TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DIPLOMARBEIT

Open Source Web-based Interaction with 3-dimensional Building Models via Human Interaction Devices - a Usability Study

ausgeführt am

Institut für Geoinformation und Kartographie

und dem

Institut für Architekturwissenschaften

der Technischen Universität Wien

unter der Anleitung von

O.Univ.Prof. Dipl.-Ing. Dr.techn. Andrew U. Frank

Univ.Prof. Dipl.-Ing. Dr.techn. Ardeshir Mahdavi

durch

Stefan Glawischnig

Hofbauergasse 6/ 203

A-1120 Wien

Wien, am 18. Jänner 2013

Kurzfassung

Der Gebäudesektor hat in den meisten Industrienationen einen geschätzten Anteil am Gesamtenergieverbrauch von 40%. Berücksichtigt man die globalen Herausforderungen in puncto Ressourcenknappheit, Energiekrisen und den damit verbundenen steigenden Energiekosten wird ersichtlich, dass der Gebäudesektor einen wichtigen Teil in nationalen Energieoptimierungsstrategien einnimmt. In Österreich beträgt das Bauvolumen pro Jahr weniger als ein Prozent des gesamten Gebäudevolumens. Die Steigerung der Energieeffizienz der bestehenden Gebäudeinfrastruktur ist demnach ein entscheidender Schritt zur Optimierung des Energieverbrauches am Gebäudesektors.

Gebäudemonitoringsysteme dienen der Speicherung und Analyse von Gebäudedaten, wie Energieverbrauch elektronischer Geräte, Wärmeleistung, Luftfeuchtigkeit, Luftzug u.ä.. Erhobene Daten sind komplex und müssen aufbereitet werden, um von Laien in einen Kontext mit dem persönlichen Verhalten gebracht werden zu können. Die folgende Arbeit erläutert die Umsetzung eines Plattform- und Hersteller-unabhängigen Gebäudemonitoringsystems.

Die Umsetzung der entwickelten Konzepte mit dem Ziel, Gebäudedaten einer breiten Masse an interessierten Nutzern zur Verfügung zu stellen, wird anhand des entwickelten Frameworks diskutiert. Am Beispiel eines Gebäudeinformationsterminals wird untersucht, welche Technologien einen möglichst schnellen Lernerfolg garantieren. Unter Einbeziehung vorhandener Konzepte der menschlichen Wahrnehmung von räumlichen Phänomenen wird das Lernverhalten mit Personal Computer, Touchscreens und die Bedienung mittels Gestensteuerung (Nutzung ohne physischen Kontakt) untersucht.

Eine mit dreißig Studenten durchgeführte Nutzbarkeitsstudie zeigt, dass ein Touchscreen die besten Lernergebnisse erzielt. Eine Prototypimplementierung befindet sich am Institut für Architekturwissenschaften der Technischen Universität Wien.

Abstract

The present contribution describes the concepts and implementation of a vendor and platform independent building monitoring framework. Building sensor data (e.g. electricity power consumption, air flow, temperature) is collected from various data sources and stored in a database. The data storage and user interface design concepts are introduced with focus on a web based representation. The example implementation of a building information terminal is used to conduct a usability study. The application implements support for various input methods. By using a personal computer, a touch screen and a gesture input device (control an application without direct physical contact), the study is conducted to specify which input device offers the best learning experience.

Acknowledgments

The research presented in this paper is supported by funds from the program „Neue Energien 2020“ within the „Klima- und Energiefonds“ (no: 834517). Additional support was provided by the division „Gebäude und Technik“(Amtsdir. Hodcek), which supplied us with real-world test beds. Moreover, the thematic link to the CAMPUS 21 project (Control & Automation Management of Buildings & Public Spaces in the 21st Century, no: 285729) provided further impulses for the realization of the research objectives.

First of all I want to thank professor Frank for the supervision of my thesis and that he awoke my interest in science through his support during my master studies - and professor Mahdavi for giving me the opportunity to take my first steps in the scientific community and allowing me to follow my ideas.

Special thanks belong to all Monitoring System Toolkit team members and my colleagues at the Departments of Building Physics and Ecology and the Department of Geodesy and Geoinformation for all the great time and conversations.

Finally and most importantly, I want to thank my parents. Their support made me who I am today.

Abbreviations

BIM	Building Information Model
BMS	Building Monitoring System
DND	Drag and Drop
DOM	Document Object Model
GIS	Geographic Information Science
GWT	Google Web Toolkit
HID	Human Interaction Device
HTML	Hyptertext Markup Language
IFC	Industry Foundation Classes
JDBC	Java Database Connectivity
JS	Java Script
JSNI	Java Script Native Interface
MOST	Monitoring System Toolkit
OPC DA	Open Process Control Data Access
OS	Operating System
RPC	Remote Procedure Call
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UI	User Interface
WebGL	Web Graphics Library
XML	Extensible Markup Language

Contents

Chapter 1

Introduction

Human life is strongly connected to space. Daily life builds upon abstractions of our surroundings (section 2.6). Experiences and knowledge about places and distances influence the interaction with our physical and virtual environments [KuFr91].

Cheap and performant hardware pushes mobile application development and offers an alternative to the conventional desktop computers. Touch screens introduce a new interaction method as an alternative to mouse and keyboard. Touch screens combine the input and output device.

Human Computer Interaction devices abstract the interaction process even further. No direct physical contact to the input device is necessary. An optical sensor (red, green, blue) and an infrared sensor capture a user's body movement. The recorded gestures are translated into commands by the driver. This work will compare the three input methods

- Keyboard and mouse
- Touch
- Gesture.

I focus on the question if the input method is influencing one's learning process. The evaluation is the result of a usability study with thirty participants. The study is using the Monitoring System Toolkit (MOST) Framework developed at the Department for Building Physics and Building Ecology at the Vienna University of Technology [⇒Most]. This framework offers an open source web based building monitoring application that is accessed via a building information terminal. An information terminal is used frequently by various people. To attract new users, the application must be understandable and easy to use. One reason to conduct the study is to find out if one input method is preferable.

1.1 Motivation

A building information terminal displays complex data. An application that is working with building performance data includes various data sources like databases and sensor networks. The representations must be understandable for users with differing educational background. The best application is useless if the logic and the interaction method is difficult to understand. Touch devices simplify the interaction process, because they combine the input and output device. Gesture input removes the necessity for direct physical contact to interact with an application.

People base their daily routines on real world concepts of space (section 2.6). This paper deals with the question if adding space concepts to a human - machine interaction process reduces the learning time.

Gesture input, primarily known from computer gaming, is also used in a number of business applications (e.g. surgery). In this case it is used to interact with a Spatial Information System. An example use case is to interact with a 3D building model and to extract information. This includes finding room numbers or ways to this rooms.

The comparison of various input methods to identify the possible impact on the cognitive human-machine interaction influences hard- and software development strategies.

1.2 Goals and Hypothesis

The goal of this thesis is to compare the three input methods

- Keyboard and mouse
- Touch
- Gesture.

A usability study evaluates if one input method is preferable because of increased learning speed and quality.

The hypothesis is:

„*The input method does influence the learning process*“

1.3 Approach

Verifying the hypothesis requires a practical approach. The development process will be introduced, focusing on the various input methods, the influences on the user interface and general development approaches. The example application implements the discussed concepts. Thirty students will interact with the application within the scope of a usability study. The measured information will be used to prove the hypothesis right or wrong.

1.4 Scientific background

The research is strongly connected to the fields of computer science and cognitive sciences. The hypothesis is build upon cognitive research. Throughout the thesis cognitive research will be discussed and referenced, primarily focusing on human spatial cognition of space. Proving a cognitive theory requires a practical approach, because the results can hardly be calculated.

1.5 Relevance of the Work

The discussed input method concepts offer new use cases for (web-) applications if practically applied. A better understanding of the human-machine interaction process is expected. An input method learning performance analysis can influence hardware and software design patterns. The basic question - „Does the input method influence the learning experience?“ - is usable in areas of computer science as well as cognitive sciences.

1.6 Organization of Thesis

The methodology is structured into the following parts.

Chapter 2 describes the necessary background. The input hardware and the application is introduced. Cognitive concepts about the human perception of space are discussed and connected to the application design principles. Chapter 3 describes the methodology of the usability study. This includes the study design, study material, sample, study procedure and information about the data aggregation. The last chapters deal with the results and the findings that conclude this thesis.

Chapter 2

Background

2.1 Building Information Systems

Buildings represent up to 40% of the overall energy usage in many countries [GrHu03]. Considering the global challenges associated with potential resources depletion, energy crises and rising energy costs, the building sector covers an important part in resource optimizing strategies.

Optimizing building performance requires information on building specific energetic and environmental performance. The majority of existing buildings do not provide this information [MaMo08]. So called „intelligent“ or „smart“ buildings often provide the necessary infrastructure, but do not offer appropriate analytical tools. Data is not monitored in comprehensive ways. Data processing is non consistent and separated into multiple, non interacting routines. The full potential that the monitoring infrastructure provides is not used.

Taking the existing building stock into consideration is vital when developing monitoring strategies. In Austria the new construction volume per annum takes less than 1% of the existing building stock's volume [⇒Statistik Austria 2007]. The development of new energy-efficient building technology focuses on integration into existing infrastructure. The process of continuous commissioning of building systems in order to improve building operation can yield savings of an average of over 20 % of total energy cost [BrRa12]. A continuous building performance evaluation is required, because the building environment changes over time. A performance analysis delivers incorrect results if not all important parameters (e.g. faults, user behavior) are taken into account [Mahd09].

2.2 Monitoring System Toolkit

The Monitoring System Toolkit (MOST) is an open source building monitoring framework. It can be categorized as a spatial information system as it implements a similar structure [Fran88].

