Institute of Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diploma Thesis Nr. 3552

Using Narration Networks to Model Distributed Tangible Systems for Cultural Heritage Sites

Robin Goldberg

Course of Study: Software Engineering

Examiner: Prof. Dr. Albrecht Schmidt

Supervisor: Dipl.-Medieninf., M.Sc. Tilman Dingler

Commenced: June 3, 2013

Completed: December 3, 2013

CR-Classification: H.5.2

Abstract

The goal of the EU meSch project is bringing back the physical experience to cultural heritage sites using tangible interactive systems connecting the experience on site with digital information in novel ways. To realize this, many distributed sensors, presentation devices and tangible objects are needed that are able to work together to create the overall experience for the visitor. Furthermore, the system should support authoring done by curators through creating digital narratives as well as online exploration of personal visits to discover more information. These requirements draw the needs for a common software architecture for on-site systems that supports fast creation of detached prototypes as well as embedded use in a bigger software life cycle. As a challenge for integration the distributed devices might use different platforms and communication technology. They must be easily maintainable and there must be a fixed set of interfaces for integration within a larger environment, getting predefined digital narratives as input and outputing a log of the visitor's interactions. This work should give an outline of the requirements and the integration of the in-place system with the whole meSch server architecture. It focuses on using narration networks to model the interaction and facilitating these networks as an exchange format between the authoring environment and the in-place system. It provides sample prototype implementations for distributed tangible systems in cultural heritage sites controlled by predefined narration networks and evaluates their use for the meSch project.

Contents

1 Introduction 9
1.1 Related work . . . 9
1.2 The meSch project . . . 10
1.3 New forms of interactions . . . 12

2 Requirements 15
2.1 Roles . . . 15
2.2 Design for interactions in cultural heritage sites . . . 16
2.3 The meSch cycle . . . 18
2.4 User stories . . . 19
2.5 Key challenges . . . 21
2.6 On maintenance . . . 21

3 System Overview and Context 23
3.1 Distributed, multi-platform computing: a look at loose coupling . . . 23
3.2 Placing the system in a bigger context: interfaces . . . 26

4 Narration Networks 29
4.1 Building blocks . . . 29
4.2 Navigation and simulation . . . 31
4.3 From sensors to context . . . 32
4.4 Including content . . . 33
4.5 File format . . . 34
4.6 Prescribing Narration Networks with templates . . . 36
4.7 Triggers and content types . . . 39

5 The Companion Novel, a tangible Prototype 41
5.1 The concept: a story of evolution . . . 41
5.2 Making it tangible . . . 43
5.3 Software design . . . 46
5.4 Implementation . . . 48
5.5 End-2-End scenario . . . 50

6 Evaluation 55
6.1 Expert evaluation . . . 55
6.2 Field-test the prototype . . . 58
6.3 Create sample Narration Networks for other concepts . . . 60

6.4 Supporting the design of templates . . . 65

7 Conclusion and Future Work 67
7.1 Narration Networks revised . . . 67
7.2 On the prototype . . . 68
7.3 Future work . . . 69
7.4 Acknowledgements . . . 71

A Appendix 73
A.1 Python example code . . . 73

Bibliography 75

List of Figures

1.1 The meSch cycle [PCD+13] 11
1.2 GUI and TUI interaction model 13

2.1 Categories of meSch concepts 18

3.1 meSch server components [Ven] 24
3.2 XML schema description for log messages 27

4.1 Narration Network building blocks 30
4.2 Navigation example for the BT prototype network 31
4.3 Basic XML schema structure for Narration Networks 35
4.4 Detailed structure for content nodes 36
4.5 Basic XML schema structure for template files 37
4.6 Detailed structure for content templates 38

5.1 Storyboard for the first prototype concept 42
5.2 Sketch for the extended concept 43
5.3 Technology embedded in the book 44
5.4 Book page for the "Favourite Spots" perspective 45
5.5 Prototype software class diagram 47
5.6 Prototype software sequence diagram 48
5.7 Example website with map of a visit 54

6.1 Pictures showing the different parts of the cemetery 59

List of Tables

2.1 meSch user roles 16
2.2 meSch user stories 20

4.1 Content entry examples 34
4.2 List of possible triggers 40
4.3 List of possible content types 40

List of Listings

5.1 Example template file . . . 51
5.2 Part of an example Narration Network . . . 52
5.3 Example visiting log . . . 53

6.1 Technical log file data . . . 58
6.2 Jewellery sample network . . . 61
6.3 AR Goggles sample network . . . 62
6.4 Projective Torch sample network . . . 63
6.5 Bird Shadows sample network . . . 64
6.6 Group Kit sample network . . . 65

A.1 Navigator code to select nodes and maintain node sets . . . 73
A.2 XMLHelper code to import narration networks . . . 74
A.3 MeSchNode code . . . 74

1 Introduction

How can we put the physical element back in the centre of cultural heritage experience? In order to answer this question the EU project meSch tries to explore new ways of creating tangible interactions for cultural heritage sites. One of the greatest challenges in the whole process is to enable museum curators and staff to create their own digital narratives and deploy them on their site. This draws the demand to find new ways of abstracting from technical terminology as well as create easy to use software and hardware that can be produced and installed by everyone with a basic understanding of technology. A variety of possible sites and platforms and a wide range of environments and conditions need to be taken into account when creating tangible, smart objects for cultural heritage sites. Common ground seems to be hard to achieve but allows the creation of easily combinable and interchangeable templates and designs for creating a tailored experience. Three main components are needed to get things started: (1) A hardware platform that supports messaging, automatic connection and self-setup qualities, (2) a way to describe the tangible experience for both the author and the technical expert monitoring the installation, and (3) the system itself running on the site. The third component lays out the scope of this work. After giving an introduction about the goals and construction of the meSch project and some background for new forms of interactions used in meSch in this chapter, the following one analyses the requirements for an integrated end to end system that serves all purposes of meSch. In Chapter 3 a picture of the system architecture is drawn to describe the setting a common language lives in. Chapter 4 picks up the idea of Narration Networks as proposed by Petrelli and Not and applies it to the meSch scenario. Prototype implementations are shown in Chapter 5 while Chapter 6 contains some more possible applications for Narration Networks for different meSch concepts. Conclusions and some thoughts about the ongoing work in meSch close this thesis in Chapter 7.

1.1 Related work

The broad, interdisciplinary approach adopted in meSch is influenced by work from diverse sources. Work on tangible interaction is presented in 1.3 and work on interactions in museums along with the requirements in 2.2. But there is also some influential work that has to be mentioned in the context of this research. The idea of using networks to organize content is mainly drawn from the work on personalization in the museum context done with the HyperAudio experience by Petrelli and Not [PN05]. It was a location based system designed to deliver personalized, natural language information to a visitor in a museum. The content was organized in a complex network to allow picking of audio snippets according to the visitors preferences and the content he had heard before and seamlessly blend these snippets together. Further insights into personalization for museums contents can be found in the work of Ardissono et al that provides a broad overview over existing solutions and categorizes them as well as pointing out problems and future challenges [AKP12]. The foundations for the notion of context and abstracting sensor data to context information can be found in Schmidt seminal work on ubiquitous computing [Sch03]. His research lays down the basic understanding of context in the technical world and provides systematic approaches to use context in an interactive environment. In understanding digital technology in museums beyond audio guides and information stations the work of Ciolfi et al proved to be very useful. They looked at design methods and requirements and tested prototypes with visitors ([CBF01], [CBF08]), including works on visitor participation ([CM12]) and feedback ([CBF08]). When trying to engage with actual visitors there can be a gap between behaviour in lab situations and the behaviour on the actual site which is addressed by the work of Hornecker and Nicol that provide good insights about visitors engagement in a museum context [HN12]. Finally, the ideas for software engineering aspects are mainly drawn from the work of Hohpe and Woolf [HW03]. Although focused on application integration in an enterprise context their basic assumptions and statements about the concept of loose coupling is widely adopted in this work as well as some of their pattern as the focus of this work is to fit in a system between a predefined server environment on one side and a sensing and actuating platform on the other side and thus has to deal with application integration issues.

1.2 The meSch project

The aim of the meSch project (Material Encounters with digital Cultural Heritage) is to find new ways to bring the wealth of digital cultural heritage information created over the last decades to visitors in an engaging and fun way. This should be done by tangible interfaces to bring the materiality back into the visiting experience. To achieve this, meSch uses a co-design approach where curators, artists and designers work together to create adaptive exhibits and devices for interactive experiences. The overall meSch idea is depicted in Figure 1.1. Curators can collect data from heritage databases and create digital narratives enriched with this heritage data. The meSch infrastructure helps the curator to deploy the created narrative in their museum using smart objects and replicas that fit the created narrative as well as the museum it is deployed in. Feedback from visitors interacting with these objects can be used to tailor new narratives or gather new information supported by automatic recommendations.

Following a 4 year timetable the twelve meSch project partners focus on different aspects of the whole project depending on their area of expertise. This enables to build the meSch system iteratively and progressively through different cycles of prototype development and testing, thus it needs regular communication and negotiation about designs and interfaces. The work presented in this thesis has been carried out in the early stages of the project. In the first year the focus has been on the exploration of ideas, concept creation, technology evaluation and architecture for the meSch server backend system. Whilst there is an exhaustive architecture for the meSch server part including integration of several cultural heritage databases, recommender engine and portal technology for user interaction on the one hand and a software platform to semi-automatically integrate sensor and actuator technology on the other, there is still a

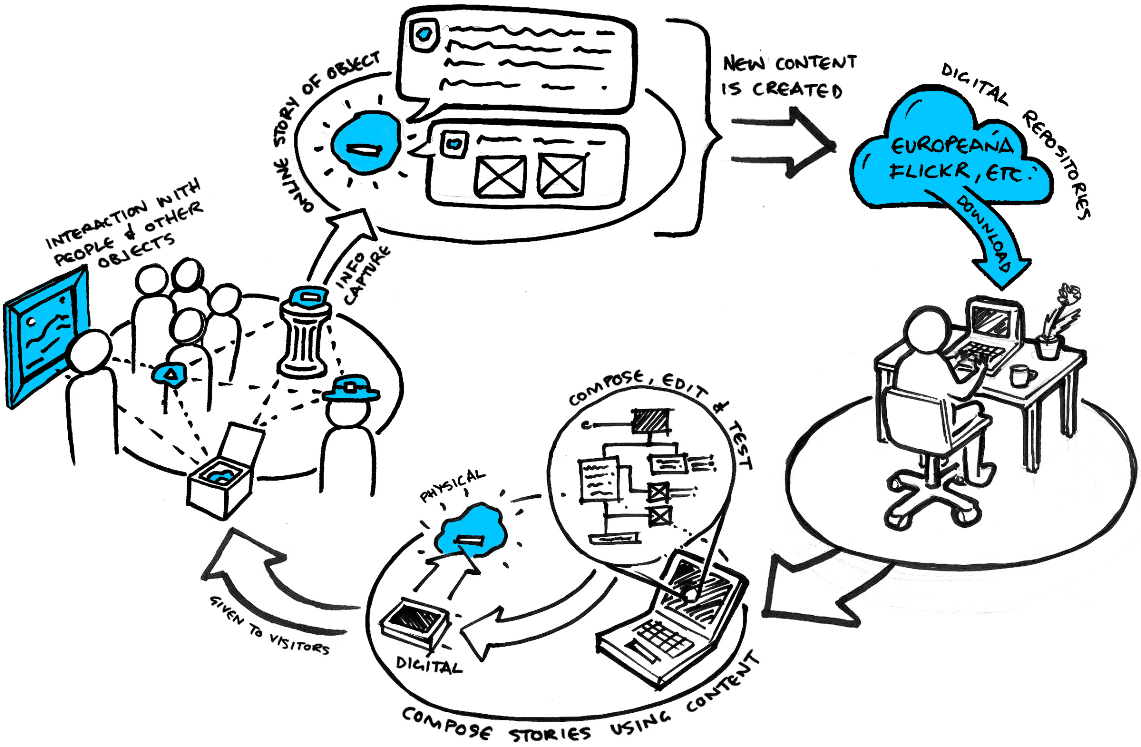

Figure 1.1: The meSch cycle [PCD+13]

gap between the meSch server environment and the real on-site experience. This gap must be crossed in two ways: How does the authored content relate to the on-site experience and how does on-site collected data find its way back to the server. The focus of this work is to fill this gap by proposing a software design that covers the whole process and implementing parts of it in an interactive prototype. As neither the meSch server nor the sensor and actuator platform is yet available, the implementation does not contain any integration with them and just lays out interfaces and requirements for later development. The core of this work is the design of a structure for describing interactive and personalised cultural heritage experiences, the Narration Network, and the implementation of a system able to instantiate a specific path within the network that matches the current context.

A network structure allows multiple paths to be described simultaneously at editing time, but it is instantiated as a single, linear story when the network is traversed during the visit. The Narration Networks enables curators to author their content into multiple stories, all simultaneously available; this network is then combined at run time into an interactive experience for visitors. A Narration Network is therefore a structure that allows to define in which context a certain type of content will be presented to visitors. Summed up the contribution to the meSch project is manifold:

1. To define the concept of Narration Network, a structure to author contents for personalised interactive experiences.
2. To define the process by which the Narration Networks integrates with the current meSch architecture.
3. To implement a software prototype for the definition and usage of Narration Networks to run an an interactive tangible designed for a cultural heritage site.
4. To provide a desktop simulator core for the prototype in point 3 to be used for experimenting with multiple path and personalised narratives.

1.3 New forms of interactions

One important point for meSch is creating devices the visitors can directly interact with. This interaction should differ from just pressing buttons or browsing information on a computer. Tangible UIs provide a way to create this new kind of interactions and bring the physical back into the interaction. Although there are earlier examples of tangible UIs, the term is stated by Ishii and Ulmer in their 97 CHI paper “Tangible bits: towards seamless interfaces between people, bits and atoms” [UI97]. They envision three types of interfaces: Interactive surfaces, coupling of bits and atoms and ambient media. The classical Graphical User Interface (GUI) WIMP paradigm splits and distinguishes physical input devices and digital representations of data one can manipulate, Tangible User Interfaces (TUI) have offer a physical representation for digital information and allow the user to edit those using direct physical interactions therefore removing the boundaries between input and output. Figure 1.2 shows the idea of TUIs compared to the classical Model-View-Controller pattern of GUI based systems [UI00]. The physical representation of digital content makes TUI ideas and technology interesting for meSch. It enables material encounters with the digital data coming from cultural heritage repositories. The meaning of the physical representation and its link to the digital content can vary. Whilst in classical TUI ideas the physical items either represent a digital object or act as a container for it [UI00], in meSch they mainly act as an interface for getting and navigating digital cultural heritage information that is tailored to the individual (or group) and their current experience. This can be done by using smart replicas or theme related objects as well as facilitating real world objects that have a strong relation to themes of discovery and exploration like compasses, goggles or torches.

Other important concepts for personalisation relate to ubiquitous or context-aware computing. Here there is not a direct connection to physical objects and their manipulation: the focus is on visitors and context. This can be as simple as responding to the visitor’s presence or incorporate gestures and even group behaviour. This follows from the observation that human communication has, beneath spoken words, a substantial interaction that can be categorised as implicit information, leading to the notion of implicit interactions [Sch00]. This opens the way to the disappearance of computing devices as it allows interpreting actions that are not aimed at interacting with a technical system. By integrating technology directly in our world we are able to complete everyday tasks that require a computer without even noticing that we are

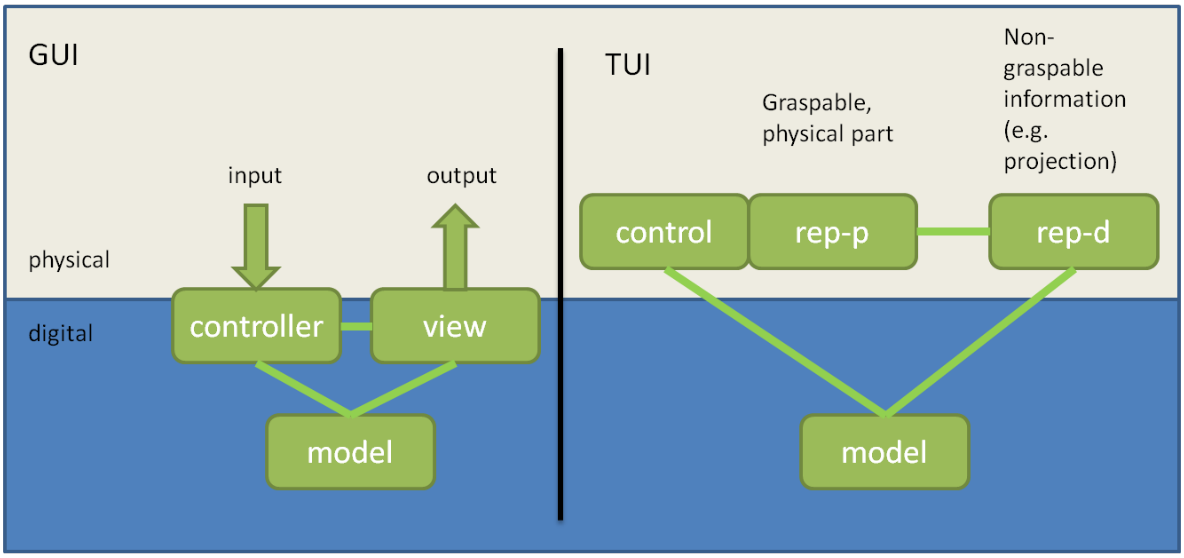

Figure 1.2: GUI and TUI interaction model

using it. This includes a fundamental change of roles, we no longer have to learn how to use the multi-purpose machine called computer, the computer has to learn about its environment and how to interpret human behaviour[Wei91].

All these interaction concepts hold potential to increase visitor engagement by offering playable and easy-to-learn access to interesting cultural heritage information. Abandoning classical computers and their input and output devices and modalities, if done right, also bring back the focus on the exhibit itself rather than putting it on information terminals and screens. When thinking about ways and formats to describe such experiences the knowledge of this basic interaction styles is crucial to understand the objects you have to deal with. Nevertheless, it is just one part of the requirements for the overall system which will be developed and presented in the following chapter.

2 Requirements

The requirements for this work are drawn from three sources:

1. Basic reading of related research about interactions in museums
2. Group discussions in an interdisciplinary team with designers, HCI professionals, electrical engineers and programmers
3. Co-design workshops with cultural heritage professionals and volunteers

While the basic reading provided insights into current forms of interactions in museums and helped to understand the basic needs and challenges, the user-centred design sessions with people directly involved with cultural heritage provided guidance for new and innovative concepts to bring various information to the visitors. From the co-design process many insights on the needs and the workflow of curators could be drawn by directly creating concepts for smart objects and stories. They also told a lot about possible types of interactions curators can envision for their museums. Based on this sessions a couple of concepts for the Sheffield General Cemetery were created by the team. A session with volunteers from the cemetery trust was organized to get feedback on the concepts, especially on the output modalities and the user involvement, which proofed very fruitful to get a deeper understanding of curators views of their own heritage site. This concepts were also used later on for validation of the flexibility and generality of the Narration Network design in Chapter 6.3.

