on the limitations of using TUIs to manipulate information. For example, a database can contain a huge number of digital elements. Using unique physical objects for each one of them is a situation that will eventually turn into a mess. What is proposed is that physical objects can be used not to be mapped to individual digital elements, but to represent expressions containing a large amount of information.

In order to investigate this concept, two prototypes were built around a real estate application and used to test. The tangible interface used to query the sample database should preserve and promote the benefits of TUIs.

The first prototype (Figure 2.2) uses parameter wheels to manipulate the queries. Each wheel is considered as a token and represents a different parameter. The corresponding parameter is marked on the token with a passive label. The wheels are divided containing continuous and discrete parameters. The wheels are placed on a series of query pads part of a query rack and upon placing one it is a part of the

active query. The interface contains two views with two different visualisations. The first one is a scatterplot and the other one a map with the real estate location. The wheels represent different parameters for the X and Y coordinates of the scatterplot giving real time feedback to the view making it easy to manipulate the application with two hands. It should be noted that, according to the paper, the parameter wheels are implicitly joined with the Boolean AND relation, thus being understood that there is not an option for other relations.

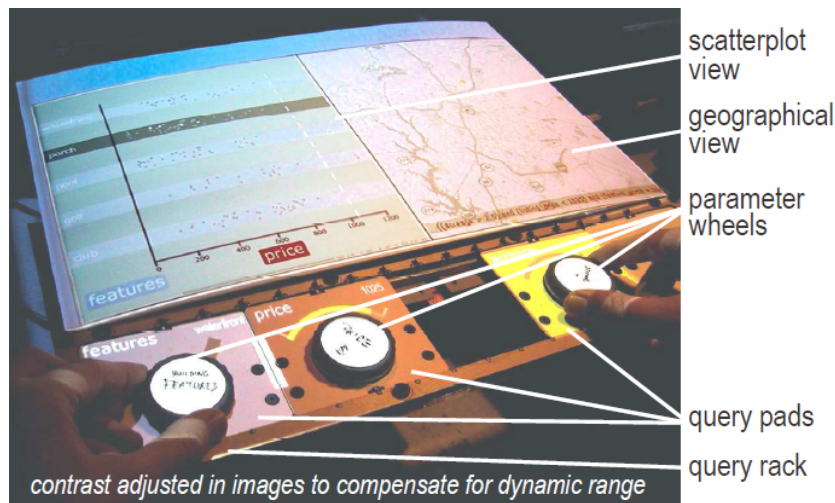


Figure 2.2: Tangible Query Interfaces - First prototype

The second prototype (Figure 2.3) is making use of parameter bars. Similarly to the parameter wheels of the first prototype, each token represents a different parameter. This time, instead of simple labels, active displays are used to label each token. With the small active displays and sliders, the upper and lower bounds of a parameter can be adjusted manually to fulfill the user's desire. The bars are also placed onto a query rack resulting in real time display of the results in the scatterplot and map views. The bars can be reconfigured while they are not on the rack, allowing for multi-user collaboration. Unlike the parameter wheels, the bars can take advantage not only of the AND but also of the OR Boolean operator. This is achieved by relating the spatial distance between the parameter bars on the rack to the desired Boolean operators. When the parameter bars are adjacent the AND operator is used. If they are spatially separated, the OR operator is this time the case.



Figure 2.3: Tangible Query Interfaces - Second prototype

Three main TUI high level approaches are considered and described. These are the interactive surfaces, constructive assemblies and the tokens + constraints, the latter being the case related to the paper. The tokens are interpreted as physical objects representing digital information, objects which are discrete and can be re-configured in terms of space. Constraints are regions where the tokens should be placed, with discrete boundaries. The constraints apply different digital properties and operations to the tokens that are into the constraints region. It is considered that the tokens + constraint approach is an intermediate approach between the other two described; interactive surfaces and constructive assemblies. The first approach is related to continuous forms of manipulation, while the second one is discrete. The tokens + constraint approach offers, as claimed, to balance these two forms by using both. The two forms occur in two phases of interaction called associate and manipulate. In the associate phase the discrete form of interaction takes place where the user places the token in the constraint. During the manipulation phase tokens are continuously manipulated within the constraints and their confines along with the properties of other tokens.

Tokens + constraint map physical to digital interactions. In these prototypes four kinds of operations are associated with the constraints. Physical presence, placement, rotation and adjacency are mapped to the assertion of the query parameter, the selection of the view, the selection of the parameter value and the Boolean operators, respectively. The queries are formed with the usual SQL syn-

tax of "select where" with additional tokens added to the "where" part. A table of grammar for mapping token + constraint compositions to digital interpretations is also given in the paper.

A preliminary user study was conducted comparing the first prototype to the GUI version of the query interface. It is claimed by the researchers that tangible query interfaces can strongly support interaction with data for exploration purposes. The experiment conducted explored three hypotheses. The first one claims that tangible interfaces using tokens + constraints are able to express simple queries. The other one is based on the case that TUIs can use parallel two-handed interactions. The last one, which actually sums up the previous two, is that TUI is faster than GUI for some querying tasks. The result of the research is quite unclear because, although the TUIs achieved a 30% longer completion time, a significant time for setting up the necessary tokens to the query rack was needed, a setup time corresponding to the associate phase. Also, surprisingly to the researchers, some tasks were more difficult to complete in TUI than GUI. The preference among the users was slightly in favor to TUIs with a success over younger users. The first two hypotheses were confirmed, but not the last one. The TUI is believed to be focusing around the objects of interest, the parameters of the queries, with a more direct manipulation than GUI. Expectations are formulated for increase in performance by using the parameter wheels with parallel two handed interaction. The adjacency based Boolean operators can be valuable by also preferring only the AND operator.

For future development, functions to save and compare queries are considered important. Interactive joins can be done by relating views to physical objects and stacking them to express visual joins. Parameter tokens can also be encoded for interactions involving sensitive information.

The preference of the AND operator is arbitrarily given without being proved in the preliminary user study, as only the parameter wheels supporting only the AND operator were used. In order to create a universal tangible query language more operators should be taken into account. The logical SQL operator of NOT was not covered by any of the prototypes. Taking into account the large amount of data nowadays being accessible to everyone and the relative familiarity of casual users with these concepts, database tangible querying should be covering more parts of traditional SQL querying.

2.2.2 TeddIR

TeddIR [30] (Figure 2.4) is a TUI which allows young primary school children to retrieve information from a library database. The starting point, as described, is the difficulty of retrieving digital information for children. Difficulties of different types, for children to use keywords, lead to use an alternative way by using a TUI.



Figure 2.4: TeddIR - The complete setup

The approach proposes that keywords used to search the library database can be replaced by tangible objects. A prototype was built based on this assumption, where tangible objects are separated into two categories, figurines and books. Figurines are related to concepts and have a visual relation to the actual meaning of the word. For example figurines representing dogs relate to dogs as a concept, similarly a heart relates to the concept of love. Books can be directly used as tangible objects for finding related ones. These objects can be placed in two boxes; the green box with the happy smiley represents concepts and books favored by the user, while the red box with the sad smiley represent the unwanted ones. The tangible objects are equipped with RFIDs of the same type as used in the books of Dutch (where the research took place) libraries. The boxes are equipped with RFID readers. Results are displayed on a passive display, returning the top twelve results. The results hold pictures of the cover along with information and the description of the books. Relevance between the tangible objects and the results is given combining similarity search and Euclidean distance.

An evaluation took place among boys and girls between 6 and 8 years old. Simple queries had to be created as tasks, asked in the form of simple questions. Participants enjoyed the use of the prototype seeing it as a game and continuing to use it after the completion of the tasks. Abstract concepts were understood by the participants and were mixed seamlessly with concrete ones. The meaning of the red box concept to represent the NOT Boolean operator was not obvious to most participants during the evaluation process. The test only included the figurines and not the search with books, the other category of tangible objects available.

The difference from similar previous work done before TeddIR, is that the main purpose was to help children with information retrieval. For future work it is considered as a useful addition to replace the passive display with a touchscreen as children tried to operate the screen as such, showing that they expected such kind of interaction. TeddIR has limitations, as it relies on databases holding rich information about the available books. Also, the number of figurines is an important question, as too few provide a small number of available keywords to relate to and too many shifts the problem from searching through too many books to finding one of too many figurines. Libraries can be equipped with two boxes to drop off books to return with immediate feedback of suggestions, a concept that can be also extended for adult users. The final results of the research indicated that TUIs can be a promising alternative to keyboard as input of database querying for children.

2.2.3 Interactive Venn diagram

EnhancedDesk [36] (Figure 2.5) is an augmented desk interface system. The paper includes two prototypes developed for the desk, the Interactive Textbook and the Interactive Venn Diagram. The Interactive Textbook is developed for Computer Supported Learning Environments. Usually digital content is used to accompany textbooks to help people understand the material to be taught. To achieve simultaneous interactive support of such multimedia content, there are matrix codes attached to the pages of books with digital content available. The matrix codes are unique for each application program. When a page containing a matrix code appears, it is recognised through a camera and the appropriate digital content is loaded from a database and projected to the desk's surface. The content can be directly manipulated with touch interactions directly to the table's surface.

Keyword searching for digital information retrieval is the case to be enhanced by the Interactive Venn Diagram. Users retrieve digital information by using one or more keywords combined with the AND, OR and NOT Boolean operators. In the past, only experts relating to computer systems used this kind of operations, but at

the time of writing the paper and also nowadays more people are getting used to these concepts because of the World Wide Web (WWW) search engines, as stated in the paper.



Figure 2.5: Interactive Venn Diagram in use

The issues that triggered the development of the second prototype are related to keyword searching. The first issue is the selection of the appropriate words to do an effective search without being too generic or having trouble because of the lack of knowledge of the target domain. The other issue is the possible problems using Boolean operators for the average user who is unfamiliar with forming more complex queries with them. The last issue is about observing different search conditions at once. Subsets of searched queries results should be able to be observed and given discretely without having to form a new slightly different search query.

In order to overcome these issues, an interactive system is proposed. The digital information is visualised using Venn diagrams to represent mathematical sets, which is a popular approach. Keyword search results are represented as sets and their combinations at the same time. CDs, books and other objects can be combined between them to retrieve information based on similar results. The concept of the prototype relies to TUI extended with Augmented Reality.

A matrix code is attached to book covers, where each one of them corresponds to multiple keywords associated with the book's content it is attached to. When

a book is put on the EnhancedDesk, the matrix code is recognised and a circle is drawn around it. The keywords associated are used to search a book database and produce results. The digital information retrieved is projected into the circle as icons. This applies for any books put on the desk. Now, if two or three circles intersect one another, data related to the Boolean AND of the two books keywords will be displayed in the intersection. When the user points to one region, the results are brought to front and a digital book can be selected to see its related information. The book can then be projected with a circle with relevant results, and used just like a real book, with the system supporting up to three real books and one digital. The possibility to type keywords and manipulate them like a virtual book would have been manipulated, is something left for future work.

2.2.4 Navigational Blocks

Navigational Blocks [8], [7] (Figure 2.6) is a tangible user interface which can be applied to systems similar to information kiosks. The initial prototype is used for the information retrieval of historical facts for a certain tourist spot. The Navigational Blocks offers exploration of information by navigating through them using physical objects. These objects are simple cubes which can be manipulated in numerous ways using tactile manipulation and by also being able to provide haptic feedback.

The prototype is based on an existing application using a computer graphics box with image mapped faces used as hyperlinks to digital information. Although innovative for its time, the system was found slow and difficult to use by users, mainly due to the difficulty of interaction through mouse control. The setup consists of four Blocks representing one category of information each, named and answering the questions of Who, What, Where, When. On each one of the cube's sides an answer is given, being a tuple of the database. The Blocks are placed on an active area one or even more at a time. The side facing upwards calls the related search written on it, while rotating the cube offers simultaneous navigation to different search terms and as a consequence to different results. Placing two or more Blocks on the active area returns information satisfying a query including all the parameters, which implies a Boolean AND operator between them. Sliding Blocks on the active area allows navigating through the results, which are actually a virtual gallery. If the Blocks' terms to be combined return results, electromagnetic forces are applied and the Blocks attract each other. If there are no results, nothing happens, but if the terms cannot be combined for different reasons (e.g. a What term does not match the When term) the Blocks repel each other. User experience showed that the interface provides affordance and can be easily understood without any instructions.