Figure 2.1 shows the framework architecture that consists of four layers. The

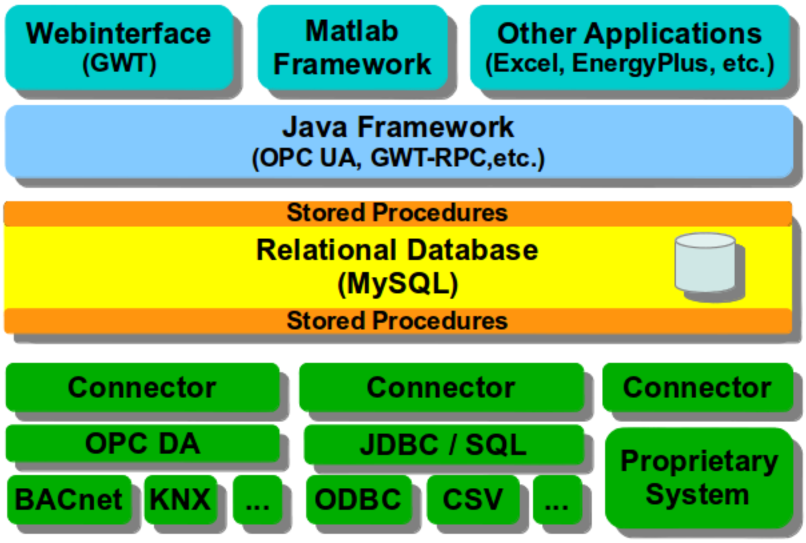


Figure 2.1: Four layer architecture [ZaGl12]

connector layer provides interfaces to various data sources. The database layer implements continuous data storage and data preprocessing functionality, the data abstraction layer offers various software interfaces for data access. The topmost layer offers client side interfaces on various platforms. The infrastructure is applied to a practical context in two buildings in Vienna. One provides certain monitoring infrastructure elements, the other one does not, allowing to test the basic setup as well as the data integration of existing data sources. Data is collected by physical sensors. Some sensor examples are

- occupancy
- electrical power usage
- contact (windows and doors)
- temperature
- air flow.

The framework implements a Matlab client and a web application client. The web application implements support for three input methods and is used by the usability study.

2.2.1 Data Model

The usability study material strongly depends on two concepts: Sensor data is organized by *Datapoints* and *Zones*(Figure 2.2). Each sensor is represented by a unique entity *Datapoint* in the database. Zones organize data points regardless of the physical structure and location. Typically a zone covers a room in a building. The current implementation limits a sensor to one zone. Combinations of zones or rooms (e.g. aggregated energy consumption for a building storey) are enabled by introducing *Subzones*. Each zone consists of multiple children.

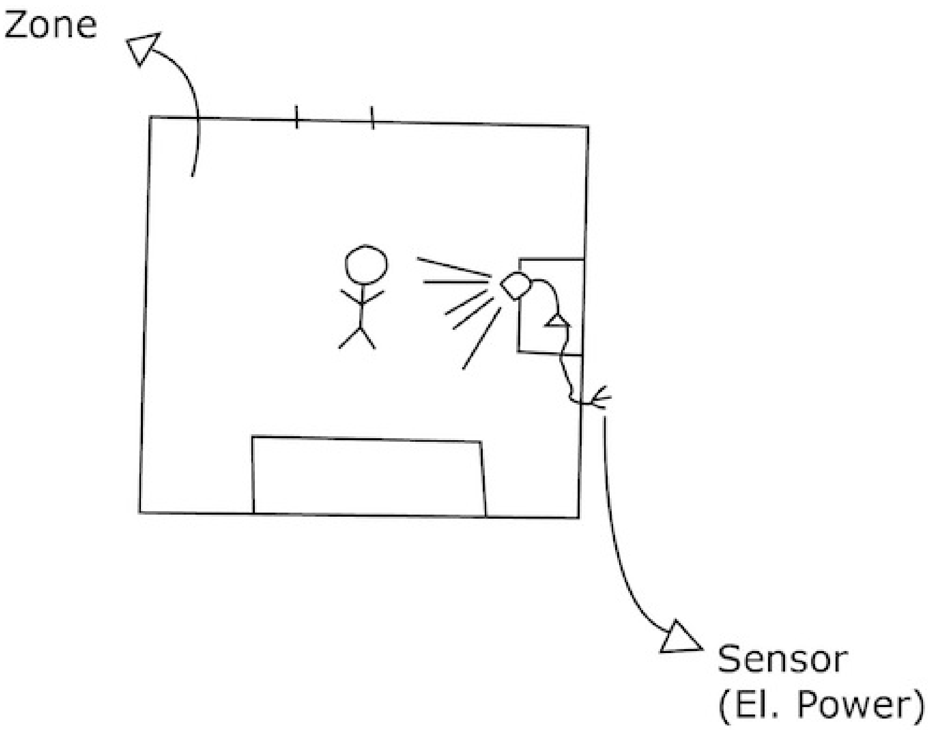


Figure 2.2: Concept of data points and zones

2.3 Input Hardware

The suggested implementation supports Microsoft Windows, Mac OSX and Linux based distributions that offer a version of one of the browsers

- Mozilla Firefox
- Internet Explorer

- Google Chrome
- Opera
- Safari
- Chromium
- Mobile Webkit.

2.3.1 Keyboard and Mouse

The standard input method for personal computers and notebooks is a keyboard in combination with a mouse or touchpad. All current browsers support the conventional keyboard and mouse input, no hardware issues must be considered.

2.3.2 Touch Input

Touch interaction is used in desktop and mobile environments (e.g. tablet computers). Mobile operating systems (e.g. Android) integrate touch support natively. Desktop operating systems need third party drivers to use touch screens. An issue is that generic drivers tend to miss certain features (e.g. multi touch support). Mobile and desktop environments interpret touch input differently. For example, certain gestures are disabled for the browser and directly forwarded to the operating system (e.g. zoom, swipe). This situation is solved by either implementing a browser plugin or integrating libraries that provide a work around. These JavaScript libraries must support desktop and mobile browsers. Figure 2.3 shows a 55 inch touch screen displaying the showcase building monitoring application in Firefox on Linux Mint.

2.3.3 Gesture Input

New generations of human interface devices (HID) implement gesture input. Optical sensors monitor a person's body movement in various spectrums (visible light and infrared). The gestures are recorded and translated into commands. No operating system natively implements gesture input support. Third party libraries are needed. Gesture input is limited to desktop applications due to the size and energy needs of the hardware.

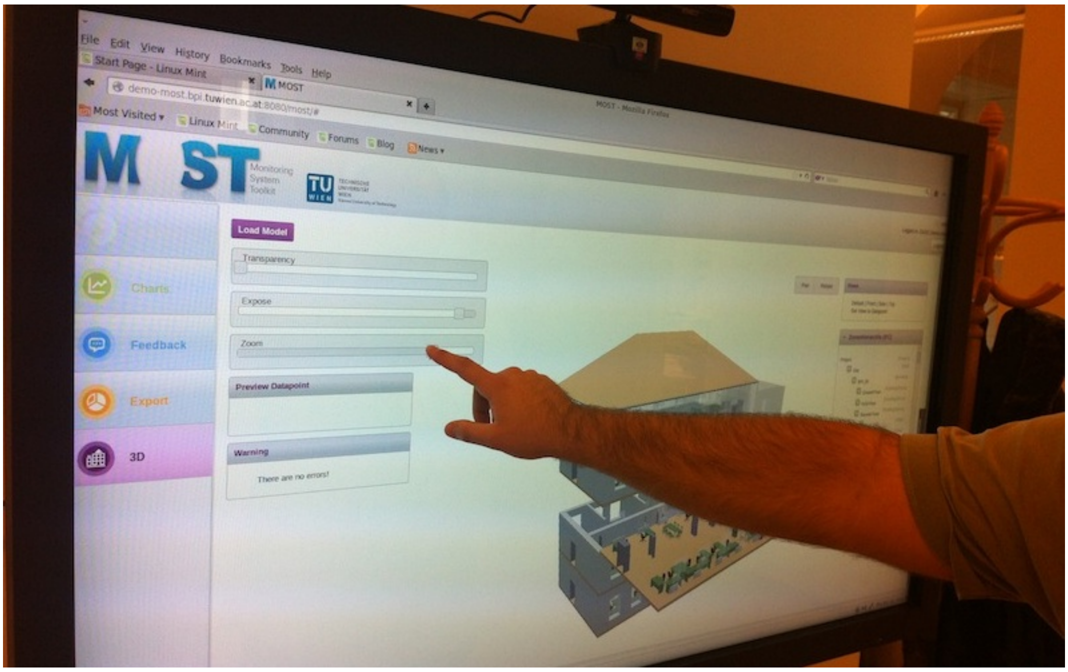


Figure 2.3: MOST showcase application. OS: Linux Mint; Browser: Firefox

Microsoft Kinect

Microsoft produces a gesture HID, the Kinect [⇒Kinect]. The sensor shown in Figure 2.4 costs about 100 Euros and recognizes human body structures from a 40cm distance onward. Initially developed for a gaming console, Microsoft now offers a version for desktop use (Figure 2.4). The open source community [⇒Open Kinect] pushes driver development for various platforms. The libfreenect software package [⇒Libfreenect] contains drivers and various wrappers (e.g. C++, C#, Java, JavaScript). In case the sensor is active an initial body scan recognizes the body structure (head, trunk, upper and lower limps - including fingers). The middleware generates a skeleton model and transforms the incoming gestures into computer interpretable commands. The sensor accuracy allows hand tracking to a level that allows a distinction between a fist and an open hand. The infrared sensor works at 30Hz and supports a resolution of

Figure 2.4: Microsoft Kinect [⇒Kinect]

1200x960 pixels. Due to the limitations of the universal serial bus connection, incoming images are preprocessed. The vision field covers

- 58 degree horizontally
- 45 degree vertically
- 70 degree diagonally,

with a minimum distance of 0.4 meters and a maximum distance of 3.5 meters. The resolution in a distance of 2 meters is 3mm on the x- and y-axis and 1cm on the z-axis [Kram12].

2.4 Navigation Operations

Each input method requires a unique set of input operations. These operations must be platform independent. For example, a mouse implements zooming functionality by rotating the mouse wheel. A touch screen or a gesture input device needs a different definition. Navigation operations are generalized and combined to a set of platform independent input operations. These operations are

- Select/ activate
- Drag
- Pan/rotate
- Zoom

Each application requires the user to select or activate an item at some point. Keyboard interaction is included in this group. Drag operations build the basis for human input in the framework. Drag and pan operations have similar physical movements but are logical distinct. A drag operation always requires a certain target point, a pan operation does not. The drag and pan movement is separated by logic into two modes (e.g. implementation via switch button). The following tables show the results of the navigation operations analysis [GIAp12] and are used to realize input method platform independency.

In Table 2.2 the term „dragging“ is defined as

- Pressing the select mouse button while moving

Input Device	Interaction
Keyboard/ Mouse	Clicking
Touchscreen	Tapping
Gesture Input	Push gesture (the hand is moved in the direction of the sensor, like pushing an imaginary button in front of the user)

Table 2.1: Interaction type: Select/ Activate operation

- Staying in direct physical contact with the touch screen while moving
- Keeping push posture while moving the arm in the desired direction.