The requirements based on this design sessions and evaluations can be divided into two groups, requirements for the overall system and integration and requirements for the design of Narration Networks. The overall system describes how Narration Networks can be embedded in the meSch process as shown in Figure 1.1. As this system defines some constrains for the design of the network it will be described first. The requirements for the Narration Network structure are mainly lead by the need to provide common ground for technical persons who have to implement them as well as designers and heritage experts that create smart objects with attached stories.

2.1 Roles

There are several user roles involved in the overall meSch process, many of them with different backgrounds, expertise, focus and goals. The important role for the Narration Network design is the curator, as terminology and concepts used in the system must be understandable for this role and tailored to their needs. The other roles represent different views on and interests in the meSch cycle and are used to capture all requirements for the design of the process,

Role	Expertise	Goals
Curator	- Exhibits and related information - Visitor profiles	- Create an interesting, educational and innovative exhibition
Museum staff	- Museum site and organization - (Technical) possibilities	- Setup and maintain an exhibition
(Interaction) Designer	- Knowledge about interactives - Impact of design decisions - Materials and affordances	- Create new and appealing interactive experiences
HCI Professional	- Understanding of human behaviour with computers - Evaluation techniques	- Ensure good usability and user experience for a new interactive experience
Electronic Engineer	- Computing, sensing and output hardware - Hardware related programming	- Create a reliable hardware system - Find new ways to deal with problems and combine technology
Programmer	- Software architecture and modelling - Software implementation	- Create a requirement-fitting software system

Table 2.1: meSch user roles

represented by the user stories assigned to each role in Chapter2.4. Exploiting the co-design approach and bringing them all together in the early stages of the project holds the power to avoid misunderstandings and find a common language when it comes to discussions about interactive, personalisation, and tangible experiences in museums. A short summary of the meSch user roles is presented in Table 2.1, yet the variety of expertise makes it necessary to abstract from explicit technical or design terms when talking about meSch concepts. This will be referred to when talking about naming in the context of the Narration Network. The roles are neither mapped necessarily to a single person nor are they in any way exclusive. Especially the designer and HCI roles are just used to illustrate a more creative or more evaluative kind of view on an interaction concept.

2.2 Design for interactions in cultural heritage sites

Years of research on technology usage in museum show three main problems that have to be taken into account when designing innovative technology proposals for museums.

The first is the isolation of individuals from their group. It is best illustrated by the various kinds of audio guides and later PDA based approaches that deliver information to one person depending on his location and path in a museum. These are essentially personal devices that hamper the social setting of the many visitors who come in groups rather than on their own. While there can still be demand for individual solutions depending on the type of museum and the visitor ethnography, designers should clearly have in mind the effects of their concept when used within a group of visitors [WAG+02].

The second problem relates to the information-terminal approach used to grant access to a large amount of information within the museum space. All this screen based solutions tend to shift the attention away from the exhibits, which in fact should be at the centre of attention [CBF01]. Interactive concepts should try to keep the exhibit in focus and enrich it rather than trying to replace it.

The third issue arises especially from the use of tangible technology to create new interactive experiences. Although there is a big difference between more hands-on focused technology and science museums compared to art galleries, museums are traditionally a place where touching things is often not allowed [CB07]. This is a special challenge designer have to be aware of when designing smart exhibits that require visitor interaction. With their design they have to make clear that visitors are allowed and wanted to touch the smart exhibits and interact with them.

Nevertheless, there is a demand in getting modern technology, social media and content contributed by visitors into museums. If museums want to stay competitive in a modern environment they have to meet the expectations of people surrounded by ubiquitous technology. And they might benefit from the intelligence of the crowd if they succeed in channelling it and ensure high quality of the given information [JABW+12].

In reviewing the meSch concepts so far there seems to be two main categories of concepts for using smart objects in museums. Both focus on enhancing the visitor experience with additional information. Core dimensions that have to be taken into account while designing a language to describe them are depicted in Figure 2.1.

- Visitor centric concepts focus on a device the visitor carries around and that connects him with special places in the museum space.
- Exhibit centric concepts where the exhibit or its display is enhanced to sense people and their actions and offer proper reactions to them.

Whilst the so far presented topics have a strong focus on dealing with interactions, there are some more, important things to take into account when designing for museums. Most of them are related to the people visiting the museum, their backgrounds and expectations and the mission of the museum itself. The way content is created and presented is core. Besides being a place for research and a sanctuary for cultural heritage, museums have an educational mission [The12]. Offering educational material for a wide variety of visitors leads to three types of personalization that are necessary in creating a meaningful experience[AKP12]:

- Offer content in different levels of details to be able to serve the needs of first time visitors with low familiarity with the topic as well as professionals coming back often.

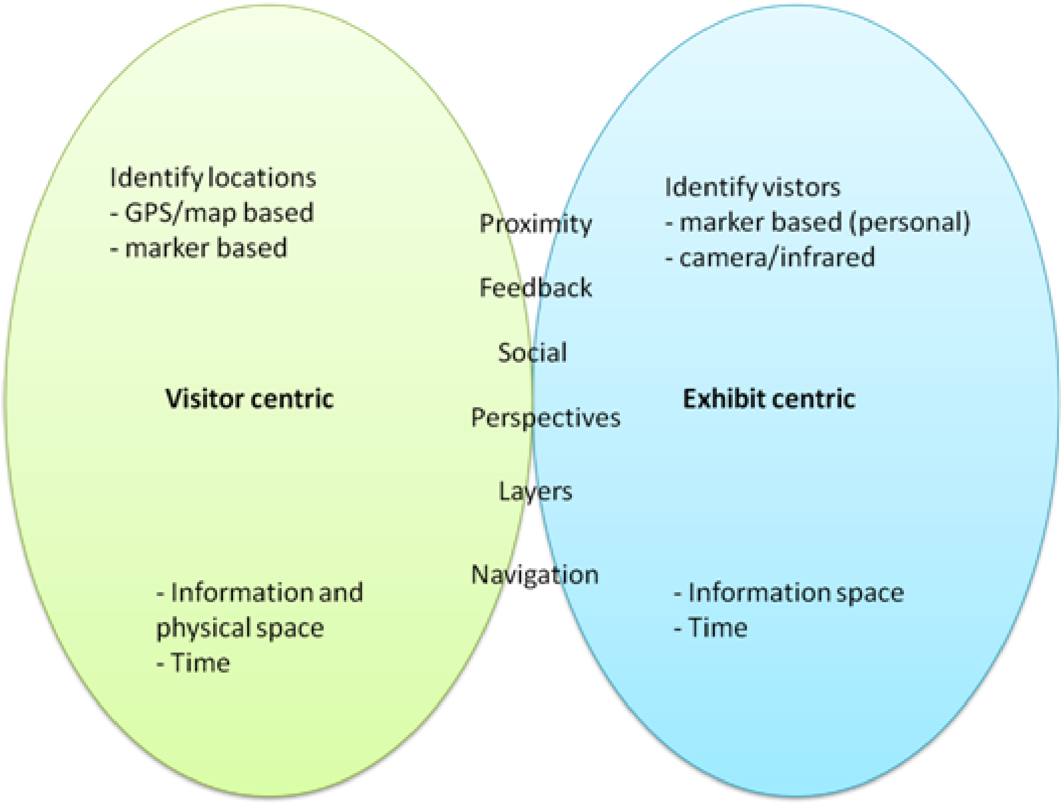

Figure 2.1: Categories of meSch concepts

- Offer content in the right language for the visitor. There is a big gap between the appropriate language when talking to a child and talking to a domain expert.
- Present different perspectives on a topic to fit visitors preferences on the one hand, but also extend their view and make them think about the exhibits and the stories linked to them.

Structuring an application to meet these requirements does not only create a system to define and author the content for smart objects, it also provides terminology and a common language between all stakeholders to talk about the design of smart objects.

2.3 The meSch cycle

The idea of meSch places all the interactions on the heritage site in a bigger context and is very important for the system design as it points out the interfaces for the software and the different roles involved in the process. The smart objects has to be designed and the technology to implement them has to be available. If this prerequisite is met, a museum curator can start to select content from various data sources and create his own narrative for the use in his exhibition. The narrative could include different kinds of media as well as data from different kind of sources, local databases as well as public heritage data bases, public image, sound and video repositories and databases from partner museums. It should be possible to simulate and test the narrative as next step to ensure it has been prepared correctly and adds value to the exhibition. Based on the design of the experience, there is the need for customization, using physical objects that relate to the heritage site and its exhibits. To enable visitors to interact with it, the technology and the narrative has to be combined and installed in the exhibition space. Whilst visitors interact with it, log data should be collected for various purposes. Logs of identified visitors can be used to create an online experience where the visitors are able to re-experience their visit and get additional information about things they have seen or even things they have missed. Log data can also help interaction designers to understand how visitors use it and help curators to learn what works with visitors and what does not. Log data could also be used to create and find new content based on visitors habits, expectations and feedback or even engagement in content creation if the interactive experience allows for it.

2.4 User stories

This section formally sums up the requirements for the overall system in general and the ways to describe and author experiences in detail. It provides the requirements in form of short user stories known from agile development models. This is less formal than complete use case descriptions, it serves the purposes of this work better and is easier to understand when working with non-expert in a multidisciplinary team. User stories are grouped by roles to give a better idea about the involvement of each role in the overall process and the tasks a role is faced with.

Role	Story	Requirements
As a Curator i want...	... to combine existing components for interaction and sensing in a manner to build a unique user experience for my site.	- Predefined, compatible components - Tailoring of encasing and appearance
	... to learn about the visitors of my museum, their behaviour and my current exhibition.	- Logging of visitor interactions
	... to present additional information from cultural heritage databases in the context of my exhibits in a personalized fashion.	- Authoring content of different kind and different media types - Introduce perspectives and layers
As a museum staff i want ...	... to setup a system of distributed components building a meSch scenario to enhance the visitors' museum experience.	- Of the shelf hardware components - Reuse of software components - Installation hints - Control over paramters
As a designer i want ...	... to create an innovative and fun experience for the visitors.	- No artificial, technology based limits - Extensible definitions
	... to grab the visitor's attention to get them in touch with enhanced information and interactions around the exhibits.	- Easy to setup and customize solution
As a HCI professional i want ...	... to ensure usability and user experience of interactive, digital installations.	- Easy prototyping capabilities - Testing capabilities
	... to learn about visitors interactions in the museum space.	- Sensor logging
As an electrical engineer i want ...	... to use the technology that makes most sense to realize a concept.	- Platform independent solution
As a programmer i want ...	... to create reusable software components with low maintenance and documentation effort .	- Well defined interfaces - Common architecture

Table 2.2: meSch user stories

2.5 Key challenges

The user stories represent the set of functional and non-functional requirements the software design has to cover to make it a useful solution. Splitting them into the user roles makes it easier to understand the suitable terminology and context for each requirement. But there is also a more general view on the challenges these requirements draw. The central challenge is providing personalized content in the right context. This is what differentiates meSch from other approaches in the past and what makes it necessary to come up with new concepts and technologies. Interactive, smart objects help to determine the current context and present appropriate, personalized content. A network of content is required to model all the different possibilities for the context and support multiple stories. As context is a very flexible and generic term a lot of sensors are needed and a lot of different combinations can be imagined. For the Narration Network design in Chapter 4 this means that an efficient way to implement it has to be find in such a way that can fit any kind of devices or augmented environment, any type of media and any type of conditions. As this networks can get complicated and a field test takes a lot of effort, the existence of a simulation tool to the networks reaction to a series of different conditions is essential.

2.6 On maintenance

Although not being in scope of this work, maintenance is a critical success factor for every meSch installation. Having a distributed system of isolated components raises the need to monitor the state of all these components. Sensors and presentation devices might get damaged, run out of power or get disconnected for some reasons. There must be a quick way to identify such components to be able to reset, repair or replace them with minimal impact on the overall system. This could be especially challenging in an environment where mainly short distance communication like Bluetooth is used. Visitors expect the systems to work properly as they may not have a second chance to come back and try it again. Interactive museum installations that are not working properly affects the museum experience and shed a negative light on the institution.

There are three perspectives when monitoring such distributed systems. Monitoring it constantly using a heartbeat or similar technique lets you know about the state of every component but requires a lot of effort in building such systems. The components must be able to report about themselves and they must be able to connect to a central service to send their data.

An on demand service removes the requirement for constant connection. One can just go around and see if all components are responding. They still need some sort of wireless connection and an API to respond but the effort is lowered. There could be also the possibility to detect broken components by analysing the log files from the interactions. If a normally regularly used component does not show up in the log for a whole day or a component normally visited between two others is missing one can reasonably guess something is not working properly. Many of these issues might be quite easy to handle in a closed museum environment, but for an outdoor place this is a challenge. It is therefore important to include in the design of the system architecture an explicit consideration of the maintenance; indeed it becomes difficult, if not impossible, to manage later when the system is fully defined and in place.

While experimenting with the Bluetooth Prototype described in Chapter 5 in a public outdoor space, additional issues came up regarding the upkeep of a solution. Power and an overall network infrastructure are often not available, which makes is harder to monitor all components. In contrast, independent powering draws the need for constantly monitoring battery status. In addition, outages are much more likely to occur due to rough conditions like weather and temperature as well as vandalism and theft. Calibration tasks can also be an issue if the equipment has to be removed and put into a safe place in the evening. Generally speaking, going to an outdoor place increases complexity as well as importance of a maintenance solution. Some of these issues are discussed in chapter 5 on the outdoor prototype.

3 System Overview and Context

To get a better idea about the complexity of designing a narrative network, it is necessary to take a look at the context of its later use. A high level architecture for supporting the whole meSch cycle is needed. In different stages of the meSch cycle people will work with Narration Networks using different tools and having different goals. The design must take all the different needs into account. It starts with a designer creating the definition of a template for the authoring tool. A curator can select this template, fill it with the content he composed and tailor it to the needs of his heritage site. All these authoring is done in the meSch server environment which is shown in Figure 3.1. It is a three tier architecture showing the components used by visitors and museum people in the top layer, the adapters to connect to various data sources in the bottom layer and an intermediate layer in between doing the necessary transformations and providing unified access to the data sources. The finished models are exported to the heritage site and used to drive a set of smart objects. At the end, collected log data from the heritage site feed back into the server environment. While the server is designed and implemented with a fixed set of technologies, the systems in the museums for the smart objects can use arbitrary platforms and technologies. Some configurations may be well connected and have wireless available across the whole place, others may be outdoors settings with no wireless connections at all. In the end, all of these systems should integrate in a seamless manner to form a system fully supporting the meSch cycle. Integrating distributed components across various platforms is a well known problem in enterprise application integration, as discussed below.

3.1 Distributed, multi-platform computing: a look at loose coupling

Loose coupling introduces a fundamental principle to deal with the multi-facet situation described above: It reduces the number of assumptions two components make on each other when exchanging information. This leads to four degrees of autonomy that can be achieved using loose coupling [HW03].

- Reference autonomy: The components do not have to know each other
- Time autonomy: The components do not have to be simultaneously available and work at the same pace
- Format autonomy: The components may not use the same data format
- Platform autonomy: The components may not run on the same platform or be implemented using the same programming language or technology

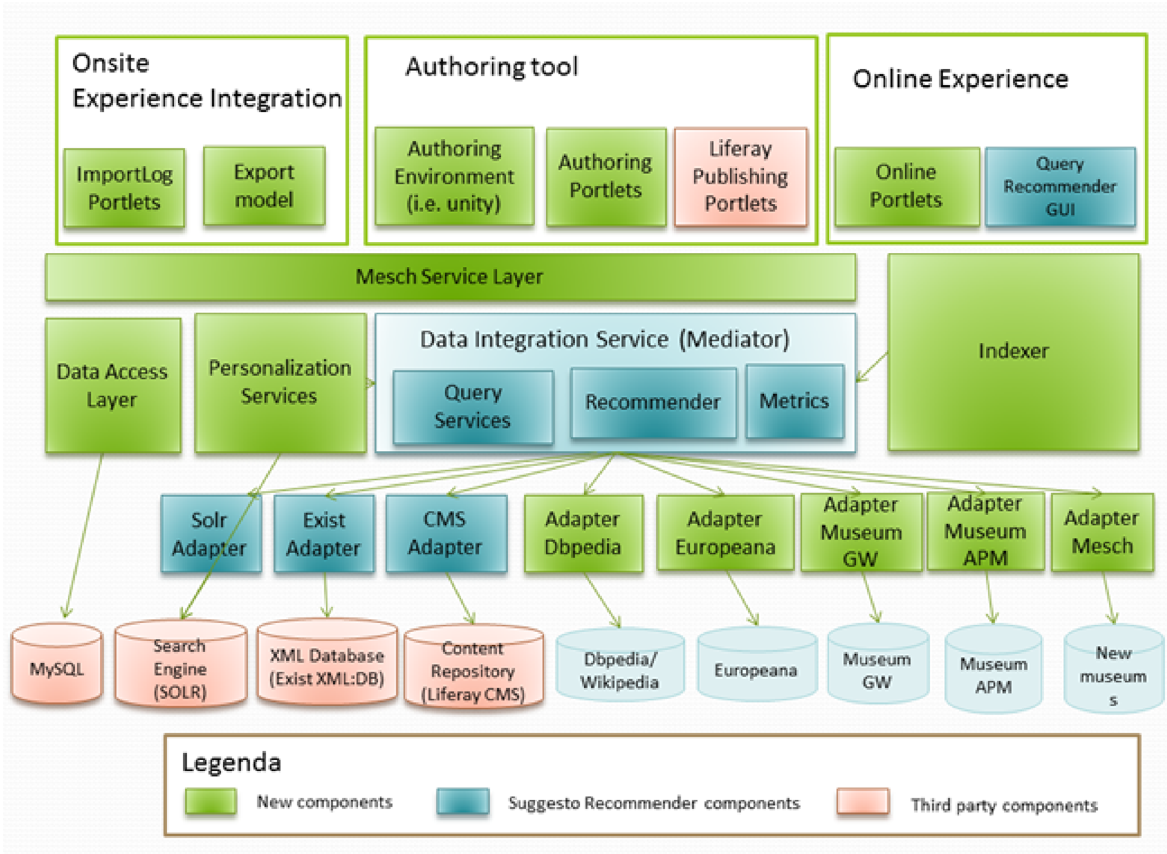

Figure 3.1: meSch server components [Ven]

Using loose coupling techniques therefore enables to keep the system located in the museum independent from the meSch server environment, creating interactive systems without the need of constant Internet connection and removing any constraints about platforms, programming languages and technologies for the implementation of the smart exhibits. The field of enterprise application integration suggests further solutions to address the challenge: a set of patterns can be used to illustrate the power and possibilities of archiving loose coupling through the use of messaging.