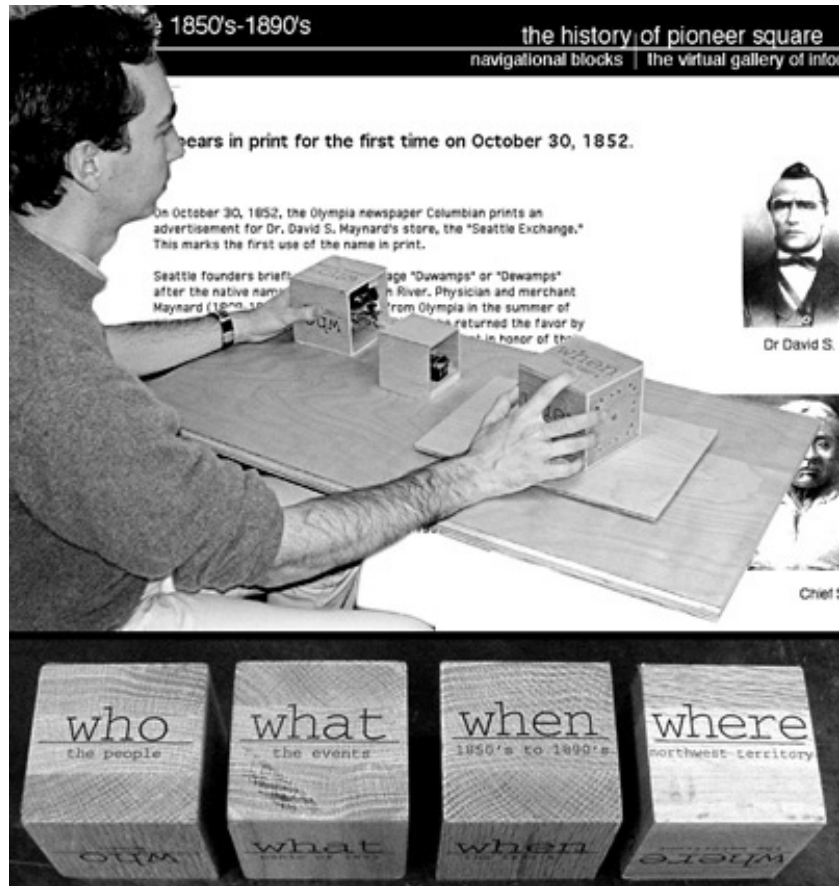


Figure 2.6: Navigational Blocks in use and their facets

The question arises that if it is important for users of Navigational Blocks to use OR and NOT operators, as the system supports only the AND operator, and how to do that in case it is needed. It is suggested that AND should be expressed by attaching the Blocks side by side from left to right. OR can be expressed similarly by placing the Blocks vertically side by side. The 'NOT' operator can be on top of a Block in a stack formation. The argument against using other operators is that adding them depending on the relative spatial positioning of the Blocks can be a burden for novice users. The other improvement that can be made is eliminating the desktop computer used to interact and substitute it by adding displays on each side of the Block. Stackables [35] can be considered to work on these ideas. The prototype was used for a tourist information kiosk, but suggestions are given for use in other environments. The Navigational Blocks can be used as tools to help in teaching language, logic or mathematics. The other scenario is using the Blocks

for home automation by changing the physical environment by rotating them.

2.2.5 Stackables

Stackables [35] (Figures 2.7, 2.8) are tangible objects used for faceted information retrieval in a variety of contexts. Searching for digital information among large amounts of data is done either individually by a user or in collaboration with other users. The solution of Stackables aims to bridge this gap in real life activities. The main motivation for this idea comes from scenarios where information retrieval uses various facets from different types of users with different needs, allowing for individual search and merging these under discussion. The search interface also needs to be fluid and easy to use, share and transfer while working with high or loose coupling.

With these in mind, Stackables were developed as physical objects to realise faceted information seeking by forming stacks without the need of a tabletop interface. Each tangible represents values which can be changed using the same one, resulting to constraining the number of Stackables and using large amount of values.

The idea is that in a group of decision makers, assistance of a digital system is given to lead to a decision. Those people are the direct users of the system by using a Stackables' tangible interface individually or in a smaller group of them. This is done in order to form their individual preferences and decisions and finally combine all these together, to see a collective result. Except for the result sets, the facet values should be available to be viewed. The system should allow users to be able to share, transfer and manipulate facet values fluidly and allow working together closely.

The scenario which was the motivation behind the idea, and helped to define the requirements is the following. In a school environment text books must be found by representatives of teachers, students and parents. In order to do that, a dataset of 1580 books is used, while each participant conducts research individually. After that, the preferences are gathered and a common solution must be found. According to this, the goals were set. A realistic dataset had to be used, with faceted browsing capabilities and simple querying. To be easily used, input and output must be separated, supporting both individual and collaborative work.

The system is comprised of facet tokens and ground plates. Tokens are generic objects containing all facets and their values. These can be configured on-the-fly by using two wheels to achieve the selection. Ground plates are used to build

queries by stacking tokens on top of each other. The operator used in these simple queries is the Boolean AND. On top of these objects, a computer is required in order to analyse and issue queries. The results of these queries are displayed in a separate output.

The shape of the tokens is an abstract box and contains a monitor to actively select the desired facet. Two turnable wheels allow the user to navigate and select these facets. Their size is big enough to fit in the hand of the end user and the monitor is the same size, for readability reasons, as a mobile phone's monitor; the monitor is actually a small mobile phone. The size and shape gives the ability to stack tokens.



Figure 2.7: Stackables - The tokens

The tokens can be combined to form a query by being stacked vertically. This condition binds the faceted tokens with a logical AND. There is no option for an OR operator, although it was asked by some participants in the final experiment, during the free exploration session. The authors state that there was an attempt to add an OR during a early prototype, but it was abandoned. The negation can be achieved by a combination of saved facet values, single and range-based, and by simply placing the token upside-down onto the query stack. The results of the

queries are shown on a display. The output consists of two main parts; one shows the representation of the stacks formed and the other the representation of the results. In the second visualisation, the first category of results shows the exact matches of the query, the second partial matches, and the third all unmatched items of the query.

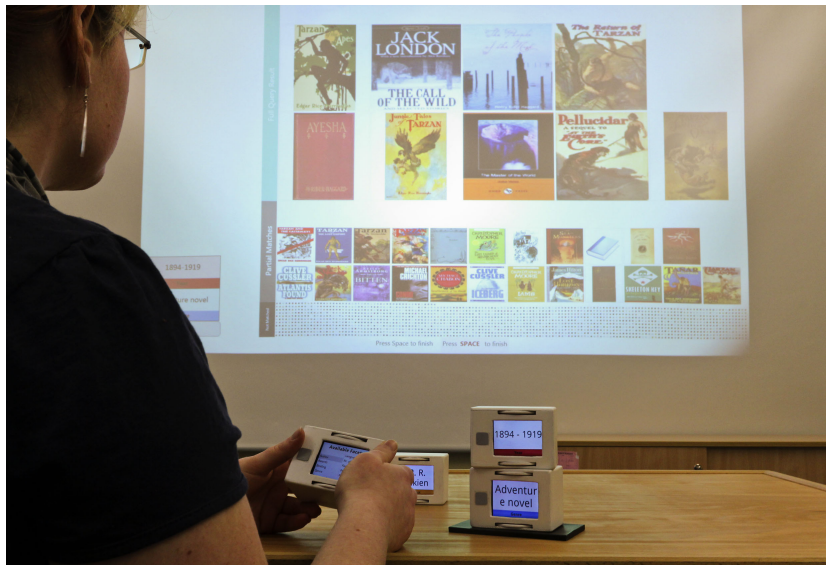


Figure 2.8: Stackables in use

The experiments and their results showed that there is potential and left a positive feeling to the authors. For future work further experiments will take place for more complex queries in a more collaborative environment. The tokens will be updated to achieve a larger set and maybe hold NFC tag support.

2.2.6 The Search Wall

The Search Wall [12](Figure 2.9) is a TUI proposed to assist children's browsing activity in a public library environment. The research that took place combines information interfaces for children and TUIs. A study took place in two German public libraries, mostly during weekends. What was observed was that children have a good knowledge of how to find known material in a library, but accessing material in unexplored areas is a troublesome procedure. This is considered as a direct result of not paying the necessary attention to textual labels and the knowledge of the structure and procedures that a library follows.

The study resulted in forming some requirements for the new system. Children's behaviour in a library is studied and the shelves' browsing is augmented in that direction, by also supporting social activity. Information should be provided interactively and graphically. The book search must allow users to apply the same criteria as used by children when naturally searching. Access to information should be given directly with a few steps and combine abstract and concrete forms, applicable to different age groups. Search is built upon children's knowledge of location of preferred material and provide ways to find the physical copies after the query, and present hints and information stimulating multiple senses. A need to represent Boolean searches also rose.

The "Search Wall" is a large vertical interactive surface designed with regard to serve child users, also according to the observed behaviour in public library environments. The vertical position of the interface and its large size (160 x 100 cm) serves to be visible from a distance while saving space. A small table is attached to the front of it, giving the capability to the users to place items used in the search context described later. On the vertical display, in the center, the results of the queries are displayed to the users, as book covers, and there is functionality on the side for up to two users. Nine of them are displayed simultaneously as 3 rows of 3 items. The rows can be browsed by scrolling to the sides using rotary knobs, independently of each other to allow more than one users to explore the results.

A fixed window with the information about the book (e.g. author, summary, location in the library, etc) exists, too. the collection can be browsed by categories, shelf and cover color, and suggestions are given depending on their age. On the side, cubes can be inserted inside "constraints" (appears as holes on the wall) to determine the categories to narrow the search for. The categorisation system is based on ICDL ². It is organised based on questions as who, what, where, when, and how.

²International Childrens Digital Library Foundation (IDCL), Manchester, MA, <http://www.icdlbooks.org/>

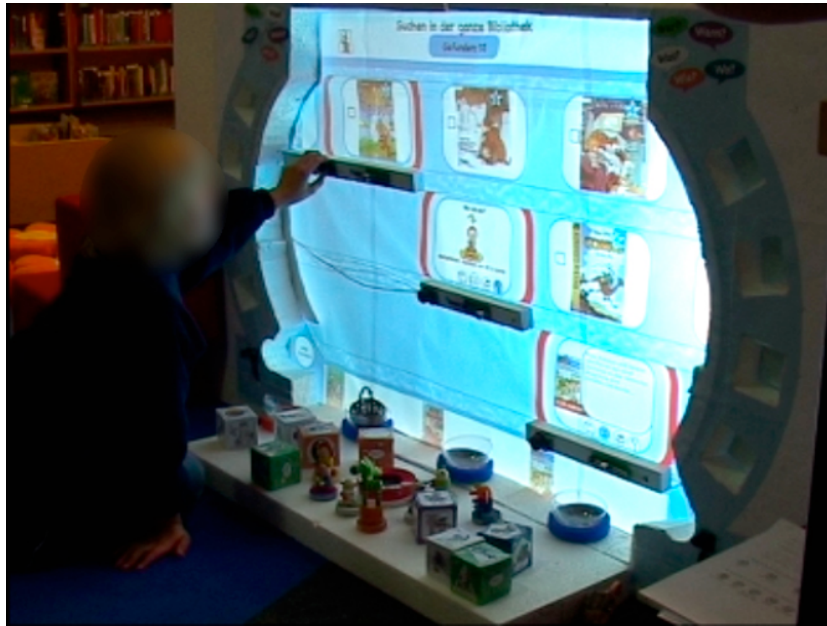


Figure 2.9: The Search Wall in use by a youngster

The search keywords are assigned to "Keyword Cubes", where each side represents a value. By placing multiple cubes on the wall, Boolean queries are composed. When cubes are placed on the same side conjunctive queries are formed, while when they are placed on the opposite sides, the queries are disjunctive. A basket is used to add the desired books, by placing them on a placeholder. After the removal, a signature is printed, indicating the book and its location. On each selection, the item's popularity increases, and placing the basket in the "suggestions" spot on the table will return the most popular items for the user's age. Shelves are also marked with an animal character. The same character, as a puppet, can be used to search only within a given shelf.