Input Device	Interaction
Keyboard/ Mouse	Clicking plus dragging the selected widget to the target point and releasing it
Touchscreen	Tapping plus dragging the selected widget to the target point and releasing it
Gesture Input	Push gesture plus dragging the selected widget to the target point and releasing it

Table 2.2: Interaction type: Drag operation

Input Device	Interaction
Keyboard/ Mouse	Clicking on a map or building model plus holding down the mouse button and moving the cursor
Touchscreen	Single touch: Tapping on a map or building model plus staying in physical contact with the touch screen and moving the cursor
	Multi touch: Tapping with two fingers plus staying in physical contact with the touch screen and moving the cursor
Gesture Input	Push gesture on a map or building model plus keeping the posture and moving the arm

Table 2.3: Interaction type: Pan/ Rotate

Figure 2.5 shows the example application with the applied navigation operation translation. The volunteer is performing a push gesture that is applied to an *Activate* procedure.

Input Device	Interaction
Keyboard/ Mouse	Using the scroll wheel of the mouse or two fingers on a track pad
Touchscreen	Single touch: using a zoom slider
	Multi touch: Placing two fingers on the screen and move them apart or closer
Gesture Input	1. Changing a zoom slider in the user interface 2. Moving the body closer/farther from the device to zoom in or out 3. Move both arms apart/closer

Table 2.4: Interaction type: Zoom

Figure 2.5: Push gesture in example application

2.5 Example Application

The example application is divided into modules. Each module is planned to support a user's concept of space with the aim to simplify the learning process. The primary interaction method between spaces (modules) is drag and drop. This limitation supports the development of spatial metaphors [Kuhn96].

2.5.1 Drag and Drop (DND)

Every user makes mistakes when interacting with an interface, partly because commands are difficult to learn and parts of the interface are inconsistent [KuFr90]. A common UI development problem is to show the set of available operations to a user. Only a small percentage of operations is used because the user [Fran93]:

- never finds out about it
- cannot identify helpful functions
- does not understand how an operation works
- is intimidated by the seeming complexity of the operation

Dix [DixA04] defines three points that influence interface design. An UI should be useful (functionality), usable (it is easy to do things) and used (attractive and available). One conclusion is that an application should prevent a user from making mistakes. The presented solution uses drag and drop functionality in combination with highlighting. Each module defines areas where the user can interact with the system logic. When a user starts dragging a graphical user interface element, the droppable areas that are able to interact with the dragged element are visually highlighted. The application screen turns grey and only certain options are usable. The example in Figure 2.6 shows a trend chart object that is dragged by a user. The chart can be dragged to two areas, where further interaction is possible.

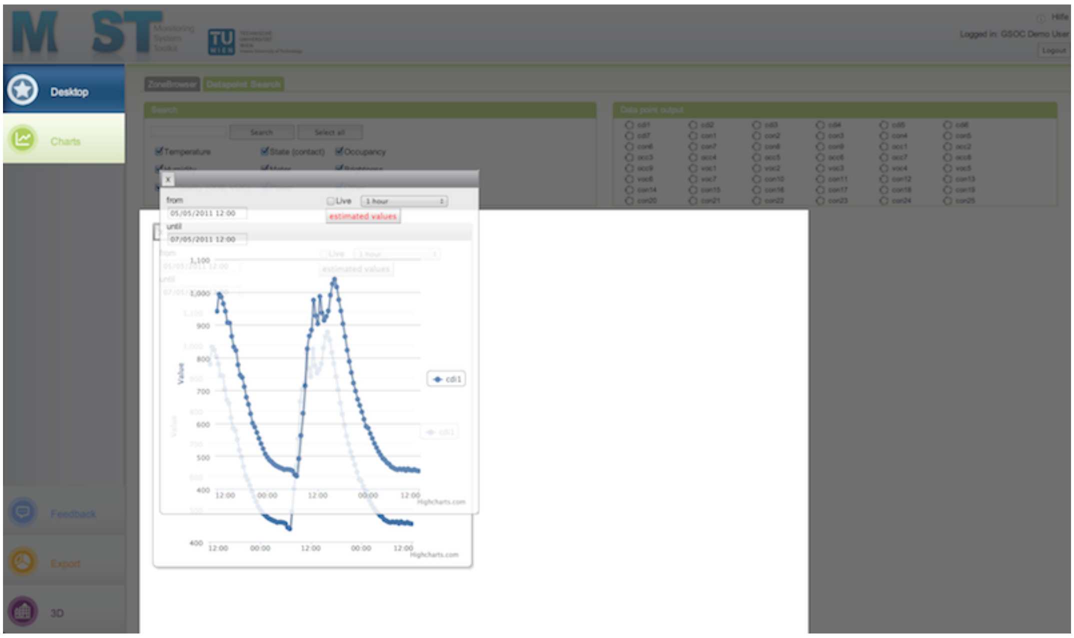


Figure 2.6: Drag and drop with highlighting example

Disabling not supported user interface parts prevents the user from using unsupported operations.

2.5.2 Desktop Module

The application entry point is the desktop module, similar to an OS desktop (Figure 2.7). Information from various other modules (e.g. sensor data and

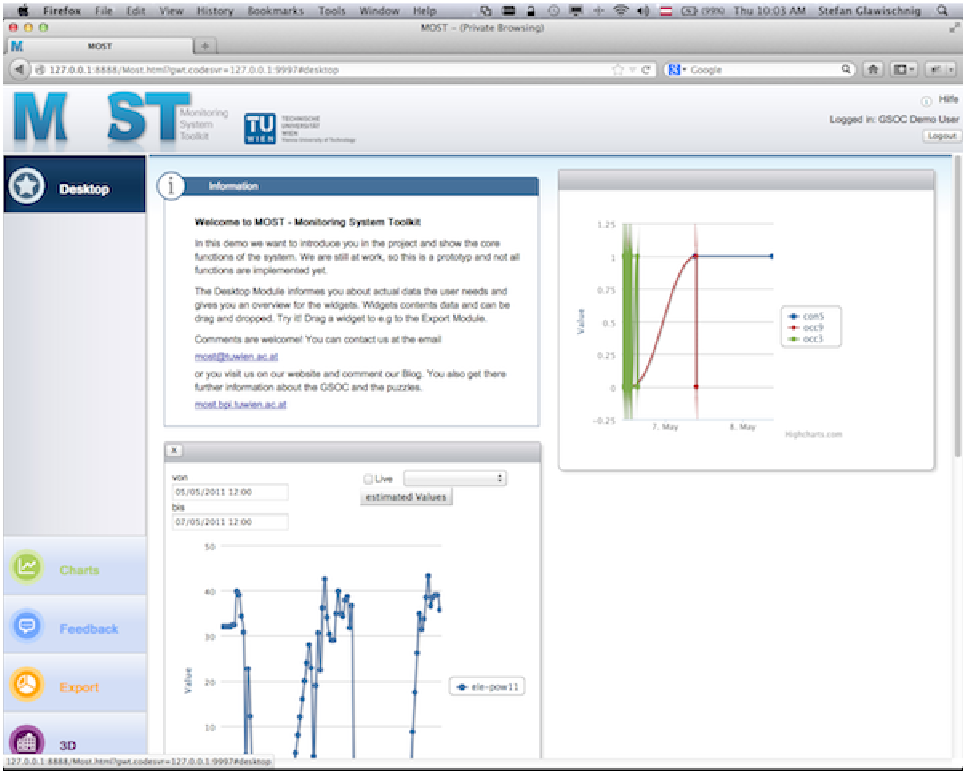


Figure 2.7: Desktop module containing three widgets.

user feedback) is stored in a centralized view port.

2.5.3 Chart Module

The chart module creates trend charts of certain data points for a specific time frame. A data point object is dropped on a specific area in the chart module and an interactive trend chart is generated (Figure 2.8).

2.5.4 Feedback Module

Personal feedback on building performance is highly subjective, but an interesting factor in energy optimization strategies. Heating costs are an example. The user answers questions about air, air movement, air temperature, personal clothing, activity, feeling and choses between predefined values. At the example of the air movement these choices are

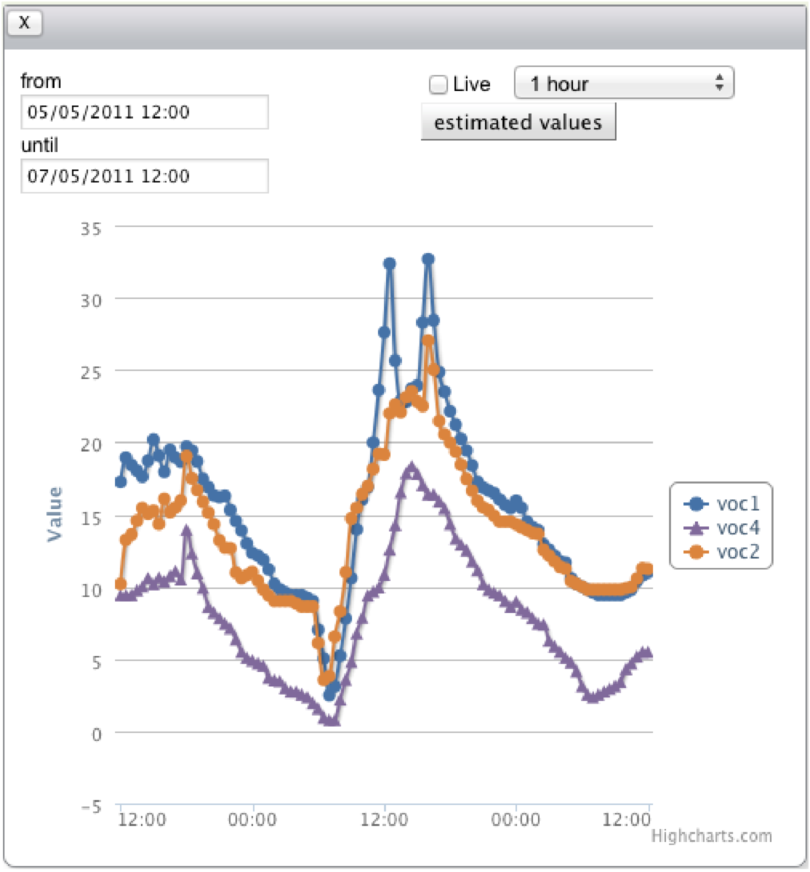


Figure 2.8: Chart widget displaying three sensors.

- very pleasant
- pleasant
- neutral
- unpleasant
- very unpleasant

2.5.5 Export Module

Certain applications depend on historical data but do not have access to web services (e.g. offline applications, student projects). The acquired historical data is exportable into various file formats including SQL and CSV. Sensors are dragged into a data collection widget and the values are exported for a requested timeframe.