Some of the most important patterns used to design the overall system presented in this work will be briefly introduced, if new problems arise during the meSch project there will probably be some more patterns that can be used to solve problems. In fact, the design proposal is not a messaging system as real messaging comes with a large overhead and should only be considered if absolutely necessary. For separating museums and smart exhibits from the meSch server, a file exchange is a suitable solution as guaranteed delivery, high availability and high performance are not requirements for the meSch system. The following descriptions are organised in simple tables, the first row contains the name of the pattern, the second row a short description, the third row its relevance for meSch. For more details on this pattern see [HW03].

Message
Q: How can two applications connected by a message channel exchange a piece of information? A: Package the information into a Message, a data record that the messaging system can transmit through a message channel.
Although the meSch system won't contain real messaging middleware it might use the idea of a message based system and a queue to transfer log data back to the meSch server.

Message Endpoint
Q: : How does an application connect to a messaging channel to send and receive messages? A: Connect an application to a messaging channel using a Message Endpoint, a client of the messaging system that the application can then use to send or receive messages.
Just write an endpoint for a platform once and make it available as a library. This enables faster development of prototypes and interactive systems communicating with the meSch server.

Canonical Data Model
Q: How can you minimize dependencies when integrating applications that use different data formats? A: Design a Canonical Data Model that is independent from any specific application. Require each application to produce and consume messages in this common format.
Define a common format for each of the two kind of data that have to be transferred, Narration Network data and log messages.

Content Enricher
Q:How do we communicate with another system if the message originator does not have all the required data items available? A: Use a specialized transformer, a Content Enricher, to access an external data source in order to augment a message with missing information.
The on-site system could use a minimal data format for logging; the meSch server saves the data about created Narration Networks and maps the data back when it receives a log message.

There are many more pattern defined that can be interesting for load balance (e.g., what if many museum systems send log data to the meSch server?) or monitoring (e.g., testing log data processing). It can also be worth a look if implementing on-site systems with distributed communication as there is the MQTT[1] protocol especially for supporting the lightweight, message based connection of sensors and actuators with many open source libraries available.

3.2 Placing the system in a bigger context: interfaces

The meSch cycle indicates two connection points where the definition of an interface is necessary. The usage of the term follows a more general notion of an interface and is not limited to its definition in software engineering. An interface in the sense used here is a system to exchange information in a loosely coupled way. Both interfaces draw some requirements that can be addressed with adoption of the pattern presented in 3.1. To be platform independent both share the idea of using XML as data format and using XML schema to prescribe the possible content in a common format. These formats can be read and written on every platform with standard libraries. For every programming language there must be one converter library producing usable objects from the XML file. For transferring the data from the authoring tool to the on-site systems we need means to:

- Define the structure of the narrative
- Define points of interest in the museum site
- Link the narrative parts to points of interest
- Define the parameters of the interaction
- Link various media content to be used as part of the narratives
- Enable personalisation of contents and experiences

The XML file format of the Narration Network will be presented in Chapter 4 as a central part of this work. An important point of the design is referencing the contents for a narrative using URIs. Depending on the content source the meaning of the URI is interpreted and used to access and present the actual content. The definition of content sources within the document allows combining these methods. Depending on the implementation of the prototype, the XML file and referenced media may need to be copied to the on-site system.

This thesis defines a XML schema description for Narration Networks, a sample Narration Network file, python components to parse this file and and a prototype system to instantiate it. For transferring log data back to the server from the on-site system we need to:

- Create a log entry every time a content node is accessed and its content is presented
- Send the entries to the server if a connection is available

[1] http://www.mqtt.org

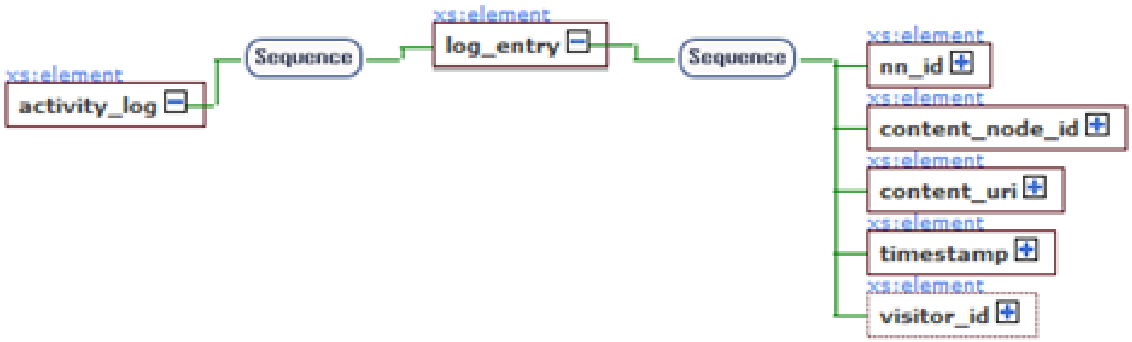

Figure 3.2: XML schema description for log messages

- Store the logs and send it later if no connection is available

To keep the log files small and efficient, especially for devices with little memory or low computing power, only references are stored. These references contain the IDs of the Narration Network and the content nodes as well as the content URIs. The system in place sends its records as XML messages back to a software component on the meSch server. Following the content enricher pattern, this component uses the given IDs and URIs to look up under which context a certain visitor has been presented with a piece of content. The approach requires saving a representation of the Narration Network on the meSch server under the same ID that is used in the corresponding XML file. As it is part of the development of the meSch server infrastructure it will not be described in further detail here, whereas the log structure with a XML schema description is defined to create a file-based log on the prototype system. The log that is located on the device has to store additional information to be able to relate it to the visitor later on. This could be an email address which could also be used to invite the visitor to review his visit or a generic number to enter in a website when he is back at home. Figure 3.2 shows the XML schema description for a log. It contains references to the network, node and the content URI that has been presented, a timestamp for the activation and a reference to the visitor if one is available.

Logging the visitor's own activities enables him to look back at his visit on the cultural heritage site. The interaction log is completed by the context determined by the triggers of the activated node. For logging sensor or other technical data there should be another logging system that can be easily monitored, turned on and off on demand and selective about what to log and at what interval. This is more interesting for designers and technicians than for curators and should reside within the sensor and actuator platform that is developed for meSch.

4 Narration Networks

After discussing the need of a structured way to capture the experience, it is time to get into the details of the proposed Narration Network. The contribution to the meSch process is threefold:

- defining a set of objects and relations the curators can deal with in the authoring environment,
- creating a file format that can be exchanged between server and on-site system and
- creating a data structure that can be used to drive the on-site experience.

In addition, an agreed set of terms to describe smart objects with attached stories within a cultural heritage context could improve communication about new concepts leading to new prototypes and templates. The key challenge in the design is ensuring the flexibility to cover a broad range of concepts and allow for quick implementation and simulation in the meSch project while keeping it easy enough to enable curators to author content in this network structure and create smart objects using this networks. This can be archived by defining a fixed set of objects curators have to deal with but make them easily extendible. The building blocks presented in the next section are designed with this requirement in mind.

4.1 Building blocks

A Narration Network in described through five items. They define the objects the network is composed of as well as the entities that are used to navigate in the network.

- **Point of Interest (POI):** A POI marks a defined physical place in a cultural heritage site. It is used to group multiple content nodes.
- **Content Node:** A content node is a part of the network that can be activated to deliver its content if the right condition is matched. The condition is defined using a set of triggers.
- **Trigger:** A trigger is an abstract description of interpreted sensor data with a discrete set of possible values. Triggers can be as simple as location or perspective but also more complex like group size or visitors emotion.
- **Context:** A context is a set of active triggers. It contains the current value of each trigger for which sensor data is available.

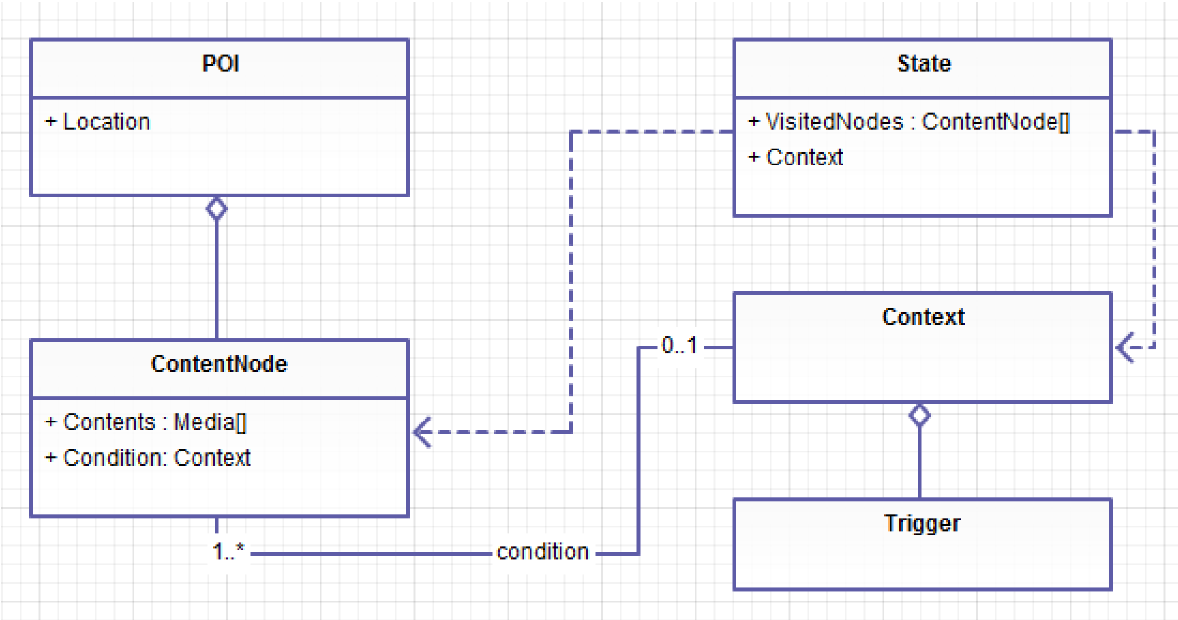

Figure 4.1: Narration Network building blocks

- **State:** As content nodes can define predecessors to form a network, the state contains the current context as well as the set of all visited nodes.

With these building blocks the construction of a Narration Network is straightforward. POI's mark the spots in a cultural heritage site that should be augmented using tangible technology. To each POI a set of content nodes can be attached, representing different layers of information about the POI. Every content node defines a context that acts as an activation condition for the node. If the state matches the context of a content node as well as the prerequisites from the constructed network the node is activated and the content is delivered. Figure 4.1 shows the building blocks and their relations.

The basic idea of organizing the content in some kind of network structure controlled by a set of triggers is drawn from the activation network concept Petrelli and Not used in their work on a hyper media audio guide [PN05]. They focused on delivering personalised content based on visitors history and interests. A complex system was used to determine the snippets to play and automatically combine them in a seamless manner. The Narration Network is a simplified approach that makes it easy to use for authors without technical knowledge. The visitors history is represented by the possibility to use predecessors for nodes. Interests can be controlled by the visitors themselves by interaction; they can be authored using "perspective" and "depth of information" triggers. While these two triggers model explicit interactions, most triggers are used to model implicit interactions. They act as an abstraction layer on sensor data. The concept of this abstraction emerged from discussions about using a set of rules to define the behaviour of smart objects in the first place. While the definition of simple rules is straightforward (if signal strength > 240 play audio file) their count and combination can get

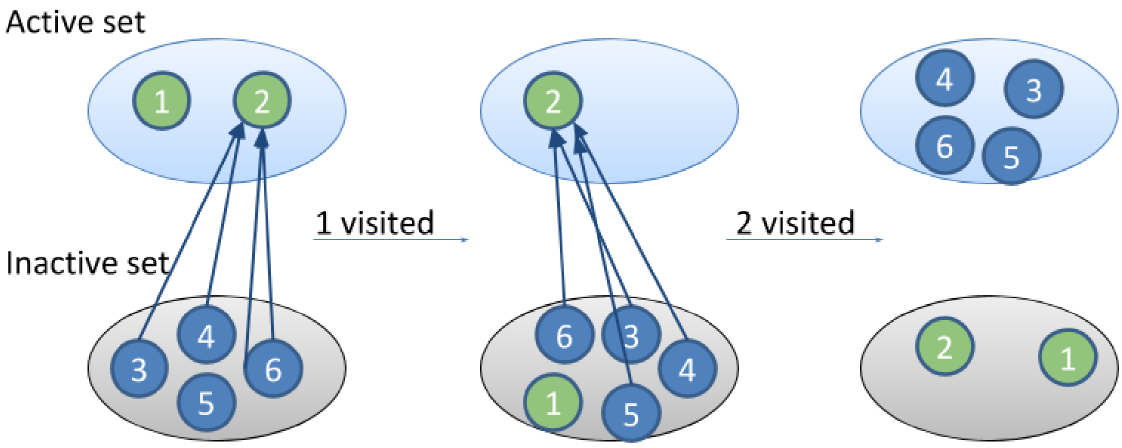

Figure 4.2: Navigation example for the BT prototype network

very complex and requires technical knowledge as well as programming skills, both skills that cannot be assumed for curators. Introducing triggers like "proximity" solves the problems rule based systems carries, but create a new drawback that is discussed in Chapter 4.3. At start there will be a fixed set of triggers drawn from the concepts and prototype proposed so far. But the trigger system is designed in a way that allows to quickly add new triggers as the need for them emerge in meSch at a later stage. This design ensures maximum flexibility and, at the same time, keeps the number of entities a curator has to deal with manageable.

4.2 Navigation and simulation

Navigation is based on a set of active nodes. This set contains all content nodes that could be activated in the current state. For the next steps, two properties of content nodes are essential.

- (1) A content node can be visited just once, or as many times as it has content pieces (if it keeps more than one piece of content), or unlimited times. This allow the curator to design different types of experience, for example a piece of content may be presented just once for each visitor, or the same piece can be played many times for the same person.
- (2) In each state exactly one node can be activated, if the current context suits more than one node the node to be activated is randomly selected. There is also the possibility to exclude a trigger from the condition of a node, meaning that the value of this trigger is ignored when evaluating the condition. This is useful in case of different perspectives or depth of information and if there is one piece of content visitors should always get.

Given these two properties, navigation can be described as follows, a graphical view of the first two steps can be seen in Figure 4.2. Initially the active set contains all content nodes without predecessors. In this example they are marked green because they can only be visited once. As a visitor walks around, the state of the system is updated constantly. For all content nodes in the set of active nodes the context is tested every time it changes. If the state matches the condition of a node from the active set the node is activated and the set of active nodes is updated, adding all nodes that have the activated node as a predecessor. When the visitor finds node 1, it is activated and moved to the inactive set. When the visitor finds node 2, the content of node 2 is presented and both sets are updated again. Nodes 3 to 6 have node 2 as prerequisite and therefore are moved to the active set.

Although the selection is simple, it raises a more complex question: When to create a new context and select a new node. Some contents like sounds or videos have a natural length, others like a displayed text or a tactile feedback does not. Other parameters include context changes during the presentation of content (e.g. the visitor changes the perspective) or nodes with multiple contents (should they be presented directly after the first piece has finished or should the visitor be given the chance to go away and come back later?). While assigning a duration for each piece of content solves most of the problems related to timing and gives the curator control over the experience of the visitor, the context change has to deal with more constraints. If the concept is implemented as a single-threaded script there is no chance to interrupt the presentation of content. An event driven environment in theory allows to interrupt content during delivery, but there might be triggers where it makes sense to do so and others where it does not. It is up to the interaction designer to implement the behaviour of the system the way he imagined it to work.

4.3 From sensors to context

While triggers introduce a level of abstraction that makes it easy for curators to create a narrative without having to deal with technical details, it draws the need to implement a piece of glue code that creates context objects out of sensor data. As sensors might contribute to multiple contexts or a context might need data from multiple sensors there can be no easy solution like a simple mapping with threshold values. For an independent solution like the Bluetooth prototype it can just be hard coded (see 5.4 for details). Creating a hardware platform that allows to plug in sensors and receive events from them with minimal setup effort needs some kind of mechanism to easily define mappings from sensors to context. Such a platform is currently under development within the meSch project: ideas about possible connections can be drawn from this implementation. Currently the sensors constantly send signals to a central component able to react to these signals. The reaction is based on series of if-then-else statements as reflected in the following quote from the work package task description [PCD+]:

> Further, a web user interface should be provided, that offers viewing and analysing logged data as well as generation of "if-then-else" rule chains in order to automatically trigger actions from incoming events.

Although the system is designed to support the full function including presentation of various media, this rule engine could also be used to create new context objects containing the appropriate trigger data to run a Narration Network. This approach would allow to combine the power of a common hardware platform integrating sensors and actuators with the authoring toolkit user to compose Narration Networks. Integration can be loosely coupled based on XML file exchange of context objects and media files or links to media files. As an alternative, the navigation component could be directly integrated in the platform software.

Using loose coupling would allow to use the platform for rapid prototyping without the need to create Narration Networks and to create Narration Networks based installations on custom hardware. Direct integration would allow to simplify handling of the system from a programming perspective and thus improve maintenance. A third alternative would be even more loosely coupled. Given a mapping from sensors to triggers, the whole Narration Network could be imported and transformed in a series of if-then-else rules automatically. Defining the mappings would be part of template creation as it requires some technical knowledge as well as a basic understanding of logical constructions. As the same sensors can be used for multiple trigger depending on the template this can not be done automatically.

4.4 Including content

Construction and navigation of a Narration Network is of course just one half of the concept. The most important part is the story itself, represented by pieces of content attached to each content node. A node can contain more than one piece of content to allow revisits in explorative environments. The definition for a piece of content can be written as a tuple consisting of (1) the content type, (2) the content source and (3) a reference to the content itself. Referencing a content using a URI allows to include contents from various sources. If an Internet connection is available on-site it can load contents directly from Internet repositories or the meSch server. In an offline environment it can point to a location in the file system. It even allows to address web services to create content in context on demand. To support all these scenarios, the content source is needed to know where to look up the given URI. This can, for example, be the file system, a local or remote document oriented database or a web address. In addition, the content source tells the system what kind of answer it can expect. The content type finally indicates what to do with the response. Play an audio or video file, display a text or image, print a document, and even use it to encode a physical reaction. Figure 4.1 shows some examples of referenced contents.