The results of an experiment that took place showed that there was positive acceptance, but the young users struggled to understand some concepts. The basket functionality and the puppets were difficult to understand. The experiment used the Wizard of Oz approach and collaborative search could not be tested.

2.3 Comparison of Related Work

After our research on related work for TQIs we gathered some key concepts that were observed on already existing systems. We also added concepts that would be in benefit of a more sound and complete TQI system that can also be mapped to more complex SQL queries. From this table (2.1) we could extract some conclusions on what is provided currently and also understand what are the possible gaps that those systems leave. Those gaps can be related to usability and user experience requirements, on top of the SQL mapping that we want to achieve. At the end of the table, we also provide, indicatively, also our systems. This way one is able to compare with the related work within a single table.

We first set the logical operators that should be essential for querying. These operators are used to create both simple and complicated queries and also concatenate result sets of more than one queries. From there on we started adding extra functionality that is either provided or mentioned by the related research or from the original ArtVis installation. We also added unique elements that we incorporated in this thesis' system.

At first glance, we observe that there is not a default expression of Boolean operators. Not all systems support every operator or use the same way to do it. One of the pioneers [52] of TQI supports the AND and OR operators. A little bit earlier, in the study of Camarata [8], [7] only the AND operator was supported and any other possibilities were left for future work. In [36] we have all of the main three operators (AND,OR,NOT) available to mix search keywords, more like a search engine context. Another research [12] mainly cover the two Boolean operators (AND,OR) or use an algorithm to determine the result set [30]. The UNION operator is only available in [35] as it is also the only research that actually can combine more than one query.

Saving queries is not supported by any of the available studies, but [52] mentions as saving the results (not the queries) as something for further development. Saving results can only be a burden when the database changes, and could become obsolete after some time. All of the systems support an active area where the querying takes place with the exception of [35], which uses active objects instead. [52] uses both and [8] suggests active objects as something to investigate in the future. The operators are determined by their position only for every study, and adding, removing, modifying objects or integrating any physical object (operators or operands) cannot be done. The metaphors that the systems use for the objects do not have a natural relation between their form and what they represent, with the exception of [30] and [36], where the latter uses the actual object. Any

physical object can be integrated in ArtVis [15], the Interactive Venn Diagrams [36], TeddIR [30], and Search Wall [12]. In the last two this can be done by replacing the RFIDs and in [36] attach a matrix code. ArtVis offers explicitly this feature by editing a simple plain text file to assign the RFIDs used for the objects.

We added some targets that also lead to the requirements analysis. We would like to support AND, OR, and NOT operators and combine them within the queries. Multiple queries should be supported and provide a UNION of the results. We also want to provide an intuitive natural TUI, appropriate for our user audience. The TQI and the objects used should also be appropriate for the museum context we are going to use it for. We would also like to avoid being too rigid in the query formation and provide more freedom towards building, re-building, and combining queries. To sum up, our system should stay within our scope of mapping the TQI with SQL querying, and still be consistent with the ArtVis concept and environment.

	Tangible Query Interfaces [52]	TeddIR [30]	Interactive Venn Diagrams [36]	Navigational Blocks [8], [7]	Stackables [35]	Search Wall [12]	ArtVis[15]
AND	✓	n/a ³	✓	✓	✓	✓	-
OR	✓	n/a	✓	-	✓	✓	-
NOT	-	✓	✓	-	-	-	-
UNION	-	-	-	-	✓	-	-
Save Query	- ⁴	-	-	-	-	-	-
Multiple Queries	-	n/a	-	-	✓	-	-
(Free) Query Grouping	-	-	-	-	-	-	-
Active Area	✓	✓	✓	✓	-	✓	✓
Active Objects	✓	-	-	- ⁵	✓	-	-
Natural Metaphors	-	✓	✓ ⁶	-	-	-	✓
Operator as Position	✓	✓	✓	✓	✓	✓	-
Operator as Object	-	-	-	-	-	-	-
Add/Remove Objects	-	-	-	-	-	-	✓
Modify Objects	-	-	-	-	-	-	✓
Integrate Any Physical Object	-	✓	✓	-	-	✓	✓

Table 2.1: Comparison of related work

³An algorithm computes relevant results

⁴future work

⁵future work

⁶actual object

Chapter 3

ArtVis

The research conducted gave us a path towards a new approach for Tangible Querying. Until the final solution was designed, there was a number of ideas and approaches that were eventually replaced by the system that this thesis is based on. In order to apply our research decisions, we enhanced ArtVis [15] (Figure 3.1), a previous reasearch on advanced visualisation techniques.

The project's target is to restate the problem of information overload in the context of InfoVis. On top of the main objective, a basic TUI was added to the system. ArtVis uses publicly available digital information about artworks from the 11th to mid-19th century AD. This information is retrieved from the Web Gallery of Art (WGA)¹ open database. The application is written in Java and uses the Prefuse visualisation toolkit and the TUI is based on the Phidgets interface. The name is chosen as such, because it brings InfoVis together with artworks. The application allows exploration and browsing of digital information, and gives the ability to the user to have a broader view on this data.

¹WGA: <http://www.wga.hu/>



Figure 3.1: The complete ArtVis for museums

3.1 ArtVis System

3.1.1 Database

The catalog provided by WGA was optimised and normalised to become the ArtVis database. There was a fundamental restructuring to better reflect the needs of the application. The relational database used is MySQL-based. The number of artworks contained approaches 27600 in total from all over Europe. The artworks cover the periods between 1000-1850 AD and include the Romanesque, Gothic, Renaissance, Baroque, Neoclassicism and Romanticism periods of artistic movements.

3.1.2 Application

The application itself, as already stated, uses the Java programming language and the Prefuse toolkit for the InfoVis part. The requirements of the analysis state that the user must be allowed to explore, analyse and browse through the available digital data. Based on these, three different visualisations are developed as three panels included in the overall main application.

The exploration is achieved through an Exploration panel on the top left side of the output with a map of the world as its main component. The information given on the map are circles where the size changes according to the number of results. These circles can represent either the artists' living area or the artworks' museum location. Both choices can be selected as facets from the top of the panel. Zooming in the map separates the circles to more and more accurate and discernible locations. Also, in the bottom of the panel there are two slider buttons used to restrict the time periods.

The analysis of the data is shown in the Analyse panel on the right side. There are three dimensions available to filter the data and have specific visualisations to extract information. The dimensions filter the artist schools by country, the artwork inspiration (e.g. religious, mythological, etc) and form (e.g. painting, ceramic, etc). The visualisation is a graph showing the period on the X-axis and the number of artworks on the Y-axis.

Browsing the database is easily done through a Browse panel on the bottom and a small details panel on the top right. The "browser" shows a number of the artworks' images according to the criteria set from combo boxes above it. The fisheye distortion technique is used focusing in the middle. Hovering the mouse above the picture zooms to reveal visual details of the artwork. The detail panel gives information about the selected artwork. The artwork's picture from there can be selected again and open a web browser showing its page in the Web Gallery of Art website, given that there is an available internet connection.

The visualisations use the same dataset and filters, thus they can be combined. Queries return combined dependent visual results. Selecting a dimension in the Analyse panel also updates the other panels, while a selection in the Explore panel updates the Browse panel, and vice versa.

3.2 ArtVis TUI

The ArtVis application is enhanced through a TUI on top of the already existing GUI. The idea behind the TUI is to use physical objects related to art to filter data and use a control pad, "disguised" as a painter's palette, to navigate the results. A rotation sensor is used to zoom in and out in the Explore panel's map of the world. To navigate through it, a mini joystick is used. A slider is used in the Explore panel to put time constraints, in 100 years intervals, and also in the Browse panel to navigate through the fisheye distorted artwork collection. The TUI is completed with the more relevant part to our study, the PhidgetRFID.

The latter is an RFID reader for RFID tags, which are used to mark physical objects, and turn them into parts of the system. These objects are used to filter the dataset and automatically update the corresponding visualisations. There are objects calling the Analyze panels filters (Artwork form and inspiration, Artist School), switching the map viewpoint (author living area and artwork museum location) and resetting the filters.

3.3 Tangible Querying

As already described, there is a TUI attached to ArtVis. Basic tangible querying is possible through the RFID tagged artifacts and Prefuse's filters. The artwork form, inspiration, and the artist school metaphors mainly serve this functionality. The panels are updated to only show the corresponding artwork that matches the query's criterion. This is achieved by having the user place the tagged object of her desire near the RFID reader, and the query is triggered immediately. The query can be done only using one object at a time, without being able to combine multiple dimensions.

Chapter 4

Requirements Analysis

4.1 Use Cases

We gathered some use cases indicative of how ArtVis can be used. These use cases helped us determine the requirements of the new TUI, and also led us to define the importance and priority of them.

4.1.1 ArtVis Compliant Use Case

A use case, for instance, can be when a user wants to see all sculptures. She will have to pick the sculpture token (the physical object metaphor) and pass it over the RFID reader. The system will have the same behaviour as when the "Artwork Form" dimension of the Analyse panel is chosen. The panel will show the desired result, the Explore panel will show sculpture-related information across the world map, and the Browse panel will show only sculptures. Similar use cases exist for the other available tokens.

4.1.2 Intersection

It is clearly understood that the querying achieved by this TUI is quite restricted as to how it can be used. The query has only one parameter at a time, without being able to combine more. The use case described above can be enhanced to contain more parameters. Let us state that the user is a Fine Arts student with a project on Italian sculpture and the impact of ancient Greek mythology on that. According to the project's requirements, she is interested to find not just sculptures, but also the ones where the artist is from the Italian art school. In addition, the artworks should also be of mythological inspiration. This cannot be done through the original ArtVis TUI (and also GUI).

In this use case, the Boolean operator that would be used in logic is the AND operator. We need to find a mechanism to map the Boolean AND also used in relational databases to the TQI of ArtVis. This will give the ability to a user to, consciously or unconsciously, trigger a more complicated query to the artworks' database and find what she was searching for.

4.1.3 Union

In another case, an art collector is interested to locate religious paintings from Scandinavian artists. As in the use case above, the user will have to use a Boolean AND operator to combine artwork form (painting), inspiration (religious), and artist school (Scandinavia). As Scandinavia is composed of three different schools (Danish, Swedish, Norwegian) the user needs to also look for an additional three operands. These operands cannot be combined with an AND Boolean operator, because an artwork in ArtVis' database cannot belong to more than one school. Thus, we need to define a way of supporting operands in a query of the same type. This can be achieved by using the Boolean OR.

4.1.4 Negation

A visitor in a museum wants to use the ArtVis installation. The user wants to see the visualisations for every artworks that were created by an artist from Europe. Given that the current database also contains different European, and American (U.S.A.) artist school artworks, the user needs to filter the content accordingly. This can be achieved in two ways. Either show content from each and every European school (e.g. show artworks of Italian or Greek or Flemish schools), or show all content except for the American school artworks (e.g. show all schools except for the American). This use case can be translated as a Boolean NOT operator in a relational database.

4.1.5 Multiple Queries

A group of 3rd graders from a nearby school visits a museum hosting, among other things, the ArtVis installation. Tasks are given to the youngsters in form of a questionnaire. They are split into groups, told to complete the tasks as teams. One group's task is to find Italian sculptures, Greek paintings, Flemish ceramics, and the results should not contain religious related artworks. In a collaborative environment, the children will either gather around the querying area and add physical objects all together to form a query, or be given the chance to do that separately by themselves. For instance, one user will form the Greek paintings query, another will look for Italian sculptures, a third one (or even subgroup of the group) will

create the Flemish ceramics query, and another user will exclude the religious artworks. This can be achieved on a relational database level by connecting all the separate queries with the UNION operator, and adding a NOT part to each one of those queries.