2.5.6 3D Building Viewer Module

The 3D module creates a three dimensional, interactive representation of buildings. The viewer application is an enhanced version of the BimSurfer 2012 open source project [⇒BimSurfer]). The initial interface depends on the use of mouse and keyboard. Using the platform independent navigation operations requires the implementation of additional interface elements (zoom slider, pan/rotate switch button).

Default map navigation operations are translated into a three dimensional context to lead the user within a building model. Frank [Fran10] points out that people switch effortless between multiple levels of detail. The current implementation defines only one level of detail. All objects are represented in the same way at all scales. Exposing and transparency functionality are introduced to improve readability. The exposing function translates building storeys along a predefined horizontal axis to isolate a number of objects. A transparency slider changes the opacity of concrete elements (e.g. walls). Figure 2.9 shows the building viewer interface of a university building in Vienna. Sensor representations are bound to physical objects. Occupancy sensors are mapped to chairs, temperature sensors to radiators and electricity power sensors to computers and other devices. Querying the model happens only sequentially, one data point at a time. By limiting the number of combinable queries, errors due to misinterpretation are reduced [EgFr92]. Interface interaction uses either native hardware

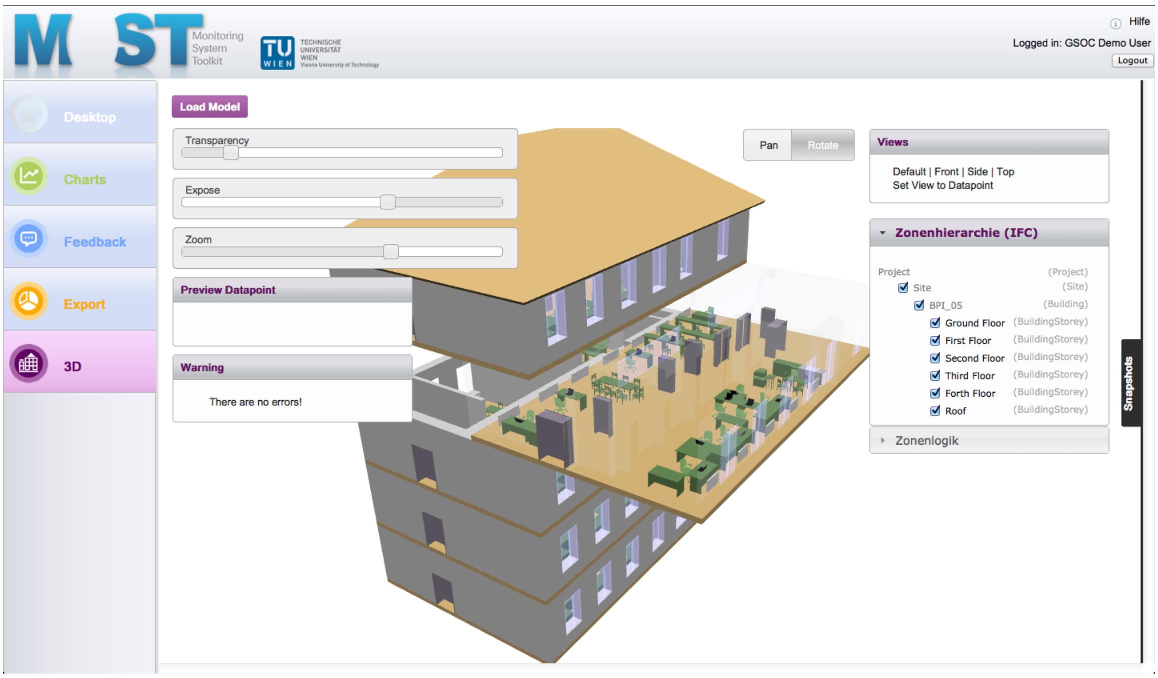


Figure 2.9: Building model with exposed and transparent storey.

functionality (mouse wheel) or the interface elements.

2.6 Human Perception of Space

Every day life is connected to space. Our environment is stored in the mind in the form of cognitive representations. The perceived objects are stored in various accuracies, some very accurate (e.g. *Home*) some fuzzy. These derivations of the infinite complex world are referred to as „Cognitive Maps“ [MoFr05]. The gathered concepts of space influence human behavior in digital environments. For example, on the internet spatial vocabulary is ubiquitous [MaMa98]:

- to *visit* a website
- *follow* a link to a new location
- *move* a file to a folder

The human mind classifies the environment into spaces. Mark [Mark92] distinguishes those spaces into several groups (e.g. *haptic spaces*, that are derived from touching and other bodily interactions). These types of spaces influence the design of spatial information systems, as they define how people intuitively conceptualize space and time [FrEg97].

2.6.1 Image Schemata and Metaphors

People learn about the environment through the senses. Seeing, hearing and feeling the environment forms individual cognitive concepts and categories. Johnson refers to these reoccurring patterns as *image schemata* [John87]. Image schemata can be seen as a generic, abstract term, that helps people to establish a connection between different but similar structured states and operations [RaEg97]. Johnson [John87] defines a number of schemas. An example is a `CONTAINTER` , that resembles a bodily experience. For example, a container could refer to a building. The proposed application includes a schematic approach at various places. Modules resemble containers. The intended reference is: „*Take something and drop it into a target container (module) to trigger some action.*“ Every time the user does a similar action he/she is supported visually by the highlighting functionality. This improves image schema creation and the learning experience.

Metaphors in interface design were established as a powerful tool to organize human-computer interaction (e.g. *DESKTOP* metaphor by Apple Computer [Appl87]). They determine how labor is distributed between a user and a system, what concepts a user has to deal with, and in what terms user and system communicate [KuFr91].

2.6.2 Spatial Concepts

Metaphors are used to derive spatial concepts from the environment. The example

> „*Theories are buildings.*“

maps the source domain „*buildings*“ to the target domain „*theories*“. We live and work in buildings, in a haptic space, while theories are abstract. The contributed work intends to develop a hierarchy of spatial concepts. The size of user interface elements support this process. For example, sensor labels are smaller than charts. A chart *contains* at least one sensor label. By structuring interface elements hierarchical regarding the size and visual appearance, the user is trained to support and understand the „*invisible*“ data structure and the application architecture.

2.6.3 Way-finding

Way-finding defines the process of a person moving through an environment from a source to a target domain. Clues and choices influence this process [RaEg98]. The building viewer serves the purpose to find a way and to identify an object of interest (e.g. room) without the need to physically move to the desired destination. To support the user at developing a cognitive map, the used 3D model is a correct geometric copy of the building it describes. The proportions and shape of objects are preserved. Objects are not added or removed depending on scale and the geometry is not distorted, which are common techniques to implement various levels of detail [FrTi94]. The user chooses which objects are displayed and which are not. The presented approach is suitable for single buildings but not for large scale maps due to the obvious limitations.

2.7 Significance

Allocating the discussed concepts of human spatial recognition to the hypothesis raises one follow up question:

Do input methods that directly interact with virtual spaces improve the learning experience?

In this context the word „*directly*“ refers to following characteristics:

- **Physical contact**
  Keyboard/ Mouse: Specific devices are needed to interact with the application.
  Touchscreen: The user is in direct physical contact with the output device and the application.
  Gesture Input: No physical contact is established. The body movement is recorded and translated into commands.

- **Hardware specifics**
  Keyboard/ Mouse: The mouse specific speed and accuracy influences the usage performance of the application.
  Touchscreen: The touch screen sensitivity holds possible usability issues.
  Gesture Input: Resolution and vision field of the sensor, necessary illumination and simulated mouse parameters are critical.

From the discussed cognitive point of view a touch input device supports the learning experience more than mouse and gesture input device. The user is in direct physical contact with the output device. The performed actions are referring to well known interaction concepts of daily life. For example, the action definitions of „*taking and moving*“ are least abstract for a touch interaction compared to mouse and gesture input. Using the mouse requires a *CLICK* operation as well as an arm movement in a space (table) that is not connected to the virtual space of the application. The brain must make a connection between two spaces before the incoming information is processed.

Chapter 3

Usability Study

3.1 Design

The study evaluates the influence of an input method on the learning performance of an application. The study layout follows an experimental approach. The participants solve computerized exercises and are directly observed. A 6x1-factorial, bivariate randomized test procedure evaluates the learning process. The six independent variables are the possible input method sequences (Table 3.1). Every exercise is repeated with every input method. For example, one participant does three exercises on a touch screen. The exercises then are repeated with keyboard and mouse and gesture input. The exercise content and the testing environment never changes (6x1 factorial). Two dependent variables measure the learning performance:

- *TIME*
- *ERRORS*

TIME is the observed time a participant needs to finish an exercise. *ERRORS* are the usage errors that happen in each exercise cycle (completion of all three exercises). This can be wrong gestures or wrong workflows. The first exercise cycle defines the experimental condition. The two repeating cycles define the control conditions. A comparison of the times and errors of all possible input method combinations should show if an input method is

- faster to learn
- more efficient.

Run 1	Run 2	Run 3
Keyboard/ Mouse	Touchscreen	Gesture Input
Keyboard/ Mouse	Gesture Input	Touchscreen
Touchscreen	Keyboard/ Mouse	Gesture Input
Touchscreen	Gesture Input	Keyboard/ Mouse
Gesture Input	Touchscreen	Keyboard/ Mouse
Gesture Input	Keyboard/ Mouse	Touchscreen

Table 3.1: Usability study: All input method combinations

Time refers to the duration a participant needs to perform each exercise. Logic errors are errors related to input operations (e.g. wrong gesture) and application logic (e.g. wrong workflow). Table 3.2 shows example times for one cycle. The index stands for exercise 1 to 3.

Time 1	Time 2	Time 3	Error 1	Error 2	Error 3
00:01:34	00:03:44	00:02:29	2	3	5

Table 3.2: Usability study: Example values

3.2 Sample

15 male and 15 female students participate in the study. The sample size is 30 (n=30). Sample persons define the characteristics. Table 3.3 shows one conceptional sample person.

Name	R. C.
Profession	Master Student - Architecture
Capacity	Needs sensor data for project work. Uses information screen when going to the administration office. Not familiar with Spatial Information Systems.
Age	25
Sociodemographic characteristics	not married, in a relationship
Other characteristics and skills	Languages: English, German is used to work with computers owns a smartphone
Use cases	Historical data access, large amount of sensors and values. Application: easy to use

Table 3.3: Usability study: example person

The students were recruited during lectures at the Department for Building Physics and Building Ecology and mainly study architecture. The sample person's characteristics are enhanced by three parallelizing conditions:

1. No experiences with gesture input devices.
2. Possession of a smartphone (mobile phone with touch screen).
3. Not familiar with Spatial Information Systems.