Content Type	Content Source	Example URI
Audio	Filesystem	`file://localhost/meSch/audio/intro.wav`
Image	www	`http://www.mesch-project.eu/wp-content/themes/meSch/img/logo_v1.png`
Document	CouchDB	`http://example.com:5984/meSch/infos`

Table 4.1: Content entry examples

Using URIs allows to flexibly address a lot of potential content sources in a uniform way. Referencing the content rather than embedding it keeps the exchange format simple and small. As URIs are a widely used technology it can be used on nearly every platform and integrated very well with modern web technologies.

4.5 File format

To describe the XML file format for saving a Narration Network, a XML schema description is used. This description acts as an instruction for how real instances of a Narration Network document are built. The meSch server can use it as a blueprint to create valid Narration Network documents, all clients on-site can validate a given Narration Network against the schema description. This reliable, agreed-upon data format allows the writing of generic software components to load and navigate Narration Networks. If a package is written for one platform and language it can be reused by other experiences in their implementation. The basic structure showed in Figure 4.3 is quite simple. The root node narration_network can contain multiple POI nodes which can contain multiple content nodes. Each of them has an ID attribute assigned which can be used for cross references, logging or reporting. It links back to an entity saved on the meSch server. The construction of the content node is more complex. It defines two additional attributes. First is predecessors, it contains a list of node Ids that have to be visited before this node can be active. Second is repeat, it determines if the contents of this node can be accessed more than just once. The trigger element defines the condition under which a node is activated, the content element the real content that is presented in that case. Figure 4.4 shows the elements contained in a content node. The triggers determine when the node is activated. The example shows just a limited set of triggers, for the meSch project an exhaustive set of triggers should be defined during the investigation

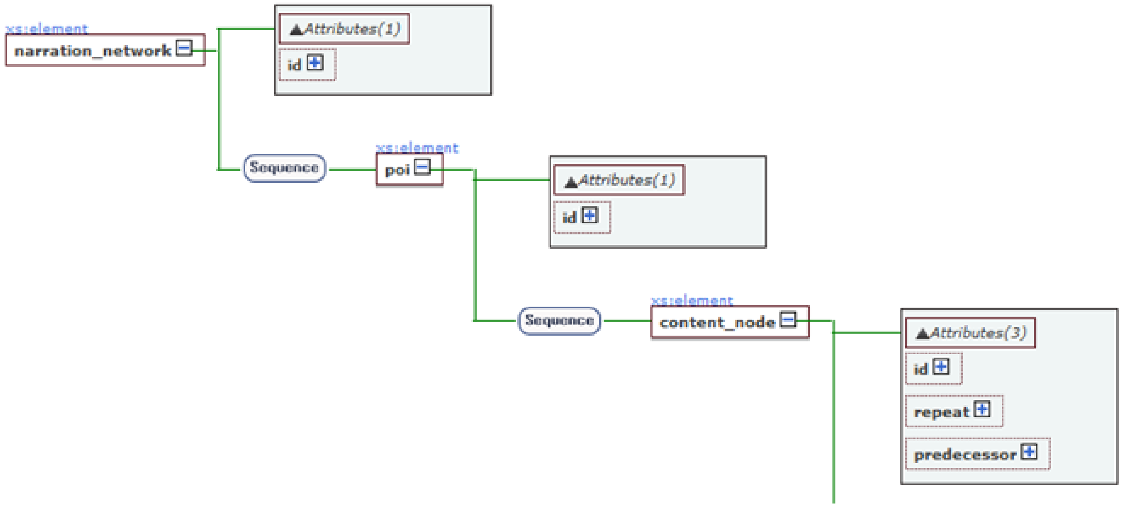

Figure 4.3: Basic XML schema structure for Narration Networks

of possible interactions and technologies. Making them optional ensures that every network just uses the triggers it really needs. As seen in the example scenario, the content of each trigger element contains the value it can activate with. The implicit assumption here is that all conditions are linked using a logical AND statement, only if all triggers have the right value a node is activated. The used triggers and their possible values can be defined in another XML file that acts as template for a certain experience, this idea is detailed out in a scenario in Chapter 5. If needed, the system can be extended by adding attributes to triggers that allows curators to parameterize the trigger behaviour. The same can be done if content sources have to be extended to add something like credentials needed for a remote login. Content types and sources should be well defined enumerations that proofed to be needed during concept investigation, pretty much like the set of possible triggers. While in terms of flexibility and platform independence the XML file approach fits the requirements pointed out in Chapter 2 very well, it creates a drawback for systems with local content. One has to put the XML file on the device used on-site. Additionally all the contents must be copied to the devices in the right folder in the right structure to match the defines URI paths. You then need a component to map the content sources and resolve the references to get to the actual content. This can be avoided by an alternative implementation of the Narration Network briefly outlined here but not implemented or tested in the scope of this work. The basis would be a document oriented database, holding all nodes as documents and contents as attachments. During creation in the authoring tool maps are created that reflect the navigation structure of the network. Navigation is performed by creating queries on the existing maps to select the node and access the content. This approach would require one software component to connect and communicate with the database which would just be one file and thus much easier to setup. As

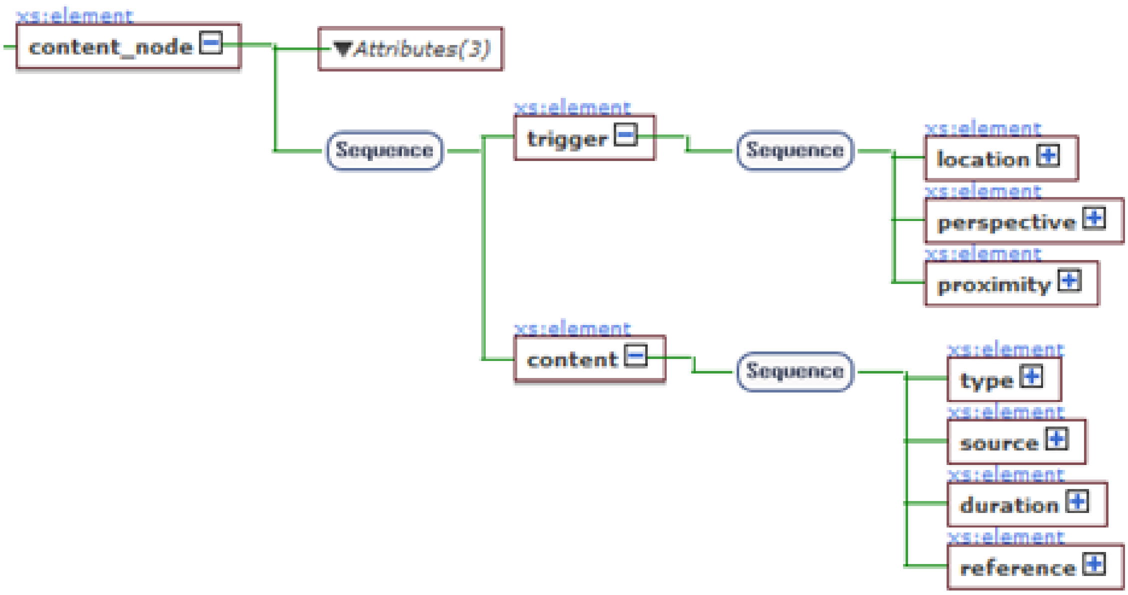

Figure 4.4: Detailed structure for content nodes

a downside it imposes the necessity to run a database management system for the document oriented database on the target platform and device. Major challenges would be the automatic creation of the maps in the authoring stage and dynamic creation of queries out of context during runtime. As requirements evolve it should be worth keeping in mind.

4.6 Prescribing Narration Networks with templates

Of course authors will not have to create XML files to create their narratives. A dedicated authoring environment residing within the meSch server architecture as depicted in 1.1 will be developed during the project. It should pick up the entities a Narration Network consist of and present them to the user in a meaningful and easy to understand way. But having a system to describe an experience draws the question which elements to use and how to do it. As the concept of having templates for certain kind of experiences came up early in meSch it seems logical to follow this route and give it a concrete shape. Having a language to describe an experience allows the definition of a meta-language that can act as a blueprint for describing a certain experience.

By definition of the network structure every instance is a subset of all possibilities to create a Narration Network, therefore the template is of restrictive nature; it puts constraints on the elements and their values. Putting constraints on elements means limiting their occurrence. This can be the total number of POIs or content nodes, the presence or absence of a certain trigger or the number of contents for a content node. Putting constraints on values for triggers

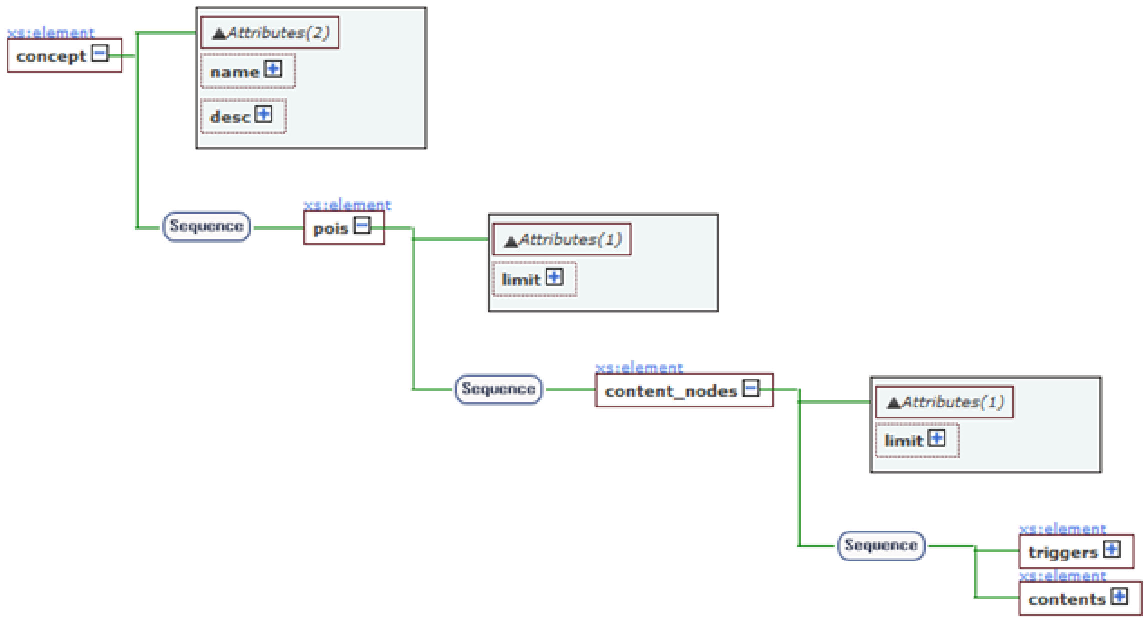

Figure 4.5: Basic XML schema structure for template files

means giving a set of possible values the author can choose from that make sense and are understandable in the context of a certain experience. In the case of content it means limiting the possible sources and types of media the author can attach to a content node. For further help and guidance to the authors, a template should also allow adding descriptions to tell about these limitations and their causes.

To put these kind of templates to real use the authoring tool has to adapt his user interface to the limitations a template contains. This can range from error messages (“ you cannot add more than 10 content nodes”) to dropdown selections for trigger values and limitations of file types that can be selected to add contents. As the authoring environment is not part of this work and is not contained in the prototype that is presented in the next chapter the contribution to meSch is in the file format for templates. Staying in line with all other file formats it uses XML as language and XML schema to define the structure of a template file. The basic structure is shown in Figure 4.5. The root node concept has a name and a description telling about the purpose of the concept. It can be used to browse a repository of concepts and find an appealing one. The limit attribute on POIs and content_nodes states the maximum number of instances that can be created, 0 is synonym for an unlimited number of instances. If a concept is designed to enable interaction with exactly one exhibit or place a limit of 1 would make this clear and the authoring tool could react by creating a POI and hiding all possibilities to add another one for example. Putting a limit on content nodes means that a POI cannot have more different kinds of content than this number. This includes

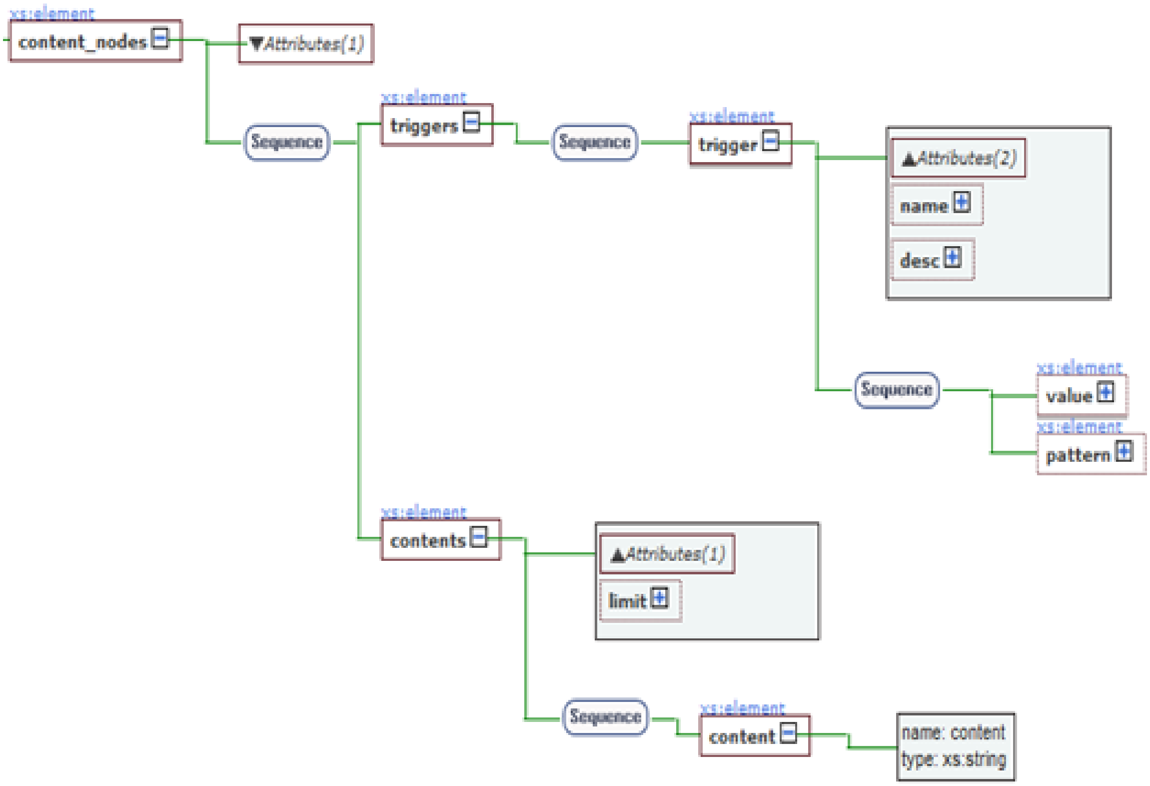

Figure 4.6: Detailed structure for content templates

addressing different layers of knowledge and interest and should therefore be used very carefully by template designers and only if there are technical constraints.

Defining triggers and contents as shown in Figure 4.6 follows the notion of a white list: Only elements that are listed here are allowed to be used during the authoring process. The definition of a *trigger* includes a name chosen from the list of available triggers defined in the XML schema for Narration Networks and a description that says how this trigger is used in the concept similar to the definitions in Table 4.2. The value of a trigger can either be a discrete set of elements that can be mapped to a dropdown selection or a regular expression used as a pattern for possible string values. If a pattern is used, the description should be available as help entry in the UI and explain to the author what kind of content can be assigned to this trigger. Perspective and layer triggers might be represented in another way than normal triggers. They could be used as flavours of a certain content rather than modelling independent nodes. Technically they would be treated the same ways as other triggers, but for the UI it appears more logical to treat them as variations of one content node.

The definition of *content* includes a limit of contents per content node and the type of media the concept is able to present. The description element of the content can be used to make further suggestions, e.g. for an audio content just use a certain file format like MP3 or make the snippets no longer than two minutes. The possible values should be the same that are used for content types in Narration Networks. As with the list of possible triggers they will evolve over time in the meSch project and should be fixed before the system goes into productive use. For the UI this information could be used to limit the file selection dialog.

4.7 Triggers and content types

As pointed out in this chapter, the idea of Narration Networks is based on a fixed set of possible triggers and content types. Doing so greatly reduces the complexity of the design for the authoring tool and the usage of it. The author just has to deal with a limited set of concepts that are taken from everyday life and that involve no technical knowledge. It seems more intuitive to work with proximity rather than working with a Bluetooth signal strength measurement for example. The limitations that are introduced on the downside of this concept are not as strict as it seems in the first place. Designed as an extensible schema, new triggers and content types can easily be added. Having the template definitions, they don't effect existing definitions. The authoring tool has to be written in a generic manner anyway so to allow to accommodate templates for emerging technology or new forms of smart objects, so adding additional triggers and content types should be easy. From the viewpoint of designing smart objects, the interpretation of a certain trigger is up to the actual implementation. Two concepts can both use proximity triggers but use different sensing technology and define different values that fit their design best. But conceptually, both concepts use an abstract definition that is presented to the author to keep him away from technical details. Analysing the different concepts the team came up with, 3 different flavours of proximity could be extracted and used for the definition of proximity in Narration Network examples. If a visitor is in range to a smart object that allows him to perceive any kind of output that draws his attention to the smart object he is in attraction proximity. If a visitor is able to perceive the content presented by a smart object he is in engagement proximity. And if a visitor directly interacts with a smart object its called the interaction proximity. Other triggers might be abstracted in a similar way to make the authoring tool easier to use while preserving the full flexibility for concept designers. The basic triggers presented in Table 4.2 already allow to define a wide variety of conditions but is meant to be extended during the meSch project lifetime.

Content types are more straightforward to define. The number of different ways to deliver information are limited by the human possibilities of perception. Available sources of content as well as possible presentation devices focus on visual or audio contents. This might be the most interesting ones from the viewpoint of curators, although in reality multimedia formats can come with strings attached such as incompatible file formats or very strict copyrights. For tactile feedback a generic action content type has been added. A basic list of content types is presented in 4.3, as with triggers, the interpretation depends on the design and the implementation of a concept.