The last use case shows a very important aspect for our research. The TQI shall support collaborative environments and, ideally, let an unspecified number of separate queries to be formed, by groups of users working separately on a common goal. Collaborative use of the system can leverage the usability of ArtVis, due to its use in a museum context among other things.

4.1.6 Saving a Query

A Fine Arts PhD student creates a complex search query, based on her research's needs. After browsing and analysing the data provided as a result of her search, the student decides to call it a day and decides to continue viewing the data another time. The problem here is that the query is too complex to be remembered by heart, with the only solution being to write down the physical objects and their combinations which produced the query. Anyway, the user must replicate the given query each and every time she wants to use it for any reason. The system shall provide a way to capture the query and replicate it at any given time. In any case, it would be also highly desirable to be able to save the query in a portable low cost object, so that a user can keep it and use it at will.

4.2 Requirements

The ArtVis system and its TQI are already described. The use cases in combination with the related work from previous chapters, reveal requirements that will help built the foundation of our research. A new TQI is constructed according to those requirements, that will enhance not only the overall user experience (UX) but, mainly, create a TQI of high functionality and capabilities for users.

id	type	requirement	priority
FR1	Functional	The system must provide intersection in queries	High
FR2	Functional	The system must provide union in queries	High
FR3	Functional	The system must provide negation in queries	High
FR4	Functional	The system must be able to connect many queries	High
FR5	Functional	The system must provide a collaborative environment	High
FR6	Functional	The user must be able to save queries in order to reuse them	High
FR7	Functional	The system results must be rapidly provided	High
FR8	Functional	The user must be able to retrieve saved queries and get the results	High
FR9	Functional	The system must allow physical objects to be easily reassigned	Low
FR10	Functional	The system must allow physical objects to be easily added	Low
FR11	Functional	The system must allow any physical object to be used in the TQI	Low
FR12	Functional	The users must be able to work individually in the collaborative environment	Medium
FR13	Functional	The system must allow simultaneous collaborative query building	Medium
FR14	Functional	The system must not restrict the number of individual collaborative "islands"	Medium
NFR1	Non-Functional	The system must not allow the users to alter the database in any way	High
NFR2	Non-Functional	The TQI must be appropriate to use in an exhibition/museum context	High
NFR3	Non-Functional	The system must be queried instantly	High
NFR4	Non-Functional	The user must be able to keep the saved query	High
NFR5	Non-Functional	The saved query must be portable	Low
NFR6	Non-Functional	The saved query must be low cost	Low
NFR7	Non-Functional	The system must provide visual feedback to the user to distinct the unique queries	Low
NFR8	Non-Functional	The TQI should be independent from the system it supports (e.g. ArtVis)	Medium
NFR9	Non-Functional	The TQI must not be specific to one SQL RDBMS	Medium
NFR10	Non-Functional	The user must be able to easily save the queries	Medium

Table 4.1: Requirements for the new TQI

Chapter 5

Design

The research conducted led to our final solution. The background leading to our design described in this chapter includes the conclusions out of the research conducted for the related work, the current ArtVis TQI drawbacks, and the requirements extracted from this research. The scenarios that were turned down were a determinant towards the final solution, too, as different requirements were either not fully fulfilled or missing.

5.1 Scenario

During our research we started, initially, designing the concept of a TQI which is described below. We finally decided to re-design the system, as it did not fulfill our requirements. The system (Figures 5.1, 5.2) did not promote collaboration and the active objects did not provide a natural feeling to the user. Also, the objects need to be set once and do not provide flexibility to convert any physical object to a TQI operator or operand.

Jigsaw puzzle pieces can be used as part of a more generic tangible querying approach. The idea of how they can be used is that queries can be formed as constructive assemblies with four or more jigsaw puzzle pieces. This idea has as a starting point the simplicity of constructing jigsaw puzzles, as well with people, even from very young ages, being familiar with the concept. By the time this thesis was written, although jigsaw puzzle pieces were used in various TUI approaches, none of them was used for information retrieval [5],[3],[2],[58],[47].

Each piece represents a predicate of a query, where in case of different expressions for the same predicate other pieces of the same shape are used. Databases can be large so different pieces can become a burden, as there can easily be hun-

dreds conveying the informational retrieval problem to the physical space. For this reason, active tangible objects can be used to enhance a puzzle piece to represent more than one expressions. Instead of keeping hundreds of the same shape as passive pieces for each possible expression, the active component can be modified according to the user's desire, by keeping only one copy of the same shape piece.

Pieces of different shapes representing different types are put in place together on a table forming a query with the Boolean AND operator. That means that predicates pieces connected to one another are using an AND relation between them. In order to achieve queries using also the OR operator, pieces put on the table but not connected to the main jigsaw puzzle, in other words pieces being spatially distant between them, can be considered as alternatives, thus they will use an OR relation. Another approach for the OR relation could be to put pieces on top of each other as a stack to represent the same relation. Pieces that represent predicates that should not be present in the results of the query, expressed by the Boolean NOT operator, can be thrown away in a part of the table separated from the rest with part of a physical trash can, embedded in it. People are expected to foresee the intended metaphor of throwing away objects that are no more needed in everyday life.

This approach can use a multi-touch table with a fiducial markers tracking framework to identify the different types of pieces. Each piece will be marked with fiducials in order to be identified and to let the system know its position. The active tangible objects except for the fiducial marker will be equipped with a small programmable electronic device with a small screen and capability of wireless communication in order to be attached to the system; this device can be a smartphone, as in our initial approach. The user will be able to type search keywords on the active component retrieving information as a list and choosing the desired one. On the table and on the active object, feedback should be provided in various ways. This feedback depends on the database used as well as with the predicate, varying from simple text to images, videos, etc.

This scenario did not fulfill most of the requirements of the Table 4.1. Also, in the comparison table (Table 2.1), one can be observe that this solution falls short towards the solution we want to achieve. The system's TQI solution did not feel natural, especially by using the active object. The metaphor provided for the operands is also not intuitive enough.



Figure 5.1: Jigsaw puzzle concept

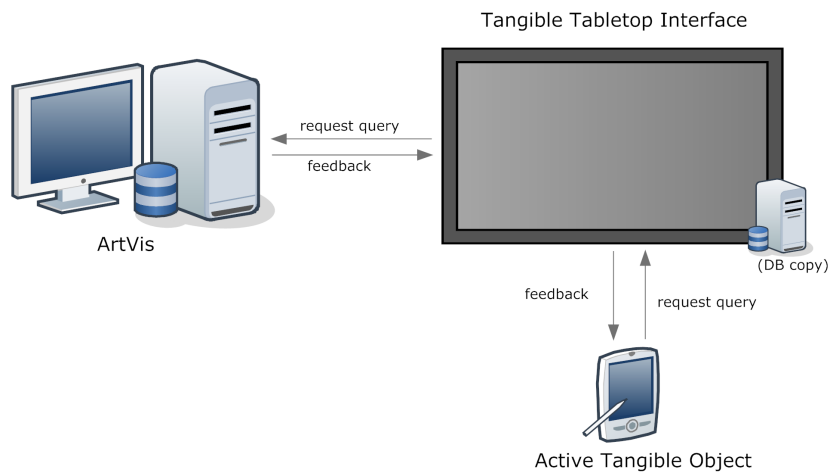


Figure 5.2: The architecture of the scenario

5.2 General Design

The final solution should fulfill the requirements captured in the chapter where the requirement analysis took place. With this in mind we moved on to reconsidering solutions to each one of the areas that each requirement addresses. The system that we need to design should not only be as extensive as possible in covering aspects of relational database querying, but also be consistent to use within a museum context. Ideally the TQI designed must be system agnostic, the system

being the one that the TQI is meant to be an interface to, in our prototype, the ArtVis.

The current console used as a TQI in ArtVis is considered inconsistent for our scope, thus a new interface must be used. We chose to use a tabletop interface on top of the current system, still being able to use the console provided currently. Tabletop interfaces were introduced as background work of this thesis used in many contexts. This kind of interface seems to be ideal in leveraging collaborative usage and gives also space for individual expressions within the same collaborative environment. With this choice of approach, we cover a serious amount of high and medium priority requirements, related in the first place to collaboration.

This tabletop interface runs as a different application on a different system. This will help in separating the TQI from being system specific, and reuse it in other context. Of course, changes should be applied on the original ArtVis system, in order to be able to incorporate the new TQI; this aspect will be discussed later on. The two systems (the main one and the TQI) may share the same machine, but due to the many applications that need to run simultaneously, we advise the separation (different machines) of the main system from the TQI.

The main idea is that different physical objects (the ones that ArtVis already uses are perfect candidates, due to their metaphoric identity) are used to represent the query operands based on the database. These objects can be grouped and form separate queries. One of the groups will also be used as an unwanted items group; the logical negation. Many users may use the system according to the requirements. The querying of the database, and the system as a whole, and its visual results should be provided in real-time, without the need for further actions.

To be able to have such functionality using a tabletop interface, we need a certain technology that will give us such capabilities. RFIDs are not a suitable solution to our problem, as location and multiple tag recognition is not adequate. For our context, fiducial marker recognition can serve our purpose. The physical objects can be marked with tags that are recognised through a camera. This way relative location and distance of the objects could be determined easily, and also a large number of simultaneously visible objects can be distinguished. The ReacTIVision framework was chosen to be used for this purpose. ReacTIVision is a well-known widely used framework for computer vision, with fiducial markers tracking capabilities.

5.3 Solution Description

The solution extracted from our research is described below, along with arguing why these choices were taken.

5.3.1 Operand Physical Objects

A physical object is assigned a dimension's value, that serves as operand of the queries to be created, thus used as "operand physical objects". For example, the sculpture object used in the earlier TQI implementation holds in our TQI the dimension of "Artwork Form" with a value of "Sculpture". Different physical objects from the former TQI were chosen to be used as operands, because of them being successful metaphors of the values that we want them to represent. Objects representing artwork forms are actually pieces of artwork that are of the same form. For instance, a sculpture, a ceramic vase, a glass vase, a metal candlestick, a miniature painting. Artwork inspiration uses cards of paintings that have the same inspiration; a portrait, a landscape, etc. Paint tubes and small paint bottles, are assigned artist schools; a tube with the Flemish flag, an Italian, etc.

5.3.2 Grouping Operands within Queries

In order to separate the different queries to be created and allow each user to work independently on her own query, the following solution was conceived. Special "query objects" are used in order to group objects/operands within a single query. These query objects are placed on the tabletop interface providing feedback of the area where the single query takes place. This area is where the objects that serve as operands should be placed in order to be part of the formed query. Users are provided visual feedback as to where is this area exactly located on the tabletop interface itself as a circle around the query object. This way, one or more users can use one or more query objects each, and query the database within a collaborative environment.

5.3.3 Boolean Operators within a Query

The two Boolean queries, AND and OR, are implied by design. When a user places operand physical objects within the circle marking the query area, AND is used by default. We will describe how both operators are used within a simple example of a use case. In a scenario, a user places an Italian paint tube and a sculpture. What will happen is a search for artworks with form of sculpture AND of the Italian artist school. Now, if one adds a Flemish tube to the above query, what will happen, as an artwork cannot be of both Italian and Flemish schools? The system will detect

this "flaw" that will normally always return empty results, and use the Boolean OR operator. That said, the query will now look like this: artworks with form of sculpture AND of Italian OR Flemish artist schools.

In our opinion, this is a clean design decision for a number of reasons. The user does not need to have knowledge of such logical concepts. Inadvertently, the user already knows and uses them in her everyday life and language. Younger people or people without a related background do not need to be trained to use our interface and its concepts, thus avoiding a crash course in Boolean algebra. Also, the system might be more well-accepted as there will be less null results from querying the database. Only queries that make sense will be sent to the database, giving the ability to the user(s) to always be able to browse their search results. Also, a truly empty result can be detected by the user, extracting more meaningful conclusions out of their research. The user might need to separate the two Boolean operators by herself, but this requires a user with high technological expertise and knowledge of the basic concepts of logic and relational databases. In our case there is also no need to do that, because, as we already said above, there will be numerous null results, if the user does not have knowledge of the database's structure.