Figure 3.1 shows the age distribution of the participants.

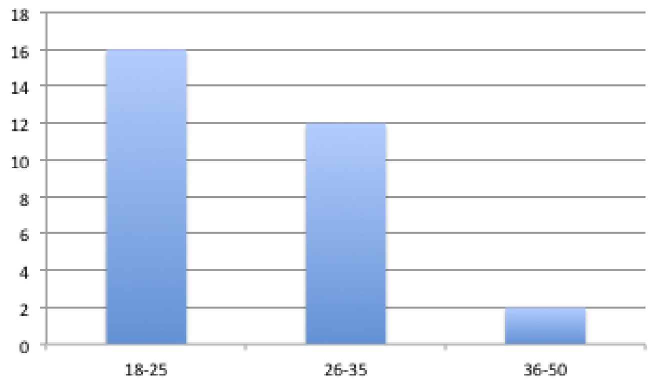


Figure 3.1: Participant age distribution.

3.3 Material

The experiment consists of three exercises that are repeated three times with various input methods. Copies of the exercises and questionnaires are appended (Appendix A). Each exercise focuses on one core question. Test case A deals with the perception of the application logic. Test case B concentrates on the perception and manipulation of virtual representations of parts of the real world (large-scale spaces). Test case C combines application logic and visual representations of large-scale spaces.

3.3.1 Test Case A

The first test case introduces the participant to the application:

- Open a model.
- Search for a specific sensor via the sensor name.

- Get familiar with drag and drop.
- Get familiar with highlighting.
- Create a trend chart for a sensor.
- Get familiar with the chart representation and extract information.
- Use the zoom functionality.

3.3.2 Test Case B

The second test case uses the 3D module to interact with the building model. The exercise is to recreate the scenery defined in Figure 3.2

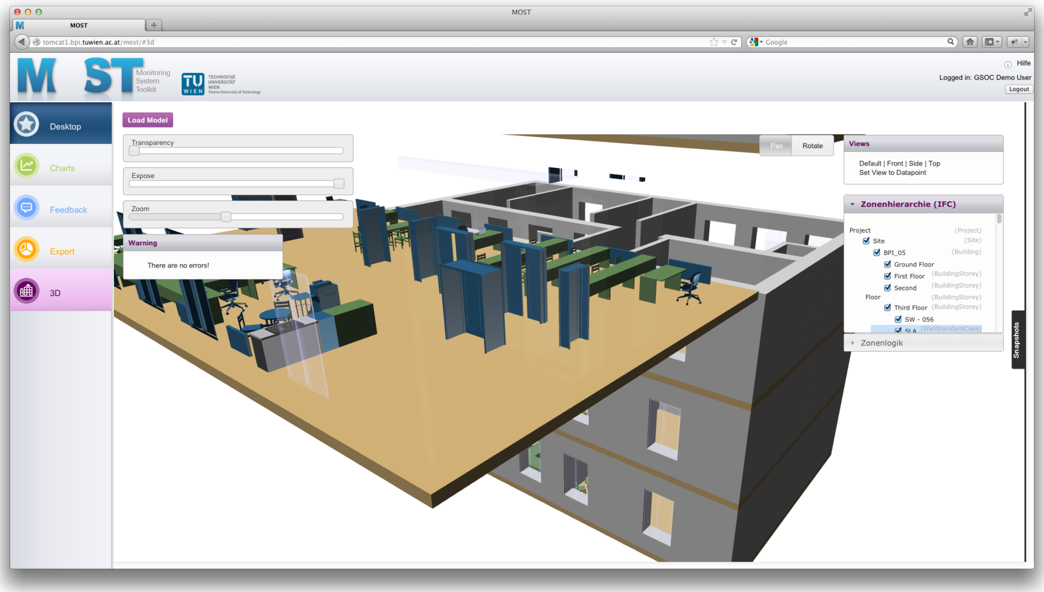


Figure 3.2: Usability study: Exercise two.

This includes following processes:

- Load a 3D model.
- Use the zone hierarchy.
- Use the exposing, zoom and transparency sliders.
- Change the navigation mode (pan or rotate).
- Use the zone logic.
- Recall visual information.

3.3.3 Test Case C

The third test case connects the principles of exercise one and two:

- Open a module.
- Search for a sensor.
- Drag a datapoint from the chart module to the 3D module.
- Interact with the building model.
- Drag a datapoint to the chart module.
- Create a chart.

3.4 Procedure

The study takes place at the Department for Building Physics and Building Ecology at the Technical University of Vienna. The students do the exercises one at a time and are observed by one person. Figure 3.3 shows an outline of the room and the positions of the participants and observers. The numbers show which positions are connected. No cycle was conducted with more than one participant at a time.

The observer is welcoming the student to the experiment and explains the procedure. This includes:

- Introduction to the research.
- Purpose of the experiment.
- Viewing of physical sensors.
- Introduction to the application concepts (not visual).
- Information on the exact procedure.
- Handing the questionnaires.
- Introduction to the input method of the first cycle.

Each input method is introduced right before the participant does the exercises. The observer hands the participant the exercise papers. The participant reads the task assignments and tells the observer when to begin. The observer records the time and errors. No student is familiar with gesture input. The training duration takes about 15 minutes. The student starts with drawing lines in a simple paint program (MS Paint). Then he/ she navigates on a web page and plays a simple flash game.

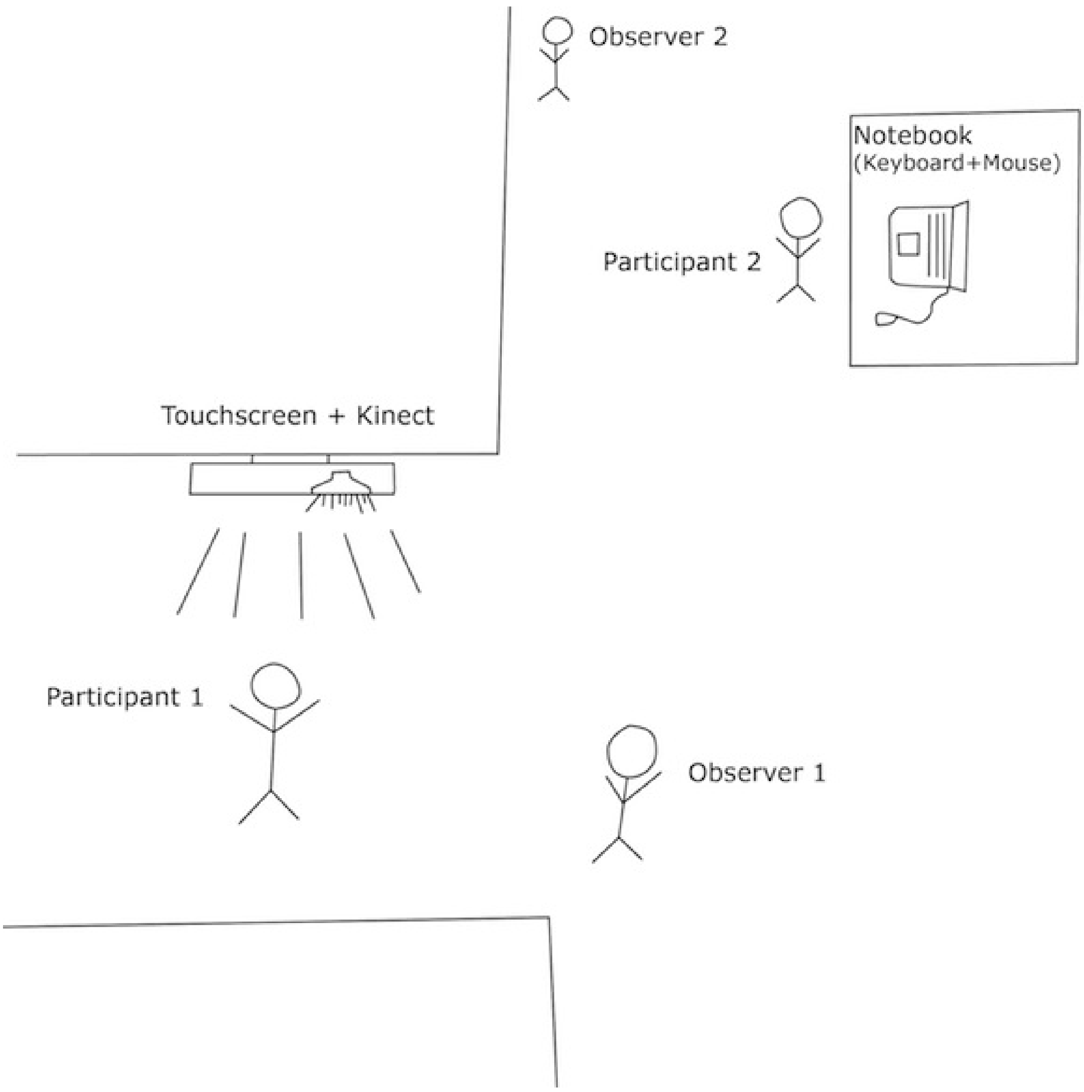


Figure 3.3: Hardware setup.

The application runs on an Apache Tomcat (Tomcat) web server. Participants access the web application via two workstations. The first computer is an Apple notebook, the second one a Mini-PC. Both have similar specifications:

- Quad-Core processor
- 4 gigabytes RAM

- Integrated graphics card

Workstation 1 is connected to the Information Screen (55 inches) and the Kinect. Workstation 2 is connected to mouse and keyboard and a 24 inch screen. The operating system is Linux Mint and the accessing browser Firefox. The testing equipment is divided into three groups

- Mouse and keyboard
- Touch screen
- Microsoft Kinect

All sessions are conducted between noon and afternoon. Electrical light is switched on to guarantee consistent conditions for the Kinect.

Chapter 4

Results

4.1 Data Aggregation

The analysis and results are descriptive. The structure of the participant group and the size of the statistical population limit the results. No explorative or inductive statistical tests are conducted.

Kinect measurements are excluded from the data analysis. A number of students did not manage to handle the physical challenges. Some students who rated the exercise difficulty between *easy* and *medium* did not finish the Kinect exercises because of:

- Physical Exhaustion: The arms got too tired to finish the exercise.
- Coordination Problems: Did not understand how to use the application with a gesture input device.