Name	Description	Example
Proximity	How far away are visitors from an interactive exhibit or place or from each other	Defining an attraction and an engagement layer based on distance measured by signal strength of communication devices
Position	An identified place in space	Use the MAC address of a device placed at a POI to identify the POI
Perspective	A point of view from which a narrative is told	Telling the story of a war from both perspectives, attackers and defenders
Layer	Depth and detail of information	Have basic information for new visitors, more in depth information for visitors with more interest
Group Size	The count of people being together in one place	Use another method of presenting content for a group of people than for individuals
Age	The age of an identified visitor	Present appropriate content for children, maybe use Kinect to look at the size of visitors
Action	A generic trigger for things like gestures	Unlock information if visitors do certain things like moving in special ways

Table 4.2: List of possible triggers

Name	Description	Example
Audio	Any kind of story, music or environmental sounds	Personal stories told by people with a relation to a topic
Video	Movies, Clips or animations with or without sound	Videos of reenactments related to a topic
Image	Photos, Illustrations and Graphics	Old photos from a place that shows the original appearance
Text	Information meant to be read	Provenance information for a certain exhibit
Action	A generic content type for tactile feedback	A device that gets warmer the nearer you come to your target

Table 4.3: List of possible content types

5 The Companion Novel, a tangible Prototype

As a proof of concept the Narration Network was implemented and used within a physical prototype, the Companion Novel. An iterative, user-centred design approach was adopted and two cycles of design and evaluation implemented. The software part was largely unchanged from the first to the second iteration that focussed on the form factor, therefore only the second prototype will be presented here. The initial concept implemented in first stage is shortly introduced in the next chapter; it did not have interactive qualities and was used essentially to test the technology and to prove its feasibility. Most of the findings presented in Chapter 6.1, however, come from this first stage and extensive indoor and outdoor testing of the prototype. It was also very useful to learn about the meSch process, the requirements for an on-site system, and the challenges of setting up an outdoor interaction in a place without constant network connection and power supply. The setup of the hardware, the physical design and production of the enclosure as well as the concept generation and the co-design workshop with volunteers were done in close cooperation with other team members and therefore will be treated as a given in the context of this work.

The need to introduce personalisation of content in context suggested to offer to the visitors the possibility to switch between different perspectives. This drew the revision of the concept to include an appealing tangible interaction for the task of switching these perspectives. This led to a new concept including a user interaction by putting a book mark onto a page and included a software redesign and implementation for importing, interpreting and navigating Narration Networks. The first revision had the network structure hardcoded and did not include a sensor implementation for the added user interaction. The focus of this work was the software design and implementation for realising and testing the suitability of the Narration Network approach.

5.1 The concept: a story of evolution

To set the scene to appreciate the prototype, a short introduction about the location and its special challenges and requirements is necessary. The meSch research team in Sheffield selected a local heritage in order to facilitate the design and test of technology under development. The team decided to opt for an outdoor setting as it complemented the more traditional heritage institutions already represented in meSch. The Sheffield General Cemetery was considered the most suitable as it offered a more intriguing and varied environment than the Botanical Gardens.

The cemetery was founded in 1836 and was an open, non-conformist cemetery until the 1970s. Today half of the cemetery is used as a park, while the other is a cultural heritage

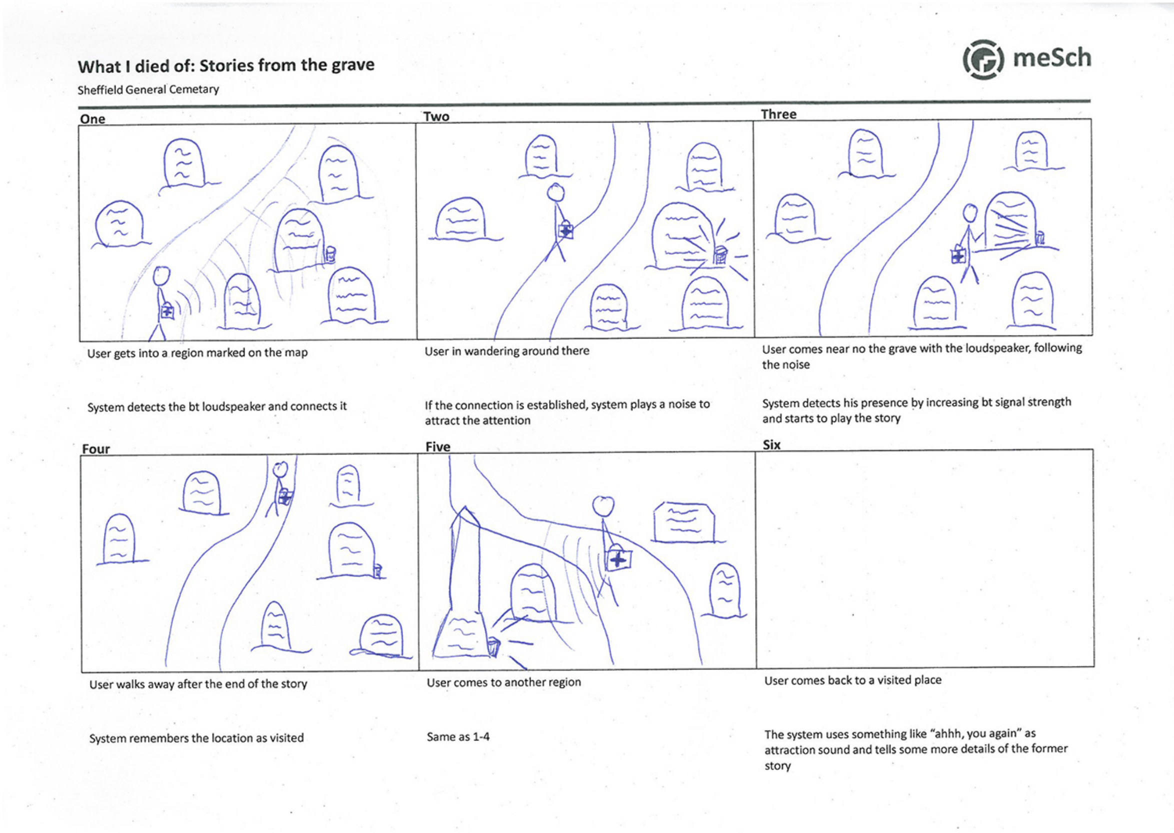

Figure 5.1: Storyboard for the first prototype concept

site preserving tombs, monuments and chapels of the old cemetery. The evocativeness of the cemetery and the presence of the many stories of the people buried there inspired the idea of a talking place. Both concepts follows the same basic design as well as using the idea of different zones of interaction and a map showing where to look for augmented spots in the graveyard. The idea of the different zones is reflected in the proximity trigger introduced in Chapter 4.7; on entering a certain range in the attraction zone a noise is played to draw the visitor to the actual hotspot. Being close to the point of interest in the engagement zone triggers playing of the actual content for this POI. On the technology side, independent powered Bluetooth loudspeakers seems to provide a cheap and easy way to realize this and the presence of a visitor at a place has to be detected anyway. A visitor carried device controlling the whole interaction was chosen to be the basic of the design. The Bluetooth speakers provided an unique identifier for every spot by using the MAC address and a signal strength measurement to make an estimation on the distance. The early concept, as depicted in the storyboard in Figure 5.1, exploits the notion of a Victorian doctors bag or apothecaries case as the object a visitor has to carry. In relation to the physical design the stories were meant to focus on medical topics like what the buried people died of (this was inspired by both a publication and organised visits that the Cemetery Trust holds). As this concept turned out to limit the choices of meaningful

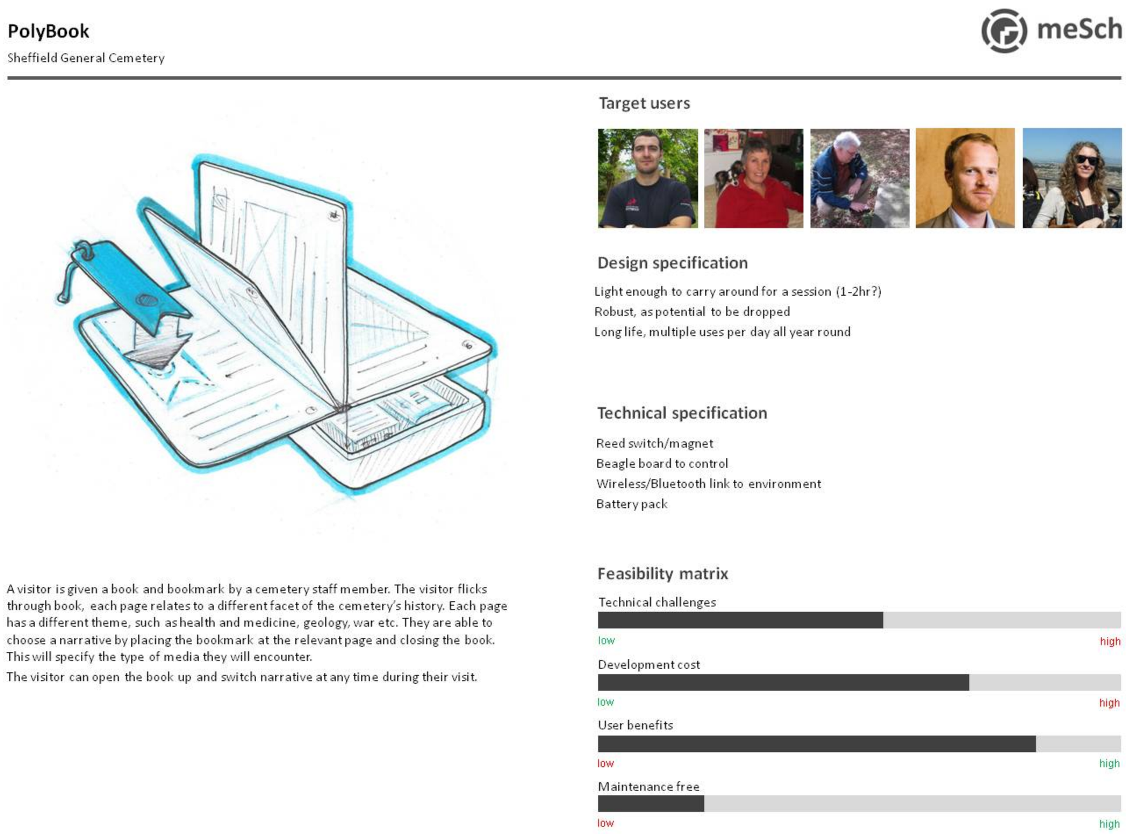

Figure 5.2: Sketch for the extended concept

perspectives that fitted the theme and the possible tangible interactions to switch between different perspectives the teams' product designers proposed a new concept, the Companion Novel, that exploits the notion of a book as shown in Figure 5.2. This concept is more generic and can accommodates several very different stories rather than a special theme (i.e. medical perspective); it opens up endless possibilities to explore a variety of perspectives. Browsing the pages of the book means browsing the available perspectives. By putting the bookmark onto the chosen page the perspective is selected. The bookmark can also be a personalised object with an ID that is associated with the visit and can be used to initiate an interaction after the visit as described in Chapter 5.5.

5.2 Making it tangible

The new concept of a book to select the perspective via a bookmark posed the physical challenge to house the technology, suggest the type of perspective each page had and implement the mechanisms to select the desired perspective by putting a bookmark in the associated page.

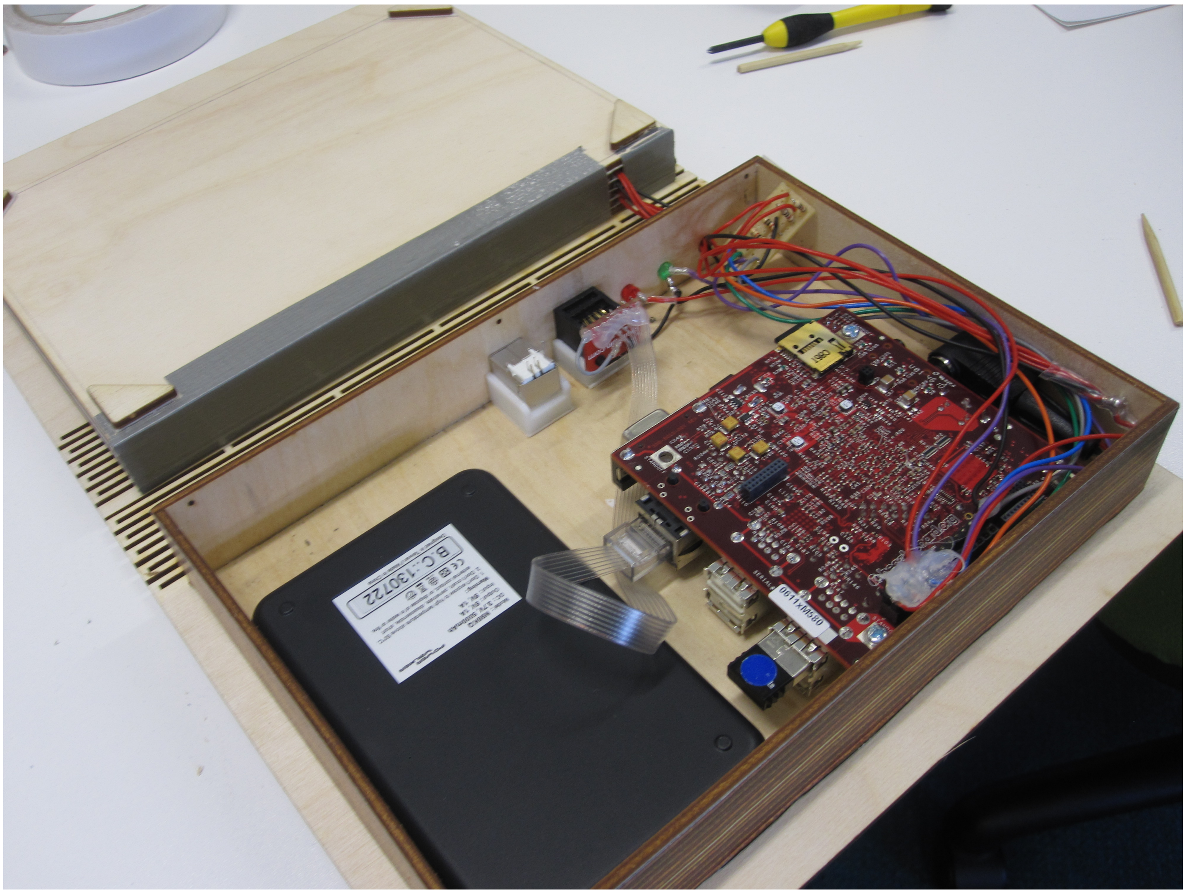

Figure 5.3: Technology embedded in the book

The case of the book is a wooden laser cut design; the BeagleBoard and the battery pack are placed in the right part of the book covered by the pages representing the perspectives. Reed switches are embedded in the pages together with additional small magnets that facilitate the placing of the magnetic bookmark into the right position to activate the perspective. Circuits are connected to the BeagleBoards GPIO to determine which perspective is selected. Figure 5.3 shows the case for the technology and an embedded reed switch. In the first draft a bigger battery pack and an enclosed version of the BeagleBoard were used, resulting in a bigger and heavier book than necessary. The battery was able to run the board for about 20 hours. By replacing it with a smaller one and fixing the BeagleBoard in place and removing its casing, the designers could make a lighter and smaller version that was easier to carry. As none of the components for sensing and presentation were affected by this redesign, software and hardware remained unchanged. This is a first hint that, if build with flexibility in mind, the physical form of a smart object can easily be changed. The pages are designed to suggest what can be expected when the perspective is selected. Each double sided page is dedicated to a perspective including a unique colour and a mark indicates where the bookmark should be placed. Every page consists of a short introductory text, a sample snippet of content and

Figure 5.4: Book page for the "Favourite Spots" perspective

some pictures. A complete page together with the bookmark in the right place can be seen in Figure 5.4. Feedback about the status of the BeagleBoard and the software two LEDs are integrated behind the binding showing if the BeagleBoard is running and if the software is running. They are mainly for debug purposes and therefore hidden from the visitor. An embedded speaker can be used to give auditive feedback on actions like placing the bookmark into the right place. The design also allows to operate the power switch of the battery pack as well as reset and user button of the BeagleBoard. Behind the binding there are connectors for Ethernet to enable remote control of the BeagleBoard over SSH and a USB socket for charging the battery pack. This allowed to easily check and update the software as well as get access to the log files during evaluation. In addition, by changing the network file or adding content, a completely new experience can be created without changing the hardware or casing, which is another advantage of the flexible, file based design.

The first implementation of the design started with simple, rechargeable, all contained Bluetooth loudspeakers as they were cheap, off the shelf and easy to use. The limited range of the integrated Bluetooth helped building attraction and engagement zones for proximity with reasonable threshold values. Lab experiments with several different speakers showed their behaviour in pairing and connection with the BeagleBoards can differ widely. The final design replaced the simple speakers due to their energy saving function (when not connected for a couple of minutes varying from 10-20 on the tested devices they deactivate their Bluetooth module and can now longer be discovered). The new devices to play sound were thus more sophisticated, they included Raspberry Pi devices with speakers attached with a normal audio cable powered by the same battery pack used for the BeagleBoard in the book. Bluetooth ability was added using USB Bluetooth dongles. Each Raspberry Pi runs on a Debian Linux and automatically logs in on boot up and starts a software making the device act as a Bluetooth audio sink the BeagleBoard can stream audio to. The battery pack powers the whole device for approximately 10 hours without recharging it. They were put in simple plastic boxes for the tests with a hole in the bottom for the speaker designed to hang them in trees and bushes at the cemetery. Having computing power and programming logic on every spot enables more complicated use cases for the Narration Networks. The content can be put on the devices in place and the playback is just triggered from the system in the book. This would only require a change in the implementation of the presentation device without affecting major parts of the network design and implementation. Although not implemented in the scope of this work it proofs the power of the loose coupling approach chosen for the Narration Network system design.