5.3.4 Negation

A useful addition is the logical complement of sets used so that the users can also drop unwanted artworks out of the results of their queries. In order to achieve that we use a "negation object" in the same way as the query object. It is a special case of a query object where all operand objects included within its radius hold a NOT logical operator towards the queries. This way users can exclude artwork with parameters described by the operands from the results. By using negation, even more capabilities to filter a big database are provided. The negation applies to every query on the table and there is no separate negation for each one. We think that using separate logical complement for each and every query would create a burden. This way would force the users to include many objects on the table with complicated positioning. Also, in a collaborative environment or when using multiple queries, the result set may include items that the users did not want to see. This would happen because the unwanted operands would be applied to one of the queries, letting another query to use it.

5.3.5 Unifying Queries

As already stated, the system gives the opportunity to be used by one user or in collaboration with other users. In order to achieve this, and still give the op-

portunity for individual expression even in a collaborative environment, we decoupled the query building. The query objects form separate query groups that form individual independent queries. In addition there is one negation group with the operands that are unwanted. To fetch a unified result created by all of these groups, the UNION clause from set theory is used. The UNION clause is also used in SQL and combines queries through unifying their results, omitting duplicate values. The second attribute is also the reason why UNION ALL is not selected to be used by our system. INTERSECT is also not appropriate, as in most occasions the result returned will be void in the considered. This way all the queries' results can be combined under one result set and be accordingly browsed, analysed, visualised, etc.

The negation object is a different, though similar, story. The group, visually and in terms of usability, similar to a query object group is used, as already described, but a UNION is not used this time. Within this group the Boolean OR operator is used for the operands to form a set of logical complements. This set is consecutively bound to each one of the individual queries appearing on the table. So each query, in addition to the operands it already holds, is augmented through the AND Boolean operator with a set of operands under a NOT operator. There is a chance that the query or negation objects might not be intuitive enough for a user, and may need to be explicitly shown, but the visual feedback provided will provide the hint. Also, the feedback on the tabletop interface is restricted, mainly because of our table's capabilities.

5.3.6 Saving & Retrieving Queries

The option to save queries is added to the system. The user is able to save a current query as it is set on the tabletop interface. The unified query produced by the individual queries and the negation group can be saved instantaneously in real-time. When the setup of the queries produces a result that pleases the user, one can save it to any RFID tag (given that it uses the same frequency as the Phidgets RFID reader). This RFID tag can later be used to retrieve the saved query and fetch the results of it. One tag can hold only one query, but one query can be saved to multiple tags. This functionality gives the opportunity to the user to keep a query she finds useful or interesting or used frequently. A query can be saved on many different tags and then be shared to many different users, too.

For usability reasons, we decided to save the queries to passive RFID tags and not in another way (e.g. print fiducial markers to use directly on the tabletop interface). RFID tags can be really a really low cost medium with a unique ID for each tag. They are also produced in different shapes (tags, stickers, cards,

bracelets, keychains, etc). This way the tag can be easily kept by a user or attached to another object. By attaching the tags, physical objects can be converted to TQI objects like the ones in the earlier TQI of ArtVis. For instance, in a use case, one saves a query about artworks of the schools belonging to Scandinavia to an RFID tag in the form of a sticker, and attach it a small map of Scandinavia. In another case, a student may save a query related to her research to a keychain attached to her personal keys and have easy access to it every time.

5.3.7 ArtVis Museum Console

The initial ArtVis Museum Console is kept as it is. The existing RFID reader is used for saving and retrieving queries, as already described. It is still used for filtering and resetting, as initially, to further filter the results of a query. The controllers on the console form a great TUI for navigating the ArtVis' visualisation panels, being intuitive and easy to use for any user group and type [15]. That said, the console stays and is used as it is with the addition of the functionality of saving and retrieving queries on demand. The RFID tag augmenting physical objects used to filter data are now, not only used as tags for filtering data, but also augmented furthermore with fiducial markers for the new TQI.

5.4 Requirements Fulfillment

In chapter 4, the requirement analysis took place (Table 4.1). From our research, the requirement gathering led to the extraction of the requirements list for our system. Fulfilling those is set by priorities (Low, Medium, High) and the system is designed accordingly. The requirements are broken into two types, functional and non-functional requirements. All of the requirements have been fulfilled, and all of them are available grouped in Table 4.1.

5.4.1 Functional Requirements

The functional requirements with a high priority are considered as essential to our system, and any design should fulfill them. The FR1 and FR2 functional requirements are applied within the radius of a query object. The intuitive selection of AND and OR Boolean operators offers the required functionality. They are considered essential as there are already systems covering those, although not always and in most occasions not even at the same time. Negation in FR3 is provided as a separate grouping function and is also essential.

A collaborative environment is something that is also considered essential and is an important aspect of TUIs. Thus, FR5 is fulfilled by giving the opportunity to one or more (FR14) people to use the TQI simultaneously (FR13). This is done in collaboration with each other for a common goal, or just to aggregate each others' search results, working as a team or autonomously. The latter fulfills FR12. There is the option to provide multiple query objects for each user. By providing multiple queries and returning one aggregated result, FR4 is also fulfilled.

FR7 requires to rapidly provide results. This is achieved through real-time querying without the need of a user-side second action, and the also real-time update of the ArtVis visualisations according to the results of the queries. FR6 and FR8 are fulfilled through the RFID related saving of queries and similar way to retrieve them. When an RFID tag is not assigned to any query, the system assigns one when scanned. If this is not the case, the assigned saved query is triggered immediately.

The physical objects used as operands and query objects do not have to be assigned as initially provided. The fiducial markers that tag these objects can be reprinted and assigned to any physical object, as FR11 suggests, and be reassigned to them (FR9). More operands or operators objects can be added by simply assigning unused fiducial markers to new objects (FR10). The new objects and their markers are matched to dimensions and their values.

5.4.2 Non-Functional Requirements

The TQI of our research needs to be appropriate to use in an exhibition or museum context as the NFR2 non-functional requirement requires. This is done by using a tabletop interface and giving the opportunity to many users to interact with it. Also, physical objects are used as metaphors of the operands to use and Boolean operators (AND and OR) are used intuitively, making it approachable to different user groups and types. The queries are real-time not needing further actions to be taken, once the query groups are formed (NFR3). Moving the objects around or adding and removing them refreshes the queries in real-time. The tabletop interface also provides feedback to the user as to what groups are currently available and their radius of grouping, by showing circles around the query and negation objects (NFR7).

NFR1 is fulfilled by keeping the same database user privileges as the ArtVis application does, thus any user of ArtVis has read-only access to the database. Also, the TQI is not specific to one SQL RDBMS (NFR9). This is achieved by actually not creating the queries within the TQI, but sending a message protocol

with the groups of the queries. This aspect will be discussed extensively in the implementation chapter. This way the TQI can be independent and agnostic of the system (NFR8), thus making it usable with other systems than ArtVis.

NFR10 requires an easy way of saving queries. This is achieved by simply scanning an RFID tag by the RFID reader. The saved query can be kept by the user (NFR4); it does not need to be left near the system (e.g. within the museum). RFID tags were chosen for this reason. They can be kept by the users as they are portable (NFR5) and low-cost (NFR6). Portable because they come in different shapes and types, and can be attached almost anywhere. Low-cost, because the monetary cost of a tag, although varying, can be really low and can be also bought and brought by the user herself.

Chapter 6

Implementation

6.1 Architecture

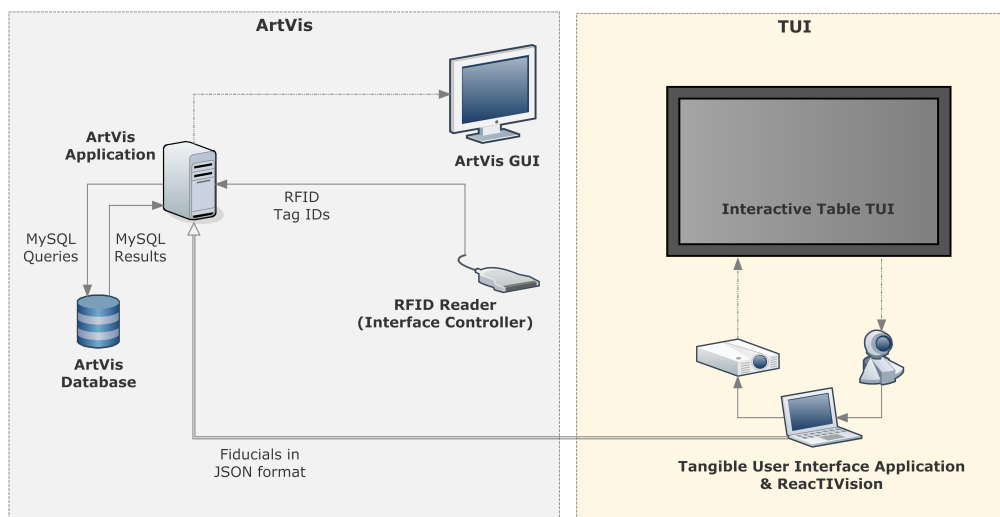


Figure 6.1: The architecture of the new ArtVis TQI

In the illustration above (Figure 6.1), one can see the architecture of our system. The two systems, ArtVis and the tabletop interface are shown separated from one another. The ArtVis part of the system shows how the application communicates with the database, the console, and the GUI. The TUI part of the illustration shows how the table and its application are setup. In-between the two parts of the system one can see that the communication is one way, and JSON messages are used to

communicate. In the next sections the components shown here are described analytically.

6.2 ReactIVision Framework

Our tabletop TQI is using fiducial markers in order to determine the quantity, identity, and relative location of the physical objects used for querying. The ReactIVision framework [32] is used towards this scope.

ReactIVision serves as a toolkit for rapid development of tabletop TUIs with its robust fiducial markers tracking capabilities. On top of this, the framework also offers finger tracking for the rapid development of multi-touch interfaces. Initially, ReactIVision was designed as the underlying sensor component of the Reactable [31]. Nowadays, the framework is provided as a stand-alone computer vision framework, with the ability to be used as an open-source platform for different projects. The current version supplied by the creators and used in our thesis is the version 1.4, which incorporated finger blob tracking for the first time. This time around, a separate fiducial marker was used to simulate finger blobs. In other words, finger tracking is natural without the need of adding extra equipment to the user's hands.

The ReactIVision framework sends Open Sound Control (OSC) messages implementing the TUIO protocol [32] via UDP port 3333 to clients on localhost that are already TUIO enabled. The IP and port can change internally. The framework can alternatively send MIDI messages. The engine is available for Windows, MacOS, and various Linux distributions. This way, the TQI of our thesis can run under different OS, thus making it available to wider audiences. TUIO clients are available for the following programming languages: Java, C++, C#, Processing, Pure Data, Max/MSP, and Quartz Composer.

The internal structure of the framework (Figure 6.2) is described analytically in several papers [32],[33]. Specially designed fiducial markers, also known as amoeba, are tracked through a real-time video stream. This way the framework determines the relative position and orientation of the visible markers, and can also identify them. ReactIVision can also track two older sets of fiducial markers, but we prefer the default ones, as they are optimised.

In order to use the framework, a camera needs to be used to recognise the markers, and the TQI application needs to use the TUIO library. The table that we used ReactIVision for is discussed in detail in another section.

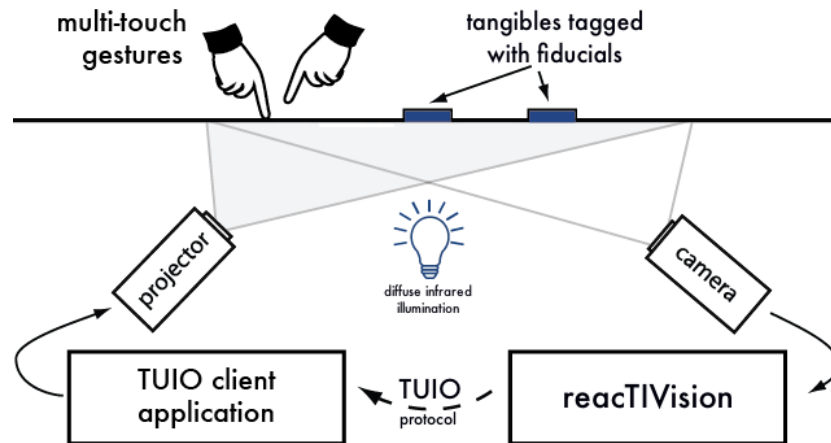


Figure 6.2: The ReactIVision framework architecture

6.3 Tabletop Interface

The main component of our TQI is the table of the tabletop interface. It is assembled according to the needs of our TQI and the capabilities of the ReactIVision framework. The table used for the prototype is rather low-cost compared to the standard ones. This is because we followed a more simplistic approach to what is usually used. The main components are the table skeleton, a plexiglas sheet, a short throw projector, a camera, a mirror, and black fabric.