The time limit to finish an exercise was set to 30 minutes. The remaining measurements delivered no usable results because:

- not enough measurements available (less than fifteen)
- feasible standardization not possible
- number of extreme outliers
- scattering of remaining measurements.

The initial study definition does not change (section 3.1). Even if the exercises are not completed, gesture input influences the learning process.

4.2 Factors

The result factors are basic statistical values. The raw data is collected according to the schema presented in Table 4.1.

Participant #	PC 1 - 3	Touch 1 - 3	PC Error 1 - 3	Touch Error 1 - 3
1	00:00:40	00:01:05	3	4

Table 4.1: Usability study: Example measurement after gesture input exclusion.

One measurement process is divided into two cycles („*CYCLE 1*“ and „*CYCLE 2*“) . *CYCLE 1* starts with keyboard and mouse input and ends with touch input (Table 4.2). *CYCLE 2* starts with touch input and ends with keyboard and mouse input (Table 4.3).

Index	Cycle 1
1	Keyboard and mouse
2	Touch input

Table 4.2: Factors: Cycle 1 sequence.

Index	Cycle 2
1	Touch input
2	Keyboard and mouse

Table 4.3: Factors: Cycle 2 sequence.

Every cycle has two time measurements and two error measurements. Following plots and values are calculated:

- The absolute time differences of *CYCLE 1* and *CYCLE 2* with reversed indexes
  e.g. compare *(Cycle 2[Index 2] - Cycle 2[Index 1])* and *(Cycle 1[Index 1] - Cycle 1[Index 2])*
- Sum, average and standard deviation of time differences between *CYCLE 1* and *CYCLE 2*.
- Time differences between *CYCLE 1* and *CYCLE 2* with matching indexes.
- Comparison of number of errors (*CYCLE 1* and *CYCLE 2* with reversed indexes).
- Comparison of the sum of error differences.

4.3 Learning Progress

The goal of the study is to show if one input method improves the learning progress more than the others. Before that it must be shown that this learning progress exists. Figure 4.1 shows the calculated error differences for one exercise within one cycle. When repeating the exercise 66.6% of the participants made less mistakes. 26.6% of the students got the same results and 6.6% made more mistakes.

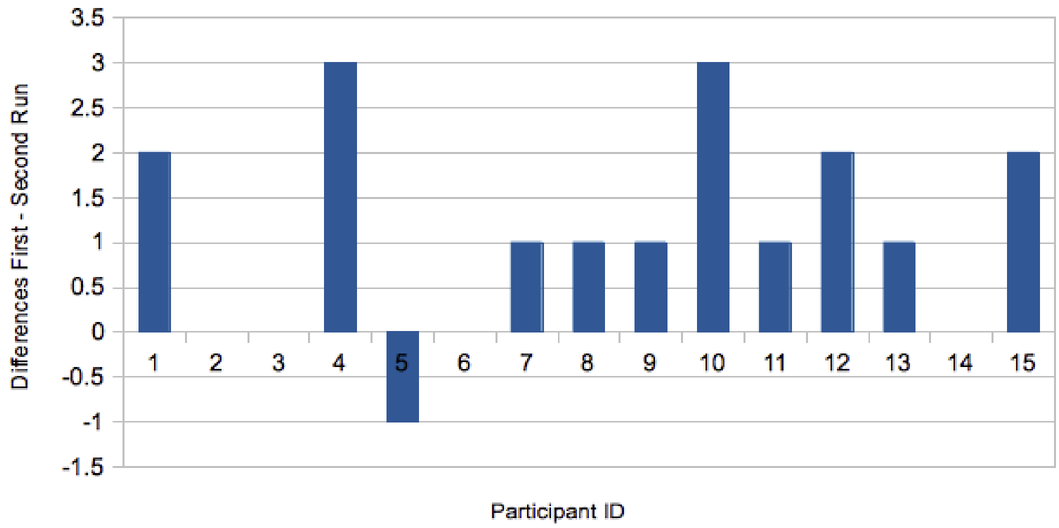


Figure 4.1: Proving a learning process within one cycle.

4.4 Input Methods

The comparison of various input method sequences shows that one input method is preferable. The plots use definitions from Table 4.2 and Table 4.3.

Figure 4.2, Figure 4.3 and Figure 4.4 compare the results of *(CYCLE 1[Index 2] - CYCLE 1[Index 1])* and *(CYCLE 2[Index 1] - CYCLE 2[Index 2])* for each exercise.

(CYCLE 1[Index 2] - CYCLE 1[Index 1]) shows the difference between the fifteen measurements starting with keyboard and mouse input and the fifteen measurements beginning with touch input. Values below zero indicate, that the corresponding input method (touch) takes longer to learn than keyboard and mouse.

The differences between keyboard and mouse and touch input after the second run *(CYCLE 2[Index 1] - CYCLE 2[Index 2])* show, that in 73% of the cases touch input delivers better learning results.

Summarizing the duration differences for each exercise iteration of *CYCLE 1* and *CYCLE 2* (Table 4.4, Table 4.5) shows, that touch input has higher differences when being the initial learning method. The students starting with the touch screen improved faster, than the students starting with keyboard and mouse.

This allows following conclusions:

- Touch input is ineffective: Keyboard and mouse are more performant and faster.
- The learning progress is faster, because touch input offers a more effective learning experience.

Exercise 1	Exercise 2	Exercise 3
00:11:54	00:15:20	00:09:50

Table 4.4: Sum of time differences $\sum(\textit{CYCLE 1} - \textit{CYCLE })$ keyboard and mouse - touch

Exercise 1	Exercise 2	Exercise 3
00:17:35	00:25:57	00:18:56

Table 4.5: Sum of time differences $\sum(\textit{CYCLE 1} - \textit{CYCLE 2})$ touch - keyboard and mouse

Figure 4.5 shows a plot of the errors from all 30 participants in exercise 3. The data range is similar but the data do not correlate. Comparing this plot to the total difference of errors (Table 4.6) shows, that the difference between Table 4.4 and Table 4.5 means that touch input offers a better learning performance.

The total amount of errors show similar structured results. Figure 4.6 shows that touch input is more effective. Less errors are repeated (larger differences) when using touch input before keyboard and mouse.

Table 4.6 shows the total difference of errors during two cycles by comparing the sequences touch - keyboard and mouse and keyboard and mouse - touch. In total nineteen errors less are committed in test case one, 18 errors in test case two and 7 errors in test case three when learning the application by using a touch screen.

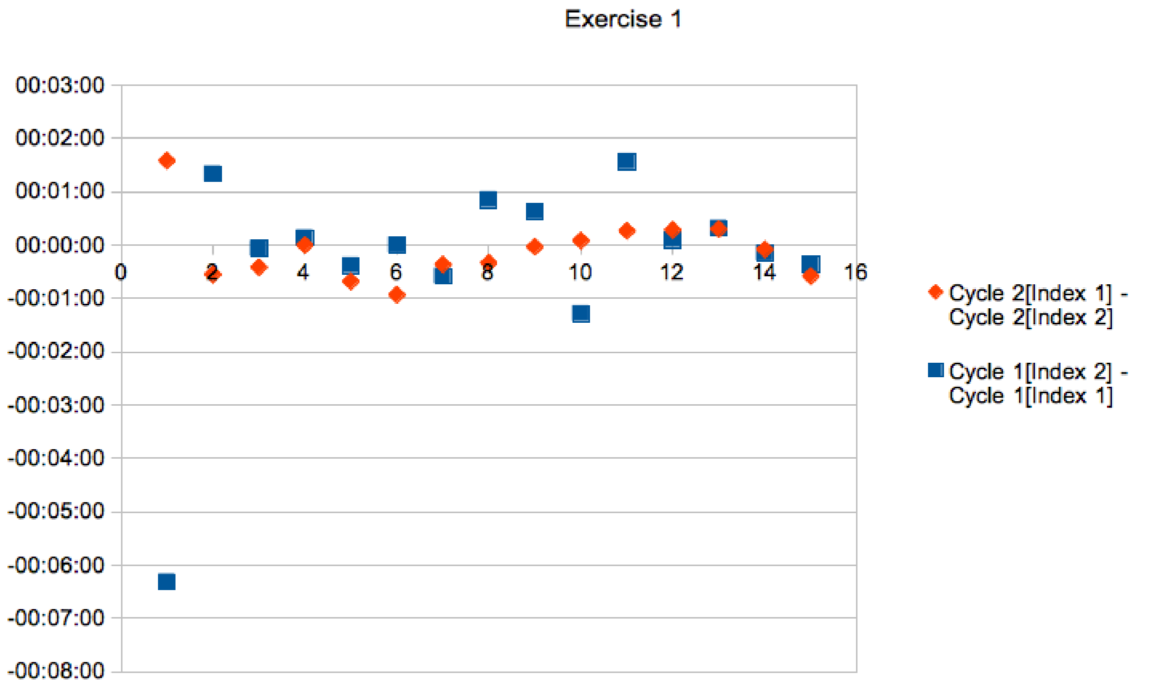


Figure 4.2: Comparison exercise 1 *CYCLE 1 and 2* with consistent input method sequence.

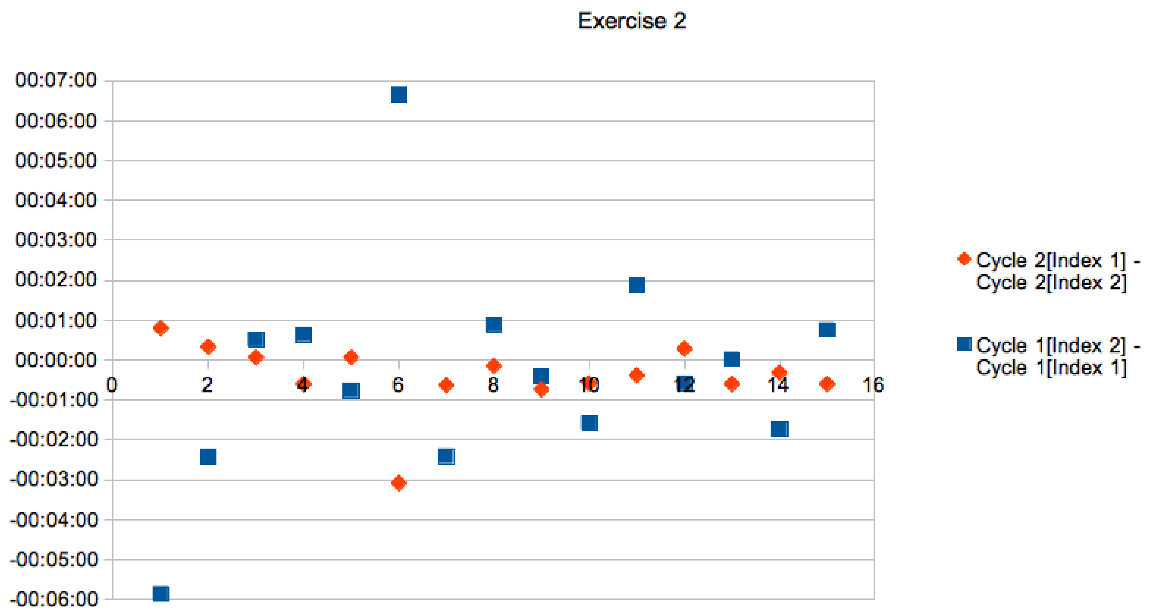


Figure 4.3: Comparison exercise 2 *CYCLE 1 and 2* with consistent input method sequence.