5.3 Software design

Following the idea of separation of concerns the design is built in an object-oriented approach where every class represents an encapsulated component. The first stage of the prototype followed the same pattern, just having the Narration Network hardcoded rather than importing it from an XML file and thus lacking the perspective trigger as well as the content class and using direct URIs as file system links. Figure 5.5 shows the components represented by classes. In principle this design allows to spread the components over several devices using proper communication mechanisms. Although the prototype software is written in Python which does not support object orientation very well, the software is designed and implemented in an object-oriented style to serve as a well structured show case for possible other implementations in meSch. The Bluetooth signal strength is measured as an indicator for proximity and the MAC address as location the BTConnection class implements both, the ISensorDevice and IPresentationDevice interface. The class is responsible for connecting to a nearby Bluetooth device; determine the signal strength and playing audio over an established connection to one of the known Bluetooth speakers. Using the sensor data the ContextEngine class creates context objects used by the Navigator class to select the next based on the current context. The Navigator class keeps sets of active and inactive nodes that are updated every time a node is selected. Only active nodes can be selected, this follows the design of Narration Networks explained in section 4.2. The instances of MeSchNodes, representing the nodes of the Narration Network, are created by parsing the XML file representing the Narration Network; it is also used for the initial state of the active and inactive set. The Context and Content class just encapsulates information necessary to describe a context or a content. Both are used in the

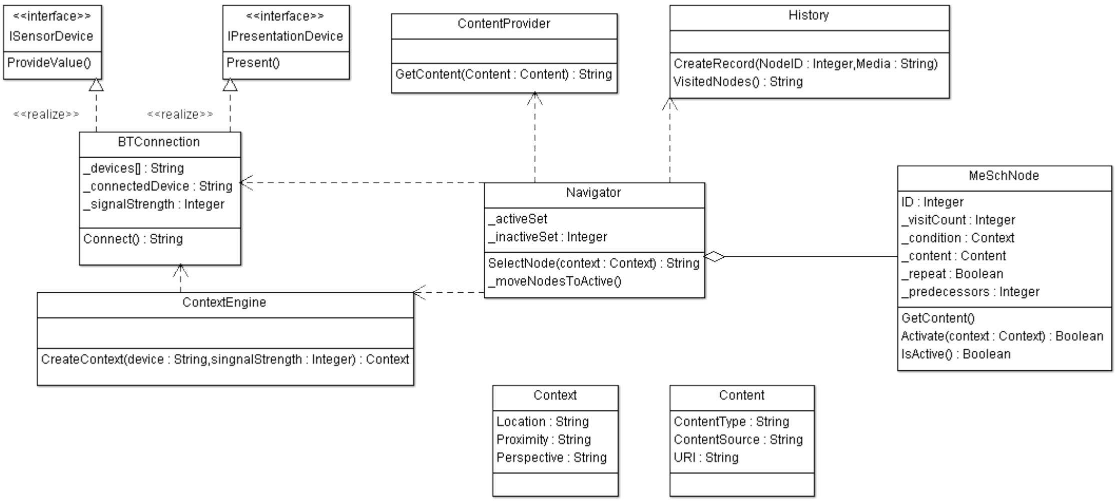

Figure 5.5: Prototype software class diagram

MeSchNode class, a context object to define the condition under which a node is activated, content objects to encode the actual content of a node. To resolve the information given in a content object to get the associated media file the ContentProvider class encapsulates the knowledge about the different content sources and how to connect and query them. Finally, the History class not only saves the history of all selected nodes, it also provides a list of visited nodes that can be used by the Navigator to determine the active set of nodes. The sequence of the calls to the methods of the different classed are depicted in Figure 5.6. The whole process is controlled by a main loop. The main loop reads the information from the sensors and hands it to the context engine to create a context object. With this context object it asks the navigator to select a node by calling the function handing over the recently created context object. The navigator takes the context object and asks every node in the set of active nodes if the context fits by calling their activate function. If a node is activated, the navigator grabs the content object and gives it to the content provider. The content provider uses the data given in the content object to select the associated content file. Before the navigator gives back the content file reference to the main loop it creates a record by calling the corresponding function from the history. In addition it updates the sets of active and inactive nodes by testing if the current node has still content that can be played on the next call and testing of the prerequisites of the inactive nodes. When finished, control returns to the main loop which calls the present function handing over the file reference. After the content has played the main loop starts from the beginning searching for nearby Bluetooth devices by calling the sensors functions.

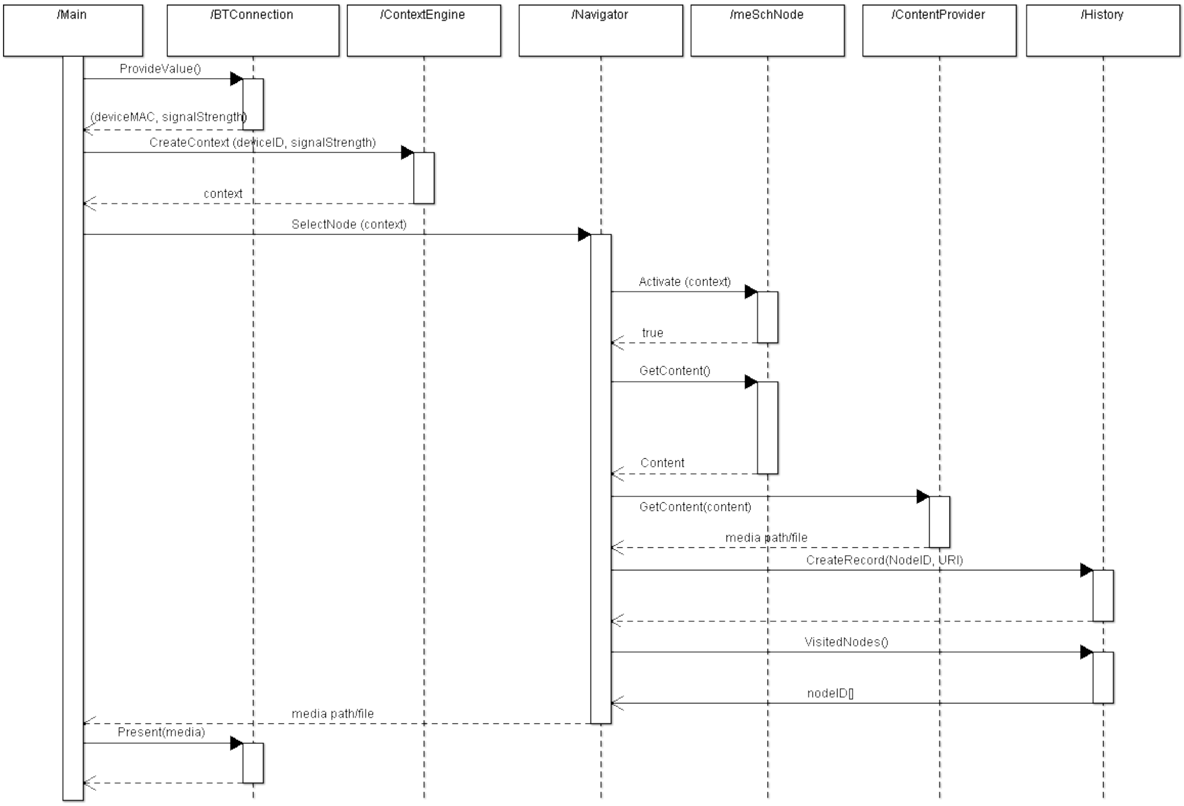

Figure 5.6: Prototype software sequence diagram

5.4 Implementation

As mentioned before the software is implemented using Python despite the fact that object orientation in Python is very basic and there are no constructs like interfaces or private members and methods. As agreed on convention, members and methods starting with an "’_"’ are viewed to be private. A BeagleBoard-xM is the basic hardware platform running a Debian Linux, Bluetooth capability is added using a USB Bluetooth stick. The board is independently powered by an external battery pack with 12.000mAH capacity and 5V 2.1A output which enables the prototype to run for up to 8 hours. The programmable button of the BeagleBoard is used for two functions. The first press of the button starts the main loop of the software after the Board has booted up, second press shuts the board down securely. This allows to operate the board without attached keyboard. To provide simple feedback without an attached monitor the board gives audio signals over its onboard audio output when it is fully booted up and ready to start the software and every time the main loop enters a new cycle. It can be monitored by attaching headphones. The button control and audio feedback were extremely helpful during testing the prototype in outdoor settings without keyboard or monitor available, in theory the software can automatically started when the board finished booting up.

The main loop runs infinitely using a while true construct. At the beginning of each loop it plays a short sound and creates a log entry for later reviews and analytics. During the loop at the beginning the sensors are checked. If a known Bluetooth device is found, a context object is created and used to activate a node and get content that can be played using the connected Bluetooth device. If an attraction sound is played the loop sleeps for 3 seconds before updating the sensor data to give the visitor enough time to get near to the point of interest. If an engagement sound is played the loop sleeps for 30 seconds giving the visitor enough time to work away without instantly presenting him subsequent audio snippets. All values were assessed trough ad hoc testing with the prototype. If no device is found the loop starts immediately again. There is an additional simulation mode in which the software can be used. It replaces the calls to sensor and presentation devices with directly reading the intended context from a text file. This allows to test the conditions and the content without setting up the whole environment or an outdoor test.

Using the bluez utility to discover and connect to Bluetooth devices introduces a basic 11 seconds delay. A discovery run for approximately 8 seconds and tries to find all Bluetooth devices in range. A list of nearby devices is returned which is compared with the list of known devices in a for-loop. The first match is taken and a connection attempt is made. Another 3 seconds are added by the connection process. If a device is already connected this process is bypassed and just the signal strength is updated. For managing all Bluetooth related actions the dbus interface is used. All devices have to be paired with the BeagleBoard and the used Bluetooth stick in advance and for every used device a PCM entry for the aplay utility has to be created. This allows aplay to stream audio output using the specified device. For getting the signal strength the hcitool utility is used. The signal strength measurement is a numeric 8 bit value but never seems to be below 200 in all tests. Both the hcitool and the aplay utility are called using Pythons command line interface.

The context engine uses simple if-then-else rules to evaluate the signal strength and create the proximity based on fixed thresholds with 210 as lower bound for the attraction proximity and 245 as boundary between attraction and engagement. The different perspectives are selected by getting the value from the BeagleBoards GPIO and use if-then-else constructs to map them to one of the four different perspectives. The GPIO inputs are wired with magnetic switches embedded into the pages of the book representing the perspective. The bookmark contains another magnet, if the bookmark is placed on a page the circuit is completed and the GPIO gets a signal from the switch.

Three tasks are assigned to the navigator. On program start the XMLHelper class is used to import the Narration Network, create node objects and put them into the active or inactive set based on the predecessor definition. Only nodes without predecessor are in the active set initially. Getting a new context object in the SelectNode() method every node in the active set is tested on activation. The first node that can be activated is taken and the content is given to the content provider to return the file reference. The active and inactive sets are updated before returning control to the main loop. If the current node has no more content to present it is removed from the active set, all nodes from the inactive set are added that had the current node as predecessor. The code for the navigator can be found in A.1. The XMLHelper uses the ElementTree package to parse the Narration Network file and create an in-memory model.

The model is traversed from the root node and MeSchNode objects are created along with their corresponding contents and condition in the form of a context object. The code for the import function is available in A.2, the _createContent() and _createCondition() functions just read the values from the XML nodes.

The MeSchNode class itself is kept simple. Activation just checks the given context object against the own condition, the equals method to support direct comparison is implemented in the Context class. If a trigger is not defined, meaning there is no value assigned, it is interpreted as a wildcard for matching the activation condition. Arrays of content objects and IDs hold the information about contents and predecessors. The repeat flag indicates if a piece of content can be played multiple times during a run. This makes content selection and active check a bit more challenging, but the central part of the navigation logic is handled in the navigator. A.3 shows the three public methods a meSch node offers.

The content provider is tailored to the needs of the prototype and just strips away the file:// prefix from the URI to create a valid path to the referenced media file in the given content object. It returns not only the path but also the duration of the referenced media file that is used in the aplay command call. Adding other, remote content sources would increase the complexity of the content provider by magnitudes but is not needed in the case of this prototype.

Finally, the History class keeps track of all visited nodes to provide this information to the navigator for determine the set of active nodes and writes a record to a log file every time a piece of content is played. Following the definition for log files a comma separated value file is created where each row equals one piece of content represented by the date and time the content is played, the references to the Narration Network and the node by their IDs and the URI of the played content. This can easily be converted in the XML format introduced in 3.2.

5.5 End-2-End scenario

Having all parts of the system presented it is time to outline how they fit together to support the overall meSch cycle. To foster understanding this is done alongside with an example for the prototype presented in this chapter. As there is a four year timeframe for meSch this scenario does not cover all details for a possible implementation, it is meant as a possible process description and points out needed documents and technology. Neither the template system nor any kind of visual editor is part of this work as they will reside in the meSch server environment.

5.5.1 Creating a template

To be able to use a design concept and share it with other meSch users the first thing to do is to create a meSch template for the concept. The core of the template is a XML file containing the description of the concept and defining the available triggers and contents. In addition

Listing 5.1 Example template file

```
<concept name="Talking places"
	desc="plays a piece of audio in every location">
	<pois limit="0">
		<content_nodes limit="0">
			<triggers>
				<trigger name="proximity" desc="Proximity from a visitor to a
				    POI">
					<value>Attraction</value>
					<value>Engagement</value>
				</trigger>
				<trigger name="location" desc="MAC address of a device">
					<pattern>([0-9A-F]{2}[:-]){5}([0-9A-F]{2})</pattern>
				</trigger>
				<trigger name="perspective" desc="Perspective from which the
				    narration is told">
					<value>Gravedigger</value>
				</trigger>
			</triggers>
			<contents limit="0">
				<content desc="PCM encoded .wav files">audio</media>
			</contents>
		</content_nodes>
	</pois>
</concept>
```

it should contain necessary software, hardware specifications and a manual for setting it up. Choosing an existing template and focus on creating the content would support curators with less technical and interaction design experience. If a museum wants to create something new they can start from scratch and create a new template. The template file serves as description for the authoring tool. It tells the authoring tool how many nodes can be created, which triggers can be used on a node and how many and what kind of content can be attached to a node. Listing 5.1 shows an example for the Bluetooth prototype. It is supposed to support an unlimited amount of nodes, every node can be triggered by a proximity and a location determined by the MAC address of a nearby loudspeaker. Every node can contain unlimited amounts of audio information. Descriptions could serve as a help in the authoring tool.

5.5.2 Creating a narrative and customizing the template

Having selected a template the authoring tool creates a GUI based on the content of the template file. The authoring process can be divided into three stages. Create a set of content nodes and group them together at points of interest is the first stage. Second stage is creating a narrative that can either consist of unconnected pieces of content or support a rich graph structure. In the third and last stage the narrative is distributed over the created nodes. As supposed in meSch the content can come from a variety of integrated sources like cultural heritage databases or public domain media repositories as well as from the museums' own data. The GUI should allow simulating the network by creating contexts of triggers and see if

Listing 5.2 Part of an example Narration Network

```
<?xml version="1.0" encoding="UTF-8"?>
<narration_network id="1">
	<poi id="1">
		<content_node id="1" repeat="false">
			<trigger>
				<location>00:15:83:46:67:61</location>
				<perspective>1</perspective>
				<proximity>attraction</proximity>
			</trigger>
			<content>
				<type>audio</type>
				<source>filesystem</source>
				<duration>10</duration>
				<reference>file:///home/debian/mesch2/sounds/digging.wav</reference>
			</content>
		</content_node>
		<content_node id="2" repeat="false">
			<trigger>
				<location>00:15:83:46:67:61</location>
				<perspective>1</perspective>
				<proximity>engagement</proximity>
			</trigger>
			<content>
				<type>audio</type>
				<source>filesystem</source>
				<duration>23</duration>
				<reference>file:///home/debian/mesch2/sounds/content1_1.wav</reference>
			</content>
		</content_node>
	</poi>
	[...]
</narration_network>
```

the right piece of content is presented. After successful testing the authored information is exported as a XML file describing the Narration Network that has been created. It then can be used to drive the experience on site in the next step. Being an independent file allows a system without any connection to the meSch server. Listing 5.2 shows part of a Narration Network for the Bluetooth prototype. For every POI there is a content node for the attraction and engagement proximity. The contents of node 4 are only played if the visitor has heard the story from node 2 already.

Besides the content the creation of suitable casings for the technology is a part of the customization process. It allows museums to create physical objects that relate to their theme and the narrative. Replicas of exhibits can play an important role in this and become “smart replicas”. Mapping of controls can be a challenge depending on the interaction style the concept covers. For the Bluetooth concept the user device is a book containing the technology and the loudspeakers are dressed up as wreath or hidden in a bird box with a solar panel to recharge the loudspeaker (see 5.1 for pictures).

5.5.3 Deploying the system on site

For deployment three tasks have to be fulfilled. The technology must be fitted into the cases and distributed on the site. Assisting technology from the template has to be installed as well. Then the system must be loaded with the Narration Network file and the content database has to be copied to the right device, if a local data source is used. The template should deliver all information that is needed to set it up. Maybe some modification of the code is required or the manual states other necessary setup actions. Before allowing visitors to interact with the system it should be tested carefully. Deploying the BT prototype requires putting the Narration Network file and the database with the audio contents on the Beagleboard[1] and putting the Beagleboard together with the battery pack into the book. The BT loudspeakers have to be put into the right places in the cemetery and there must be a point where visitors can get the book and their personal bookmark and leave the book when they finished the visit.

Listing 5.3 Example visiting log

```
<?xml version="1.0" encoding="UTF-8"?>
<activity_log>
        <log_entry>
                <nn_id>1</nn_id>
                <content_node_id>1</content_node_id>
                <content_uri>file:///home/debian/mesch2/sounds/digging.wav</content_uri>
                <timestamp>2013-08-15T16:29:34</timestamp>
                <visitor_id>john.doe@example.com</visitor_id>
        </log_entry>
        <log_entry>
                <nn_id>1</nn_id>
                <content_node_id>2</content_node_id>
                <content_uri>file:///home/debian/mesch2/sounds/content1_1.wav</content_uri>
                <timestamp>2013-08-15T16:29:47</timestamp>
                <visitor_id>john.doe@example.com</visitor_id>
        </log_entry>
       [...]
</activity_log>
```

5.5.4 Collecting visitor logs

The logging and the rationale behind it are described in more detail in 3.2. Keeping track of the visitors interactions allows the creation of an after visit online experience, as well as studies about the exhibition and the interaction concept itself. They are sent back to the meSch server as XML records, live if a connection is available or summed up in a later point of time if not. The BT prototype transfers all log data from a visit after the book has been handed back. Listing 5.3 shows an example log file. It contains references to a network and a

[1] http://beagleboard.org

Figure 5.7: Example website with map of a visit

node, knowing the associated network structure allows creating a meaningful picture of the visit from the collected data.

5.5.5 Creating an online experience after the visit

Knowledge about the visitor's interactions on the site enables analysis for curators and designers but, more important allows relationship with the museum to be built by offering personalized content after the visitor has left. It can cover the visitor's path through the site, add additional information to be explored and maybe collect some feedback or even user created content. To foster this kind of connection to the museum after the visit it might be a good idea to give the visitors a physical object as a reminder that also can be used as a tag for identification using a web cam or carry a personalized code. The BT prototype covers this idea by having a personalized bookmark for every visitor. The bookmark has a QR code printed on it and a hint where to find the website. The code can be scanned using a webcam and is used to create a map reflecting the visitor's history on the site by showing hotspots where he spent time and blurring the rest of the map as depicted in 5.7.