The main structure of the table is made out of a storage shelf. This shelf is made out of metal and wood. Its purpose is to be used for storing heavy objects of hundreds of kilograms. The shelf is 1,80 m high and of rectangular shape. We split it in half, so that to have two smaller shelves of 90 cm each. We removed all the wooden shelves but the bottom ones and rearranged some of their metal skeleton pieces. These two shelves were screwed together in order to form a big square piece of furniture of 90 cm x 90 cm x 90 cm. The top of it was left void, in order to place the plexiglas "monitor" and a wooden panel around it. The whole square shelf is covered in double thick black fabric, so that no light from the inside can go outside, and the table's internals are left hidden from the public.

The table's monitor is on the top. It is actually a projection surface sturdy enough to hold the physical objects of our TQI and the weight of people's hands and arms leaning on it. For this reason a sheet of plexiglas is used, where one side has a matte finishing. It serves all the above requirements. It is thick enough to withstand the objects' weight and some pressure from the users. Also, the fact that it is matte from one side serves to both being a good projection surface and hiding the internals of our table. The fiducial markers can be recognised by the camera using this level of matte finishing.

The internal structure of the table is mainly dependent on the ReactIVision software. The camera that is used for tracking the markers is a PS3Eye (Figures 6.5, 6.6). It is placed in a 90° angle and faces the plexiglas, in such way as to capture the whole surface. The short throw projector (a projector able to display within a shorter distance than usual) is placed below the camera facing the side of the table. In order to be able to project on the surface, a mirror is placed in front of the projector lens in an angle capable to reflect the projection on the plexiglas (Figure 6.3).

In order to align the camera image capturing, the projection and the application's graphical environment of the tabletop interface, ReactIVision should be calibrated. To achieve that, a rectangular grid with an ellipse image must be projected on the surface, the image being bundled with the framework. When this happens, the calibration mode should also be triggered in the ReactIVision application. While the camera captures the real-time video stream, the same grid image is shown on top of the video streaming. Now the user should align the grid and ellipse lines of the application to coincide with the ones of the projection. This way the movement of the markers will have a one to one response and location with the projection and the graphical environment of the application, regardless of the camera capture angle.

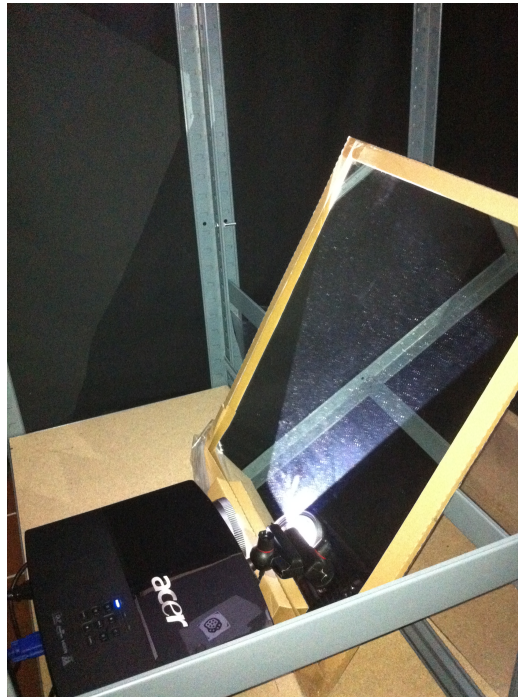


Figure 6.3: The projector and PS3Eye

6.4 TQI Application

Following the requirements analysis, an application featuring the new TQI was developed. It is written in Processing, which builds on Java, using also purely Java classes. The TUIO library for Processing is used in order to make use of fiducial markers for the ReactIVision framework. The reason to choose Processing is that it uses simplified syntax and techniques to use graphics. Future work can benefit from building on top of the current application with a more advanced graphical interface, with less difficulties. It is really helpful and important that Processing is written in Java and applications can use both languages.

6.4.1 Key Elements

The fiducial markers, known as TUIO objects, are captured when appearing on the table. Adding, removing or moving a fiducial marker are events that trigger actions (ArtVis results of these actions can be seen in Figure 6.7). The IDs of the fiducial markers that have the role of a query or negation object are declared.

Grouping

When a TUIO object is a query object, a group of objects is created with a root, the root being the query object itself. The exact same procedure also occurs for the negation object, as the separation does not take place in the TQI-side of the system. The root of a group serves to determine which TUIO object is the main object "leading" the group, making it also unique. Query objects being removed, remove the group from the table's groups. Adding operand markers on the table-top triggers a distance computation, in order to determine if this marker belongs or not to a certain group.

A default radius (Figure 6.4) and is set for the area where a query takes place when placing a query object. This radius is chosen to best fit as many objects within groups in correlation to as many groups on the table as possible. When an operand object's status is updated (moved, added), the distance between that and the query objects is computed. If this distance is less than the radius, the closest group is chosen and the operand is added to it. If the operand is removed from the table or moved away from the group's radius, it is removed from the group. Moving the object closer to another group, within the radius of it, will add the operand to this group. The total groups are being tracked and kept as the table's groups.

Sending to ArtVis

A thread runs during the whole runtime connected to the main ArtVis application through sockets. When there is a change in the group status (updated group members or whole groups added/removed), a message is sent to the main application. This message gets all the table's groups and translates the Java objects to JSON messages. These messages are captured by ArtVis (or any other system attached to the other end) and translated there.

6.4.2 Message Protocol

As already discussed, the TQI communicates with ArtVis through JSON messages of the table's groups of objects. When a change occurs in the status of the groups, a message is sent to the ArtVis application. This JSON message holds arrays of the groups formed at the time they are formed. Each array holds just the IDs of the fiducial markers, which is the only information needed on the other side of the system (the ArtVis application). This way we reduce complexity and perform better, by only sending arrays of ID strings. Each array holds as its first element the ID of the root of each group, which is the ID of the fiducial marker of a query

or negation object. Every consecutive ID belongs to an operand object.

Using a JSON message with our own syntax as a protocol to let the TQI communicate with the main application is something important. From a purely technical perspective, by doing it this way, we avoid the complexity of serialisation of Java objects, achieving better performance and being more reliable. Communication, from a more functional point of view, is cleaner and does not need to be "deciphered" to find the actual meaningful information from the side of the main application. Thus, we feed the ArtVis application with a clean view of what are the users' queries to be constructed.

We have already mentioned that we need an SQL RDBMS independent TQI that will not change according to the RDBMS the application it is connected to uses. This way, we might not have a directly plug-n-play interface, but it can still be used as a black box, moving this operation to writing glue code from the other side of the system.

6.4.3 Testing

For testing the application there is no need to run the whole tabletop interface each time. ReactIVision provides a TUIO simulator which simulates a tabletop interface and a number of fiducial markers. The user/developer can simulate the use of the real table and test the results produced by the application. Before moving to the real TUI a fair amount of testing during and after development took place on the simulator.



Figure 6.4: The table in action



Figure 6.5: Under the table

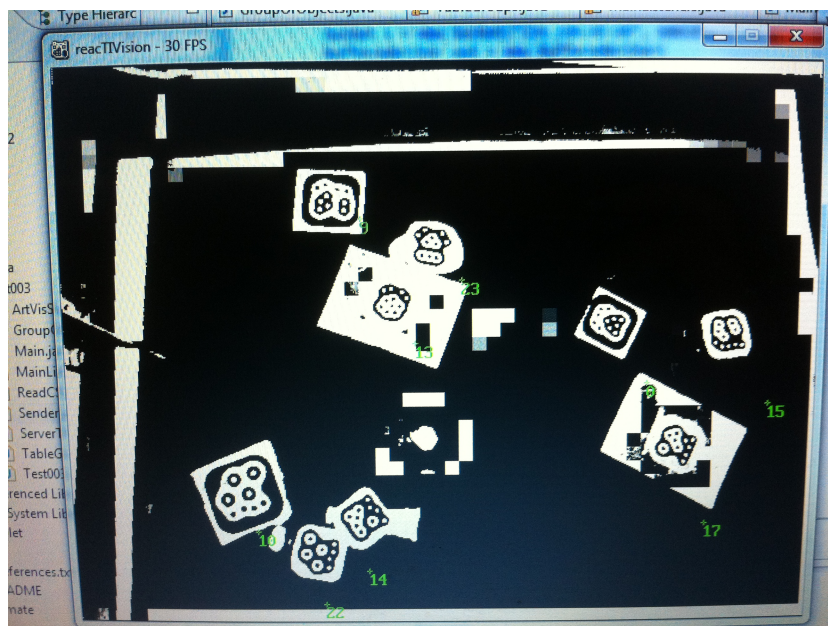


Figure 6.6: What ReactTVision "sees"



Figure 6.7: The result on the ArtVis canvas

6.5 ArtVis Changes

The new TQI requires changes in the ArtVis application, in order to handle the TQI and add functionality. The additions to the code of ArtVis might vary from direct MySQL queries to glue code. Changes related to the Prefuse Java-based toolkit [21] need to be applied, as it is the InfoVis toolkit used in ArtVis.

6.5.1 Prefuse

The prefuse toolkit is a visualisation framework for Java, with capabilities of building interactive InfoVis applications, making use of Java's 2D graphics library. Prefuse follows a common software architecture pattern for visualisation toolkits: the information visualisation reference model [11].

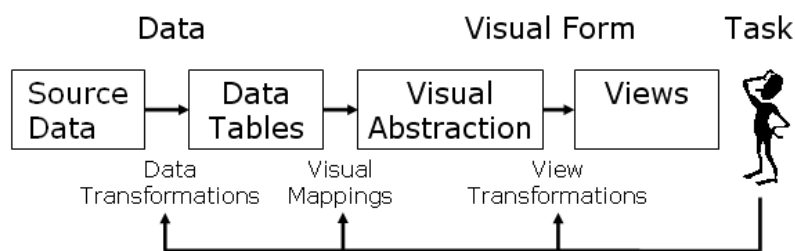


Figure 6.8: The information visualisation reference model ¹

A source of data that should be visualised is used, in our case the ArtVis optimised WGA database. This source of data is used to construct data tables, which are internal representation of data to be visualised. These data tables are then subject to a mapping to a data model with visual models known as visual abstraction. There, all the information for drawing the visual representations of our source data is held. The (interactive) views are the actual rendered data visualisations based on the visual abstraction. The user may then interact with the visualisation and cause changes in the data transformations, visual mapping, or view transformations processes.

6.5.2 Querying the Data

Our research's purpose is to create an interface to query data using a TUI. To achieve that in relation to ArtVis, we need to give the ability to the user to query

¹Source: <http://prefuse.org/doc/manual/introduction/structure/>

the available data and get the results. In our case, ArtVis returns the result sets of the digital information through visualisations. We wanted to map the tangible queries to SQL queries, more directly, so we decided to inject our MySQL queries directly as the source data in the very deep foundations of the information visualisation reference model. This way the already created and used visual abstractions and interactive views can benefit from using similar data tables, but with different numbers of items. Instead of using the data tables of all artworks, we, in every query, set the data source as the filtered data produced out of our query. For instance, in an attempt to demonstrate what happens, for a query returning 100 artworks as a result set, we create data, load this set as the data source and create data tables of 100 artworks, instead of, approximately, 27600.

6.5.3 Receiving the Query Groups

As already discussed extensively, the TQI application captures the table's groups of objects and sends them to ArtVis as JSON messages. In order to capture the messages that are transmitted through sockets, a socket server thread runs in the ArtVis application. The JSON message are translated to a Java object, a vector (the whole table) of vectors (the groups) of integers (the fiducial markers' IDs). The application then handles this group vectors and their integers in a specific way. The first element of the groups' vectors is always the ID of the root of the group, the query or negation object. This way it can be identified if a group is a negation or a query group. This way the queries can be constructed, including logical complements if applicable, and then unified towards a full query.

An example of a JSON message is the following:

```
[[9, 29, 6], [0, 2, 1], [10, 28, 3]]
```

The message is comprised of three groups and the query/negation objects have the IDs of 9, 0, and 10. Each group has some operand objects attached. The first one has 29 and 6, the second one 2 and 1, and the last one 28 and 3.

6.5.4 Query Processing

The query is created through the analysis of the received JSON message of the groups, after it is "deciphered" back to a vector. Potential negation is prepared to be added to the queries, the individual queries are created, and then, finally, the unified query is created.

Negation

First, potential presence of a negation group is detected. If it exists a part of a MySQL query is created, to be appended to the full queries created. Within this group, the Boolean operator OR is used to set the logical relationship in between the operands. These operands are surrounded by the NOT operator. This piece of logical complement is using the Boolean AND to be connected to each individual query. If there are no other query groups, the negation is applied to the whole data set of artworks.

The structure of the negation is as follows:

```
AND NOT (ArtworkForm = "Form1" OR ArtworkInspiration
= "Inspiration2" OR ArtistSchool = "School7" OR ...
OR ArtistSchool = "School9")
```

Individual Queries

We have already described on a high level what happens within the query groups, and how Boolean AND and OR operators are used. Whenever there is a unique occurrence of type of an operand (Artwork Form, Artwork Inspiration, Artist School), the AND operator is used by default. This is not applicable when more than one same type of operands occur, as the result set of the query will always be null if the AND operator was always used. In this case, the OR Boolean is used in between the identical type of operands automatically. The algorithm traverses through the vector of markers and when an operand appears for the first time, it is noted. Now, every time an operand of the same type is found, instead of adding an AND operator before it, an OR is added. Operands of the same type are grouped under an and operator AND using the OR operator in between them.

A general approach of how an individual query is handled follows:

```
SELECT * FROM table WHERE (ArtworkForm = "Form1"
OR ArtworkForm = "Form3" AND (ArtworkInspiration
= "Inspiration1") AND (ArtistSchool = "School1" OR
ArtistSchool = "School2" OR ... OR ArtistSchool =
"School7")
```

Final Query

The final query is created out of the combination of the individual queries and the negation. All individual queries get the negation, if applicable, appended to them. These queries are then connected through the UNION operator. The final unified

query will return a result set with all the results from each individual queries, with all duplicate records removed. We chose to remove duplicates as opposed to UNION ALL to not return repetitive results.

The query that is applied to the ArtVis database looks like the following:

```
(SELECT * FROM table WHERE (ArtworkForm = "Form1"
OR ArtworkForm = "Form3") AND (ArtistSchool =
"School1" OR ArtistSchool = "School2" OR ... OR
ArtistSchool = "School7") AND NOT (ArtworkInspiration
= "Inspiration2")) UNION (SELECT * FROM table WHERE
(ArtworkForm = "Form2") AND NOT( ArtistSchool =
"School5")) UNION ... UNION (SELECT * FROM table
WHERE (ArtistSchool = "School4") )
```

6.5.5 Saving and Retrieving Queries

Saving and retrieving queries takes places using the Phidgets RFID reader and RFID tags. The same Java classes of the original ArtVis for the Phidgets RFID were used. We created, based on a Java interface for the RFID tags, a functionality to save and retrieve the queries. Each time a tag approaches the RFID reader, the tag's ID is searched throughout the tags' plain text configuration files to find what is the functionality behind it. In our case, the system checks if the tag exists in it, and if not then the tag should be used to save the current query on the TQI. Else, if the tag is a saved query, the query is ran by the system. To achieve that, a plain text file is used to store the ID of the tags and the SQL that is saved under each tag.

Saving

If the tag that is scanned by the RFID reader is not assigned to any functionality, the tag is used to store a query by default. As already described, when the TQI application sends a message, a thread running a sockets server in the ArtVis application receives and compiles the SQL query. At the same time, this query is stored temporarily in a file. This file is accessed when scanning the "blank" tag, and the query stored there is matched to the tag's ID. This pair of information is then, in its turn, saved within the special plain text file, for storing queries.

Retrieving

Now, if the tag is scanned by the RFID reader, but this time it is assigned to a saved query, the query is triggered. The query, only, is then saved to a file. The

thread running the sockets server checks the file and if a query is loaded, the ArtVis database is queried. The result set then is returned to the GUI as if the TQI's tabletop had the physical objects arranged and grouped in the same way, as to trigger this query.

Chapter 7

Conclusions and future work

7.1 Future Work

The research that took place and the final product of this thesis, allows multiple opportunities for future work. The opportunities allow our TQI to be further extended and improved. More elaborate work will allow the system to be expanded towards other environments than museum or exhibitions, and be used extensively.

Our prototype allowed exploring facets as artwork inspiration and forms, and artist schools. There is also room to expand towards more operands, like artist name or collection name. Also adding fiducial markers and RFID tags, can be further optimised by offering an administration console. The latter is already mentioned as future work in the original ArtVis research, but it is out of the scope of this thesis.

An evaluation of the TQI should take place. From there we can extract conclusions and feedback towards our system. Also, we can actually test whether using special objects to form queries is intuitive enough, especially for users with limited technological expertise.

We consider that our approach can be further extended and become more complete, by moving some functionality from the ArtVis palette (the Phidgets console) to the table. An indicative functionality could be to adjust the timeline through the TQI, by either rotating the query or negation object by itself, and use the 50 year time span, or, alternatively, use two different physical objects to adjust both the lower and upper bound. Interactive paper can be also applied in our TQI and leverage our systems natural interface by using its handwriting recognition capabilities. The user may write an artist's name on a piece of paper and use it as an

operand. Saving queries can be done by writing something on a piece of paper, and then retrieve the query by re-writing it.

Adding more feedback on the tabletop interface is also an important parameter to consider, which might overcome possible negative user acceptance for the query objects. This can be applied using additional graphics on the table's projection area. An alternative way to group objects can be done by allowing a user to draw a circle grouping the desired physical objects operands on the tabletop interface. This can be done by both using his finger or a stylus. Both improvements, the additional feedback and the grouping, require a high-end multi-touch table to be used. This hardware could give us the opportunity to explore other possibilities, too.

The TQI we realised may apply to systems with a different user base and another database structure. For instance, if there were situations where AND and OR operators made sense, we may need a slight differentiation in our approach. The user may want to separate the query objects to use AND and OR Boolean operators separately for each individual query. This of course requires that the system would not return a lot of null result sets and that the user has the SQL and logic expertise and knowledge to be able to handle this correctly and successfully.

Finally, this thesis and its research context will benefit from a real-world application of its realisation. Using the ArtVis application with our TQI within a museum or exhibition from real users (children, elderly, students, etc) will show us the real achievements and challenges of the system we realised.

7.2 Conclusions

In this thesis, we developed a Tangible Query Interface based on ArtVis. This TQI is used by users of different ages and technical experience, built to be used mainly in a museum or exhibition context. The interface is built in a separate system than the original ArtVis and is designed to be used in a collaborative environment and also leverage the querying process of the already existing basic TQI.

We did extensive research in the context of available TQIs, along with TUIs in general, information visualisation, and tabletop interfaces, the TQIs getting the most attention. Related work query through tangible interfaces was extensively covered. With this information we gained throughout our research, we managed to do a comparison between the different existing systems that are closely related to our desired solution. This helped determine the requirements of an effective TQI,

through exploring the important characteristics, their drawbacks, and what they are missing.

Based on the requirements we extracted, we conceptualised scenarios fulfilling those. Our approach introduces an environment that is both individual and collaborative-friendly for users. We give the opportunity to construct queries closely related to SQL to extract digital information using physical objects. The implementation took place between two separate systems: the tabletop TQI and ArtVis. This helped to also avoid the TQI from being too application specific.

Below, we are re-introducing the comparison table (Table 7.1) from Chapter 2, enhanced with our initial scenario and our final prototype. Our system covers every category in the table, except for the active objects. The latter was used in our initial prototype, a feature that we decided would feel unnatural to the user. An active area is used, which is the tabletop interface. The four SQL operators we introduced are all included in the final solution. Saving a query and constructing multiple simultaneously is also available. Flexibility in query grouping is offered using operator both as position and object. Any physical object can be used as an operand, and we also offer adding, removing, and modifying objects. Further analysis took place in the previous chapters.

	Tangible Query Interfaces [52]	TeddIR [30]	Interactive Venn Diagrams [36]	Navigational Blocks [8], [7]	Stackables [35]	Search Wall [12]	ArtVis[15]	Jigsaw Puzzles	Prototype
AND	✓	n/a ¹	✓	✓	✓	✓	-	✓	✓
OR	✓	n/a	✓	-	✓	✓	-	✓	✓
NOT	-	✓	✓	-	-	-	-	✓	✓
UNION	-	-	-	-	✓	-	-	✓	✓
Save Query	- ²	-	-	-	-	-	-	-	✓
Multiple Queries	-	n/a	-	-	✓	-	-	✓	✓
(Free) Query Grouping	-	-	-	-	-	-	-	-	✓
Active Area	✓	✓	✓	✓	-	✓	✓	✓	✓
Active Objects	✓	-	-	- ³	✓	-	-	✓	-
Natural Metaphors	-	✓	✓ ⁴	-	-	-	✓	- ⁵	✓
Operator as Position	✓	✓	✓	✓	✓	✓	-	✓	✓
Operator as Object	-	-	-	-	-	-	-	-	✓
Add/Remove Objects	-	-	-	-	-	-	✓	-	✓
Modify Objects	-	-	-	-	-	-	✓	-	✓
Integrate Any Physical Object	-	✓	✓	-	-	✓	✓	-	✓

Table 7.1: Comparison between related work and our systems

¹ An algorithm computes relevant results

² future work

³ future work

⁴ actual object

⁵ only for NOT

7.2.1 Contributions

Question 1: How does the state of the art address tangible querying?

In our thesis we analyse extensively the work provided by related research. We gathered our findings and compare the said work amongst one another and also ours in a comparison table. This way we managed to determine what is provided and what are the inconsistencies of the state-of-art tangible querying systems. This was a determinant to formulate our work and how we should proceed, in order to contribute the most to our field.

Question 2: What are the links between tangible querying and InfoVis (Information Visualisation)?

ArtVis is a system providing advanced visualisation techniques based on an artwork database, as described in Chapter 3. Our work made use of these techniques and manipulated the visualisations through tangible querying. We provided a technique to query data through real-world tangible physical objects and reflect the provided digital information changes to the visualisations. The objects that we used are tightly linked to the visualisation. Also, we provide a collaborative environment to query and visualise data by supporting input from multiple queries at the same time. This queries may come from one or more users. Finally, our system allows the use of any physical object as input, providing better metaphor to visualisations mapping.

7.2.2 Query Operands

The operands of a query are physical objects that have a form closely related to what they actually represent. We used the ArtVis objects used for filtering by attaching a fiducial marker under them, in order for the tabletop interface to recognise them. These objects' forms have a relation between the type and the value of the operand they want to represent. We believe that this approach will provide a more natural feeling to the users' actions with a more intuitive query setting. Seeing the table and the objects for the first time should give a first hint as to how our interface works.

7.2.3 Query Operators

We decided to use the AND, OR, and NOT operators within the individual queries, though the negation has a more general approach. The UNION operator is used to connect all individual queries. These queries are determined through special query

and negation objects. A possible limitation is that it might not be a natural enough way to create queries, but we believe it serves its purpose. A user evaluation could give us some feedback on this. Some more feedback on our tabletop interface could help towards providing a deeper understanding to the user.

7.2.4 Saving and Retrieving Queries

Saving and retrieving queries is also a key feature of the product of our research. Users can save queries for future reference and retrieve them immediately, without having to rearrange the whole TQI's physical objects. This may benefit advanced or even just frequent users. We believe this may leverage the system, getting closer to GUI using tangible physical objects.

Bibliography

- [1] H. Agrawal and K. Sorathia. AstroGrasp: A Tangible User Interface for Teaching Basic Astronomy Concepts. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction, APCHI '13*, pages 123–128, Bangalore, India, 2013.
- [2] A. N. Antle, M. Droumeva, and D. Ha. Thinking with Hands: An Embodied Approach to the Analysis of Children’s Interaction with Computational Objects. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems, CHI EA '09*, pages 4027–4032, Boston, MA, USA, 2009.
- [3] A. N. Antle, N. Motamedi, K. Tanenbaum, and Z. L. Xie. The EventTable Technique: Distributed Fiducial Markers. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI '09*, pages 307–313, Cambridge, UK, 2009.
- [4] P. Baudisch, T. Becker, and F. Rudeck. Lumino: Tangible Building Blocks Based on Glass Fiber Bundles. In *ACM SIGGRAPH 2010 Emerging Technologies, SIGGRAPH '10*, pages 16:1–16:1, Los Angeles, California, 2010.
- [5] J. Bohn. The Smart Jigsaw Puzzle Assistant: Using RFID Technology for Building Augmented Real-World Games. In *Workshop on Gaming Applications in Pervasive Computing Environments at Pervasive 2004*, Linz/Vienna, Austria, 2004.
- [6] E. H. Calvillo-Gómez, N. Leland, O. Shaer, and R. J. K. Jacob. The TAC Paradigm: Unified Conceptual Framework to Represent Tangible User Interfaces. In *Proceedings of the Latin American Conference on Human-computer Interaction, CLIHC '03*, pages 9–15, Rio de Janeiro, Brazil, 2003.
- [7] K. Camarata, E. Y.-L. Do, M. D. Gross, and B. R. Johnson. Navigational Blocks: Tangible Navigation of Digital Information. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems, CHI EA '02*, pages 752–753, Minneapolis, Minnesota, USA, 2002.

- [8] K. Camarata, E. Y.-L. Do, B. R. Johnson, and M. D. Gross. Navigational Blocks: Navigating Information Space with Tangible Media. In *Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI '02*, pages 31–38, San Francisco, California, USA, 2002.
- [9] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [10] L. K. Y. Chan and H. Y. K. Lau. A Tangible User Interface Using Spatial Augmented Reality. In *Proceedings of the 2010 IEEE Symposium on 3D User Interfaces, 3DUI '10*, pages 137–138, Washington, DC, USA, 2010. IEEE Computer Society.
- [11] E. H.-H. Chi. *A Framework for Information Visualization Spreadsheets*. PhD thesis, Minneapolis, MN, USA, 1999.
- [12] K. Detken, C. Martinez, and A. Schrader. The Search Wall: Tangible Information Searching for Children in Public Libraries. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI '09*, pages 289–296, Cambridge, UK, 2009.
- [13] P. Dietz and D. Leigh. DiamondTouch: A Multi-user Touch Technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, UIST '01*, pages 219–226, Orlando, Florida, 2001.
- [14] A. Dix and G. Ellis. Starting Simple: Adding Value to Static Visualisation Through Simple Interaction. In *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '98*, pages 124–134, L'Aquila, Italy, 1998.
- [15] B. Dumas, B. Moerman, S. Trullemans, and B. Signer. ArtVis: Combining Advanced Visualisation and Tangible Interaction for the Exploration, Analysis and Browsing of Digital Artwork Collections. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces, AVI '14*, pages 65–72, Como, Italy, 2014.
- [16] J.-D. Fekete, J. J. Wijk, J. T. Stasko, and C. North. Information Visualization. chapter The Value of Information Visualization, pages 1–18. Springer-Verlag, Berlin, Heidelberg, Germany, 2008.
- [17] G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95*, pages 442–449, Denver, Colorado, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

- [18] M. Fjeld, F. Voorhorst, M. Bichsel, K. Lauche, M. Rauterberg, and H. Krueger. Exploring Brick-Based Navigation and Composition in an Augmented Reality. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 102–116, Karlsruhe, Germany, 1999. Springer-Verlag.
- [19] G. W. Furnas. Generalized Fisheye Views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '86, pages 16–23, Boston, Massachusetts, USA, 1986.
- [20] J. Y. Han. Low-cost Multi-touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 115–118, Seattle, WA, USA, 2005.
- [21] J. Heer, S. K. Card, and J. A. Landay. Prefuse: A Toolkit for Interactive Information Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 421–430, Portland, Oregon, USA, 2005.
- [22] B. Hengeveld, C. Hummels, K. Overbeeke, R. Voort, H. van Balkom, and J. de Moor. Tangibles for Toddlers Learning Language. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 161–168, Cambridge, UK, 2009.
- [23] U. Hinrichs, H. Schmidt, and S. Carpendale. EMDialog: Bringing Information Visualization into the Museum. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1181–1188, Nov. 2008.
- [24] M. S. Horn and R. J. K. Jacob. Designing Tangible Programming Languages for Classroom Use. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, pages 159–162, Baton Rouge, Louisiana, 2007.
- [25] M. S. Horn, E. T. Solovey, and R. J. K. Jacob. Tangible Programming and Informal Science Learning: Making TUIs Work for Museums. In *Proceedings of the 7th International Conference on Interaction Design and Children*, IDC '08, pages 194–201, Chicago, Illinois, 2008.
- [26] E. Hornecker and J. Buur. Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 437–446, Montreal, Quebec, Canada, 2006.

- [27] H. Ishii. Tangible Bits: Beyond Pixels. In *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, TEI '08, pages xv–xxv, Bonn, Germany, 2008.
- [28] H. Ishii and B. Ullmer. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 234–241, Atlanta, Georgia, USA, 1997.
- [29] R. Jafri. Electronic Braille Blocks: A Tangible Interface-Based Application for Teaching Braille Letter Recognition to Very Young Blind Children. In K. Miesenberger, D. Fels, D. Archambault, P. Pez, and W. Zagler, editors, *Computers Helping People with Special Needs*, volume 8548 of *Lecture Notes in Computer Science*, pages 551–558. Springer International Publishing, 2014.
- [30] M. Jansen, W. Bos, P. van der Vet, T. Huibers, and D. Hiemstra. TeddIR: Tangible Information Retrieval for Children. In *Proceedings of the 9th International Conference on Interaction Design and Children*, IDC '10, pages 282–285, Barcelona, Spain, 2010.
- [31] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactTable: Exploring the Synergy Between Live Music Performance and Tabletop Tangible Interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, pages 139–146, Baton Rouge, Louisiana, 2007.
- [32] M. Kaltenbrunner and R. Bencina. reactIVision: A Computer-vision Framework for Table-based Tangible Interaction. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, pages 69–74, Baton Rouge, Louisiana, 2007.
- [33] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. TUIO - A Protocol for Table Based Tangible User Interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, 2005.
- [34] D. A. Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan. 2002.
- [35] S. Klum, P. Isenberg, R. Langner, J.-D. Fekete, and R. Dachsel. Stackables: Combining Tangibles for Faceted Browsing. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 241–248, Capri Island, Italy, 2012.

- [36] H. Koike, Y. Sato, Y. Kobayashi, H. Tobita, and M. Kobayashi. Interactive Textbook and Interactive Venn Diagram: Natural and Intuitive Interfaces on Augmented Desk System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 121–128, The Hague, The Netherlands, 2000.
- [37] A. Mazalek, G. Davenport, and H. Ishii. Tangible Viewpoints: A Physical Approach to Multimedia Stories. In *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, pages 153–160, Juan-les-Pins, France, 2002.
- [38] A. Olwal and A. D. Wilson. SurfaceFusion: Unobtrusive Tracking of Everyday Objects in Tangible User Interfaces. In *Proceedings of Graphics Interface 2008*, GI '08, pages 235–242, Windsor, Ontario, Canada, 2008. Canadian Information Processing Society.
- [39] J. Patten, H. Ishii, J. Hines, and G. Pangaro. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 253–260, Seattle, Washington, USA, 2001.
- [40] J. Rekimoto, B. Ullmer, and H. Oba. DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 269–276, Seattle, Washington, USA, 2001.
- [41] S. D. Scott, M. S. T. Carpendale, and K. M. Inkpen. Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, CSCW '04, pages 294–303, Chicago, Illinois, USA, 2004.
- [42] O. Shaer and E. Hornecker. Tangible User Interfaces: Past, Present, and Future Directions. *Foundations and Trends in Human-Computer Interaction*, 3(1–2):1–137, Jan. 2010.
- [43] O. Shaer and R. J. Jacob. A Specification Paradigm for the Design and Implementation of Tangible User Interfaces. *ACM Transactions on Computer-Human Interaction*, 16(4):20:1–20:39, Nov. 2009.
- [44] O. Shaer, N. Leland, E. H. Calvillo-Gamez, and R. J. K. Jacob. The TAC Paradigm: Specifying Tangible User Interfaces. *Personal Ubiquitous Comput.*, 8(5):359–369, Sept. 2004.

- [45] D. Shenk. *Data Smog: Surviving the Information Glut*. HarperCollins Publishers, 1997.
- [46] C. Smith. The Hand that Rocks the Cradle. *Informatica Didactica*, 1995.
- [47] T. Speelpenning, A. N. Antle, T. Doering, and E. van den Hoven. Exploring How Tangible Tools Enable Collaboration in a Multi-touch Tabletop Game. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II, INTERACT'11*, pages 605–621, Lisbon, Portugal, 2011. Springer-Verlag.
- [48] M. Spindler, C. Tominski, H. Schumann, and R. Dachsel. Tangible Views for Information Visualization. In *ACM International Conference on Interactive Tabletops and Surfaces, ITS '10*, pages 157–166, Saarbrücken, Germany, 2010.
- [49] B. Ullmer and H. Ishii. The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology, UIST '97*, pages 223–232, Banff, Alberta, Canada, 1997.
- [50] B. Ullmer and H. Ishii. Emerging Frameworks for Tangible User Interfaces. *IBM Systems Journal*, 39(3-4):915–931, July 2000.
- [51] B. Ullmer, H. Ishii, and D. Glas. mediaBlocks: Physical Containers, Transports, and Controls for Online Media. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, pages 379–386, 1998.
- [52] B. Ullmer, H. Ishii, and R. J. K. Jacob. Tangible Query Interfaces: Physically Constrained Tokens for Manipulating Database Queries. In *Proceedings of Interact 2003*, Zurich, Switzerland, September 2003.
- [53] B. Ullmer, H. Ishii, and R. J. K. Jacob. Token+Constraint Systems for Tangible Interaction with Digital Information. *ACM Transactions on Computer-Human Interaction*, 12(1):81–118, Mar. 2005.
- [54] J. Underkoffler and H. Ishii. Illuminating Light: An Optical Design Tool with a Luminous-tangible Interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '98*, pages 542–549, Los Angeles, California, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

- [55] J. Underkoffler and H. Ishii. Urp: A Luminous-tangible Workbench for Urban Planning and Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 386–393, Pittsburgh, Pennsylvania, USA, 1999.
- [56] M. Waldner, J. Hauber, J. Zauner, M. Haller, and M. Billinghurst. Tangible Tiles: Design and Evaluation of a Tangible User Interface in a Collaborative Tabletop Setup. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, OZCHI '06, pages 151–158, Sydney, Australia, 2006.
- [57] P. Wellner. Interacting with Paper on the DigitalDesk. *Communications of the ACM - Special issue on computer augmented environments: back to the real world*.
- [58] L. Xie, A. N. Antle, and N. Motamedi. Are Tangibles More Fun?: Comparing Children's Enjoyment and Engagement Using Physical, Graphical and Tangible User Interfaces. In *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, TEI '08, pages 191–198, Bonn, Germany, 2008.
- [59] D. Xu. Tangible User Interface for Children - An Overview. In *Proceedings of the SIXTH Conference in the Department of Computing*, pages 579–584, Preston, Lancashire, UK, 2005.