4.5 Practical Results

This work compares three input methods at the example of a spatial information system. The study shows that a touch screen is the best choice for the

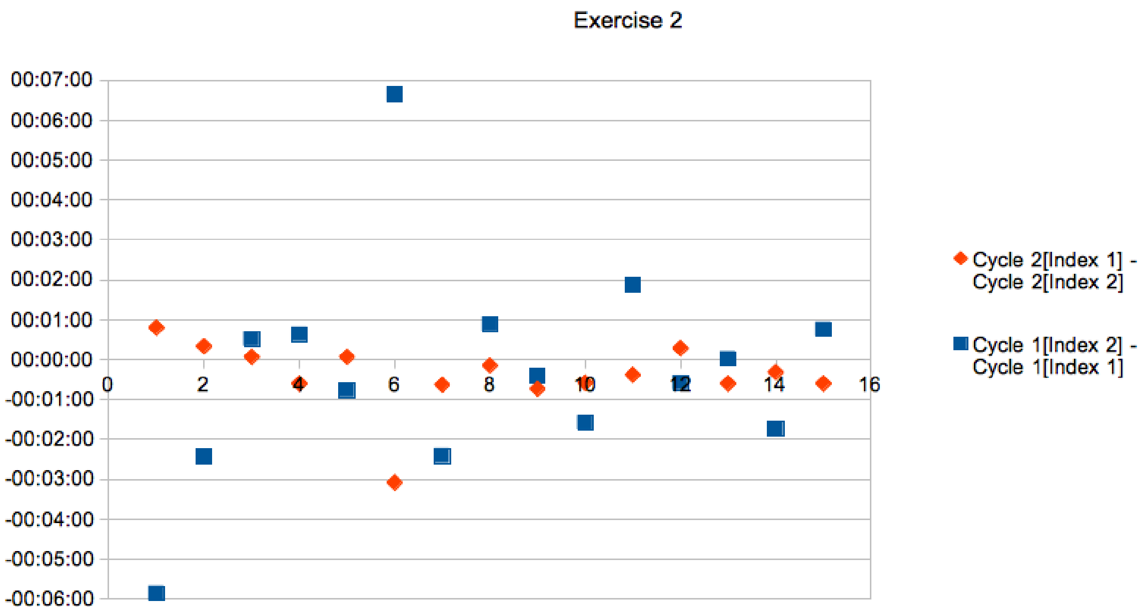


Figure 4.4: Comparison exercise 3 *CYCLE 1 and 2* with consistent input method sequence.

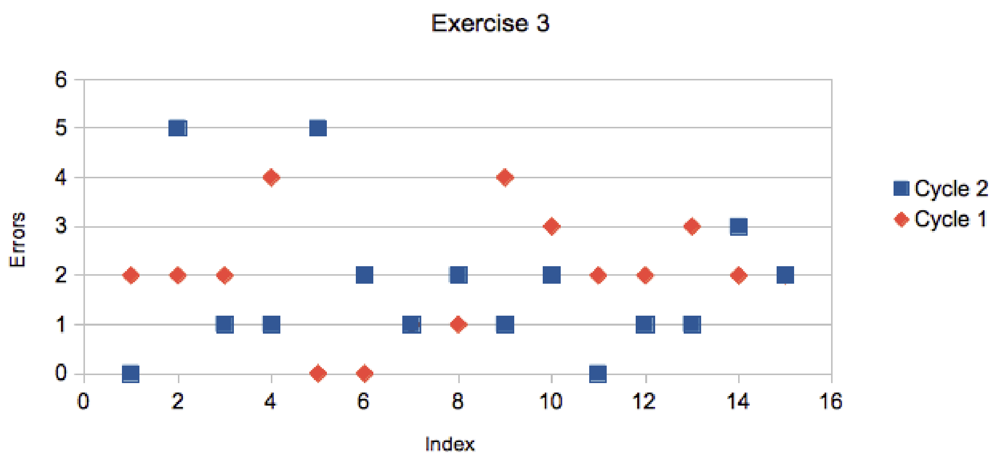


Figure 4.5: Plot of all 30 error measurements from exercise 3.

Exercise 1	Exercise 2	Exercise 3
19	18	7

Table 4.6: Error differences by comparing touch - keyboard and mouse AND keyboard and mouse - touch

implementation of a building information terminal. Gesture input is limited due to hardware based problems (e.g. necessary illumination, range) and software specific issues (e.g. open source drivers not sufficient). Participants did not use gesture input efficiently. Probable causes are

- insufficient hardware support

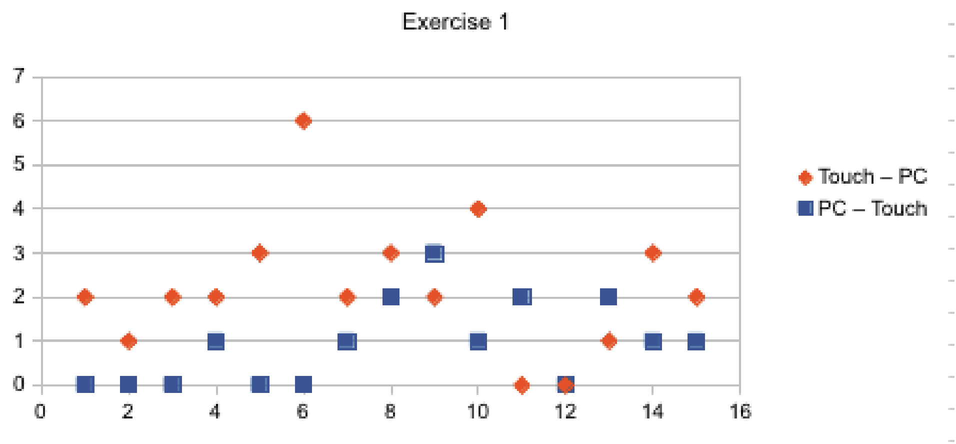


Figure 4.6: Comparison of error differences between *CYCLE 1 and 2*

- user interface design flaws
- driver problems.

Gesture input is not performant in combination with common interface elements (sliders, buttons, lists). The usability study showed that all operations must be mapped to certain gestures, otherwise navigation gets too complicated. All participants found the introduced highlighting concept helpful. The questionnaires show that touch screen and personal computer are chosen to be equally easy to use (each 50 %). When asked for personal preferences, 65 % of the participants enjoyed using the touch screen most.

Drag and drop limits the application functionality. At the current state, multi-level drag and drop is not possible. This includes dragging objects into a submenu structure.

Chapter 5

Discussion

5.1 Summary

This paper documents a usability study that compares the three input methods

- keyboard and mouse
- touch screen
- gesture input.

Thirty participants take part in the experiment and test a building information terminal. They do simple exercises that test the ability to perceive spatial information. The exercises are repeated with each input method. The time each exercise needs and the errors are monitored by an independent observer. The study showed that a touch screen is comparable to keyboard and mouse and delivers better learning results. Gesture input is not included in the final analysis, because some students did not finish the exercises with this input method. This happened for two reasons:

- Physical Exhaustion: The arms got too tired to finish the exercise.
- Coordination Problems: Did not understand how to use the application with a gesture input device.

5.2 Conclusion

The usability study resulted in a number of implications for improving the work done so far. Drag and drop as the primary input method is not sufficient enough at the current state. Dragging and dropping gets tiring when repeating the same (physical) movements frequently. Due to learning performance and participant feedback the touch screen is the current information terminal device of choice. The application design and the input method do influence the learning experience. User interface design principles must not change frequently.

Cognitive sciences in (spatial) applications can improve application usage performance. The contributed work documents an approach to organize a building data collection in a comprehensive way. Due to the scalable design, the concepts are applicable to various project cases. The discussed approaches offer the possibility to build a data collection for real-time monitoring applications as well as long term historical data analysis. Based on the structure of the data abstraction layer, data access is generalized for online and offline applications. Trying to generalize and abstract the framework into coherent components resulted in a number of development guidelines that are reusable in other projects.

The data abstraction framework provides a way to structure building data in a generic manner. Implementing a web based sample application increases the range of possible users. The discussed techniques and concepts aim to provide non-expert users easy and understandable access to building data. The overall goal is to increase awareness of building users to rethink one's own energy saving performance. The best working system cannot replace a rational thinking person.

Building information terminals are used frequently and need a good learning experience. If the application or input method is not easy and understandable a user will not spend time to learn it. The usability study investigates if one input method is preferable at the example of the MOST framework. Touch input delivers the best results. Gesture input was excluded from the statistical analysis, because a number of students could not do the exercises due to physical exhaustion.

I doubt that touch input will replace traditional desktop computers with keyboard and mouse. Similar to gesture input, touch screens require a user to repeat certain physical movements.

5.3 Outlook

The findings of the usability study are combined to a improvement catalogue that will be applied to the current application design. The database model is continuously improved. One current effort is to implement NoSQL support into the existing structure. The measurement data will be moved to a NoSQL cluster, because transaction security is not a vital requirement for sensor measurements. The next step is to lift the developed building performance analysis concepts to an environmental (urban) space. One interesting point is to apply the presented ideas and findings to a larger scale. This step requires an combination of GIS and custom building data collections and will be dealt with in my Ph.D. thesis. The discussed study is a first step to compare the learning effects of various input methods. Gesture input support will be improved to be included into the analysis. If this is working, a statistical model can be developed to test if the results are inductive.

Appendix A

Usability Study

Datum:
Department of Building Physics and
Building Ecology
Vienna University of Technology,
www.bpi.tuwien.ac.at

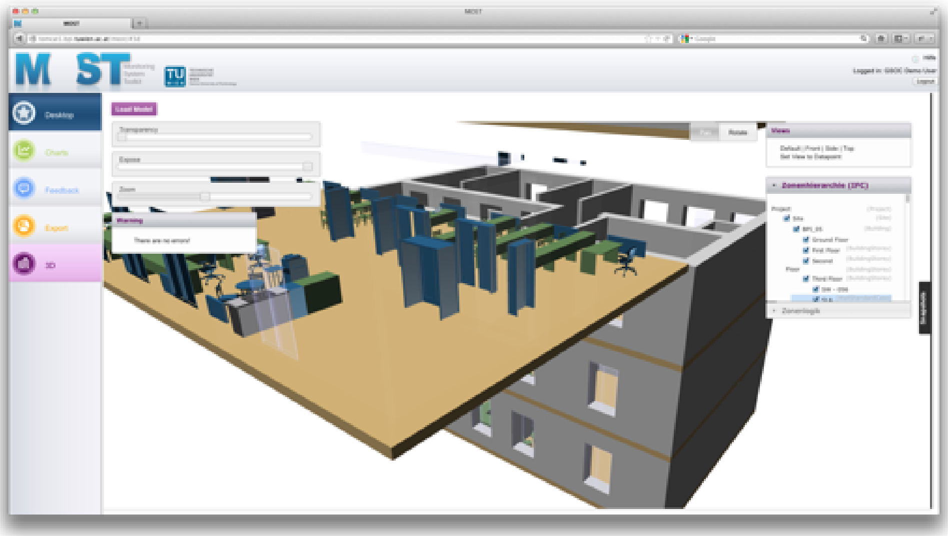
Exercise 1

1. Log into the application. (Login)
2. Move to the Chart Module (Charts).
3. Search for the data point **voc5**.
4. Take the data point and drag it to the white/ highlighted field.
5. Take a look at the generated chart window.
6. Zoom in to the data point with the highest value.
7. Tell the value to your supervisor.

Datum:
Department of Building Physics and
Building Ecology
Vienna University of Technology,
www.bpi.tuwien.ac.at

Exercise 2

1. Move to the 3D Module.
2. Look at the following image and try to reconstruct the scenery. Tell your supervisor as soon as you are finished.



Datum:
Department of Building Physics and
Building Ecology
Vienna University of Technology,
www.bpi.tuwien.ac.at

Exercise 3

1. Move to the chart module.
2. Search for the data point **con5**.
3. Take the data point, open the 3D Module and drop it onto the building model.
4. After the animation
   a. Take the appearing item with the title **con5,** open the chart module and drag it to the white field.
   b. Drop the item at one of the white fields inside the chart module
   c. Close the diagram.

Background Questionnaire

Profession:
O Scholar
O Student of ______________________
O working
O ________________

Age:
O 18-25
O 26-35
O 36-50
O 50-75
O > 75

Do you own a smartphone?

O Yes O No

If yes, which one? ______________________________

Do you own a Xbox Kinect?

O Yes O No

If yes: For how long already? ______________________________

What do you think a **Building Monitoring System** is?

____

____

____

____

Conclusion

How do you like the Website (Austrian school marks 1-best, 5-worst)?

1 2 3 4 5

Difficulty of exercises:

Exercise 1:

O Easy O Medium O Hard

Exercise 2:

O Easy O Medium O Hard

Exercise 3:

O Easy O Medium O Hard

How do you like the highlighting functionality?

1 2 3 4 5

Is the highlighting functionality helpful?

O Yes O No

How do you like the Drag and Drop functionality?

1 2 3 4 5

Which input method is the simplest to use?

O Mouse/Keyboard

O Touch Screen

O Kinect

Which input method provides most fun?

O Mouse/Keyboard

O Touch Screen

O Kinect

Do you have any amendments or suggestions?

____

____

____

List of Figures

List of Tables

Bibliography

Books

[Appl87] Apple Computer, Inc., *Human Interface Guidelines: The Apple Desktop Interface.*, Reading, MA: Addision-Wesley, 1987

[DixA04] Dix A. J., Finlay J. E., *Human-Computer Interaction*, 3. Auflage, Essex: Pearson Education Limited, 2004

Articles

[BrRa12] Bruton, K., Raftery, P., Aughney N., Keane M. M., OÕ-Sullivan D., *Development of an Automated Fault Detection and Diagnosis tool for AHUÕs*, 23-26. October, International Conference for Enhanced Building Operations ICEBO 2012, Manchester, United Kingdom, 2012

[EgFr92] Egenhofer, M. J., Frank, A. U., *User Interfaces for Spatial Information Systems: Manipulating the Graphical Representation*, Geologisches Jahrbuch, edited by R. Vinken, 59-69, 1992

[Fran88] Frank, A. U., *Requirements for Database Managements Systems for Large Spatial Databases*, Geologisches Jahrbuch, A 104: 75 - 96; Hannover, FRG, 1988

[Fran93] Frank, A. U., *The Use of Geographical Information Systems: The User Interface Is the System*, In Human Factors in Geographical Information Systems, edited by David Medyckyj-Scott and Hilary Hearnshaw, 3-14. London: Belhaven Press, 1993

[Fran10] Frank, A. U., *How do People Think About Space - Position Paper*, Schloss Dagstuhl Seminars, Dagstuhl, Germany, 28.03.2010 - 01.04.2010, In: „Seminar 10131 Spatial Representation and Reasoning in Language: Ontologies and Logics of Space“, J. Bateman, A. Cohn, J. Pustejovsky (eds.), (2010), pp.5, 2010

[FrEg97] Freundschuh, S., Egenhofer, M. J., *Human Conceptions of Spaces: Implications for Geographic Information Systems*, Transactions in GIS 2 (4): 361-375, 1997

[FrTi94] Frank, A. U., Timpf, S., *Multiple Representations for Cartographic Objects in a Multi-Scale-Tree - An Intelligent Graphical Zoom*, In Computers and Graphics; Special Issue on Modelling and Visualization of Spatial Data in GIS, 18 (6), pp: 823-829., 1994

[GlAp12] Glawischnig S., Appel R., Zach R., Mahdavi A., *Recent advances in the development of MOST: an open-source, vendor and technology independent toolkit for building monitoring, data preprocessing and visualization*, 23-26. October, International Conference for Enhanced Building Operations ICEBO 2012, Manchester, United Kingdom, 2012

[GrHu03] Graubner, C., Hüske, K., *Nachhaltigkeit im Bauwesen*, Ernst und Sohn, ISBN 3-433-01512-0, 2003

[John87] Johnson, M., *The Body in the Mind: The Bodily Basis of Meaning, Imagination and Reason*, The University of Chicago Press, Chicago, 1987

[Kram12] Kramer J., *Hacking the Kinect*, New York: Springer Science and Business Media, 2012

[Kuhn96] Kuhn, W., *Handling Data Spatially: Spatializing User Interfaces*, In Proceedings of 7th International Symposium on Spatial Data Handling, SDHÕ96, Advances in GIS Research II, (Kraak, M.- J., Molenaar, M., eds.), in Delft, The Netherlands (August 12-16, 1996), Published by IGU, Vol. 2, pp: 13B.1 - 13B.23, 1996

[KuFr90] Kuhn, W., Frank, A. U., *Human Interaction with GIS/LIS: Editing Geometric Models*, In Proceedings of 19th

Congress of International Federation of Surveyors, 10.-19. June 1990, in Helsinki, Finland, pp: 420-434, 1990

[KuFr91] Kuhn, W., Frank, A.U., *A Formalization of Metaphors and Image-Schemas in User Interfaces*, In Cognitive and Linguistic Aspects of Geographic Space. (Mark, D.M.,and Frank, A.U., eds.), NATO ASI Series, Dordrecht, The Netherlands, Kluwer Academic Publishers, pp: 419-434, 1991

[Mahd09] Mahdavi A., *Patterns and Implications of User Control Actions in Buildings.*, Journal Indoor and Built Environment, 18, 5; S. 440 -?? 446., 2009

[MaMa98] Maglio, P. P., Matlock, T., *Metaphors we surf the web by*, Workshop on Personalized and Social Navigation in Information Space, 1998

[MaMo08] Mahdavi A., Mohammadi A., Kabir E., Lambeva L., *Shading and Lighting Operation in office Buildings in Austria: A Study of User Control Behavior*, Building Simulation, Volume 1 Number 2 June 2008 (2007), S. 111 – 117, 2008

[Mark92] Mark, D., *Spatial metaphors for human-computer interaction*, In Bresnahan P, Corwin E and Cowen D (eds) „Proceedings Fifth International Symposium on Spatial Data Handling, Charleston, SC“: 104-112, 1992

[MoFr05] Motello, D. R., Freudenschuh, S., *Cognition of Geographic Information*, In R. B. McMaster and E. L. Usery (Eds.), A research agenda for geographic information science (pp. 61-91). Boca Raton, FL: CRC Press, 2005

[RaEg97] Raubal, M., Egenhofer, M. J., Pfoser, D., Tryfona, N., *Structuring Space with Image Schemata: Wayfinding in Airports as a Case Study*, In Proceedings of Spatial Information Theory - A Theoretical Basis for GIS (International Conference COSIT'97), (Hirtle, S., and Frank, A., eds.), in Berlin-Heidelberg, Published by Springer-Verlag, Lecture Notes in Computer Science Vol.1329, Vol. 1329, pp: 85-102., 1997

[RaEg98] Raubal, M., Egenhofer, M. J., *Comparing the Complexity of Wayfinding Tasks in Built Environments*, Environment and Planning B, 25 (6):895-913, 1998

[ZaGl12] Zach R., Glawischnig S., Hönisch M., Appel R., Mahdavi A., *MOST: An open- source, vendor and technology independent toolkit for building monitoring, data preprocessing, and visualization*, 25-27 July, European Conference of Product and Process Modelling (ECPPM 2012), Reykjavik, Island, 2012

Internet

[⇒BimSurfer] Bim Surfer, December 2012, http://bimsurfer.org

[⇒Kinect] Microsoft Kinect, November 2012, http://www.microsoft.com/en-us/kinectforwindows/

[⇒Libfreenect] Open Kinect Libfreenect Project, November 2012, https://github.com/OpenKinect/libfreenect

[⇒Most] Monitoring System Toolkit, December 2012, most.bpi.tuwien.ac.at

[⇒Open Kinect] Open Kinect, November 2012, http://openkinect.org/wiki/Main_Page

[⇒Statistik Austria 2007] Bestand an Gebäuden und Wohnungen, November 2012, www.statistik.at/web_de/statistiken/wohnen_und_gebaeude/bestand_an_gebaeuden_und_wohnungen/index.html