6 Evaluation

The evaluation is divided into several parts that map the iterative process of development and the multiple purposes the prototype implementation serves. Specifically, the prototype was assessed by experts and this is reported first; then a number of lab and field test were carried out to assess the technology overall. This included feedback from the cemetery volunteers who took part in a workshop. The purpose of this second set of tests was to ensure the creation of a reliable system and to test different technologies for their general suitability for the problem domain. As this work is concerned with the software and the system to create smart objects in a flexible and generic way, the focus is on technical aspects. Indeed it does not matter if the object is a book and the perspective selection is done via a bookmark onto a page or if the object is a apothecary case and the selection is done by putting a bottle in a specific place: what is core is that the software is the same and works with both physical designs. Therefore mainly the technical and setup side of the prototype has been evaluated in detail in the context of this work.

The further two sections focus on the concept of Narration Networks rather than the actual prototype implementation. The creation of sample networks for other concepts was done to test if the network concept can be generalised to cover the whole range of possible meSch designs for smart objects. Creating further concepts by porting existing technology or using the simulation software to alter content and network design closes this chapter. The aim of this wide scope of evaluation is to see if the whole construction is able to address the key challenges for the meSch process pointed out in Chapter 2.

6.1 Expert evaluation

This section covers ad-hoc testing in the laboratory and in the field with both stages of the prototype for both hardware and software. It is split in: (1) technical insights, aimed to support progress in the meSch project and further prototyping and (2) insights about interaction, setting and place influencing the definition of Narration Networks. All tests included extensive logging on the device as well as documentation of sites and process with video and pictures. The speakers were put in several places during both, lab tests as well as tests in outdoor open spaces, to measure maximum connection ranges and timing. All tests included time and range measurements, but also included passing judgement on how well attraction and engagement audio can be heard over distance.

6.1.1 Technology and prototyping

This section relates to problems with the Bluetooth technology used and findings about testing prototypes in an outdoor environment.

Pairing
Before they can be used, Bluetooth devices have to be paired which means getting to know each other, agree on communication paradigm and authorise each other. Different Bluetooth loudspeakers might use different methods for authorisation and the bluez utility has to be configured in the right way to support the currently needed one. After unplugging or exchanging the Bluetooth stick, devices have to be re-paired every time.

Discovery
The discovery process runs for a fixed time which is usually 8-10 seconds. It is not reliable, a nearby Bluetooth device might but must not be discovered immediately in the first run, sometimes it takes several runs to find all nearby Bluetooth devices. During the test runs with isolated speakers the detection range appeared to be up to 25 meters. When the speakers were paired with Raspberry Pi devices the range increased to over 30 meters with a much higher initial signal strength.

Connection
The connection process proofed to be very reliable and robust during the tests. Once connected the signal is very stable as long as no physical obstructions come into play. Problems occur when using forced disconnection and trying to immediately reconnect after it, as the connection is established but the playing does not occur. Using Raspberry Pi devices problems were encountered with audio being muted sometimes after 15-20 seconds of playback. This was tracked down as a pulseaudio error and could be fixed by automatically logging in after startup.

Feedback
In the first prototype the software was starting up automatically and provided no feedback. During outdoor tests, this made it impossible to monitor the status of the device and to pinpoint errors if the Companion Novel was not working as expected. As a consequence, the control and feedback features mentioned in Chapter 5.4 were introduced. The last version added a speaker and LEDs integrated into the book for status handling and tester/user feedback.

Shutdown
Having no keyboard or display available during outdoor tests makes it hard to properly shut down the BeagleBoard after usage or between two tests to save battery. Unplugging the power supply causes the corruption of the SD card used as hard drive for the BeagleBoard because of write commands that did not complete properly when abruptly switching off the board. This led to the implementation of a shutdown command triggered by the BeagleBoard's user button. In the last version the button was used to start and stop the software, to shut down the BeagleBoard and a laptop was used to remote control the BeagleBoard over Ethernet.

6.1.2 Interaction and Narration Networks

This section relates to findings about the suitability of the technology used for the intended interaction styles and requirements and details that came up regarding the design of Narration Networks. The prototype was assessed for the interaction as a whole by the Sheffield meSch team. The expert evaluation used a walkthrough method: the different steps in the interaction were considered, possible interaction issues were noted, and the design was revisited accordingly.

Feedback
One items that emerged in the walkthrough was related to the lack of feedback for setting a theme via the bookmark. In the revised version of the book the attraction sound for that theme is played when the bookmark is placed on the page (an internal control bypass this step if the bookmark is moved while on a POI as it is understood that the visitor just want to listen to another story).

Orientation
The key question here was how to find the points of interest in the 14 acres of the cemetery. Some kind of orientation has to given to the visitor. To support orientation and navigation in the cemetery a map showing the points of interest has been etched on the back of the book.

Proximity
The Bluetooth signal strength measurement used as indicator for proximity turned out to be highly dependent on the environment as well as the direction of movement. This indicates that neither a metric range measure nor a configurable parameter is suitable for this kind of design and draws the need for more generic forms of proximity that can be used in the authoring environment. It also draws the need for a uniform design of playback devices in the points of interest.

Volume
This goes straight along with problems of proximity measurement. If the attraction is picked up on 25m range, visitors might not be able to hear the attraction sound. If volume is increased and the attraction is picked up 5m away visitors might get frightened by it. The nature of the sound also had a great influence; while a horse whinny worked very well it was hard to hear the sound of someone digging with a shovel. Additionally the overall background noise plays an important role which might even vary based on the season or the time of day when there is more noise from birds or traffic.

File type
Using the aplay utility on a Debian Linux without additional software just allows to play PCM encoded .wav files over a Bluetooth connection. Although this is a problem that can be addressed using different technology it seems reasonable to offer a designer possibilities to constraint the potential media usage in his template definition when there are technical limitations.

Listing 6.1 Technical log file data

```
Searching: 2000-01-01 01:53:33.900511
Set: 00:15:83:46:67:61,255,2000-01-01 01:53:51.493803
2000-01-01 01:53:51.591612,2,6,file:///home/debian/mesch2/sounds/content1_c.wav
Searching: 2000-01-01 01:55:58.468260
Update: 00:15:83:46:67:61 255,2000-01-01 01:55:58.503264
OOC:00:15:83:46:67:61|engagement|c, 2000-01-01 01:55:58.897154
Searching: 2000-01-01 01:56:06.762755
Update: 00:15:83:46:67:61 255,2000-01-01 01:56:06.798644
OOC:00:15:83:46:67:61|engagement|c, 2000-01-01 01:56:07.144621
Searching: 2000-01-01 01:56:15.015959
Update: 00:15:83:46:67:61 255,2000-01-01 01:56:15.051207
OOC:00:15:83:46:67:61|engagement|c, 2000-01-01 01:56:15.454192
Searching: 2000-01-01 01:56:23.322691
Update: 00:15:83:46:67:61 255,2000-01-01 01:56:23.358427
2000-01-01 01:56:23.391966,2,2,file:///home/debian/mesch2/sounds/content1_a.wav
```

6.2 Field-test the prototype

A series of tests in the Cemetery with the complete system (hardware and software as well as attraction sounds and content) were conducted to check the prototype in its intended field of use. A total of 7 points of interest were defined along with 4 different perspectives for the content. Two of the points of interest were used as way points, playing the same content independent from the selected perspective, to guide the visitor through the cemetery. These way finding snippets are played every time a visitor comes by, while the attraction and content snippets for points with many perspectives are only played once. For each perspective an attraction sound was defined. A total of 20 content snippets were recorded with a length between 26 and 121 seconds. The cases with the batteries, the Raspberry Pies and the Bluetooth loudspeakers were put into trees or bushed near the gravestones that marked the points of interest. Two researchers then tested the system, one walked the Cemetery ground, the other said close to the loudspeaker to check when the attraction sound first and then the story were played.

These tests were aimed to find out if the whole interaction concept works out well and how much effort it is to find the right parameters for attraction and engagement zones as well as volume of the audio and length of the attraction and content snippets. Data collection mainly relied on the log files, personal observations and notes during the tests. The log files were split into one log focusing on interactions and designed to be imported into statistical analysis tools for later use in user evaluation and a technical log recording the system events. The list 6.1 shows part of the technical log file from a lab test: it simulates that a visitor is at the first POI, hears the content from perspective c and then switches the bookmark to get content from perspective a. The *OOC: (out of content)* entry means that there is no more content to play for the current POI in the the current perspective. The log also shows errors from Bluetooth connection and perspective selection.

As discussed above, the tests showed that the signal strength measurement and the zones created from it are a big challenge. The cemetery is a mixture of wide open spots as well as

Figure 6.1: Pictures showing the different parts of the cemetery

overgrown and tidy spots packed with gravestones. For this huge variation in the cemetery setting from one area to another, the attraction distance varies greatly and makes it hard to find a common set of threshold values for attraction and engagement that works well across all the spots. Figure 6.1 shows two panorama pictures, the upper one from the more open part of the park, the lower one from the overgrown part of the cemetery. Depending on the time of day and the exact place in the cemetery the background noise also varies greatly. In the upper parts there is much noise from a major close by street, especially in the morning and in the evening during rush hour. In the parts with more trees there is a lot of bird singing noise in the morning. Given all these challenges only a few spots worked reasonably well during the tests. On some spots no audio could be heard although log analyses showed that sounds have been played. On other spots it was possible to pass by the spot without getting a connection in time and, for fast walkers, to be out of the connection range again before any sound has been played. In the spots where the prototype worked properly the Companion Novel turned out to be very strong in terms of attracting the visitor and making the place talk to them as in a demo video recorded in the cemetery in the Summer[1]. The horse sound drags the visitor from the path to the grave under a tree where he is told the story of the buried soldier.

The feedback from the volunteers complemented the field tests. Issues discussed were the need for a minimal set of instructions on what to do with the book (in the revised prototype the first page explains what the bookmark does and invites visitors to take their preferred path), and the volume of the sounds. The volunteers' concern with sound was not, as one might expect, if it annoyed other visitors: the Cemetery is a large, public space and visitors may just divert to a quieter path if sound is played nearby. Rather, the sound was deemed possibly too quiet to be heard as the place is loud with birds. They also observed that the vegetation in Summer is dense and it may further dampen the volume. On the bases of this comment the type of attraction sounds selected were very loud and unexpected for a park such as a traffic jam, a piece of opera, a rap, and a high neigh.

The feedback from the field tests was directly used for the iterative improvement of the prototype: the current ongoing work encompasses a revision of the Bluetooth communication protocol to create a reliable and robust technology infrastructure for a field user evaluation. Analyses of the logs collected in the wild test proofed the software navigating the Narration Network worked correctly: based on the state of the current context the right content was played, already played snippets were removed and not played again. Perspective switches were carried out correctly and the perspective defaulted back to the 'a' setting if the bookmark was removed with on new position recognised. Finally, the controls to reset and shut down the prototype without opening the case proofed to be very helpful.

6.3 Create sample Narration Networks for other concepts

Based on the experience with the Companion Novel the concept of Narration Networks worked well: the model of the narrative was simple to create and the interaction was as expected. But as meSch has to cover a wide variety of possible concepts, the Narration Networks have to support them all. The whole design was carefully crafted with this core requirement in mind: that novel ideas and templates that will emerge in the project lifetime should be modelled and rendered throughout he Narration Network and its software. To add some proof, this section will present sample trigger definitions and Narration Networks for other concepts generated by the SHU team (for some the hardware is currently under development). The following examples showcase and explore the possibilities of Narration Networks, some of them will be implemented using the same software as the Companion Novel. The concepts discussed include the mourning jewellery that guides a visitor to a specific grave, augmented reality goggles that lets visitors take a look inside the catacombs or see the park as it was, a projective torch bringing gravestones to life, a shadow projecting bird house and a victorian group exploration kit. The networks shown here are only one possible solution, designers may want to hardcode some of the parameters instead of offering them as triggers.

[1]http://www.youtube.com/watch?v=GP0wAPO84Qo&feature=player_embedded

Listing 6.2 Jewellery sample network

```
<?xml version="1.0" encoding="UTF-8"?>
<narration_network id="2">
	<poi id="1">
		<content_node id="1" repeat="true">
			<trigger>
				<location>936DA01F-9ABD-4D9D-80C7-02AF85C822A8</location>
				<proximity>far</proximity>
				<action></action>
			</trigger>
			<content>
				<type>action</type>
				<source>script</source>
				<duration>0</duration>
				<reference>file:///home/debian/mesch2/actions/vibrate3sec.sh</reference>
			</content>
		</content_node>
		[...]
		<content_node id="4" repeat="false">
			<trigger>
				<location>936DA01F-9ABD-4D9D-80C7-02AF85C822A8</location>
				<proximity>near</proximity>
				<action>cover</action>
			</trigger>
			<content>
				<type>action</type>
				<source>script</source>
				<duration>0</duration>
				<reference>file:///home/debian/mesch2/actions/stop_vibrate.sh</reference>
			</content>
		</content_node>
	</poi>
</narration_network>
```

6.3.1 The Jewellery

The mourning jewellery concept is designed in the style of Victorian mourning jewellery and is linked to a specific grave. Each jewellery is associated to a person buried in the cemetery. While walking the cemetery it helps the visitor to find the grave by vibrating or progressively getting warmer the nearer one gets to the grave. When on the right spot, the visitor's then acknowledge being there by, for example, blowing on the jewellery to "cool it down". This design requires triggers for proximity and location. Obviously proximity needs several values to define the right behaviour, many more than the two used for the Companion Novel. Generic action triggers can be used to model the interactions. List 6.2 shows the sample network, the actions are encoded in scripts forcing the jewellery to vibrate every 3, 2 or 1 seconds. The action trigger for cover is activated when the visitors holds the jewellery in his hand for at least 3 seconds. A duration of 0 indicates to run the action as long as the condition is met.

Listing 6.3 AR Goggles sample network

```
<?xml version="1.0" encoding="UTF-8"?>
<narration_network id="3">
	<poi id="1">
		<content_node id="1" repeat="true">
			<trigger>
				<location>/home/debian/mesch2/img/marker1.png</location>
			</trigger>
			<content>
				<type>model</type>
				<source>filesystem</source>
				<duration>0</duration>
				<reference>file:///home/debian/mesch2/models/cave.max</reference>
			</content>
		</content_node>
	</poi>
	<poi id="2">
	[...]
	</poi>
</narration_network>
```

6.3.2 AR Goggles

The catacombs in the graveyard are visible as a set of arcs looking like entrances. Originally they were open to view and locked through iron gates. If a visitors looks at one of this arcs with the augmented reality goggles, he will be presented with a model of the room inside fitted in the arc, so with a view of how it was in Victorian times. The arc the visitor looks at can be seen as a location trigger. For replacing the wall with a 3D model of the room behind it a new content type has to be defined referencing the respective 3D scene or model to display. Locations can be defined by markers which are needed for the technology to work as early experiments suggest. The meSch logo proved to be a good marker in the field test. The network shown in 6.3 is very simple, it defines a marker and a 3D model file to display if the marker is detected.

6.3.3 Projective Torch

A torch is a tool of exploration, designed to shed light on things and make them visible. This can be used to project contextual information on gravestones. Texts, images and even short video clips are possible contents. Changing the colour of the light can be used to switch between different perspectives. As proximity is integrated by using the projector, this system only needs to identify locations and handle perspectives. Graves might use an infrared marker seen by an infrared camera mounted on the torch. This would ensure the torch activates only if it is pointed on the right gravestone. The network in 6.4 defines two different types of content for a place depending on the selected perspective.

Listing 6.4 Projective Torch sample network

```
<?xml version="1.0" encoding="UTF-8"?>
<narration_network id="4">
	<poi id="1">
		<content_node id="1" repeat="true">
			<trigger>
				<location>/home/debian/mesch2/img/marker1.png</location>
				<perspective>1</perspective>
			</trigger>
			<content>
				<type>image</type>
				<source>filesystem</source>
				<duration>0</duration>
				<reference>file:///home/debian/mesch2/img/portrait_mark_firth.png</reference>
			</content>
		</content_node>
		<content_node id="2" repeat="true">
			<trigger>
				<location>/home/debian/mesch2/img/marker1.png</location>
				<perspective>2</perspective>
			</trigger>
			<content>
				<type>text</type>
				<source>filesystem</source>
				<duration>0</duration>
				<reference>file:///home/debian/mesch2/img/firth_bio.txt</reference>
			</content>
		</content_node>
	</poi>
	<poi id="2">
	[...]
	</poi>
</narration_network>
```

6.3.4 Bird Shadows

This concept differs from the others by not requiring the visitor to carry any device. It is meant to be mounted on a tree and projects shadows of birds according to the season and the time of day when visitors pass by. A binary proximity trigger can be used to model presence of visitors. The Cemetery volunteers suggested to change the shadows depending on the season or the daytime: triggers then have to be added to cover the idea of changing bird shadow. Based on the desired degree of realism a video or image would be the content to present. The sample in 6.5 shows two nodes for moving shadows in the morning depending on the season. Defining more than one bird for a node allows to show alternating samples, even to people that stay a little while next to the bird house. The proximity trigger might even be skipped and the function hard coded in the software because it is the same for every node.

Listing 6.5 Bird Shadows sample network

```
<?xml version="1.0" encoding="UTF-8"?>
<narration_network id="5">
	<poi id="1">
		<content_node id="1" repeat="true">
			<trigger>
				<proximity>attraction</proximity>
				<daytime>morning</daytime>
				<season>autumn</season>
			</trigger>
			<content>
				<type>video</type>
				<source>filesystem</source>
				<duration>10</duration>
				<reference>file:///home/debian/mesch2/videos/hawk.mp4</reference>
			</content>
			<content>
				<type>video</type>
				<source>filesystem</source>
				<duration>10</duration>
				<reference>file:///home/debian/mesch2/videos/raven.mp4</reference>
			</content>
		</content_node>
		<content_node id="2" repeat="true">
			<trigger>
				<proximity>attraction</proximity>
				<daytime>morning</daytime>
				<season>summer</season>
			</trigger>
			<content>
				<type>video</type>
				<source>filesystem</source>
				<duration>10</duration>
				<reference>file:///home/debian/mesch2/videos/swallow.mp4</reference>
			</content>
		</content_node>
		[...]
	</poi>
</narration_network>
```

6.3.5 Group Exploration Kit

The kit consists of three victorian styles items, a hat, a stethoscope and glasses. All three have integrated headphones to be able to present pieces of content. To foster interaction in a group, every item plays only part of the story and the visitors have to work together and tell each other the parts to get a complete picture. To coordinate the information flow, each piece has to be location aware and has to define an action to activate the playback. In a basic implementation each device could run its own Narration Network, making implementation easier. If they run from a central source to be able to link the nodes played on different devices, an additional trigger for the kind of device is needed. The sample in List 6.6 shows only a

Listing 6.6 Group Kit sample network

```
<?xml version="1.0" encoding="UTF-8"?>
<narration_network id="6">
	<poi id="1">
		<content_node id="1" repeat="false">
			<trigger>
				<location>53.36887,-1.48798</location>
				<action>auscultate</action>
			</trigger>
			<content>
				<type>audio</type>
				<source>filesystem</source>
				<duration>17</duration>
				<reference>file:///home/debian/mesch2/audio/story_med_1.mp3</reference>
			</content>
		</content_node>
	</poi>
	<poi id="2">
	[...]
	</poi>
</narration_network>
```

small part for the stethoscope, the location is based on GPS coordinates to define a place, the action to trigger the playback is auscultating a gravestone.

Although just showing parts of the networks the overall conclusion is that interactions and contents of all concepts can be described using Narration Networks. Additional triggers and content types had to be added, as it can be expected when new interaction design concept is created. What is important is that the model designed and implemented seems able to serve these other very different scenarios that are not particularly similar to the concept they were introduced for.

While the notion of input-action triggers works well and is easy to understand, defining tactile feedback does not fit very well into Narration Networks. The workaround for the jewellery, defining scripts for the tactile feedback, seems to be complicated to author for non-technical users. Some of the used triggers as well as actions could be coded into the template software rather than offered to the author. The proximity trigger for the bird shadows concept does obviously not give much choice or control to the author and might just be a hard-coded feature described in explanation for the template. The jewellery feedback could also be completely fixed and kept internal; what the author has to do is then to associate a piece of jewellery with a place on the site. It is up to the designers and template creators what to offer to authors.

6.4 Supporting the design of templates

The simulation tool for Narration Networks proofed to be very useful for designers to experiment with different narrative structures and see "how does it feel". It was initially used to create and test the context for the Companion Novel. Further experiments showed the potential for experimenting with new story templates. Two of them have been created and tested with the help of the simulation tool. The first one contains a strict sequence of audio snippets. The simulator was used to test which type of content is more effective when the visitor is out of track: a friendly and simple "this is the wrong station, go back to 2" or a more cheeky message like "you won't trick me to tell you the end of the story before the time! go back to station 2!" The second template was more complex in the sense that the layout of the story looks like a simple hypertext and the visitor is offered the choice of where to go next. In this case the challenge for the interaction designer and the storyteller is to guarantee that the story is fluid and coherent no matter which path the visitor will take. Feedback from designers showed that the simulator is used to find out which is the best network and to identify and remove mistakes (wrong content played or wrong condition set). In a more general sense it supports moving the attention from hardware and software details onto more interactive and emotional aspects of design.

As portability of solutions to other heritage sites was one of the key challenges when designing the system in this thesis, a port to another heritage site seemed to be a reasonable test to evaluate this part of the design. The Italian Historical War Museum, one of the meSch project partner, has an outdoor part showing structures from the first world war in the mountains of Trento. There is a tour through the trenches where a concept like the Companion Novel fits well. Three steps were defined for a successful port: (1) define a Narration Network and test it with included command line simulation tool, (2) design a case and a perspective switching mechanism fitting into the place and theme and (3) implement the interface for the custom perspective selection mechanism. Full images for both, the Debian system on the BeagleBoard as well as on the Raspberry Pies where provided along with a copy of the Python software including the simulation tool and sample data for some preliminary experiments with Narration Networks. On the hardware side technical specifications and setup instructions were compiled to allow rebuilding the whole hardware setup quickly. Due to time constrains there was unfortunately not enough time to do a full port within the scope of this thesis.

7 Conclusion and Future Work

This work provided insights into three different topics of the meSch project: (1) how can an on-site system be integrated into the meSch cycle, (2) how can a generic description for smart objects with attached stories look like and (3) how can this description be implemented in a way that supports easy porting. The integration part framed the design of Narration Networks and the prototype; it also pointed out requirements and challenges for the ongoing meSch development. Creating a system that, on the one hand, supports various design concepts for different types of museums and, on the other hand, is easy to use for museum curators to setup smart objects with attached stories seems to be one of the key success factors for meSch. In providing a loosely coupled, platform independent approach, this work can give good guidance on the way to achieve this goal. The usage of predefined templates would make it easier for curators to focus on the tasks of content assembling and story design. A critical issue will be the integration with the hardware platform outlined in Chapter 4.3. It must be as easy as possible to get from a chosen template to a full network and to reliably working smart objects in the museum space. As this system just started to evolve there is not much that can be said so far about the suitability of the proposed solution. The next two section will therefore focus on the other two topics before the thesis closes with some ideas for future work that can be done to further develop the concepts, designs and implementations presented in this thesis.

7.1 Narration Networks revised

During the creation and test of the Companion Novel the concept of Narration Networks proofed to be robust and easy to handle. With only a few building blocks it allowed to define complex structures and the delivery of personalised content. With no more than a text editor the networks were easy to define and edit on the one hand and easy to parse and use in the software on the other hand. The navigation concept could be implemented in Python with just 50 lines of code. The sample networks in Chapter 6.3 showed that a waste of different concepts can be covered by just adding a few triggers and content types. Some conclusions on that topic have been made there and will not be repeated here. In general, the ability to define content nodes with different triggers and prerequisites allows to flexibly create complex network structures and extend them as needed. The unified access to content snippets over URIs serves approaches with local data sources in outdoor settings as well as highly connected scenarios with data from central servers or the internet. Based on the building blocks used to describe Narration Networks, an easy to use, graphical user interface can be envisioned that makes the authoring task appealing for non-technical persons like curators.

On the downside the concept showed some weaknesses when the input is not just a set of sensor measurements or the output goes beyond simple media. Defining a generic action trigger and content type can address these problems but creates new challenges for template designers and requires a very generic implementation in the authoring tool. As seen in the evaluation of different concepts in Chapter 6.3 there will also be a need to regularly add new triggers and, not as often, content types, which might introduce further maintenance and versioning dependencies for the meSch server and the authoring tool. But as every existing implementation just uses the triggers and content types it needs, there will be no impact on existing systems.

It is also hard to model the behaviour of smart objects that depend on interactions between visitors. The system of predefined triggers is too static to model complex interactions depending on the state of several devices. Even the behaviour at shared resources like two people approaching the same loudspeaker cannot be modelled with the existing triggers, it has to be hardcoded into the implementation of a smart object and therefore cannot be changed by the curator by just using the authoring tool. But as this complex interactions are hard to understand it might be a good idea to leave this to the hand of designers and template creators.

Experimenting with augmented reality brought up two more complications. The input and output for augmented reality is hard to define with the simple descriptions used in Narration Networks. In addition, Narration Networks do not integrate with external toolkits which are widely used when creating augmented reality applications. If a toolkit handles input and output completely internal there is no possibility to define triggers and react according to changes and therefore no possibility to use the Narration Network navigation concept. This also prevents logging of user interactions for later use. There might be other technologies where toolkits are widely used and that will be hard to integrate into the meSch concept. If they provide an API to use their functionality directly, the Narration Network implementation can act as a wrapper. If there is no API, the toolkits can still be used for rapid prototyping, but for usage in the meSch environment they have to be re-written.

7.2 On the prototype

While the implementation of the Narration Network in Python was fairly easy, the Bluetooth technology turned out to be a big challenge, indeed quite temperamental. Most time on the prototype was spent on getting a reliable connection and figuring out the right threshold values. Some of these problems were based on the fact that we used cheap off-the-shelf hardware to build the prototype. The Bluetooth implementation on Debian using external dongles seemed to be unreliable and not designed for frequent connects and disconnects with different Bluetooth devices. The speakers did only support an old Bluetooth specification and even the same model of speakers behaved differently in the authorisation and timeout process. Initial tests with the Bluetooth capability on Android devices was more robust but did not allow access to all Bluetooth functions and included some security settings and checks that made it impossible to guarantee a consistent reaction time. The advantage in using Bluetooth was the direct integration of signal strength measurements in a reasonable distance which could be used to define proximity without the need of additional technology. Most of the problems could be solved for prototype use, but for productive use other technologies will have to be chosen to realise the concept. A reliable and more precise technology for range detection will be needed as well as a better technology to play the audio over speakers in place.

On a more general view the prototype was very successful in pointing out the requirements for using smart objects in an outdoor place. This applies to the interaction design itself as well as to testing the smart objects in the wild. On the side of the interaction design it showed the strong influence of weather, time of day or even vegetation in Summer or Winter can have. Noise and light levels can have a big influence too when there is a blockage of signals or line of sight to objects depending on the vegetation. Other issues like powering and maintenance or security of the equipment were realised but not discussed as no deployment for productive use is planned for the near future. On the side of testing smart objects outdoors the need for debugging feedback and control without a screen or keyboard available were clear findings of the test sessions. Even for embedded devices the current state of the device and the software must be obvious without connecting other monitoring hardware to allow field tests. Status LEDs and special debugging sounds proofed to be very useful during the test process. All of these findings also feed back to the system design in terms of being able to run independently and keeping the content on a central device rather than spread it over a number of devices that only uses short range communication technology.

Although not in the focus of this work, the intended interaction style of a talking place telling visitors several stories depending on their interests received mainly very positive feedback in informal feedback rounds, conversations and presentations. Compared to classical audio guides it has the potential of being much more emotional and engaging. Combined with the possibility to create complex networks it creates a very personalised experience which might help to create a strong link to the cultural heritage site and foster usage of online tools that allows to review the own visit.

7.3 Future work

Following this thesis there are clearly two main challenges ahead to use Narration Networks within the meSch project. The authoring tool would have to support generic creation of the user interfaces based on predefined templates and create a file based network structure that can be used in the on-site system. If the GUI is created based on the templates with the defined triggers and content types this includes the possibility to add new triggers and content types without much effort. XML schema descriptions can be facilitated to provide a common set of elements every template designer can use. The deployment process needs special attention, it must be easy to put a Narration Network file and related content onto a set of smart objects without having to deal with folder structures and file names. This is mainly a GUI design and workflow challenge, integrating the Narration Network structures with the meSch platform infrastructure for sensors and actuators is a more technical challenge. If the platform acts on a set of if-then-else chains the navigation structure of the network has to be translated. This requires a piece of glue code to map sensor values to context data and perform the necessary navigation steps. For commonly used platforms for smart objects like Linux or Gadgeteer it would be worth writing libraries that provide the functions to deal with Narration Networks like loading a file, navigation, content retrieval based on URIs or log file creation. As the libraries can be reused, they would speed up future prototyping and development of smart objects driven by Narration Networks.

Another open topic is dynamic content creation. With the possibility to address content via URIs it is possible to directly connect to a web service and create content on the fly using the visitors history and recent decision. This opens the door for powerful personalisation features as well as integration of user generated content. This requires a stable network connection and the web service needs access to the history but the content could flexibly be compiled from up-to-date data sources. However this on the fly content creation is a field of research on this own. As the prototype only used local content, even network or internet content has to be tried out to see if the system can fulfil the needs it was designed for.

Finally there are still open issues with the Narration Network design, especially regarding the integration of gestures as triggers and tactile feedback as content types, that should be investigated. This thesis is a first step in an ongoing process, pointing out the requirements and key challenges in closing the gap between the meSch server infrastructure and the on-site hardware platform. The requirements might change over the next three years of the meSch project but this work should still provide a good starting point and overview for future development. So far parts of this work inspired and were used in academic publications. Participation of curators and heritage professionals in the design process was addressed in the Nodem 2013 paper "Exploring Historical, Social and Natural Heritage". The TEI 2014 paper "Prototyping Tangibles: Exploring Form and Interaction" focuses on rapid prototyping tangible devices with minimal effort. Both the Nodem and the TEI paper were accepted for publication. A further paper submitted to CHI 2014 "A Personalised Companion System for Heritage Sites" about the Companion Novel presented in Chapter 5 is currently in review at the time of this thesis (27.11.2013). There are further plans for publications focusing on personalisation of content in context and on the challenges of designing and testing smart objects for outdoor usage. The openness and flexibility of the chosen approach will allow meSch to experiment with as many creative ideas as possible and will help define what is the most effective combination for each user and experience.

7.4 Acknowledgements

This work was part of the meSch project[1] funded by the European Community's Seventh Framework Programme 'ICT for access to cultural resources'[2]. Creation and Evaluation of the prototype was done in close cooperation with the meSch team from Sheffield Hallam University (UK), Daniela Petrelli, Luigina Ciolfi, Helen Grantham, Melanie Levick-Parkin, Fabio Caparelli, Nick Dulake, Matt Willox and Mark T. Marshall. I also owe them my thankfulness for the good ideas and valuable feedback they provided on this thesis.

[1] http://mesch-project.eu
[2] http://cordis.europa.eu/fp7/home_en.html

A Appendix

A.1 Python example code

Listing A.1 Navigator code to select nodes and maintain node sets

```
	def SelectNode(self, context):
		# select active node
		for node in self._activeSet:
			print str(node.NodeID)
			if node.Activate(context):
				nodeContent = node.GetContent()
				self._history.CreateRecord(node.NodeID, nodeContent.URI)
				self._moveNodesToActive()
				#remove nodes from the active set that are no longer needed
				if not node.IsActive():
					self._activeSet.remove(node)
				return self._contentProvider.GetContent(nodeContent)
		return False

	def _moveNodesToActive(self):
		#check prereqs and move nodes to the active set
		#if their predecessors have been visited
		temp = []
		for node in self._inactiveSet:
			a = set(node.GetPredecessors())
			b = set(self._history.GetVisitedNodes())
			if len(a) == len(a & b):
				self._activeSet.append(node)
				temp.append(node)
		for node in temp:
			self._inactiveSet.remove(node)
```

Listing A.2 XMLHelper code to import narration networks

```
def ImportNarrationNetwork(self):
        nn = ElementTree.parse("narration_network.xml")
        root = nn.getroot()
        self._networkID = int(root.get("id"))
        for poi in root.getchildren():
                for contentNode in poi.getchildren():
                        condition = self._createCondition(contentNode)
                        content = self._createContent(contentNode)
                        nodeID = int(contentNode.get("id"))
                        if contentNode.get("repeat") == "yes":
                                repeat = True
                        else:
                                repeat = False
                        try:
                                predecessors = map(int,
                                    contentNode.get("predecessors").split(","))
                        except AttributeError:
                                predecessors = []

                        node = MeSchNode(nodeID, condition, content, repeat,
                            predecessors)
                        self._nodes.append(node)

        return (self._nodes, self._networkID)
```

Listing A.3 MeSchNode code

```
def GetContent(self):
        retVal = None
        if len(self._content) == 1:
                retVal = self._content[0]
        else:
                #return the current content, for repeatable content the
                #modulo operation is needed
                retVal = self._content[self._visitCount % len(self._content)]

        self._visitCount += 1
        return retVal

def Activate(self, context):
        #see if condition for this node is met
        if context == self._condition:
                return True
        else:
                return False

def IsActive(self):
        #check is this node can play content again
        #print self._repeat
        return self._repeat or self. _visitCount < len(self._content)
```

Bibliography

[AKP12] L. Ardissono, T. Kuflik, D. Petrelli. Personalization in cultural heritage: the road travelled and the one ahead. *UMUAI - User Modeling and User-Adapted Interaction*, pp. 73–99, 2012. (Cited on pages 10 and 17)

[CB07] L. Ciolfi, L. J. Bannon. Designing hybrid places: merging interaction design, ubiquitous technologies and geographies of the museum space. *CoDesign*, pp. 159–180, 2007. (Cited on page 17)

[CBF01] L. Ciolfi, L. J. Bannon, M. Fernström. Designing and evaluating out-of-storage solutions. pp. 595–606. Archives & Museum Informatics, Philadelphia, 2001. (Cited on pages 10 and 17)

[CBF08] L. Ciolfi, L. J. Bannon, M. Fernström. Including visitor contributions in cultural heritage installations: Designing for participation. *Museum Management and Curatorship*, pp. 353–365, 2008. (Cited on page 10)

[CM12] L. Ciolfi, M. McLoughlin. Designing for meaningful visitor engagement at a living history museum. pp. 69–78. ACM, New York, 2012. (Cited on page 10)

[HN12] E. Hornecker, E. Nicol. What Do Lab-based User Studies Tell Us About In-the-Wild Behavior? Insights from a Study of Museum Interactives. pp. 358–367. ACM, New York, 2012. (Cited on page 10)

[HW03] G. Hohpe, B. Woolf. *Enterprise Integration Patterns.* Addison-Wesley Professional, 2003. (Cited on pages 10, 23 and 25)

[JABW+12] L. Johnson, S. Adams Becker, H. Whitchey, M. Cummins, V. Estrada, A. Freeman, H. Ludgate. *The NMC Horizon Report: 2012. Museum Edition.* The New Media Consortium, Austin, Texas, 2012. (Cited on page 17)

[PCD+] D. Petrelli, L. Ciolfi, D. van Dijk, E. Hornecker, E. Not, A. Schmidt. meSch description of work. Internal meSch description of work document. (Cited on page 32)

[PCD+13] D. Petrelli, L. Ciolfi, D. van Dijk, E. Hornecker, E. Not, A. Schmidt. Integrating Material and Digital: A New Way for Cultural Heritage. *ACM Interactions*, pp. 58–63, 2013. (Cited on pages 7 and 11)

[PN05] D. Petrelli, E. Not. User-centred Design of Flexible Hypermedia for a Mobile Guide: Reflections on the HyperAudio Experience. *UMUAI - User Modeling and User-Adapted Interaction*, pp. 303–338, 2005. (Cited on pages 9 and 30)

[Sch00] A. Schmidt. Implicit human computer interaction through context. *Personal Technologies*, 4(2-3):191–199, 2000. (Cited on page 12)

[Sch03] A. Schmidt. *Ubiquitous computing-computing in context.* Ph.D. thesis, Lancaster University, 2003. (Cited on page 10)

[The12] *Trends Watch 2012. Museums and the Pulse of the Future.* 2012. (Cited on page 17)

[UI97] B. Ullmer, H. Ishii. Tangible Bits: Towards Seamless Interfaces. *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pp. 234–241, 1997. (Cited on page 12)

[UI00] B. Ullmer, H. Ishii. Emerging frameworks for tangible user interaction. *IBM Systems Journal, 39 (3.4)*, pp. 915–931, 2000. (Cited on page 12)

[Ven] A. Venturini. D5.2. System Architecture. Internal meSch design document. (Cited on pages 7 and 24)

[WAG+02] A. Woodruff, P. M. Aoki, R. E. Grinter, A. Hurst, M. H. Szymanski, J. D. Thornton. Eavesdropping on Electronic Guidebooks: Observing Learning Resources in Shared Listening Environments. pp. 21–30. Boston, 2002. (Cited on page 17)

[Wei91] M. Weiser. The Computer for the 21st Century. *Scientific American*, pp. 94–104, 1991. (Cited on page 13)

All links were last followed on August 22, 2013.

Decleration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature