FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Kooperative Arbeitsabläufe in kleinen und mittleren Filmproduktionsfirmen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medieninformatik

eingereicht von

Ing. Jakob Frohnwieser BSc
Matrikelnummer 0726397

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dr. Hilda Tellioğlu
Mitwirkung: Univ.Ass. Mag. Roman Ganhör

Wien, 27. April 2015

Jakob Frohnwieser	Hilda Tellioğlu

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Cooperative Workflows in Small and Medium Sized Film Production Companies

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Media Informatics

by

Ing. Jakob Frohnwieser BSc
Registration Number 0726397

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Associate Prof. Dipl.-Ing. Dr. Hilda Tellioğlu
Assistance: Univ.Ass. Mag. Roman Ganhör

Vienna, 27th April, 2015

Jakob Frohnwieser

Hilda Tellioğlu

Erklärung zur Verfassung der Arbeit

Ing. Jakob Frohnwieser BSc
Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 27. April 2015

Jakob Frohnwieser

Danksagung

Ich möchte all jenen danken welche mir auf die einen oder andere Weise geholfen haben diese Arbeit zu vollenden. Ohne sie hätte ich die Arbeit nie fertigstellen können. Besonderer Dank gilt den Experten, welche bereit waren mir ihre wertvolle Zeit für die Interviews und die anschließenden Benutzertests zur Verfügung zu stellen. Ich danke ihnen, dass ich von ihrem Wissen und ihrer jahrelangen Erfahrung profitieren konnte und bin zutiefst in ihrer Schuld da meine Arbeit zu einem großen Teil auf ihrem Wissen aufbaut. An der Technischen Universität Wien hat Associate Prof. Dipl.-Ing. Dr. Hilda Tellioğlu meine Arbeit begleitet und stets vorangedrängt, sodass die Arbeit schließlich zu einem Ende kommen konnte. Univ.Ass. Mag. Roman Ganhör stellte die Kontakte zu den Experten her und half mir den Überblick zu bewahren. Er verbrachte zahllose Stunden in Diskussionen mit mir, trug großartige Ideen bei und gab mit immer wertvolle Rückmeldungen. Das gesamte Konzept des Systems, wie auch die gefundene Lösung wären ohne ihm nie möglich gewesen. Ich danke auch meiner Schwester MMag. Anna Frohnwieser, sowie meinem Kollegen Dipl.-Ing. Philip Helger, für das Korrekturlesen und ausbessern von Fehlern und somit verbessern meiner Arbeit. Insbesondere bin ich meiner Familie, sowie meinen Freunden dankbar welche mir fortwährend psychologische und finanzielle Unterstützung gaben; nicht nur während dem Schreiben dieser Arbeit sondern auch während des gesamten Studiums. Danke, dass ihr mir meinen Rücken freigehalten habt, sodass ich mich freier auf mein Studium konzentrieren konnte! Schlussendlich stehe ich auch in der Schuld der Technischen Universität Wien selbst, welche mich durch zahlreiche Semester von harter, aber nicht selten auch erfreulicher, Arbeit geführt hat und mir stets alle benötigten Materialien und Geräte zur Verfügung stellte. Mein Dank gilt all jenen welche dort arbeiten und studieren und welche mir die großartige Möglichkeit gegeben haben mein Studium durchzuführen und es letztendlich abzuschließen.

"Wenn alles gesagt und getan ist, ist der Weg das Ziel."
- Randy Komisar, The Monk and the Riddle

Acknowledgements

I wish to thank all those who helped me in some form or another to write this thesis. Without them, I could not have completed my work. I owe special thanks to the experts who were willing to spend their valuable time for the interviews and were also available for the user tests later on. I am grateful that I could draw from their expertise and years of experience and am deeply indebted for my work is to a large extent based on their knowledge. At the Vienna University of Technology Associate Prof. Dipl.-Ing. Dr. Hilda Tellioğlu guided my work and kept pushing forward so that the work could finally come to an end. Univ.Ass. Mag. Roman Ganhör established the contacts to the experts and helped me keep the overview. He spent countless hours in discussions with me, contributed great ideas and gave always valuable feedback. The whole concept of the system as well as the final solution of the implementation would not be possible without him. I also want to thank my sister MMag. Anna Frohnwieser and my colleague Dipl.-Ing. Philip Helger for proofreading and correcting errors and thereby improving my work. I am obliged to my family and friends who gave me enduring psychological and financial support, not only during the process of writing this thesis but also though the course of my study. Thank you for covering my back so that I could concentrate more freely on my studies! Finally, I am also in debt to the Vienna University of Technology itself which lead me through numerous semesters of hard but not seldom enjoyable work and provided me with all the needed materials and equipment. My thanks are directed to all the people who work and study there and who gave me the golden opportunity to conduct and finally complete my study.

"When all is said and done, the journey is the reward."
- Randy Komisar, The Monk and the Riddle

Kurzfassung

Die Produktion von Medieninhalten ist ein stark arbeitsteiliger Prozess. Von der ersten Idee bis zum finalen Werk sind viele Menschen involviert, welche mit großen Mengen analoger und digitaler Daten arbeiten. Trotz dem weit verbreiteten Einsatz von spezialisierten Werkzeugen gehen viele der dabei erstellten Artefake, wie etwa Texte, Bilder und Video- oder Klangdateien während der Produktion verloren, weil sich nicht alle verwendeten Technologien nahtlos in den Produktionszyklus integrieren lassen. Neueste Entwicklungen in der digitalen Aufnahme- und von internetbasierten Technologien versuchen diese Probleme zu überwinden. Hierfür werden Media Asset Management (MAM) Systeme benutzt um die erzeugten digitalen multimedialen Inhalte zu verwalten. Nichtsdestotrotz, in einem modernen, interaktiven Arbeitsumfeld müssen MAM Systeme mehr bieten als nur simple Archivierungsmechanismen und dateibasiertes Prozessmanagement. Noch immer besteht die Herausforderung die kooperativen Arbeitsabläufe zwischen den Menschen und all die dabei verwendeten Technologien miteinander zu verbinden. Obwohl hoch spezialisierte Programme, welche die meisten Anforderungen der BenutzerInnen aus der Medienbranche abdecken, verfügbar sind, fehlen noch immer leistbare und leichtgewichtige Anwendungen für kleine und mittlere Unternehmen. Solch ein leichtgewichtiges System zu entwerfen ist nicht trivial, da die Arbeitsprozesse in diesem Arbeitsumfeld komplex und zeitlich und räumlich stark verteilt sind. Die vorliegende Arbeit untersucht daher die sozialen Arbeitsprozesse in der Medienbranche. Sie erforscht was professionelle MitarbeiterInnen der Medienindustrie benötigen um ihre Arbeiten durchführen zu können und identifiziert die Herausforderungen die sich BenutzerInnen in solch einem hochtechnisierten Bereich stellen müssen. Durch das Zusammenführen aktuellster wissenschaftlicher Literatur und Interviews mit sieben ExpertenInnen aus verschiedenen Medienunternehmen wird eine neuartige Lösung für ein MAM System erarbeitet, welche es ermöglicht, die wichtigsten Funktionalitäten, die bei der vorangegangen Recherche identifiziert wurden, abzudecken. Basierend auf dem Entwurf dieses Systems ist ein Prototyp implementiert worden, welcher anschließend mit fünf BenutzerInnen aus der Industrie evaluiert wurde. Die Ergebnisse dieser Entwicklung zeigen wie Computerprogramme entworfen sein müssen, um den Bedürfnissen und Erwartungen von AnwenderInnen aus der Medienbranche zu entsprechen.

Abstract

Creating a media production is a highly specialised process. From the first idea to the final product many people are involved who work with vast collections of both analogue and digital data. Despite the widespread use of specialised tools many of the thereby created artefacts, such as texts, images and video or sound files get lost during the production process since not all technologies used integrate seamlessly into the production cycle. Advancements in digital recordings and web based technologies try to overcome these problems. For this, Media Asset Management (MAM) systems are used to administer the thereby created digital multimedia content. Nevertheless, in modern, interactive workspaces, MAM systems must go beyond providing simple storage mechanism and proving file-based workflows. Incorporating the cooperative workflows between people with all the technologies used is still a challenge. However, although high level programs which meet most of the needs of the users from the media industry are readily available, affordable and lightweight programs for small and medium-sized companies are still missing. Designing such a lightweight system is non-trivial since work processes that are going on in this sector are complex and vastly distributed in time and space. This thesis examines the social work processes that take place in the media business. It investigates what professionals working in the media industry require to perform their tasks and identifies the challenges users face while working in such a highly technologised environment. Compiling state-of-the-art literature and interviews with seven experts from different production companies a novel solution for a MAM system is proposed which is able cover key functionalities identified in the preliminary research. Based on the systems blueprint a prototype is implemented and evaluated with five users from the industry. The results of this process show how computer software has to be designed to meet the needs and expectations of users from the media industry.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In professional media companies a production is traditionally carried out in three consecutive steps, namely pre-production, production and post-production. These steps are furthermore divided into numerous sub-steps [Sta08]. Many people are involved in this production process beginning with scriptwriters, set assistants up to actors and producers. Keeping track of their actions and the processes that are going on in the background manageably is a complex task. One of the many challenges lies in coordinating the vast collections of different types of media files, both analogue and digital. Especially the first two production steps, where a lot of paperwork is produced, can be effectively digitalised, but also the last, post-production step can benefit from it.

For this, all of the artefacts created throughout the project must be stored at a central location to make them accessible for other project members or for further productions. Such a digital workflow management and archiving system also provides more independence for the staff members in media production companies. For instance, when using such a system journalists or cutters do not depend on time-consuming editing actions of others and can spend more time on their research or can do more operations at once [GALSH04]. Hence, technical solutions which provide such functionality in a user-friendly environment are needed. Therefore, Media Asset Management (MAM) [Saw01] systems are used to administer the digital artefacts which emerge during the production process. These artefacts can include texts, images, videos and sound files. Moreover, MAM systems also provide means to manage file based workflows.

MAM systems have many advantages over simple video platforms and are not only used to organize and publish multimedia data (c.f. [JCG+05]) but also have limited workflow management capabilities [Gon10]. They also incorporate metadata catalogues to make it easier to find and retrieve assets and to speed up the search processes. Hence, such systems are widely used in the media production industry where large collections of up to hundreds of terabyte (TB) of data have to be analysed, transcoded, archived, searched and released to make them easily available, e.g. for multi-media productions or in online Video-on-Demand (VoD) media libraries [SWL11]. They are usually used to

find and (re-)use content for news reports, documentations or also TV shows and movies. Hence, they have become indispensable in the media production industry. Companies using MAM systems span from institutions like small broadcasters and media production companies [Sch11] to major TV stations [SWL11].

Solutions for MAM systems exist since the late 1980s and have become a practical mainstream technology [Saw01]. Since the advent of the first tape-based systems MAM systems have been continually improved and their functionality has been extended. Despite such systems existing for several decades advancements in digital recording devices and web-based technologies since the 2000s led to MAM systems facing new problems regarding preservation and reuse of analogue and digital content, transcoding from and to different codecs or the retrieval of assets over the internet [SWL11].

Traditional MAM systems are often only used to store digital assets but do not offer advanced tools to also manage cooperative tasks or make use of intelligent agents (c.f. [Sch91]). They tend to leave the management of digital assets and the organization behind the files mostly separated from other tools and programs [Saw01]. Hence, they are often used as (advanced) archiving systems only without support for cooperative tasks and other technologies. For instance, planning and writing must be done in other specialized programs, whereas the MAM is only used as an isolated storage container [Pal04] - just like online video platforms.

On the other hand, various groupware programs which support users in specific tasks, such as co-authoring [NKCM90] tools for writers, have emerged. Unfortunately these tools often do not have any further capabilities to be usable in a broader context. What is missing are MAM systems which are tightly integrated into the entire media production workflow and are able to manage the workflows in which the digital artefacts are used [SW03]. Thus, in modern, interactive workspaces, MAM systems must go beyond providing simple storage mechanisms. They should support multi-modal workflows which include not only digital but also analogue data, enable fluent document sharing and the execution of numerous, parallel tasks at once [NKCM90]. Accordingly, more sophisticated and advanced workflow management technologies which are able to support all levels of the work at hand have to be integrated into the management systems.

Recent research shows that such systems can have many significant benefits on various fields [SW03] and can also help to reduce costs [Gon10]. For that, such systems have to cover many different requirements. Whilst in online news editorial offices the speed of the production is one of the biggest success factors [MB09], in documentations, for instance, it can be more important to have good and rich documented production phases. A feasible workflow management system must be able to meet all of these demands. It must offer more than conventional MAM systems. It must be designed to support all work processes from the first exposė to planning and producing up to the (multi-channel) distribution and marketing. Moreover, it should also offer tool integration, multi-user collaboration and a centralized, fail-safe infrastructure to make it possible to cover all associated tasks in an end-to-end workflow management for the whole company [SW03]. Such a tool can then be a valuable part in managing the collaborative work processes that take place at media companies where many people are involved.

Including techniques from Computer Supported Cooperative Work (CSCW) Systems (see [BS91]) can have many benefits in comparison to using only traditional MAM systems (c.f. [PRM+04]). CSCW based systems improve productivity, reduce costs, increase the value of the work and help to complete tasks in less time [SW03]. Furthermore, CSCW systems allow users to work together from different geographic places and reduce social barriers while increasing the group participation [Bor97].

Although there are high level programs which meet some of these needs readily available, affordable and lightweight programs for small and medium-sized companies are still missing. Current solutions are often too complex, use proprietary formats and are too expensive to be usable for small productions. Since the tasks and processes that are going on in the media production are non-trivial, a software solution which covers all of the requirements must be well designed and implemented in respect to the people who are going to work with it. Introducing new software which will replace old procedures is hard, especially when many users are affected [Nor02]. New technologies must be adopted and used in the right way. Only installing them at the workplace is not enough [PRM+04]. Thus, it must first be understood how people from the media industry work, what they actually require in order to perform their tasks and how computer software can assist them to achieve their goals.

This thesis investigates such collaborative work environments for video production systems and examines the social work processes that take place behind them. It makes use of recent research and provides a critical view on CSCW systems [Sch91] in regards to how workflow management models are designed [AS94]. This is done in order to comprehend and address problems of MAM systems which are currently in use and describe possible solutions for them. The study presented here is based on various subsequent scientific research methods which are explained in detail in Chapter 3.

A preliminary, profound and well-informed literature research stood at the beginning to function as a basis for further research techniques and to build a new design of a MAM system upon. To get deeper insights into this field seven experts from the media industry were interviewed to answer the questions of how people work in this collaborative environment, what social processes take place and how computer science can help to support people in the media industry at their everyday work. Thereby different solutions for MAM systems were evaluated against each other and based on the statements of the experts and the before conducted literature research a new design for a MAM system is conceptualized.

The system is designed to be simple and fast but to also cover a wide range of features. Moreover, it is designed to be extendible to meet diverging needs in upcoming fields of application. For this, the proposed system model makes use of state-of-the-art technologies. It consists of a central cloud based service hub and a dedicated client program installed at the users' own hardware. Thus, the application is moved to the cloud (c.f. citeMG11). This novel setup is new for MAM systems since usually MAM systems use only one central standalone application at this time [Sch11]. By combining these two very different approaches it gets the best of each system. It uses more powerful

hardware in the cloud but also overcomes network limitations of internet based systems. Additionally, by outsourcing services to the central cloud service costs for the individual tenants are reduced . This makes it possible to share access to services and data among users over the internet but also keeps raw video data secured in the local data storage. For this, users do not have to have their own IT-infrastructure any more [Cos11] or maintain the software used [TMAS11].

To test the assumptions made an open and easy-to-use prototype of such a collaborative work software for multimedia productions based on the described system design was partially implemented. The use cases which are included in the program were decided during the research and planning phase in an iterative approach. Thus, it was possible to flexibly react to the actual needs of the users and upcoming requirements which arose during the realization. These selected features are implemented vertically, i.e. so that every aspect is implemented as it is in a real-world program. Therefore the use cases included in this prototype are fully functional and can already be used in a productive system.

However, the software is also implemented and documented in a way that others can easily continue the work and adapt it to their needs as a basis for future extensions without the need to pay expensive licence fees. This makes it especially suitable for small and medium-sized businesses. The generated software meets current standards of software engineering techniques regarding security and usability and makes use of state-of-the-art research of Human Computer Interaction (HCI).

The application was tested with five experts from the media industry and from the Vienna University of Technology directly in the field. Conclusions drawn from these user tests were evaluated and used to take a critical reflection of the work. Major advantages but also weaknesses of the solution are identified and further improvements of the system for future releases are proposed.

To give a systematic overview over the investigation and the proposed solution this thesis is organized as follows. First, state-of-the-art implementation for MAM systems and software engineering techniques are examined in Chapter 2. With these findings in mind further investigations were carried out as described in Chapter 3. Based on the herein contained summaries of the expert interviews and design generating methods a novel architecture for an MAM system is introduced in Chapter 4. A detailed description of the implemented system is found in Chapter 5. Here, the techniques and algorithms used are exemplified to provide a complete overview of the created prototype. Afterwards, the evaluation with the experts is summarized in Chapter 6. Finally, the thesis continues with a critical reflection on the work in Chapter 7 where advantages and drawbacks of the implemented system and the feedback from the user tests are included. The summary in Chapter 8 takes a close look on what was achieved and proposes open questions for future inquiries and improvements for upcoming implementations of the system.

CHAPTER 2

State of the art

This chapter reviews the most important aspects in regard to the presented work. It discusses MAM tools as an ubiquitous component in media production. It emphasizes on online services and the various considerations to be made such as usability, scalability, monetary costs and security. This is done in order to comprehend current techniques of the field in question and to find solutions for pending problems.

Hence, the research starts with an investigation of state-of-the-art techniques and methods. Based on this, subsequent research phases can follow to answer upcoming questions.

2.1 Media Asset Management

As discussed in the introduction MAM systems offer multiple functions to manage digital assets. An overview over the general features of a MAM system can be found in Figure 2.1.

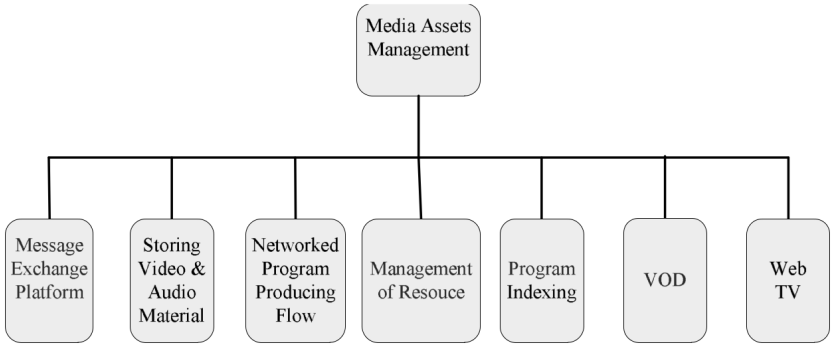


Figure 2.1: The main features of MAM systems according to [SWL11]. These features define organizational demands but not technical requirements.

The functions shown in Figure 2.1 include storing and archiving, managing, indexing and sharing of multimedia artefacts - i.e. assets. While these features are drawn from existing solutions for MAM systems, according to [SWL11] several unresolved problems still exist. These issues arise due to non-standardized descriptions for information in the archiving system, different choices of storage devices, digital rights management issues as well as the definition of standardized software interfaces (APIs) to other systems.

2.1.1 Logical function framework

Consequently, based on these requirements and issues the authors derive a logical function framework - as shown in Figure 2.2 - which is designed to meet all the requirements for a MAM system and solves the problems discussed above.

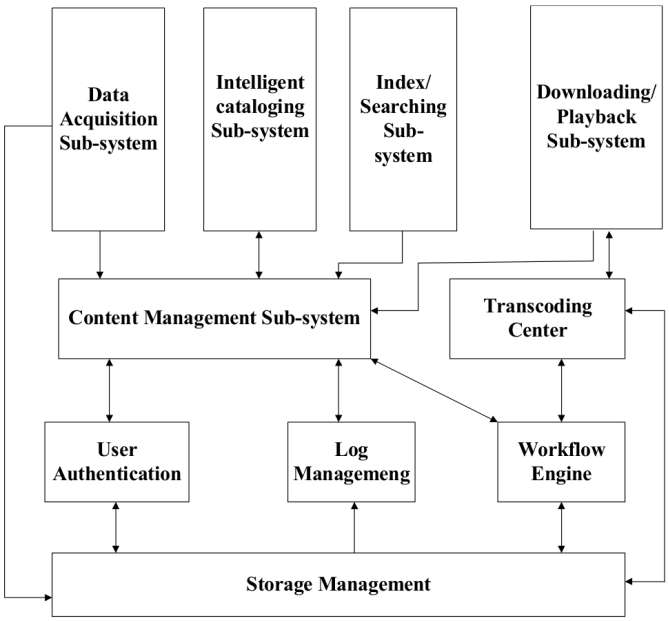


Figure 2.2: The logical function framework of MAM taken from [SWL11]. Each one of the various subsystems offers a specific function. Dependent subsystems are linked with a directed line.

This framework offers all functions as defined by the basic requirements above and to do so, combines numerous different subsystems, according to the actual implementation and demands. Still, the main focus lies on content management as the core functionality of a MAM system. For that purpose, the authors define six key functions of this framework:

1. **Storage subsystem**
   This is the main storage device where all assets and metadata are stored. It performs automatically optimizations for the storage mode and also hosts the application software.

2. **Content management**
   The content management subsystem is the interface between the stored files and other application subsystems. It provides functions such as searching, retrieving or cataloguing of stored assets.

3. **Transcoding center**
   In the transcoding center all assets are transformed, i.e. encoded to provide consistent data formats, bitrates and codecs for the MAM system. It is primarily used for from all kinds of sources imported data but can also be used to transcode data to different formats for downloads and streams.

4. **Data collection**
   In order to be able to work with raw files, i.e. material files, the data collection is needed. It digitizes assets, saves a history for each file and transcribes files to different storage devices such as analogous cassettes.

5. **Catalogue and retrieval**
   A catalogue structure is established which provides methods to describe metadata information in order to make the assets searchable. Additionally, this catalogue is also accessible for other media producers. This makes it possible to search in asset collections of other companies in order to buy and use assets from different sources. For this, a standard catalogue format has to be defined and a database access must be provided.

6. **Workflow engine**
   The workflow engine orchestrates all workflows behind the files. It is used to define and monitor work processes and distribute tasks to different employees. It can also be used to analyse problems and to improve the overall system performance.

With these key functions defined it is possible to plan and implement a working MAM system. Since the system is designed gradually, i.e. as an extensible collection of independent subsystems it supports all kinds of applications and operations needed by the user.

2.1.2 Hybrid Software as a Service solution

Schneider [Sch11] presents a different approach. His design for a MAM system consists of a cloud based hybrid approach in a Software as a Service (SaaS) solution. Here, the word hybrid states that the system partly incorporates a local installation and one or more cloud based services which are provided over the public internet. An overview over this system is given in Figure 2.3.

While a MAM system is built to handle raw materials uploading them to an external server is impractical. As an example, a one hour high-definition (HD) video typically has a size of 54GB [Sch11]. For that reason, this setup is designed to keep these large files (which can have a size of up to a hundred TB) local where fast access to them is given.

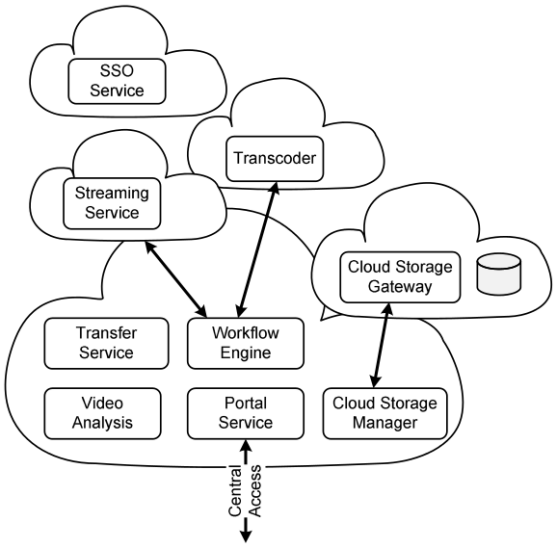


Figure 2.3: An overview of the hybrid SaaS solution for a cloud based MAM system as proposed by [Sch11]. The central cloud in the middle encapsulates the central service hub; external services are located in different clouds above.

Otherwise they would need a lot of bandwidth and a long uploading time. Uploading 10TB of data with a 20Mbits/sec connection would take more than 45 days [AFG+10].

File based operations such as analysing and transcoding are therefore still done on the users' own hardware while the processed files are then transferred to the external cloud based systems. Other standardized services for indexing, analysing, searching, data backups and general workflow management are outsourced to a central service hub in a cloud system. The service hub then manages all connections and invocations to different cloud services which provide the required functionality. For that reason, these third party cloud services are provided by different companies and do not induce any additional expenses except for eventual user charges. To make these cloud services callable by different platforms they are called with standardised Webservice protocols such as REST[1] or SOAP[2] (more on Webservices in Section 2.6.3 below).

The major advantage of this model in comparison to [SWL11] solution is that time-expensive operations with large files (see example above) are still under control by the user whereas common services can be used to extend the range of functions with different, readily available services from the public internet. This does not only save license fees, reduces local maintenance needs and minimizes system downtimes but also the most valuable data, i.e. high resolution materials (usually HD up to ultra-high-definition (4K)) and private information are still safely stored at the users' premises.

Since this system is designed to make use of already available services over a Single

[1] see http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
[2] see http://www.w3.org/TR/soap

Sign-On (SSO) network the user does not need to logon for each service individually. In addition, other services can be subsequently included. Since this system is designed to be usable for multiple tenants hardware and software costs are shared among all contractors and all user accounts are managed centrally. Data can also be shared among users or through the internet with other media companies and customers [TMAS11]. Managing the software in a central point for all users also has the advantage that bugfixes and updates are faster installed [TMAS11]. Since the system is designed to run on an external cloud server the users do not have to possess and maintain their own IT-infrastructure [Cos11].

2.1.3 Performance/Scalability

The performance of computer systems is defined as the amount of work a system can execute with given resources in a specific timespan. Hence, if a system is highly performant it is able to accomplish more work with less resources and - or - needs less time to do the same amount of work. Performance and the adaptation to new demands is a curial success factor for media based businesses [Dur10]. Thus, the right computing infrastructure has to be chosen carefully.

Since both discussed systems are not available for statistical measurements exact numerical values of the systems properties are not at hand. Moreover, the performance of the systems strongly depend on the specific software and hardware used. Hence, a direct quantitative comparison of both systems is complex. To that end, it is only possible to compare the systems regarding qualitative measures. In their papers Shan et al. [SWL11] and Schneider [Sch11] already present profound facts about their approaches but to make the comparison even more objective other sources are also consulted too.

The performance of the application depends entirely on the provisioned server capacities (see also example in Section 2.4). If circumstances change the server must adapt to them by adding new resources or releasing unused ones. For the in-house solution [SWL11] installing new server racks can take several weeks [AFG+10]. In contrast, cloud server can also better adapt to unexpected or seasonal events, e.g. traffic peaks in online shops before Christmas, since computing resources in the cloud appear to be unlimited [AFG+10]. It is also more efficient to scale to rising or decreasing needs than to provide a large data center at all times [AFG+10].

As a consequence Schneiders' hybrid model [Sch11] takes advantage of a local client and central services in the internet cloud. For this, large, i.e. high resolution, files are already processed in the local client and then transferred to the external servers. This does save bandwidth and upload-time, especially where internet connections are limited. Additionally, this setup also makes it possible to add or remove new resources or services to the central service hub in the cloud afterwards [AFG+10] and makes the MAM system more scalable then the traditional implementation.

As a drawback, in certain cases the usage patterns of one customer can lead to performance losses for others when using a shared computing infrastructure. This has to be addressed by the cloud service vendor who must ensure that the service is robust and always available [BYV+09] [Dur10]. Additionally, the availability of the service strongly

depends on the quality of the underlying internet connection. Using multiple cloud service providers can solve these issues [AFG+10].

Even so it can be assumed that if the cloud service is correctly configured no performance losses have to be expected. The ability to rent and release resources at any time and the elasticity of adding and removing external services at will makes cloud computing a powerful alternative to providing an own IT-infrastructure. This is especially useful for smaller organizations [BYV+09].

2.2 Usability

Usability measures the quality of use of a User Interface (UI) of an application [Nie94]. It is used to improve the ease-of-use of a program and is defined by certain key factors, such as the efficiency or learnability but also the utility of the software, i.e. if the application does what the user expects in order to perform a certain task [3]. Usability is an important aspect for successful applications. Users will not accept an unusable program, regardless how well functioning it is. For instance, if a product a user searched for in a webshop is not easily findable the user is not able to easily log in he or she will not buy anything there. Moreover, usability also enhances the productivity of the users and reduces cognitive stress[4]. This is usually the case when the human mind is more challenged than expected through poorly designed systems, inconsistent site navigations or malfunctioning control fields. Unexpected behaviour of an application such as strongly delayed response times causes further distractions. This is especially objectionable for novice users which will turn away from the application but has also major drawbacks for experienced users which face stress when using the software.

An IT system must therefore support the user in his or hers work tasks but not interrupt their train of thoughts and not generate additional stress. In the media industry where many people with different levels of technical expertise have to work together a program used in this context must be designed to be usable by all of these users in an easy and user friendly way but still be powerful enough to perform all needed tasks. Usability research methods and their design techniques try to encompass these requirements to make the software accessible, user-friendly and efficient.

For this, to start with, usability can be enhanced by studying current designs of old versions of the application in question or of products from competitors. Further improvements include user tests with prototypes or field studies where the intended users are examined in their usual work environment [Nie94]. Of course, such improvements can never be really concluded and have to be consistently evaluated in an iterative design process where each step provides an improved solution, closer to the optimum. Of course, even the final design should be tested to verify the findings. This process also draws from outcomes of previous studies and design principles. Both Nielsen and Norman state therefore various design principles which summarize the most important ideas

[3] http://www.nngroup.com/articles/usability-101-introduction-to-usability

[4] http://blog.trymyui.com/2015/03/new-horizons-bringing-ux-to-the-survey-design-field

about usability [MN90] [Nor02]. These ideas include using consistent interaction and presentation patterns through the application, displaying control elements in a clear way so that it can be comprehend what the element does and how it is to be used or giving instant and direct feedback to the users so that they can understand what is happening.

In addition to this, a software product has to be simple enough in order to be usable and understandable for many users without a steep learning curve. Yet it also has to be able to perform complex tasks [Nor02]. Covering these conflicting requirements can be hard. A good interface and interaction design therefore hides the underlying technical system to avoid misconceptions about the offered functions. The creator of an application has to find a balance between a simple design and the capabilities, i.e. richness of features, of an application. Thus, during the design and implementation process features of an application may be discarded whereas others can be added. During this cycle the scope of an application can shift but must be kept clear and narrowed [Mar03]. Good usability design manages the technical complexity behind a program and abstracts the underlying processes to give the user understandable and direct feedback.

Following this, an application which is designed to administer digital artefacts does not necessarily have to include other features such as an email mailbox or an accounting functionality. This functionality can be split into several sub modules with each only providing a specific function in a clear and cohesive way.

With these principles in mind, it is possible to prevent the most critical pitfalls. Still, user test should be conducted where users are observed when given representative tasks. This does not only help to verify the worked out design but it also clarifies misconceptions of the users, as well as wrong assumptions of the designers and programmers [Nor02].

It is suggested that 10% of a project's budget should be spent on enhancing usability[5]. Devoting such a big share of a project's finances on usability is still not common today, but on the other side studies show that enhancing usability increases the outcomes of a software product of about 10-100% [DWN99]. Thus, improving usability usually far exceeds the costs of implementing it.

2.3 Security

In computer science the term security includes all processes and mechanisms by which digital information and accesses to services are protected, i.e. so that resources cannot be read, used or altered by unintended or unauthorized individuals. This is usually achieved by using data encryption and enforcing access restrictions on all digital devices and files. Security therefore ensures the confidentiality, integrity, and availability (i.e. the so-called CIA triad[6]) of all resources of a computer system.

Leaks or illegal publications of assets would lead to high losses for the holder of the copyright or for the owning company [Sch11]. Thus, in the media industry too security is a major concern. To achieve this, Shan et al. [SWL11] and Schneider [Sch11] follow

[5] http://www.nngroup.com/articles/usability-101-introduction-to-usability

[6] http://www.techrepublic.com/blog/it-security/the-cia-triad

different approaches. Shan et al. [SWL11] keep all data, i.e. all media files, processed as well as unprocessed and metadata-information in a local, in-house server. Hence, the security of the traditional MAM system solely depends on the underlying hardware and network protection. Unfortunately, the authors do not make any suggestion about further safeguarding the application.

2.3.1 Determining the overall Systems security level

With Schneider's [Sch11] approach determining the level of security is more complex. Schneiders model keeps all high-resolution files local, where security measures can be as diligent as with the traditional approach, while processed files and metadata-information are transferred to an external server over the public internet. Here, the author suggest some basic security measures such as using a secured SSL connection for the upload, file and access key encryption and considerate session handling. Still, the questions persists if cloud computing can be as secure as an in-house server.

A meta-study based on 55 papers [ID12] where nine security factors (shown in Figure 2.4) for cloud computing were investigated came to the conclusion that two thirds of the security factors examined are almost in all cases very well covered by multiple mechanisms. However, other areas such as recovery and prosecution are yet not well researched.

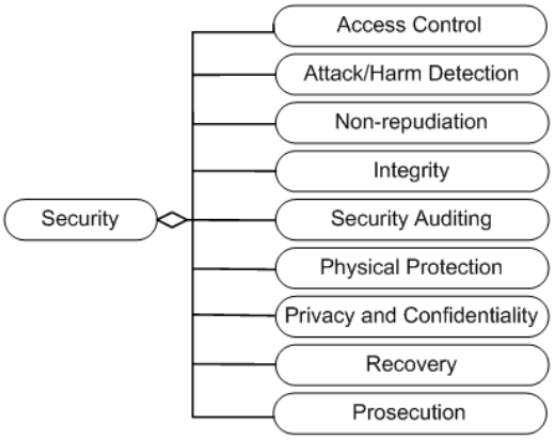


Figure 2.4: The security model as taken from [ID12]. The taxonomy defines nine quality factors for general system security.

When using cloud services a third party or external provider is responsible for the applications security. On one hand, this makes it easier for the users when specialists take care of the MAM system but on the other hand the stored data needs also be secured against the provider itself [AFG+10]. Thus, as Schneiders' [Sch11] model involves various service providers each one of them must guarantee a certain level of trustworthiness.

From this point of view it is hard to say which solution is more preferable. Both systems can be configured to be highly secure but still other - non technical - mechanisms

such as digital rights management (DRM) [SWL11] have to be promoted to assure that the digital assets are safe and secured.

2.3.2 Access control

Access control is needed to selectively restrict users from accessing resources or sites. This covers restrictions on viewing or writing digital files or executing specific tasks. By authorization a user gains the permission to perform an indented task. Thus, to secure the assets and the stored information from unwanted accesses a mandatory system-wide security model must be established. This model must ensure that each user (authenticated as well as anonymous) gains access only to those resources which he or she needs for a certain task and where he or she has the rights for. All accesses to other resources must be strictly blocked.

For that reason, each user which needs to execute certain tasks must be authenticated in the system. For this, the MAM system needs to implement a dedicated user store where users are registered and can be authenticated with a login mechanism such as with a username or an email address and a password[7]. Nonetheless, other public or anonymous users can be defined which then get only access to public sites. Of course, these users need not to register at the system and can be treated as one. To further discriminate between different types of user permissions each user needs to have at least one role assigned which states his or her privileges within this system.

To that end, often a hierarchically organized security model such as introduced by LaPadula and Williams [LW91] is used. Their model consists of various levels which are equivalent to certain roles. Users on an upper level have more rights than users on a lower level. Therefore, users on an upper level usually subsume all rights from all levels below and receive additional rights which are needed for their role. In that regard, the levels are not defined by the trust each user is placed upon but merely by the needed rights a specific task needs to have to be executed. Accordingly, each level of security is defined by what needs to be done and not by what one can do. Users must be placed in this hierarchy according to their organizational duty and demands. For this, certain roles are defined, often containing roles such as (from top to down) super-user, admin, registered user and undefined. Clearly, the number of roles used depends strongly on the systems demands. Other roles can be added as well. The role a user has assigned then restricts accesses to certain information such as web pages or administrative tasks. Such a role-based access control is proven to be highly efficient and trustworthy and eases the administration of the systems permissions [FCK95].

Moreover, the MAM system also needs to secure read and write accesses to the stored assets and sets. Since defining a specific role for each resource would be too complex but using only a limited number of roles for all resources would be too inaccurate a more flexible and fine-grained security model is needed.

For this purpose, users can be collected in groups with specific rights for one or more resources. This is in contrast to roles which only collect permissions [SCFY96]. The

[7] `https://www.owasp.org/index.php/Authentication_Cheat_Sheet#User_IDs`

group principle is well know from Unix style operation systems security model where groups of users can have read, write or execute permissions for files. There, users can usually be assigned to an unlimited number of groups.

In the MAM system, only read and write permissions for the files are needed since the files can either be viewed or stored but no programs can be executed (administrative tasks such as defining new users are covered by the role-model). Each group is therefore defined as a set of users with a set of read and - or write permissions for a collection of files. Thus, before access to a certain file can be given the system simply needs to check if the user is contained in at least one group which holds the needed rights for this task.

Additionally, other non-technical, i.e.organizational, security policies must also be met. In the scope of this work the principle of least privilege [SS75] which states each user should only have the least set of privileges which is necessary for his or hers work, is most important. Other common principles like the separation of duty [BE01] or the chinese wall [BN89] can also be applied. Thus, a fine-grained role and group model which is flexible enough to be adjustable for a wide range of demands must be implemented. The main goal of this role and group based setup is to limit the users rights as much as possible without interfering with their work to establish a system wide trustworthy security mechanism.

2.3.3 Authentication

For the authentication, users should use complex and hard to guess passwords. These passwords are then stored in the database with irreversible, i.e. one way, hash functions. This is done to encrypt the passwords so that if the database is hacked the passwords cannot be decrypted. For this, strong hash functions with large bit sizes, e.g. 256bits or more[8] must be used.

Of course, an attacker might also try all possible combinations of letters to guess the password. To prevent such brute force attacks on stored passwords user accounts have to be disabled after a defined number of failed logins[9]. This is done to prevent the attacker from making unlimited guesses about the passwords which can be easily executed automatically. Because of this, user accounts are often locked after a few unsuccessful login tries. Actually any one to four digit number of attempts could be allowed since a brute force attack usually would need more than a few thousand tries to be successful. Since it can be assumed that nobody enters a wrong password for dozens of times user accounts are usually locked after a few, e.g. three or five, failed attempts. Moreover, since an attacker will not use a standard web browser to conduct the attack he or she can easily open a new session for each try. For that reason, the failed login count cannot be stored in the session but has to be stored in a persistent database entry. Thus, before each login attempt is evaluated a login attempt counter for each user has to be increased in a special database field and only reset if the attempt is successful.

[8] https://crackstation.net/hashing-security.htm

[9] https://www.owasp.org/index.php/Authentication_Cheat_Sheet#Prevent_Brute-Force_Attacks

Moreover, an attacker might also try to not attack a specific user with different passwords, but try one (or more) passwords on different users. This approach is also referred to as a reverse brute-force attack[10]. For this, statistics with often used passwords are readily available[11]. Trying such passwords on different users can be successful in many cases. Hence, it is also necessary to store the IP addresses of the users and lock all accounts for a specific IP address if it fails on many logins. Certainly, an attacker still can make use of different proxies to hide his or her traces but nevertheless he or she would need an extensive set of usable IP addresses to circumvent this mechanism.

Then again, an attacker might exploit this mechanism for an application layer DDoS attack. That is, given that the attacker can guess the usernames, he or she can use this security mechanism to lockout all users of a system. This would lead a company to be out of service. To prevent this, the user accounts should only be disabled for a short period of time, such as for instance 20 minutes. This mechanism is also referred to as a tar pit since it effectively slows the brute force attack down and makes testing hundreds of thousands of passwords almost impossible[12]. This technique also helps to decrease servicing costs since the accounts do not have to be manually unlocked. Additionally, the timeout can be increased for subsequent login attempts to further slow the attacker down.

Other improvements include remembering the failed passwords to not count duplicate password attempts for each user or to store hints about the passwords. This is done to not lock users who accidentally misspell their password in the same way many times because an attacker obviously will not try the same password more than once and to enable users to reset their passwords at their own if no timeout is set. Nevertheless, passwords hints are prone to social engineering and therefore need to be handled with care. More sophisticated measures such as multi-factor authentication where, for instance, one gets send an one-time token (a Transaction Authentication Number (TAN)) to his or her mobile phone via SMS which he or she then has to enter to login, can be used too. However, these mechanisms are hard to implement and cause further costs for the service provider.

Authentication is not only needed to authorize the user for specific tasks but should also be mandatory when the user changes critical settings such as his or her account password.

2.4 Cost-efficiency

In order to decide if a company should use one of the discussed solutions or the other, commercial considerations also have to be taken into account. Both MAM systems are

[10] http://www.cs.virginia.edu/~csadmin/gen_support/brute_force.php

[11] e.g http://gizmodo.com/the-25-most-popular-passwords-of-2014-were-all-doomed-1680596951

[12] https://www.owasp.org/index.php/Authentication_Cheat_Sheet#Prevent_Brute-Force_Attacks

designed for different requirements but nonetheless a comparison regarding cost-efficiency is possible. Shan et al. [SWL11] standalone solution is designed for major TV broadcasters whereas Schneider [Sch11] focuses on comparatively smaller media production companies which use cloud services via the internet. Thus, the question is when buying and operating an in-house server is more efficient than renting cloud computing resources.

Armbrust et al. [AFG+10] found a simple formula which is based on the server utilization running a specific application for a given timespan to calculate the cost efficiency of a cloud computing solution. Their formula assumes that in order to serve all requests at all times for an in-house server the maximum estimated server capacity must always be present. If peak loads are not correctly anticipated under-provisioning at peak times happens but on the other hand server resources are wasted at periods of rest, as shown in Figure 2.5. Thus, dividing the maximum utilization needed by the average utilization at all times results in a cost-efficiency factor.

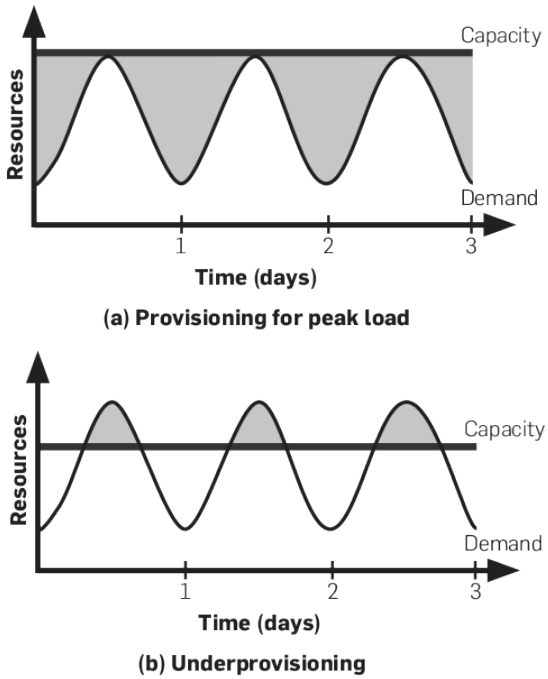


Figure 2.5: Possible server utilization patterns as taken from [AFG+10] (excerpt). In order to compensate for peak loads server resources are wasted (shaded areas) at periods of rest (a). If the maximum needed server capacities are not provisioned less resources are wasted but under-provisioning happens (shaded areas) (b).

Since cloud services are able to adapt to peak demands as shown in Section 2.1.3 using them no resources are paid when not needed. Therefore, if the costs of the cloud service exceed the costs of the in-house server multiplied by the calculated factor renting a cloud server is advised. As [AFG+10] show this factor is usually more than 1.7 which

means that even if given that the cloud service is more than 70% more expensive than the in-house solution cloud computing is still significantly more cost-efficient. Of course, for the calculation exact knowledge of current and future demands is needed. Nevertheless, cloud computing is a promising solution for businesses in small and large markets [Dur10] where it provides an alternative to reduce IT-infrastructure costs [BYV+09].

2.5 Semantics

The heart of a MAM system is its ability to store not only the file content itself but to also enrich the stored assets with additional meta-information. Thus, the stored files do also include rich information about themselves. This distinguishes it from simple storage systems such as FTP servers or One-Click-Hosters (OCH) where stored files do not contain much information and are therefore hard to find and often barely usable by others. Hence, this enhancement is not only needful to make the resources searchable but it is also crucial when it comes to provide useful and significant descriptions of the files for other, not involved, users as well. The information can also be later on used to search and retrieve assets to support different tasks than the initially intended ones. In turn, this too increases the likelihood to find specific resources. For this, once knowledge about a certain resource is available it needs to be shared in order to make it reusable for others. Therefore, Web 2.0[13] technologies draw from many user-based techniques such as voting, tagging and commenting to annotate data and generate such information. In addition to it, one can make use of automatic mechanisms to further collect more information about multimedia files too.

2.5.1 explicit knowledge

Voting is a simple technique to identify the relevance or acceptance of resources. Voting is well known from social media sites such as Facebook (e.g. the Like Button) but is also found on many comment areas on other sites. This strongly subjective technique gives a quick insight into the preferences of the users and helps to better understand their needs and likes or dislikes.

On the other hand, tagging is used to provide a short, usually one word, description or classification for a text or a resource. This technique is very powerful to get information about the content from the users themselves because of its simplicity and ease of use. This principle is widely known for its use in Twitter messages but can also be found on many video hosting sites and social media platforms. Tags are often only stored as a list of words along with the resource. To be able to identify tags in a continuous text, tags have to be specially marked. For this, often special characters are used to identify the beginning of a keyword, such as for instance a hashtag ("#") in Twitter. This principle can also be used for additional resources like other users ("@" in Twitter) or currency

[13] http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

signs ("€" or "$") for monetary amounts. By using more than one tag a resource can be linked with various other meanings and can be identified even better.

Lastly, commenting is used to provide extended descriptions as enhancements to resources. This is the most time-expensive but also the most enriching technique. Commending is also used to give recommendations or reviews for products (Amazon) or personal opinions (online newspapers) and encourages communication among different people.

These three techniques are not separated from each other and can be used together as well. For instance, newspaper sites such as derStandard.at often provide a comment section for each article. There, users can leave comments and up- or down vote other comments. This helps to quickly identify the best - or worst - comments at a glance. Of course such comments could also be enriched with hashtags to provide further meaning and classifications for the articles.

Further information can also be automatically extracted from meta-information which is already included in digital files, such as Exif, IPTC or XMP data from images[14] or from MXF video file format headers[15]. This metadata does not contain any semantic meaning but is used to describe technical features of the resource such as length of exposure or the GPS location of the recording. This is important since a manual creation of such metadata entries is very time expensive but can be easily automatised. Using such automatic techniques is therefore many times less costly in comparison to manually processing the files.

Finally, basic data such as the type or the size of the file can also be extracted from the asset files itself. This very simple measures provide only rudimentary information and do not add much additional information to the resource but are viable meta-information about the file itself.

2.5.2 further processing the knowledge

Drawing from these sources a lot of metadata can be added to the stored assets. Without further processing, this unsorted data can be overwhelming and thus useless for indexing, searching or reasoning. Hence, to structure the collected data it has to be further processed with semantic techniques [SR03] where the gathered data is linked to each other on basis of its semantic meaning to make logical sense of the whole dataset. Mechanisms which draw from collective intelligence and link the data with each other to enhance its quality are further applied to enrich the quality of the dataset. In addition to this, such techniques are also very scalable for huge datasets or knowledge stores and can be used for vast knowledge databases [SS09].

One way to achieve this is to use semantic web technologies such as information description models like XML based RDF or OWL languages [SS09]. These so-called

[14] http://www.myphotocentral.com/articles/exif-iptc-xmp-metadata
[15] http://www.digitalpreservation.gov/formats/fdd/fdd000013.shtml

ontologies provide frameworks of formal descriptions and relations for information and knowledge in hierarchical structures. They do not only provide meaning for humans but also make the concepts processable in hindsight of their meaning for machines. Implementing ontologies makes it possible to add meaning to the meta-information gathered and paves the way from only user generated content based Web 2.0 sites to knowledge based Web 3.0 sites [BLHL01]. This also introduces a transformation from content-centric applications to user-centric ones where the users provide meaning to the datasets and determine the value of the resources. Using knowledge-based systems has therefore many advantages also for small and medium sized companies [ED13].

Hence, starting with basic files users and automatic processes can add meta-information and give semantic meaning to them. This can then be used to create user generated, collaboratively ontologies or taxonomies - so called folksnonomies - which classify the resources and describe whole datasets. When such a semantic meaning is provided the MAM system does not only supply bare files but also adds knowledge about them for sophisticated knowledge based processings. For instance, such a system can make assumptions about the search and suggest related items based on the input or provide additional information about the resources.

2.6 Software engineering

Software engineering refers to all kinds of disciplines within the field of software programming such as designing, implementing and maintaining computer software. When the outlines of an application are defined one must find the right methods and techniques to implement the planned solution. For this, technologies, design patterns, guidelines and conventions exist from which one can chose. In order to find the fitting models in the following a brief overview of state-of-the-art software engineering techniques is given. Of course, it is to be understood that this list is far from being complete.

2.6.1 GUI widget toolkits

GUI widget toolkits are used to create programs which are directly runnable on the users machine without using any further technologies in between. A widget toolkit usually provides its own rendering engine and is runnable in an own window directly on the local machine. Widget toolkit libraries contain standard IO components such as buttons and input fields which can be themed or skinned to match the look and feel of the operating system or to implement an own style. Thus, using GUI widget toolkits is the common way to develop installable PC software with graphical control elements. GUI widget toolkits exist for all kind of programming languages and operating systems. One of the best known are Cocoa for Objective-C, Qt for C++, Swing for Java, which is at the time of writing replaced by JavaFX, or also Adobe Flash.

Since the release of Java 8 JavaFX is included in the standard JRE/JDK libraries and replaces Swing as the standard GUI toolkit in Java[16]. JavaFX provides more

[16] http://www.oracle.com/technetwork/java/javafx/overview/faq-1446554.html#6

sophisticated features than Swing like animations and basic media support and runs cross-platform. It can be layouted with the XML based markup language FXML and styled with CSS. For this, JavaFX defines its own CSS classes for its standard components but lets programmers override them or lets them define their own classes. This makes JavaFX easier to style and better adaptable in comparison to Swing. Nevertheless, JavaFX components can still be included in Swing applications and vice versa to enhance the interoperability between both technologies and make it easier for developers to use JavaFX features in older Swing applications.

2.6.2 Webtechnologies

Building complex dynamic web pages is difficult by using only HTML. Since the underlying HTTP protocol is stateless webpages need also methods to handle sessions, asynchronous calls or techniques to reuse code and site templates[17]. For that purpose, Web Application Frameworks (WAF) are used to support the implementation of websites and also simplify commonly used activities. WAFs work beneath the surface of a webpage and are not directly visible to the users. They provide means to manage sessions, ways to use site templating, enhance the usability of the site, have built in security mechanisms and often also handle database connection pools within internet applications. They strongly depend on standardized web technologies such as (X)HTML, CSS, JavaScript, AJAX and the usage of Webservices. WAFs enforce clean coding standards and hence separate between the content, formatting, data models and business logic. Often the Model View Controller (MVC) or similar design patterns [GHJV94] are applied to keep the data model, the presentation and the controllers separate.

The range of functionality of a WAF can be extended by using component suites which provide more sophisticated components for webpages such as sort- and filterable tables or extended features for standard HTML elements.

In contrast to content management systems (CMS) WAFs do not supply complete websites which can be extended by using plugins but provide a skeleton for an application, such as a set of code libraries with a simple API, which is easily usable by a programmer. Hundreds of WAFs can be found for each specific programming language, computing platform and intended field of application. WAFs range from simple static blog templates up to highly dynamic, AJAX based websites with an elaborate business logic and database system in the backend. As an example, notable WAFs are AngularJS, ASP.NET, JSF, jQuery, PHP, Ruby on Rails, PHP and many others. Clearly, implementing a complex MAM system is only possible by using the right WAF.

2.6.3 Webservices

WebServices (WS) are used for direct machine to machine communication to exchange data between two endpoints over a computer network such as the internet[18]. Webservices

[17] http://docforge.com/wiki/Web_application_framework
[18] http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice

are loosely coupled which means that the interface of the Webservice is described in a programming language independent way. This is in contrast to direct Remote Procedure Calls (RPC) used in inter-process communication (IPC) over a network such as in Java Remote Method Invocation (Java RMI)[19]. Here, the calls are not language independent and thus not compatible with most other systems. Contrary to this, Webservice clients can be implemented in different software systems and thus Webservices can be invoked from many different platforms.

Webservices usually use markup languages such as JSON or XML to describe their interfaces. Common Webservice protocols are REST (Representational State Transfer) which directly sends XML or JSON data via HTTP or SOAP (Simple Object Access Protocol) which is based on XML structured information sets and uses HTTP (or similar protocols) only for the transmission. In comparison to SOAP REST is much simpler since SOAP uses a more tightly structured approach whereas REST only provides an architectural style and guidelines for creating the Webservice[20]. Opposed to this, SOAP Webservices are described in a XML based and machine processable format called Web Services Description Language (WSDL) which is standardized by a W3C recommendation[21]. WSDL is platform, language and protocol agnostic and is understood by a wide variety of programming languages. It describes available functions which are provided by the server and defines all specific non-primitive data types in an attached XML file. This enables programmers to interact with the Webservice by only using the from the server provided WSDL file. This is a major advantage in contrast to REST based Webservices which do describe the interface in a machine processable format and hence should also always provide the client API. To automatically generate SOAP Webservice clients from the provided WDSL files many frameworks exists such as the Java API for XML Web Services (JAX-WS) or the possibility to add a service reference in Visual Studio for C# clients.

2.6.4 URLs

Unified Resource Locators (URLs) are a specialised form Uniform Resource Identifiers (URIs). They are used to specify the locations of resources in a computer network and how - with which protocol - to get them. In the internet URLs are used synonymous as the address of a website. Different mechanisms to construct URLs exist which cover different aspects of an application's demands. Since URLs are the main way to access websites the creator of a website has to carefully chose the right model. In the following two important schemas are discussed.

Semantic URLs To enhance the user experience and the websites accessibility semantic or also cool or pretty URLs[22] should be used to identify pages and resources on the website. Semantic URLs replace technical symbols and terms or complex site paths

[19] http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html
[20] http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/
[21] http://www.w3.org/TR/wsdl
[22] http://www.w3.org/Provider/Style/URI.html

with meaningful and human readable strings. For instance, the page path: /pages/secure/site.xhtml can be rewritten to /site. This enables users to intuitively identify the site's content at a glance, makes the URL more memorable and also improves the Search Engine Optimization (SEO) of the site. Such URLs are also used to decouple the page's URL from the servers internal structure. If the structure of the site changes, i.e. resources are moved to other folders, the URLs for the pages can be kept and broken links are avoided. This is especially helpful to maintain bookmarks and often accessed sites but also enhances the security of the site because the internal structure of the application is not disclosed[23]. This makes the site more stable and consistent.

Capability URLs Additionally, sometimes resources must be concealed from unwanted accesses. In MAM systems this is especially important for published assets which are publicly available but should not be findable by others. However, also private assets which are protected by the user or group access control model or even public ones should be available under a distinguishable, unique URL.

Several techniques for such secure- or one-shot URLs established during the last decade. These URLs contain an unique, often Base64 or Base36 encoded string which identifies the resource and is long enough so that it is hard to simply guess it. This is usually used to hide one-time URLs which expire after a specified time, such as for instance password reset confirmation sites, but can also be used to create public sites which are only findable by those who have the right link. Many Web 2.0 sites where no logins are required are based on this technique such as Doodle, TinyURL, public Google calendars or download links from Dropbox. This approach is signed as a W3C working draft since February 2014 where these URLs are called capability URLs[24].

Since capability URLs are only hidden but not protected by any more sophisticated measures W3C recommends some basic provisions to enhance the systems security. First of all the capability URL must be unique and not guessable. For that reason, it must be avoided to use short strings or easily deducible IDs such as consecutive numbers to identify the resources. Thus, using Universally Unique IDentifiers (UUIDs) or similar techniques is recommended[25].

With the URL hidden it must also be ensured that the URL does not get leaked or unintentionally publicised. W3C recommends using only a HTTPS secured connection to prevent exposure of the URLs or that they are shown in plain text in the network traffic. This is also crucial for pages where capability URLs are linked from. Furthermore, the pages behind such URLs should not include links to third party sites or mechanism to insert any links to external sites for instance via a comment field. This is necessary to prevent the browser to set the URL in the referrer header. If this is not possible the link should set the rel="noreferrer" flag at least. Moreover, it is also necessary to include the URLs regex in the Robots.txt to prevent them from being listed by search engines. Further measures include expiring the URLs after a given time or making it possible for

[23] http://www.nngroup.com/articles/url-as-ui
[24] http://www.w3.org/TR/capability-urls
[25] http://www.w3.org/TR/capability-urls

an authenticated user to reset the URLs at will.

Of course, other organisational arrangements must be met too. The URLs should not be published on any external or uncontrolled site and handled only with great care when sending to third party users.

2.6.5 HTML5

HTML (HyperText Markup Language) is the standard language to describe webpages. Put simply, in the end browsers do only interpret HTML and hence WAFs always render websites as HTML no matter how complex the composition of the code behind the webpage is. HTML is well known to many people and thus will not be discussed in any further detail here. However, since October 2014 the final HTML5 standard is available which is the first update of the HTML standard since 1997. HTML5 introduces new features and major improvements in comparison to the previous HTML version. Even the new standard was discussed and evaluated for many years but nevertheless, at the time of writing, all major internet browsers such as Chrome, Firefox, Internet Explorer, Safari and others support all features of HTML5. Hence, some of the new features are described in the following section.

One improvement in HTML5 is the initiation of dedicated semantic elements to structure the webpage[26]. Semantic elements are used just as tags and can be evaluated by the browser. They describe the structure of the site and the semantic meaning behind each section or content of the webpage. This is especially important for accessible clients which can help users with handicaps to find a relevant part in a site fast. These tags replace commonly by programmers used div tags such as , which is now denoted as the semantic element <header>. Other possible semantic elements are for instance <main>, <nav>, <footer>, <article> or also often used elements such as <figure> along with <figcaption> and <time>. Alone by using these tags it is possible for machines to comprehend what the section contains. A possible structure of a HTML5 valid page is depicted in Figure 2.6. The illustration shows how a webpage can be typically layouted by using HTML5 semantic elements.

HTML5 also adds new syntactic features such as the replacement of Flash as the defacto standard for playing videos on the web. For this, HTML5 introduced the <video> tag which is used to directly embed videos into webpages just as it is done with images by using the <img> tag. There were many reasons for this[27]. Flash proves to be a security threat and needs a lot of battery capacity. Even Adobe, in the anticipation of HTML5, already discontinued the development of Flash for mobile clients in 2011 [28]. Using HTML5 videos is also more performant since the video is played directly by the browser on the users hardware and thus using a plugin is not necessary. As of early 2015 latest browser versions only support videos of formats MP4, OGG or WEBM[29]. Each format was developed by an other company and so far the only format which is supported

[26] http://www.w3.org/TR/2014/NOTE-html5-diff-20141209

[27] http://www.apple.com/hotnews/thoughts-on-flash

[28] http://blogs.adobe.com/conversations/2011/11/flash-focus.html

[29] http://www.w3schools.com/html/html5_video.asp

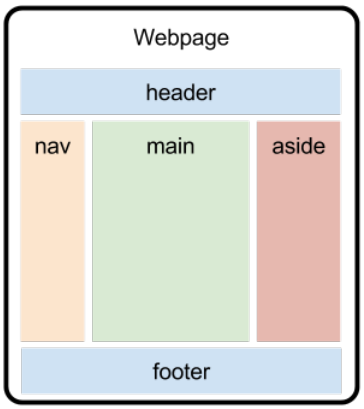


Figure 2.6: The illustration depicts a possible usage of some HTML5 semantic elements in a webpage.

by all browsers is MP4. Still, by using MP4 problems may arise because all codecs are not usable by all browsers. Hence, in order to support a broad range of different browser all of the above mentioned formats should be provisioned, so that the browser then can decide which format to use. For this, distinct formats can be included with the <source> tag. To support different devices and different types of network connections the videos should also be available in different resolutions. Videos with smaller resolutions can be used to decrease the loading time for slower internet connections or for devices with smaller display resolutions, whilst high resolution videos are more suitable for larger screens.

Additionally, similar tags exist for playing audio content (<audio>) and displaying scriptable graphics (<canvas>)[30]. HTML5 also provides new APIs such as methods for simple input validations directly in the input elements. This can be used for regular expression pattern matching for instance for email address verifications. Before HTML5 pattern checks were usually done with JavaScript which is slower in comparison than to do it directly in the browser. Input checks can be used in the form of <input pattern="regular expression">.

2.6.6 CSS

Cascading Style Sheets (CSS) is the standard technique for formatting web pages. By using CSS a clean separation of the documents content and presentation (look and feel) throughout the application is guaranteed. Every element of the webpage must be formatted with CSS classes which are also supported and used by HTML and enforced by most WAFs. Using CSS different styles for each component according to all kinds of different user demands can be implemented. As at the time of writing this thesis in early 2015, the latest CSS standard is CSS Level 3 (or CSS3) which's finally version is

[30] http://www.w3.org/TR/2014/NOTE-html5-diff-20141209

not yet published as a recommendation by W3C[31]. Nevertheless, many CSS3 modules are already supported by current browsers and can be used for up to date websites.

With CSS3 it is now possible to make website usable for all kinds of devices and types of browsers. This does not only eliminate the need to implement dedicated mobile sites or even mobile applications but does also enhance the overall systems accessibility. Thus, Responsive Web Design (RWD)[32] is the state-of-the-art technique to achieve this.

RWD is a programming paradigm which's goal is to implement websites in way so that they are viewable on all kind of different devices with all kind of different screen sizes. For this, it is necessary to change sizes of elements and reposition them in the website for different display sizes to fit them better. For that, all elements must use relative size properties, such as percentages, to fluidly adapt and scale the layout to all screen sizes. This is in contrast to adaptive web designs where fixed views of a webpage are programmed only for certain predefined screen sizes.

To do so, CSS3 introduced Media Queries which can be used to detect media features such as the viewport size and filter and overrule CSS classes for different devices with different screen sizes directly in the CSS code. Hence, it is possible to completely change the layout of a page for each screen size used.

On most mobile devices it is more convenient to scroll vertically than horizontally so placing elements beneath each other is usually more usable. As an example, this can be used to structure a website according to the available screen width. As an default setting the navigation (<nav>) and the main content (<main>) may be floating right next to each other. This is done with "float: left", or respectively "float: right". To be then able to scale to smaller screens a Media Query can be defined for smaller screens such as displays with widths below 768px which removes the floating of the elements. This forces the browser to re-render the main div below the navigation bar for smaller screen sizes, as exemplified in Figure 2.7.

Another new property which is introduced in CSS3 is the Flexbox layout. Flexbox defines a container which's items can be flexible arranged. For this, it also provides means to order items within such containers in a specific sequence and aligned them to the boundaries of its parent. Using Flexboxes for the layout items can be reordered in the Media Query to fit the screen size better. For instance, this can be used in a nav section to reorder a logo and a navigation bar. As an example, if the viewport width is smaller than 768px the logo is reordered to be displayed underneath the navigation bar (which is then horizontally aligned) as shown in Figure 2.8.

This can be done for vertical as well as for horizontal boxes and elements can be also numbered which allows using even more flexible arrangements.

By default all browsers apply predefined CSS values to webpages, such as additional paddings and margins, when such styles are missing[33]. This causes inconsistencies across the site which are also different for all kinds of browsers. When these values are not

[31] http://www.w3.org/TR/2000/WD-css3-roadmap-20000414
[32] http://alistapart.com/article/responsive-web-design
[33] http://sixrevisions.com/css/the-history-of-css-resets

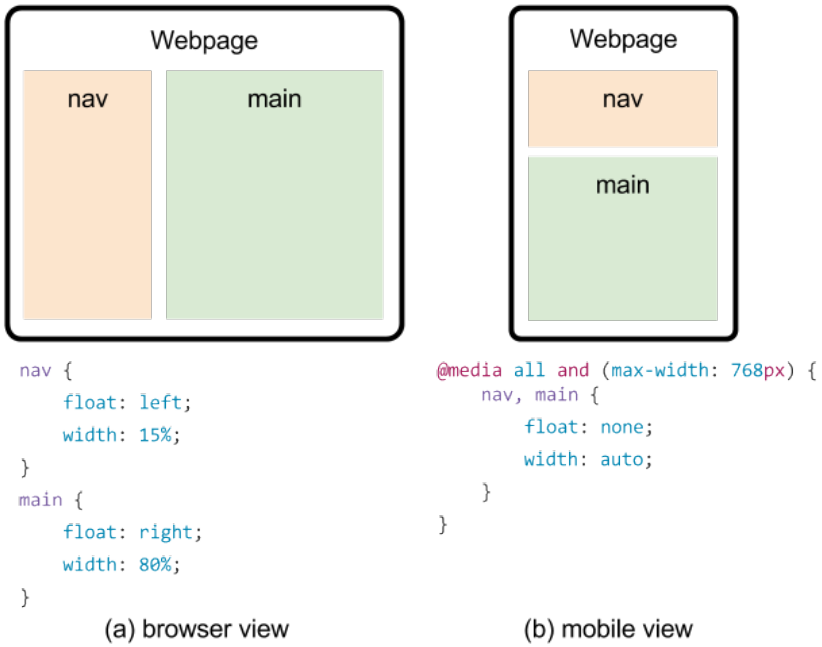


Figure 2.7: A Media Query is used to restructure the layout of a website. If the screen's width is greater than 768px the navigation bar is displayed beside the main content (a). If (b) the viewport shrinks below the threshold the navigation bar is positioned on top of the main content.

overridden different positioning and deviating sizes for many elements are displayed. Some of these default properties are very salient whereas others are hard to find and therefore hard to reset. To circumvent these problems a CSS reset stylesheet is often applied prior to the websites stylesheets to clean all predefined values. The most minimal CSS reset stylesheet, but also one of the most effective ones, is Andrew Krespanis's[34] hard reset from 2004 which resets all distances for all elements on the site:

```
* {
        padding: 0;
        margin: 0;
}
```

This simple code snippet is used in some form or the other in many websites. Of course, further, more sophisticated reset stylesheets can be found online.

2.6.7 Persistence

To organize the persisted data websites usually use databases in the backend. Providing the store data entries in such a structured, safe and secure way makes it possible to handle

[34] http://leftjustified.net/journal/2004/10/07/css-negotiation

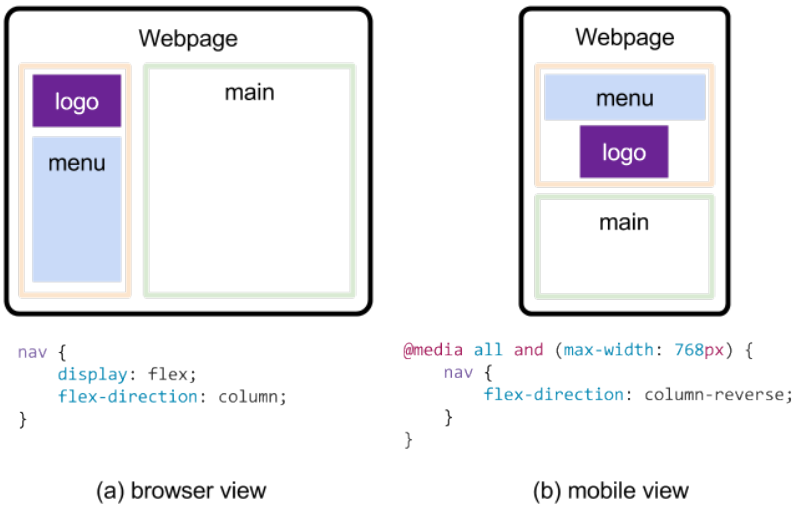


Figure 2.8: By using a Flexbox container for the navigation sidebar the structure of the elements can be adapted to the screen size. Normally, the logo is displayed above the menu (a). When the viewport width is smaller than 768px the logo is rendered beneath the navigation menu (b).

huge and complex data sets like they are used in a MAM system. For that reason, the database works at the very core of an application as a central and critical component of the MAM system. To administer databases, Database Management Systems (DBMS) are used as an interface between the users or applications and the databases [CB05]. These DBMSs provide accesses to the stored data and are used to efficiently interact with the data collections. DBMSs allow users to manipulate data entries such as storing, retrieving, updating or removing them from the database and additionally enforce access restrictions on these operations. They provide a coherent view of the data and means to access it from different applications which is often done by using Structured Query Language (SQL) for relational databases. Thus, during the last four decades Relational Database Management Systems (RDBMS) became the defacto standard for databases [CB05]. However, with the increase of big, complex and fast changing data collections in websites such as in social networks or from sensor data old database models face problems [FTD+12]. Such big data collections are no longer manageable within a relational database. Consequently, database models emerged which overcome the limitations of RDBMSs by using faster key-value stores or document-oriented structures. These new kinds of databases are generally referred to as NoSQL (Not Only SQL) databases [Lea10].

Such databases implement very different approaches and usually do not provide a standardized interface to query the data. Data is not stored in tabular relations and thus NoSQL databases do not require fixed table schemas. Due to their unstructured nature data is denormalized to avoid expensive joints. Each entry usually is identified by a globally unique key which makes the entry identifiable across large and distributed

data collections. These new database scale horizontally and for that reasons are very fast to query which makes them also usable for real-time web applications or in large distributed systems.

Depending on the application field and the requirements NoSQL databases can have performance advantages over standard SQL databases especially for huge data collections [Lea10]. Twitter (Apache Cassandra)[35] and the NSA (Apache Accumulo)[36] are two examples where NoSQL are deployed in real world scenarios. However, NoSQL databases can prove to be valuable even for smaller applications [Lea10]. It can be estimated that about 80% of a websites execution time is spend in the backend, which is usually for querying the database[37]. By using faster and more flexible NoSQL databases also smaller websites can make use of the advantages of NoSQL databases. Thus, using such technologies already proofed to be successful for MAM systems [NM03].

[35] https://blog.twitter.com/2010/cassandra-twitter-today

[36] http://www.informationweek.com/applications/nsa-submits-open-source-secure-database-to-apache

[37] http://www.speedawarenessmonth.com/when-8020-becomes-2080

CHAPTER 3

Methodology

To find a suitable solution for a novel MAM system various different scientific research methods were conducted. The methods were used in an iterative approach where, based on the outcomes of each method, subsequent researches was carried out. In the following, an overview of the methods used is given along with an explanation why they were chosen. This is done in a chronological order of how they were used throughout the process of writing this thesis.

3.1 Literature research

To begin the research the available literature about MAM systems was studied, already established techniques in the field of the media industry examined and existing implementations of MAM systems were investigated. This was done in order to orientate in the field in question but also serves as a basis and theoretical background for the further researches. Thus, digital and printed literature from scientific as well as from non scientific sources was surveyed. Used sources include conference and journal papers, ebooks and newspaper articles but also blog posts and private websites from various research fields and groups of users.

Throughout this process findings were consistently evaluated and the different aspects found compared with each other. So found conclusions were discussed with the advisers to get additional feedback. In this process already known problems in MAM systems were identified but also new questions were posed which were answered in the subsequent research phases. The outcome of the literature research is mainly incorporated in Chapter 1 and Chapter 2 but continues to be used throughout this work. Without these profound knowledge of the field in question further investigations could not have been possible. With this knowledge in mind the right questions to ask for the expert interviews were found.

3.2 Expert interviews

In order to gather further qualitative data seven experts from the media production industry were interviewed. These experts include people who work in different areas in the media production industry and can draw from a year long experience there. For this, already established contacts with professional users from the business who were willing to take part in an interview were used.

All of the in this thesis interviewed experts work in different companies including small documentary filmmakers, advertising companies up to medium sized TV show producers. Some of them actually work with MAM systems or with similar software, others use different tools or no software programs at all. All of them live in or close to Vienna and work on all levels in the production cycle. Hence, the interviewees include a screenwriter, an organisational assistant, a project leader, two production managers, a TV show producer and a cutter. These experts provided invaluable insights into this field and made major contributions to the solution found. They helped to get a good overview about the problems and challenges people face who are working in media production companies. They talked not only about the technical side but also about organisational as well as social problems that occur in such an environment.

The experts were interviewed in an informal and freely way. Thus, the interviews were conducted less as question asking but rather likely as conversations. The interviews took place either in coffee houses or bars or at the workplace of the experts where they were able to show directly how and with what tools they work. During the interviews the focus of interest lied in what the experts do and how they accomplish their tasks and how technology can help them to do their work. Hence, typical questions asked concerned if and if yes which kind of tools they use for their work and where they see drawbacks of these tools. This helped to recognize problems of commercial systems which are used in professional environments but to also comprehend what these systems do better than their competitors. In that regard, another typical question asked was what kind of tool they would like to have for their work if they could make a wish. This provided insights which helped to identify which kind of system is most desired by professional users and how such a system should be implemented.

The interviews were carried out during the year 2014. A detailed summary of the interviews can be found in the Appendix. Conclusions drawn from them are found in Chapter 4. To keep the experts anonymous no names either of the interviewees or of their companies are stated in this thesis.

3.3 Modelling

Based on the outcomes of the literature research and the expert interviews a novel solution for a MAM system was planned. The specification of the system can be found in detail Chapter 4. The proposed system lays a special focus on usability such as the GUI and User Experience (UX) design to enhance the ease of use of the system. Therefore, different GUI models of the technical solutions for both the client and the webapp for the

central server were created beforehand to evaluate different solutions for the GUI right at the beginning. This was done by creating wireframes which represent the outlines of an application and act as visual guides also for non technical versed viewers and future users [Gar10]. The wireframes also illustrate technical features of the projected application and depict the systems graphical design for the different interfaces and navigation cases. They were used to elaborate the generated ideas for the MAM systems' GUIs and to get first impressions of how the realization could look like. Accordingly, the modelling started with low fidelity wireframes on paper and afterwards digital high fidelity wireframes for salient design ideas as exemplified in Figure 3.1 were created.

Figure 3.1: Two examples of wireframes for the MAM systems' client. To elaborate different design ideas several layouts for distinct courses of action were created. Whereas (a) shows a layout where each consecutive action to be made is done in a different view, wireframe (b) was created to show all necessary control elements in one window. Draft (b) was later on refined and used as the basis for the actual implementation of the client.

During this process the individual wireframes were refined or later on discarded until it was agreed upon one. Inappropriate designs were sorted out at the very beginning so that further implementation details could be clarified and the overall look and feel of the product communicated. In this stage of the design process first decisions about the implementation design such as the layout, color schema and control elements of the prototypes were already made.

3.4 Prototyping

Afterwards, a Proof-of-Concept Prototype of the client and the server application was implemented in order to demonstrate the usability of the solution and make it possible to use the system for subsequent evaluations with the experts. The prototype contains major functions of the proposed MAM system and represents the capabilities of the finished product. Thus, a subset of selected features, i.e. so-called use cases, which would be needed for the finished product to be usable in a real-world scenario are already included in the prototype.

The prototype is implemented fully functional in a vertical design, i.e. each implemented function works through all levels of the system, but with a broad horizontal range since almost all projected items are implemented with only about 15-20% filtered out. Hence, this high fidelity prototype provides a fine graded resolution and is close to the finished product. It can be readily installed on a PC, or respectively on a server in the public internet and is usable for extended user tests and benchmarks to evaluate complex user interactions and the systems utility. Thus, with this prototype at hand substantial characteristics of the system can be already evaluated in the field. For these tests, the server application is installed on a public server to make the system accessible to a large user base over the internet. Furthermore, the test users were provided with an already set up laptop where the client program is installed and test files are supplied. Screenshots of the two applications can be found in Figure 3.2 and Figure 3.3 .

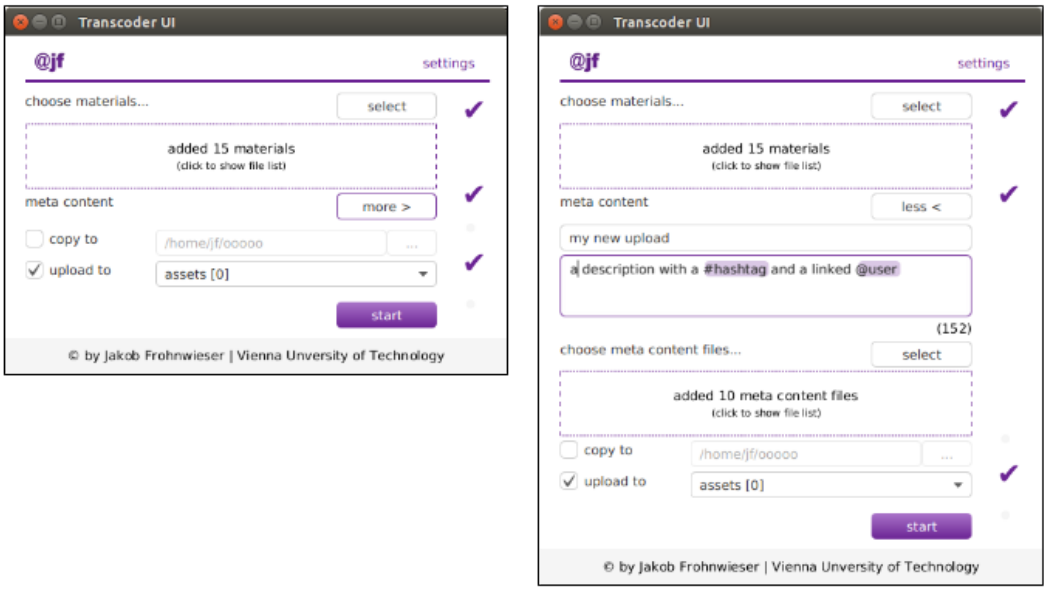


Figure 3.2: Two screenshots of the local installer client. (a) shows the main view. In (b) the optional controls for the meta-content are toggled.

The prototype was implemented in an agile and iterative development cycle with close consultation with my advisor. During several face to face meetings (one to two per month) and in numerous email conversations the systems' features were further refined, the look and feel of the system discussed or also other functionalities added or discarded. The implementation was carried out beginning in the autumn of 2014 and finished in early 2015. A detailed description of the system and the implemented techniques and algorithms can be found in Chapter 5.

The prototype is designed to be extendible for future requirements in order to be usable in real-world production companies. Therefore, it is implemented and designed in a way that others can easily continue the work and adapt it to their needs as a basis for future extensions. The implementation is kept simple but also rich in features and performant. This makes it possible to use the software in small and medium-sized media production businesses without the need to rent powerful and expensive hardware in the cloud. Thereby, an important requirement met was that the written code meets current standards of software development regarding security and usability and makes use of state-of-the-art developments in Human Computer Interaction (HCI) design.

3.5 User tests

After the implementation of the prototype was completed five experts, partly from the previously conducted inquiries or from different institutions, which were willing to test the application and give feedback about it were interviewed. The interviews were carried out in February and March 2015.

For the qualitative inquiries the usability design technique Thinking Aloud [LR08] was used. For his, typical tasks to carry out in the system were given to the interviewees. While they performed these tasks the probands were observed and asked for direct feedback about their thoughts. The outcomes were captured in rich notes. Throughout this process the testers identified advantages but also weaknesses of the implementation and gave valuable insights of what should be revised and improved in upcoming implementation iterations. This contained mainly usability concerns but also problems with the systems architecture and the naming schema. The outcome of this evaluation is recorded in Chapter 6 and Chapter 7. Some, especially needful, suggestions from this process are already included in the current implementation version to quickly improve major shortcomings of the system.

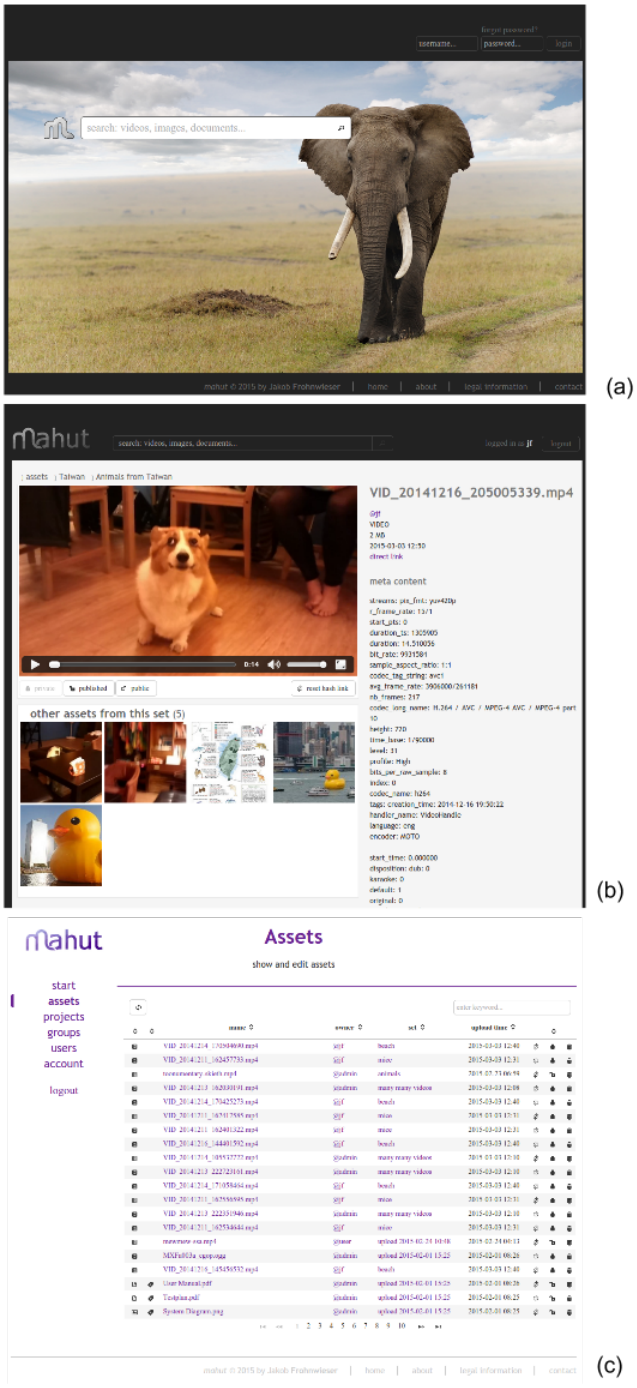


Figure 3.3: Three screenshots of the webapp. Figure (a) shows the main view of the central server application UI with the search bar and the login field. In (b) a view of an asset is depicted. (c) shows an administration page. To make the distinction between public and private pages more apparent public pages are generally dark whereas private pages for the backend administration are white.

CHAPTER 4

Analysis

During the interviews it became apparent that despite the widespread use of specialized tools in the media industry many unresolved problems still exist. Most of these problems seem to emerge because many of the commonly used tools do not integrate seamlessly into the production cycle and are not able to work together. For instance, most available mobile applications are not able to connect to in-house storage systems, proprietary cutting and archiving tools are not open to other software applications, whereas online based video platforms usually do not support all video file formats, codecs and sizes. Therefore, information gathered during the production phases often gets lost and is not available for future applications. Thus, many filmmakers rely on outdated tools such as spreadsheet programs or simple paper notepads to preserve information which would otherwise be lost forever. These solutions usually bear other problems since existing software and editing programs do not represent individual organisational structures and workflows. Consequently, there is often a large range of additional tools in use in order to be able to connect one program to another and to archive all ideas and artefacts created. Nevertheless, a lot of manual processing is still necessary which is often time consuming and prone to failure. Specific information is therefore frequently not recorded and only known to a single person. Hence, as found during the interviews, often the whole company relies in some parts on one person alone.

In the rest of this chapter these problems are addressed and a solution is proposed to eliminate these problems. For this, a novel system is introduced which integrates all of these requirements seamlessly into the production cycle without the need to add any additional software. This system is capable to cover most processes and is flexible enough to fit the needs of individual users from different organisations.

4.1 Problem description

In order to be able to find a workable solution for a novel MAM system the problems behind the current solutions have to be understood first. During the interviews it

became apparent that most professionals often work with very basic, standardized software programs only. Almost none of the persons interviewed used any specific or more sophisticated tools than what is usually pre-installed on most computers and smartphones. The most common programs used by the experts from the media industry are word processing and spreadsheet applications, social networks and instant messaging software and services such as email and, of course, telephone and SMS. This might be due to the fact that most people use these type of programs for private purposes too and are therefore already familiar with them. Especially social media platforms have become a powerful tool to attract and coordinate users for businesses [KH10] and are therefore readily used by many companies. Still, it is recognised that whilst most of the used programs are workable for their intended purpose they often do not work together for the larger workflows behind the production processes.

For instance, most filmmakers (from authors and assistants up to producers) often need to share spontaneous ideas and artefacts. These can include pictures from the set or short videos from ongoing productions as well as rough cut videos for their colleagues or contractees. A great variety of software products are used for this. As a production manager reported, he sometimes needs to share photos from the set with other team members in order to communicate concepts or to clarify orders. In order to do so, he uses an instant messaging app on his smartphone which also downsizes the quality of the files. Other filmmakers use different programs for that propose such as email or MMS. While at a first glance this information may seem to be only usable for the current occasion it is also viable when the video files resulting from this process are archived later on. Any information is helpful when it comes to search for specific video tapes or takes. Regardless which program is used all of those have in common that once the information is sent it is then lost since it is not archived and linked to the task anywhere. Yet, this happens with textual notes and audio or video recordings at all times during the entire production cycle.

It is also noticeable that most of the interview partners used already existing services for rather private audiences such as Vimeo to share video snippets among their partners. Although these platforms are quite user friendly another producer confessed that they do not know if the files uploaded there are still under their copyright. Moreover, since they usually use only one corporate account for the whole company proper user administration is not possible for them. Another problem he pointed out is that these platforms offer only a very limited webspace and so they have to frequently delete older files. While on such platforms video files are easily uploaded and shared among other users it is also not possible to download these files from there or even connect the platform to an in-house archiving system. Files hosted on such platforms need to be manually duplicated and saved elsewhere in order to preserve them for future uses. Furthermore, these sites only offer a very basic tagging system for metadata and stored files often cannot be found later on.

Other experts from larger media production companies noted that they have in-house FTP servers where they store and share video files. They are able to manage the server on their own and can keep everything to the company's own responsibility. With the

right FTP client program files can be up- or downloaded and automatic backups can be performed. Moreover, an extensive and fine grained user management is also possible here. Yet, all of the interviewees said that they hardly ever used these systems because such systems are too complicated for them or their external partners. In FTP, users are administered directly in the operating system on the server and thus the user needs special knowledge for the user administration what most often only computer experts have. Also, metadata entries are not supported by those systems and a project structure is mostly only established via a folder hierarchy on the server's file system. Hence, the location of each file is often only known by a small number of users. Consequently, they still rely on video hosting platforms when it comes to share their data, for instance, to present rough cuts to their contractees.

Another interviewee's company built its own storage system which is also available for purchases online via a webshop. This system not only archives and searches for video files but it also enables customers to view and purchase selected clips over the internet. For that reason, a database with metadata information exists with metadata entries for each video snippet. This system was designed to perfectly fit their needs which is mostly due to the fact that it was development by one of their employees. It is user friendly and thus used as their main archiving system to find old video snippets. However, it still needs to be maintained regularly in order to meet new requirements. Unfortunately, since the company faced severe financial problems during the last years this was hardly ever done which became a major drawback of this system. Seeing this, pre-built commercial or open source platforms have the advantage that constant development is done externally with the arising costs shared among several other tenants. Upcoming technical requirements and improvements can be implemented regularly with no need of intervention by the individual users.

To replace such ill fitting systems a specialised and open multimedia file platform which combines the advantages of these systems is needed. Such a system should combine the user friendliness of video hosting platforms with the extensive possibilities of in-house FTP servers along with the capabilities of metadata databases. Such a system is easier and faster to use and - depending on the design - open for further company specific developments and extensions. This makes it possible to implement it as a company owned archiving system which stores numerous meaningful metadata tags for the digital artefacts. Most of the interviewed experts requested this must happen automatically without the need to manually process the files. In addition to it, if required, it must be possible to enter further information afterwards as well.

As discussed, current commercial or open source video hosting services do not offer these functions and therefore do not fit the requirements of the media industry where a lot of large, up to several Terabyte big files have to be stored and managed. Therefore, the need for an extensible multimedia file storing platform which is used to search for and share digital files with different formats and sizes but is also capable for long term archiving of such files is apparent.

This system should be able to run on in-house servers as well as on remote webservers in the public internet. The users then can decide where their files are stored and can

resolve copyright issues and security problems according to their needs. Such a system has to be centrally managed by external experts so that the users do not need to worry about the technical side of the application but can concentrate on their work. Depending on the decided environment it can offer a virtually unlimited webspace and is available also for remote users from external companies. The requirements for such a highly reliable system which satisfies many of the discussed wishes and needs of the professional users in the filmmaking business are stated below.

4.2 System requirements

Based on the problem description in the last section requirements for a novel solution for a MAM system are identified. Since the new platform is described very flexibly it is expected to meet most of the needs expressed by the interviewed experts. The system's key points are defined as follows:

- The whole system must be designed as a central service provided over the internet. The main processing program will be installed in a cloud system online or alternatively on company owned webservers. Hence, it can be managed centrally.
- This will make the program available everywhere and at every time so that it is also reachable when the company's office is closed or if a specific user is not in the office.
- For this, the platform will be accessible with all kinds of up-to-date hardware appliances, especially desktop computers and mobile devices such as smartphones. The stored multimedia files must also be viewable on many different devices.
- For that reason, digital video files must be converted and stored with appropriate formats, resolutions and bit rates to fit different kinds of displays and in order to be streamable to all users at every place, even without high speed internet access available. Existing video sharing platforms such as Vimeo or Kalture can be integrated to share videos on pre-existing channels.
- Thus, special clients for each platform, such as a desktop program or a mobile application and an responsive website are planned.
- An additional interface for an in-house archiving system can be implemented with which the aggregated data is backuped and stored in a storage cluster. This also helps to make the stored data available for future productions.
- To make it easier to search for specific information the video files will be stored along with metadata entries for each file. This metadata must be readily accessible, easy to generate and must support a great range of different search algorithms. It must be possible to use already existing data from the pre-production and planning phase and automatic sensor data from the recording devices such as IPTC information or EXIF entries.

- Altogether this system must be runnable on almost all current devices, accessible and easy to use with great attention to usability and performance.

4.3 Proposed solution

Combining all of the above stated requirements a novel MAM system is designed which is able to store, manage and release digital assets. The basic concept of the system is taken from Schneider's [Sch11] hybrid SaaS solution for a MAM system (see Chapter 2) but it also incorporates ideas and methods of CSCW systems to enhance the system's usability [PRM+04]. The system is designed to be reachable via a public available website but also supports various clients for different hardware devices. Furthermore, it supports automatic backups to an in-house storage system where the stored data can be linked back to the original material files via their generated unique IDs. Hence, the system consists of three major parts as depicted in the system overview in Figure 4.1. An additional fourth part, the backup interface for the local in-house storage system is left out of scope for the purpose of this thesis. The individual parts of the system and main concepts are described in detail below.

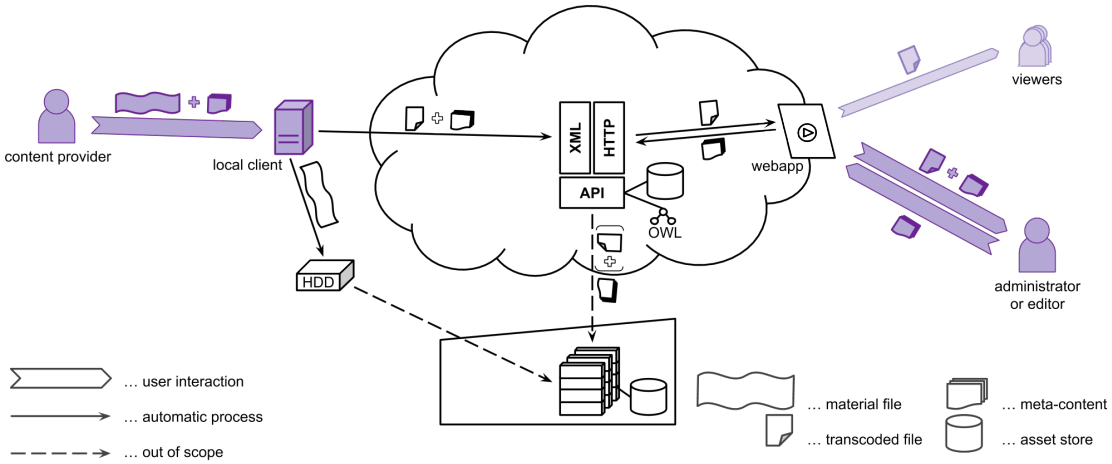


Figure 4.1: An overview over the proposed system. In the center, the main component, i.e. the central server application where all data is processed and stored is depicted. An additional web interface (right) and a machine readable client interface (left) are provided to extend the functionality and ensure a direct access from various different devices. Below that, the processed files can also be mirrored to an in-house backup system.

4.3.1 Storage concept

The MAM system is used to archive digital assets and provide means to find and retrieve them. In this context assets are all kinds of digital artefacts such as videos, images or

documents. For the purpose of this system a distinction between primary videos, so called material files, and meta-content files is made. The meta-content files are secondary files to describe the transcoded materials and thus the meta-content files are used to add further meaning to the materials. These can include behind the scene images from a filming spot or textual dispositions for a specific day of shooting. Each of these artefacts gets a unique ID assigned which is used to identify the resources and to also link them back to the externally archived raw material files.

The assets are pooled in special collections which are referred to as "sets". These are containers which are used to organize the contents and are very similar to folders in a filesystem but do also contain textual metadata information with marked keywords (so called "hashtags") that can be searched for. A virtually unlimited number of sets can be added whereas each set can also contain unlimited sub- or so-called "child sets". This can be used to build hierarchical structures of the sets and assets. For instance, one might create a new set which is regarded as a folder for a new project. Now, users can add new child sets with their data to the project and create a rich documentation for the work. For this, adding metadata is a powerful way to make the sets and their contents searchable and faster to find. For each set a user group must be entitled to have either read or write permissions to make the set viewable or editable for them. So stored assets and sets can be shared via a capability URL (obscure or unguessable URLs which are only know to entitled users[1]) to make them accessible for external users. For this, assets and sets can have different share modes. Possible share modes are: "private" for sealed resources, "publish" for shared resources which are only findable with the right URL and "public" for resources which are freely accessible by everyone.

Moreover, assets and sets can be marked and found with hashtags. To that end, an OWL ontology is used to provide knowledge about the domain and make automatic suggestions for searched terms. If hashtags are specified for an asset or a set these hashtags are then saved in a special list which links back to the resource. If the provided ontology contains terms which correlate with these hashtags the hashtags then can be used to make automatic guesses about related items. For instance, if a user searches for a hashtag, based on the provided ontology the system can suggest available related items. This also helps if a user searches for a special term which cannot be found in the system. Here the ontology can be used to show related items from the database which are similar to the searched term. If, for example, a user searches for a giraffe for which no data is available the system might suggest to also view videos of an elephant. Other intelligent search algorithms based on the ontology can be added further on.

Since the system is used to handle large collections of data the image of an elephant (c.f. [FTD+12]) obtrudes. Thus, it is named after the Sanskrit term "Mahout" which defines "a person who takes care of, rides, and controls a tamed elephant"[2]. Since the English translation "Mahout" is already taken by Apache[3] I use the German translation "Mahut". Because of that, as show in Figure 4.2, the 'm' in the mahut logo depicts a

[1] http://www.w3.org/TR/capability-urls

[2] http://www.thefreedictionary.com/Mahout

[3] http://mahout.apache.org

little elephant. In the following section each of three implemented parts of the system are described in detail.

Figure 4.2: The mahut logo. The "m" is designed to depict an elephant.

4.3.2 Central cloud service

The central part of the video hosting platform is a server application which runs on a dedicated webserver and stores all artefact files along with metadata and additional information in a database. The basic function of this server is to retrieve, process and store data and represent it in an appealing way to the user. This server is compiled to run on both in-house as well as on external cloud servers in the internet. This makes it possible to have an IT-infrastructure "without hardware" [Cos11].

The system is designed to be highly performant to avoid wasting of resources while also being highly secure to ensure the integrity and authenticity of the stored data. To handle the complexity of it, it consist of several submodules. In the backend a database management module is deployed along with a powerful search engine which is based on a domain-specific ontology. Implementing such an intelligent search mechanism makes the assets easy to search and supports the users to accomplish their tasks. For the frontend, a website which is directly accessible over the internet (via HTTP/S) and different XML based Webservice interfaces with which client programs can access the system directly are available. Both communication channels are described in detail below. An optional interface for an offline archiving system where the data can be exported is envisioned too but left out of scope for the current implementation.

It is also envisaged that once the server gets a video file it converts the file to different sizes, i.e. encodes it with different resolutions, to make it streamable to all kinds of devices and usable for different internet connections. Standardised web video formats, such as MP4, OGG, or WebM are used to achieve this. Using state-of-the-art HTML5 web technologies the use of Flash or any additional browser plugins is avoided.

According to the interviewed experts it can be expected that one project needs about 20TB of data. Since today cloud webspace is fairly cheap it is estimated that using this system as a cloud service is cheaper than operating it in an own computing center [Dur10]. Of course, prices may vary according to the needed webspace, availability, processing power and bandwidth. If this system is installed on an already existing in-house server costs could be additionally reduced compared to renting a cloud server.

4.3.3 Webapp

A complementary web interface is provided where users can view the assets and manage the rights for them. The website is implemented in a way that makes it fluently scalable to all screen sizes and therefore also usable on mobile devices without the need to use specialised apps. Since it is designed to provide a fast and easy to use entry point to the storage system there is no need to offer all possible functionalities here. For that purpose, a separate installer client is available. The website focuses particularly on the basic functionality with special attention to the presentation of the assets and the management of permissions. Nevertheless, it is also possible to edit metadata tags or delete files there. Usability is especially important here since the website is also viewable by external users from different companies who might not be familiar with this system.

Users can access sites by opening different capability URLs with a link for the viewer and optional administration controls for an entitled editor. A basic user management is provided with fine graded setting options for each user or user group. To perform special tasks, e.g. create a new project or change the view rights of an asset, a user must authenticate him- or herself via a login at the site. To manage the system, simple asset, set and group administration sites are provided. Any further configuration of the site is not needed. The webapp can be run on the same web server along with the main server application and does not need much further resources.

4.3.4 Client program

The MAM systems local counterpart is conceptualized as a standalone fat client which is installed at the user's premises and is able to communicate with the central server installation over the internet. To be able to upload extensive data collections to the server a processing of the files has to take place first. These computing expensive operations have to be done beforehand to the upload at the user's premises to speed u the uploading of the files. As stated during the interviews, in the media business one has to deal with extremely large video files in the range from GB up to a TB. In order to save bandwidth where high speed access to the internet might not be available this client program is provided to convert video files to an HTML5 web video format which is used by the central cloud service and while doing so reduce their size.

However, the most valuable data, i.e. the raw material files, are kept locally and are not available to the public. Only a downscaled version of them which is not further commercially viable is available online.

Additionally, the client also offers means to backup the raw material files to a specific folder or hardware device for archiving so that the materials are secured at a different place. It is also possible with this client to add metadata information to the uploaded video files to provide semantic meaning for them. Here it is differentiated between additional metadata files such as images or documents which are related to the materials and a textual description of the uploaded set. The metadata files are treated as additional assets but are referred to as meta-content files. However, these files can be searched for and displayed like the material files.

An optional textual description is saved along with each uploaded set. This description can contain keywords marked as hashtags ("#") that can be searched after. Along with these manually created metadata entries automatically generated entries are stored for each file in order to be usable by the search engine and the archiving system. This happens on the fly during the uploading without the need to enter information manually. Possible sources for the metadata information can be previously created text files (e.g. from dispositions), buzzwords or descriptions and other non-explicit data such as the recording time, GPS coordinates or the username of the user. Thus, a lot of information is automatically derived without ever needing the user's active intervention. Furthermore, the client also reads information from video file headers which allows it to always automatically ensure a certain basic level of information for each file.

Special attention is paid to the usability of this client. It is usable without many interactions and has therefore a very basic user interface. To give a natural mapping [Nor02], the individual elements are logically arranged in their needed execution order, beginning at the top and continuing to the bottom. Each step is only enabled if a possibly dependent preceding step is successfully finished. To reduce the complexity of appearance [Nor02] the input fields for the meta-content are hidden by default. Therefore, input controls which are not necessarily used are not always visible but can be displayed by pressing a button. Direct feedback is given for each control field and an overall status is depicted with coloured check-marks on the right. The client supports drag and drop for all files and does not need much interaction. All needed configuration values are configured once at the initial startup of the program and therefore it is not necessary to enter these values, such as passwords or connection URLs, again. Every choice a user makes, e.g. chose an upload set or select a backup folder, is automatically saved and therefore it is possible to upload data based on previous interactions with only one click. For the transcoding of the video files and for the reading of the in the video files contained metadata entries the client uses FFMPEG in the background which must be therefore installed at the users machine. Whilst converting and uploading the program gives detailed feedback to the user with different progress bars for each action such as the file copying or the transcoding. A screenshot of the client can be found in Figure 3.2.

The client is written in Java and installation bundles for the most important operating systems, i.e. OS X, Linux and Windows are available. Therefore, it is runnable on most computers and can be readily installed and used. In addition to this, it is also considered that this application - or parts of it - are ported to mobile devices. This would enable users to also access, manage and use the cloud system on the go from remote places. Since mobile devices do not have the processing powers of desktop computers it will not be possible to offer the full potentials of this program here. However, current mobile devices usually do not record with resolutions comparable to professional equipment so it is not necessary to scale the there created video files with processing expensive algorithms. For the current implementation the mobile clients are left out of scope. Yet, the interfaces to the central cloud service are already designed to be also usable for other clients written in different programming languages.

CHAPTER 5

Implementation

Based on the system description which is based on the expert interviews the implementation of the mahut system began in autumn 2014 and continued up until early 2015. The implementation process was accompanied by several in-person meetings with my advisor where the latest development status was discussed and further features planned. The whole implementation was then evaluated with five experts. Afterwards, further improvements based on these interviews were conducted.

5.1 System overview

The implementation of the proposed system consists of two major parts - the central cloud service with the website (webapp) and the local client user interface (UI). To make it possible to reuse certain functionalities and to ensure the correctness of the Model-View-Controller (MVC) pattern the system is designed as a combination of seven submodules as shown in Figure 5.1. Each submodule comprises a specific logical function whereas the dependencies for each are transitively resolved. That means that if a submodule is linked to another submodule all submodules which use this one also contain the other submodule(s). The submodules are defined as follows:

mahut-parent-pom
This is the parent for each submodule and consists only of the pom.xml where the version for each library which is used in this project is defined and executions of build-plugins are configured. It does not implement any functionality but is used by Maven (see below) for each submodule to resolve all dependencies and to configure and execute the build of the submodules in the right order. Using a parent POM (Project Object Model) is a standard Maven technique to ensure a coherent and clean build structure. This makes it easier to update libraries at a central point in the project or reconfigure build processes.

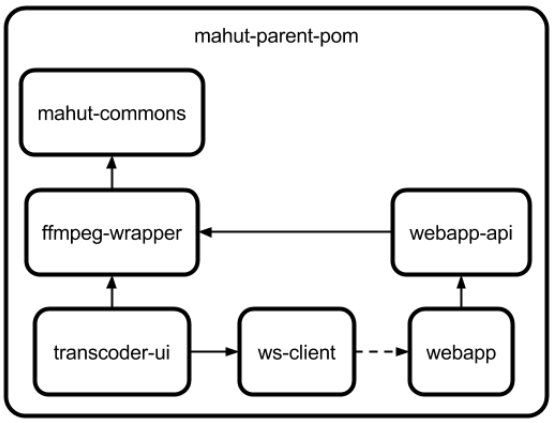


Figure 5.1: An overview over the whole mahut implementation. Each box shows a separate submodule. Dependent submodules are linked with directed edges. Dependencies are transitively resolved.

mahut-commons
This is a basic module which mainly defines common interfaces and contains often used classes such as a manager for persistent configuration files, encryption utilities, an email client or a factory for unique identifiers. This module is used in all other submodules except in the ws-client.

ffmpeg-wrapper
To be able to convert different video formats and for other video file based operations the system makes use of FFMPEG[1]. FFMPEG is a collection of various freely available programs and libraries for manipulating of video files. To make FFMPEG available within Java programs some JNA[2] or JNI[3] wrappers already exist which are able to directly call the FFMPEG code. Unfortunately at the time of the implementation all of these libraries were outdated, i.e. they were implemented for older versions of FFMPEG and therefore not usable within the scope of mahut. For that reason, the ffmpeg-wrapper module implements a custom FFMPEG wrapper for mahut. It calls the FFMPEG installation on the users premises via the Java Command Line Interface (CLI) with a "ProcessBuilder" and executes the needed functions. Output lines are then read back to get the textual output from FFMPEG in order to give feedback to the user about the progress of the executed operations. To keep the implementation modest only the calls needed within mahut are implemented with hardcoded call parameters. The implemented FFMPEG calls are the ones for the transcoding and the thumbnail generation of video files and the ffprobe metadata call. The module is used in the transcoder-ui module as well as

[1] https://www.ffmpeg.org
[2] https://github.com/twall/jna#readme
[3] https://docs.oracle.com/javase/8/docs/technotes/guides/jni/spec/jniTOC.html

for the transcoding of the video files to upload and is also needed to retrieve metadata-information from these files. The metadata is parsed from the command line and then stored as a JSON string attachment to the transcoded video. In the webapp-api this module is needed to generate thumbnails of the uploaded videos for the set and asset overviews.

transcoder-ui

The transcoder-ui module is the Graphical User Interface (GUI) for the file upload client. It can be either run directly on the Java Virtual Machine (JVM) or installed at the users computer. It is implemented with JavaFX and follows the MVC pattern. The UI is layouted with FXML and CSS and dedicated controllers for each UI component. During the development attention was directed particularly to usability, ease of use and simplicity of the client. Thus, the UI consists of only two views: the settings and the main view. All input data from the user is centrally stored in a data model class which also persists the input in a XML configuration file in the users home directory in the folder "/.mahut". Therefore, when the program is closed all values are saved for the next session which makes it possible for the user to fast continue the last work or reuse specific settings to accelerate future uploadings. The transcoder-ui module uses the ffmpeg-wrapper to process the materials and then uploads the coded video files along with the raw meta-content files and meta-information which is extracted from the materials to the server with the ws-client module.

webapp-api

The webapp-api module provides an interface to the database, the servers file system and to the ontology. It defines the database objects (so called Plain Old Java Objects - POJOs) and executes file based operations. It is the basis for the central cloud service and used by the webapp module. For this it provides a set of manager classes with specific functions for each database object, for the file system and for the ontology. It uses a persistent XML configuration file (located in the "conf" directory at the Servlet containers' installation) where the path to the data folder (where all assets, the database files and the ontology file is stored), the settings for the email client and the password for the database are configured. All database transactions as well as read and write operations in the file system are done through this module.

webapp

The webapp module provides the UI for the central server system. Here, stored assets can be viewed and administered and shared with third party user over the internet. It contains many subpages, such as for instance a page to search for assets or one to administer users and groups. Most pages are only available after a login where it is verified that one has the right permission to do so. To underline this separation, public pages are held in a dark color schema whereas private pages are in white. Again, a major goal of this site was to improve usability and the ease of use. The website is implemented using Responsive Web Design (RWD) so that it scales to all resolutions and screen sizes, also on mobile clients such as smartphones and tablets. Thus, it is not

necessary to implement a special mobile version of the website or a dedicated mobile app. Moreover, many enhancements are implemented to display each type of data in the best fitting way. To prevent users from accessing restricted resources the webapp has a built-in mechanism which redirects the users always to the last page shown if the resource is not available. Thus, it is also impossible for the user to get a HTTP 404 error. In addition to it, semantic or clean URLs are used to improve the accessibility of the website. Such URLs abstract from the actual internal structure and implementation details of a website and display only logical, and human readable URLs. For instance, instead of using /pages/assets.xhtml the webapp rewrites the URL to /assets. This improves also the readability of bookmarks. Moreover, if a page is relocated on the server the URL must not be changed. The webapp module also implements and hosts the Webservice implementation as the web interface for the machine to machine communication. This is used to automatically create sets on, upload assets to and retrieve specific information from the server. Since this project is still under development the module is adjusted to automatically send an email to a pre-configured email address if a user triggers an internal exception (this is done with a custom ExceptionHandler) to track possible bugs. Nevertheless, if an error occurs the user is redirected to an error page but will not see any further error messages or stack traces.

ws-client

The ws-client is the client for the Webservice implementation in the central cloud. It is used in the transcoder-ui module to retrieve information about sets for which the user does have write permissions (used in the set dropdown in the client UI), to create new sets in the central server's database and to upload assets to it. Consequently, it is the link between the local client and the cloud service and operates over the internet via the SOAP protocol. It uses the generated WSDL file from the webapp module to automatically generate most of the needed classes during the build process. Therefore, both modules are only weakly linked via the WSDL file but do not depend on each other. Furthermore, the ws-client is designed to be usable also for other projects, e.g. a mobile app, in order to also bind other platforms to the mahut system. Similar clients for other programming languages and computer systems can be generated with the WSDL file in a resemblant way.

5.2 Used technologies

For the implementation of the mahut system only the latest technologies and libraries available were used. Thus, the system uses Java 8 which was released just before the development began. Java 8 introduced many new features and improvements such as functional programming with lambda expressions, JavaFX included into the Standard Development Kit (SDK) or a completely revised Date and Time API which is based on Joda-Time. Hence, for the fat client JavaFX is used which replaces Swing as the new standard Java GUI widget toolkit. To implement the webapp JavaServer Faces (JSF) is used. Although JSF is said to be old and complex it still is the standard Java

Platform, Enterprise Edition (J2EE) web application technology and faster than many other, even newer web application frameworks such as Apache Wicket[4]. JSF also supports the MVC pattern and many readily available powerful component suites for it can be found. Using JSF also has the advantage that it integrates seamlessly into the Java EE stack and is runnable on all Java EE web containers such as Apache Tomcat, Glassfish or Jetty. Moreover, JSF also provides many higher security mechanism already out of the box which alleviates the need to explicitly implement them later. For instance, since the version 2.0 (with improvements in 2.2) JSF has got a directly built in protection against Cross-Site-Request-Forgery (CSRF) attacks. What was previously achieved by implementing complex HttpSession tokens for each site is now implicitly included in each <form> element. For this, JSF assigns each form element a hidden random token which is taken care of by JSF. Thus, the programmer does not need to be concerned about this any further.

To manage the dependencies and build processes I used the build automation tool Maven. Maven provides a comfortable way to manage the whole build lifecycle starting from the dependency management, build configuration up to the release of the generated software. It uses a flexible plugin-based architecture which enables automatic build processes for many files and types of archives - such as Java ARchive (JAR) or Java Web application ARchive (WAR) files. Maven projects are configured with a POM XML file in a pre-defined folder structure. POM files can also be used to define parent modules which then determine specific project settings used in all child modules. Using a parent POM also makes it possible to divide large projects into several submodules and configure and build them from a single point.

To keep the generated distribution archives as small as possible only a few external libraries are used. Whenever reasonable it was tried to re-implement as much as possible. All external libraries which are used within this project are listed in Table 5.1.

Additional plugin-libraries which are used during the build process are listed in Table 5.2.

5.3 Implementation details

In the following section selected components of the application are described in detail to further depict the implemented solution

IDs

Identifiers are a central part of the mahut system where - as with most database systems - all stored entries are identified with unique IDs as the primary keys. For that reason, the POJOs implement the IHasId Interface from the mahut-commons module which determines that each POJO instance does have an unique identifier which is used to retrieve the entry and is immutable throughout the lifetime of each object. Thus, each ID must be a globally unique value of the Java primitive type long (64bit signed

[4] c.f. `http://lu4242.blogspot.co.at/2012/05/understandingjsf-2-and-wicket.html`

library name	description
org.apache.commons:commons-collections4	extensions for Java collections
org.apache.commons:commons-lang3	helper utilities for the java.lang API
commons-io:commons-io	utilities for IO functionality
org.bouncycastle:bcprov-jdk15on	Java cryptography resources
com.google.code.findbugs:jsr305	Java static code analysis
com.sun.mail:javax.mail	Java mail API
com.sun.xml.ws:jaxws-rt	JAX-WS reference implementation
com.sun.faces:jsf-api	JavaServer Faces specification
com.sun.faces:jsf-impl	JavaServer Faces implementation
net.minidev:json-smart	high performance JSON processor
org.mapdb:mapdb	embedded Java database engine
javax.servlet:javax.servlet-api	Java Servlet API
junit:junit	framework for Java unit testing
net.sourceforge.owlapi:owlapi-distribution	Java API for processing OWL ontologies
org.swinglabs:pdf-renderer	Java PDF Renderer
org.primefaces:primefaces	UI component suite for JSF

Table 5.1: The table lists all third party libraries used in mahut. Transitive resolved third party libraries used by the libraries listed above are not displayed here.

plugin name	description
org.apache.maven.plugins:maven-compiler-plugin	Java compiler for Maven
org.apache.maven.plugins:maven-source-plugin	creates source file JARs
maven-source-plugin:maven-eclipse-plugin	generates the Eclipse IDE files
org.jvnet.jax-ws-commons:jaxws-maven-plugin	Maven adapter for JAX-WS
com.zenjava:javafx-maven-plugin	creates distribution bundles for JavaFX

Table 5.2: The table lists all third party plugin libraries which are used during the build process.

integer). Creating such a value is often done by increasing a sequence number in a single "AtomicLong" instance which can be backed up by a persistent file. In distributed systems such as mahut where each client can create its own IDs for the assets this method is harder to implement since the IDs then have to be centrally administered by the server instance. This can lead to performance issues when a great number of clients tries to process many assets at once. Other - naive - methods such as simply using Java timestamp epoch millis as an ID would lead to inconsistencies because such IDs can be assigned more than once in such distributed and also parallel processes.

Hence, an other, more sophisticated approach, to generate the needed IDs is used. For this, a modified version of Twitters snowflake ID generation algorithm[5] is implemented. Here, each ID is assembled from three different parts: a shortened timestamp, a machine id and a sequence number to handle ID collisions. The implementation is based on Predictors javasnowflake algorithm with can be found on Github[6]. In this implementation the 64bit long ID consists of a 41bit timestamp (decodes as 69 years), a 10bit machine ID (identifies 1024 different machines) and a 12 bit sequence number as shown in Figure 5.2. The individual parts of the ID are placed inside the long value with the shift operator («).

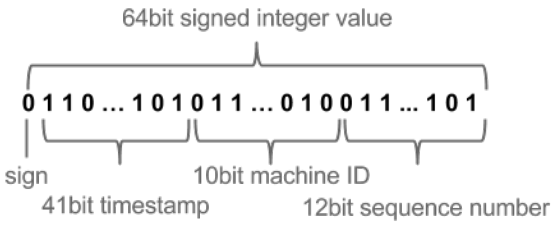


Figure 5.2: The structure of the IDs used within mahut. The 64bit signed long integer value consists of 41bits from a Java timestamp, a 10bit machine id and a 12bit sequence number.

The timestamp is implemented as a truncated Java epoch millis timestamp, the machine ID is generated from the machines primary network interface MAC address and the sequence number is a long value with a maximum value of 4096.

Using this technique has the advantages that it can be created on any machine but the Ids are still ensured to be globally unique. Using the first 41bits of the value as a timestamp the IDs can also be sorted by their creation date. With the machine ID attached it is also possible to identify the machine used to upload a certain resource.

The implementation of this algorithm can be found in the "IdFactory" class in the mahut-commons module. Such long IDs are often used throughout the system and therfore are often generated for multiple purposes. They are not only used to identify database entries but are also used for the Base64 encoded capability URLs which link to specific resources such as assets and sets but can be reset at will at any time.

[5] https://dev.twitter.com/overview/api/twitter-ids-json-and-snowflake
[6] https://blog.twitter.com/2010/announcing-snowflake

To make it possible for users to be identified with a human readable and memorizable ID, users are also identified by a username. This is only in addition to the numerical ID form the database and hence can be changed afterwards. For the username W3C advises to use an email address[7], because they are easy to remember and are ensured to be unique. However, for testing purposes a dedicated username is implemented which is easier to handle since it can be randomly assigned and is also shorter. For further uses of the system this mechanism can be easily changed in the code since email addresses are already registered in the database.

FFMPEG

In order to transcode videos from different sources which can have all kinds of different formats and codecs a powerful software is needed which is able to handle the needed codecs for encoding and decoding. For that and other purposes FFMPEG is widely used within the media industry where many standard video editing and playback software applications are based on it. Since it is shipped under an open source licence and can be installed on Windows, OS X and Linux operating systems it is an ideal choice to be used in mahut to process all kinds of video files. Mahut implements a custom FFMPEG wrapper which calls the installation via the CLI. Hence, FFMPEG must be installed at the cloud server and the users premises and be callable via the command line, i.e. an environment variable must be set to point to the FFMPEG installation. To enable all needed features of FFMPEG it must be configured and compiled as follows:

```
./configure --enable-gpl --enable-nonfree --enable-version3
--enable-libfaac --enable-libfdk-aac --enable-libmp3lame
--enable-libopus --enable-libtheora --enable-libvo-aacenc
--enable-libvorbis --enable-libvpx --enable-libx264
```

Compilation and installation guides for different operating systems can be found at the official FFMPEG Website[8].

Threads

To parallelly process tasks and speed up the execution of commands both the client and the webapp make use of various process threads. In the webapp the parallel processing of requests from different agents and clients is handled by the web application framework. Since the webapp uses a performance enhanced database (see the paragraph on MapDB below) resource intensive tasks are processed synchronously to the main thread and do not slow the execution of the website down. Operations, such as opening or deleting files, which are usually very time-consuming are executed without much delay and thus the response time for the users are low, i.e. below 0.1 seconds which is regarded as an instantaneous feedback[9] (see also performance discussion below).

On the contrary, the client makes time and resource expensive operations directly at the users premises. These operations include copying of files, transcoding of them and

[7] https://www.owasp.org/index.php/Authentication_Cheat_Sheet
[8] https://trac.ffmpeg.org/wiki/CompilationGuide
[9] http://www.nngroup.com/articles/response-times-3-important-limits

uploading of the transcoded files to the server. Since these operations strongly depend on the file system's access times, CPU intensive transcoding and network limited uploading times these processes are handled within dedicated threads to not block each other or the client's UI. The individual implementations of these threads can be found in the io package in the classes "FileCopyProgressThread", "TranscodeProgressThread" and "UploadProgressThread". Each of these threads is started individually and performs its specific actions separately. Thus, the client can start up to four parallel threads (including the main thread for the GUI). This is also necessary to give the user direct feedback about the programs status[10]. For that purpose, each of the three progress threads updates a progress bar to give direct feedback to the user as shown in Figure 5.3. Additionally, the overall progress is updated from the last thread to run (see explanation below).

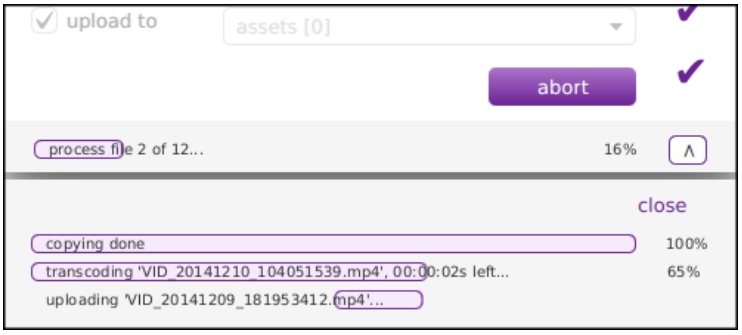


Figure 5.3: When the client executes the selected operations, i.e. copying and / or transcoding and uploading of the assets each thread updates a specific progressbar in a foldable window. The overall progress is shown in the footer of the main window.

Since copying and transcoding can be done at the same time, but uploading of a transcoded file only after the transcoding has finished techniques have to be implemented to ensure the right order of the processes. Accordingly, the copy thread runs completely decoupled from the other threads since no other thread depends on its outcomes. On the contrary, the transcoding thread has to notify the upload thread if an asset is transcoded and thus ready for uploading. For this, the transcode thread writes into a "BlockingQueue" which is then read from the upload thread when a transcode process has finished.

The overall progress which indicates how many files are already processed is updated either from the file copy thread if the upload is disabled or otherwise from the upload thread since this is always the last thread to run.

If the user aborts a running execution all threads are terminated or killed to instantly stop the client from performing the tasks. However, this does not revert previously made changes. Hence, already uploaded files might have to be deleted manually afterwards.

[10] http://asktog.com/atc/principles-of-interaction-design

MapDB

Deciding which database to use is a hard decision to make. Since it is estimated that about 80% of the websites page load time is spent in the backend[11], i.e. for database queries, using an SQL database would have been too slow for a system which is designed to be highly performant. Due to that, using a NoSQL database seemed to be the right decision. Many NoSQL databases are available as open source projects such as Accumulo, Berkeley DB, Cassandra, MongoDB, Neo4J, OrientDB and countless others. Many of them are implemented in Java or provide a Java API which makes them easy to use in Java applications.

Nevertheless, the learning curve of these databases can be quite steep since up to the time when this system was implemented theses databases do not provide any structured low-level query languages such as SQL or standardized interfaces. Using a database which is designed to handle huge datasets does also generate a bigger overhead which is not necessary when implementing a relatively small application such as the mahut system. Thus, other databases which are more tightly integrated into the Java API and do not cause much overhead were investigated.

The main idea behind such databases is to provide a framework which keeps all POJOs at any time in the main memory (RAM) and objects are therefore always in-process. This makes querying the dataset very fast in comparison to traditional SQL databases even when they are run in an in-memory mode. To permanently persist the entries such database engines usually serialize the POJOs directly onto the filesystem and load them again on startup. Such database engines for Java include Prevayler[12] and JDBM[13]. Since they are written in Java and directly run within the application they do not need any further Object-Relational Mapping (ORM) or Database Connectivity (JDBC) and queries in these engines is therefore up to 9000 times faster than in standard SQL databases [14]. Unfortunately since Prevayler is not able to grow beyond the provided RAM memory size because it assumes the "Prevalent Hypothesis", i.e. that at all times all objects can fit into the systems RAM and is not developed any more since 2013 it was decided to use the JDBM successor MapDB[15].

MapDB is written and maintained by the Czech software developer and amateur astronomer Jan Kotek[16] who also developed JDBM. MapDB runs completely in-memory but can also be extended to read from or write to a persistent file in the filesystem. MapDB is very fast in comparison to traditional SQL databases and according to Kotek is has a comparable performance to C-based database engines. MapDB is written in Java and extends the Java Collection API with persistent concurrent Maps, Sets or Queues. These collections are provided through the persistence engine and directly hold all POJOs. Since such structures are familiar to Java developers using them is naturally very easy. MapDB has also a limited setting capability which is used to configure the

[11] c.f. http://www.speedawarenessmonth.com/when-8020-becomes-2080
[12] http://prevayler.org/
[13] https://github.com/jankotek/JDBM3
[14] http://prevayler.org/http://int3.de/res/Prevayler/Prevayler_en.pdf
[15] http://www.mapdb.org
[16] http://www.kotek.net

database engine. Possible settings can include using a database encryption, enabling checksums and some additional performance options. MapDB does use write-ahead-logging (WAL[17]) and can be configured to be transaction save and is therefore safe to use in critical applications. The usage of the database engine can be found in the webapp-api module in the "DBConnector" class and in the various "Manger" classes for the POJOs.

stream servlet

Stream servlets are used to stream files from a server which are not located at the web containers root. Since in mahut all files are located at a pre-configured external folder in the file system this servlet is necessary to access all available resources. The stream servlet extends the JSF "HttpServlet" class and is registered in the web.xml of the web application. Hence, every time a request for a specific file is sent the request is evaluated within this class. This makes it also possible to check if a user does have the needed permissions to view (i.e. download) a specific file. If the user does not have the rights the request is blocked so that this file is secured and cannot be accessed. However, if the user does have the rights do so the file is streamed back in the HTTP response. This ensures the validity of the accesses and the data's security within the system. The stream servlet is implemented in the "StreamAssetServlet" class in the webapp module.

rewrite URL filter

The rewrite URL filter is an extension of the JSF Filter class and used to rewrite internal URLs to accessible human readable URLs, so called semantic or pretty URLs. This is done to get user friendly, bookmark-able and SEO optimized URLs throughout the application. It is registered in the web.xml and executed when the filter chain is processed during the JSF lifecycle. The implementation of the rewrite URL filter is found in the "RewriteUrlFilter" class in the webapp-module. It intercepts each HTTP Request and checks the validity of the requested URL. If the URL does not exist it redirects the user to the last page shown or if such a page is not present to the home page. Otherwise it checks if the URL is a registered semantic URL and in case of success let the filter chain continue to process the request in the downstream filters where the request is then processed and silently forward to the real page. This process is exemplified in Figure 5.4. Additionally, as a special case, if the requested URL is a valid internal URL it will send an redirect to the actual semantic URL where the request is then processed in the next cycle. This is used to hide the servers internal site structure.

Of course, third party libraries, such as PrettyFaces[18], can be found which are able to rewrite URLs to user friendly site paths. Since the implementation of mahut is designed to be simple and the resulting WAR compact such libraries are not included and hence the functionality is implemented by hand.

authentication filter

The authentication filter is an additional JSF Filter registered at the web.xml and called

[17] c.f. `https://www.sqlite.org/wal.html`

[18] `http://www.ocpsoft.org/prettyfaces`

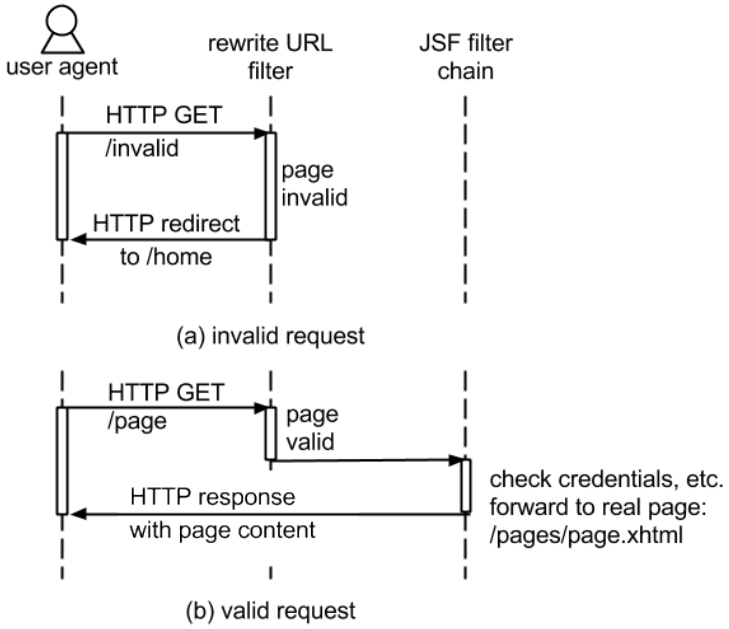


Figure 5.4: Two possible ways how a HTTP GET request is resolved. If (a) the URL is invalid, i.e. the requested page is not registered in the system a redirect to a valid page is sent. When the requested URL is valid (b) the rewrite URL filter will let the filter chain continue which then silently forwards to the real page.

after the rewrite URL filter. It intercepts every request and checks if the user does have permissions to access a requested path or not and blocks users from forbidden accesses. If a user does not have the rights to access a restricted path he or she is redirected to the last page he or she opened or to the home page if no such page is available. This method is implemented to enhance usability (it is almost impossible to get HTTP 403 or HTTP 404 errors on the website) but to also conceal protected paths from external users. Thus, this servlet saves each last opened page and works as a whitelist for all paths in the application, such as pages and resources. Only paths which are registered can be viewed by a user, and thus all other paths on the server are invariably blocked. Thus, using an authentication filter is an easy, yet effective way to enforce a system wide access restriction. It is implemented in the "AuthenticationFilter" class in the webapp module. Therefore, it was not needed to include a heavyweight third-party authentication framework such as Spring Security and the application could be kept light and simple.

Besides, if a user does not have permissions to view certain parts of a webpage the section can be shielded with the JSF "rendered" argument. For this, no extra page must be implemented and validated through the authentication filter.

encryption

Encryption is used on various places within the system. All passwords used in the

different parts of the application are encrypted to secure the user accounts and assure their integrity. Additionally, the whole database of the webapp is also encrypted by MapDB which uses a 265bit key XTea algorithm. The password for the database can be configured in the webapp's configuration file.

In mahut encryption is used in two ways: once as a cryptographic hash function for storing and verifying passwords in the webapp to authenticate users and once as a symmetric key algorithm to encrypt and decrypt persisted passwords from the configuration files. Here, the passwords are needed in plain text within the programs in order to make them usable for other services and therefore cannot be encrypted with a one way hash algorithm but must be also decryptable. Such passwords are needed in the client where it persists the password for the Webservice as well as the password for database in the webapp. Methods for both functions are found in the "EncryptionUtils" class in the mahut-commons module. This class provides methods to generate the salt needed for the cryptographic hash function and an encryption and an authentication method for it as well as encryption and decryption methods for the symmetric key algorithm.

The cryptographic hash function uses a PBKDF2WithHmacSHA512 (512bit) hash function with a random 16byte salt which is generated with an SHA1PRNG algorithm. It is configured to hash for 20.000 iterations to secure the hash function against brute force attacks. Of course, the salt has to be stored along with the user accounts in the database but since it is unique for every user a leaked salt is only problematic for one user and can be replaced fast.

Using the symmetric encryption is more difficult. Since a symmetric key algorithm always needs a password for the encryption and for the decryption it needs an additional password to cipher the passwords. Storing this password is problematic because it must also be secured against attacks - and therefore must be encrypted. Some solutions for this problem contain using a hardcoded password (which is useless and hard to replace once it is leaked) or storing the password at a secret place in the filesystem where it is secured by the operating system. This has got the advantage that every user has an own, unique password. Still, the password location can be guessed and exploited.

For that, I found another solution for the password problem which is based on an algorithm which is able to generate an unique but still reusable password on the fly. The used password for the encryption is generated from three distinct parameters which ensures that the risk of unwanted leaks is spread. First, it uses a general password which is hardcoded in the application. Since this is used for all applications (or at least for a specific series of them) it must be kept secret by the developer but can also be changed for every later release. This password is then concatenated with the MAC address of the machine to clearly identify the machine used. Since a certain machine can be used by more than one user the third part of the generated password is drawn from the logged in username. Decoding a stored password is only possible if the right person is logged in on the right computer and uses the right version of the client.

This generated password is then used to encrypt and decrypt the user's password. This is done with a symmetric AES algorithm. The cipher is then stored as a Base64 encoded string in the configuration file. To enable the configuration reader to automatically

detect an encrypted password the cipher is wrapped with the prefix "enc(" and postfix ")". Thus, when the configuration reader detects a string of the form "enc(ABC..123)" it automatically tries to decrypt it with the generated password. Unfortunately, this method has the drawback that these configuration files are not portable to any other machine or usable by other users. Still, if the password cannot be read any more it can always be re-entered by the user and saved as a new encrypted password.

login

To prevent brute force attacks each login attempt is stored in the database. The login controller saves the user ID (if an existing user was requested) and the IP address of the client in a special database list. This list is used both for the login at the webapp UI and at the Webservice. If the login is successful the entry gets deleted right away, otherwise an increased failed login attempt count will be saved along with the current timestamp. After a predefined number of failed login attempts (currently set to five) the user account and the client's IP address are suspended for 20 minutes from further login attempts. The user then has to wait until he or she can login to any account again. This is done to effectively prevent attackers from performing brute force attacks on the stored passwords and secure the application without the need to manually unlock disabled user accounts. Of course, other login attempt counts and timeouts can be defined to adapt the application to each companies demands.

user roles

Role based access control is a common technique in web applications to ensure that each user can only visit sites where her or she is explicitly authorized. To do so, certain levels of user roles are defined where each level or role represents certain rights within the application. Theoretically each user could have more than one role assigned but since a hierarchical role model is implemented where each level inherits all rights from the levels below each user of mahut can only have one role assigned. Possible roles are (from general to most specific): "public", "user" or "admin". Of course, other roles could be defined and added to the model to add further nuances to this model.

The roles are defined in a hierarchical enum ("ERole") in the webapp module. A hierarchical enum is an enum where each item also stores a reference to its parent. Thus, it is possible to traverse the so build hierarchical tree structure from the bottom to the top as shown in Figure 5.5.

It is also possible to build specific branches for special cases. Thus, the root node of the tree is the most general role, i.e. public, whereas each role is more specific the closer it is to the bottom. In the current implementation the admin role is the most specific one which also inherits all privileges from the above roles.

The roles are then used in the "EPage" enum which is used to register sites within the webapp. There, each site has a role assigned which defines the necessary basic role which is needed to view this site. Thus, if a site has the role public assigned it can be viewed by everyone, a site which is defined as user can only be view by logged in users and admins and an admin site only by an admin user. Thus, each user has a role assigned which is

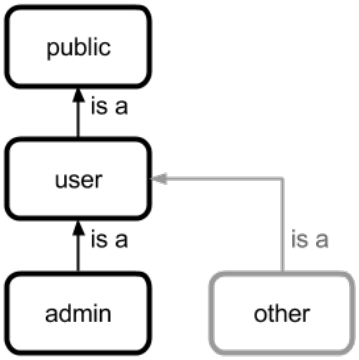


Figure 5.5: The structure of the hierarchical enum for the user roles. On top, the root node defines the "public" role, whereas at the bottom the "admin" role with the most rights is defined. Additionally, with this approach it is also possible to implement specific branches such as for separate user pages.

saved along with his or her data. After a successful login the users' role is loaded from the database and used to enforce the access control. Accordingly, registered users can only get the roles "user" or "admin" assigned. The "public" role is only used to define public sites and has no further relevance within the system. The roles are also used for an additional access control for specific areas of a site via the rendered attribute in JSF. For instance, a group of elements such as menu links to administration pages are only rendered if the user has the role admin.

groups

By using a supplementary group-based administration model a finer-graded access control for the management of sets and assets is implemented. In mahut, users can be grouped and receive read or write permissions for specific sets. This is done with the "Group" class which is persisted in the database. Each group instance contains two lists, one with the added users and one with the sets and the permission for it to map the assigned rights to the users.

In mahut there are only read and write permissions (in contrast to Unix style systems where also execute rights can be assigned) since the user can view or store assets but not execute any third party programs. If a group has write permissions for a set assigned it will automatically also be granted read permissions since it is assumed that the writer of a file can also read it. Thus, there are only three possible types of permissions for a set: none, only read or read and write.

Write permissions are necessary to write into a set, i.e. upload a new childset with assets. For this, the client fetches all sets where the user has write permissions at start-up so that the user can select a writeable set. When a new childset is uploaded it will automatically inherit all read permissions from its parent. Thus, the set can be viewed by all users who have read or write permission for the parent but initially nobody except

from the owner of the set is able to write into it. To that end, write permissions have to be set afterwards from an administrator.

Additionally, a set can also be configured as "private", "published" or "public" to share it with third party users or keep it secret. For these special cases the configuration tasks can be executed only by an administrator or by the user who uploaded the set, i.e. the owner of this set, in the webapp. All other set and asset related administration tasks can only be executed by an administrator on dedicated administration pages in the webapp. When a user or set is deleted it is also unassigned from the group to keep the data coherent.

settings

Most static variables such as usernames and passwords for services, email settings or the file storage path in server can be configured via a configuration file. For that reason, a XML configuration file for the client, located at the users home directory at ~/.mahut/properties.xml and one for the webapp, located at the Tomcats conf folder at $TOMCAT_HOME/conf/mahut.xml are used. Most settings are preconfigured in hardcoded values in the application itself. If the configuration file is not present at startup these values are then used to write a default configuration file where most settings are already present. This principle is called convention over configuration[19] and ensures that only absolutely essential options are needed to be set by the user and the system is runnable at startup without much preconfiguration needed. However, if the user wishes all settings can be overridden to configure individual parameters.

The settings file for the client is written every time the user enters some input. In order to do so, it always saves the last state of the program which enables the user to continue fast his or hers work in the next session - even if the application crashes. Only once at the initial startup of the client the application prompts the user to enter his or hers username and password for the webapp. All other necessary settings are preconfigured but can be changed at any time in the settings view or when data is entered into the input fields.

For the webapp the settings have to be changed manually on the server and are only loaded at the startup of the server. Since the settings file configures only the most basic system variables such as the file storage path or the email clients settings a restart of the Servlet container is in any way inevitable if these settings are changed.

webservice

A central functionality of mahut is its ability to exchange data between the external cloud application and the local client over the internet. For this, the webapp module provides a Webservice which is then used by the client to get information about the users write permissions and to upload assets. The Webservice is implemented with the XML based SOAP protocol to make it also usable for other services. It is created with JAX-WS specific annotations in the implementation class which are then read by Maven to generate the WSDL file automatically during the build process. This WSDL file (along with an XSD which describes the complex types used in the Webservice) is then used in the

[19] http://www.techopedia.com/definition/27478/convention-over-configuration

ws-client module to automatically generate the client implementation. The Webservice implementation can be found in the "IWSEndpoint" interface and "WSEndpointImpl" class and is available on the server under the URL path "/ws".

Each Webservice call is secured by a login mechanism to prevent unwanted accesses by attackers. Thus, the client must send a username and password in the HTTP header. The header is then read in the Webservice implementation on the server where the credentials are checked in the same way as for the login at the website. If the login is not successful the server will throw a "FailedLoginException" exception which is encoded in the server's response as a SOAP fault. Only when the login is successful the Webservice processes the request and sends a confirmation of type "boolean" or the requested data in the SOAP response.

Since the Webservice operates solely with XML files sent via the HTTP protocol it is not possible to directly upload large binary files such as videos and images within the SOAP message due to the restriction that XML cannot encode binary data. To circumvent this limitation two solutions are available.

One is to encode binary files as Base64 strings and include them directly in the XML or in the MIME part of the XML message. This approach has its drawbacks since Base64 encoding is time consuming and the encoded strings are so 1.37 times larger as the original files which causes longer upload times. Consequently, in 2005 the W3C endorsed the recommendation for the SOAP Message Transmission Optimization Mechanism (MTOM)[20] to enhance uploading of binary data in SOAP. MTOM is based with XML-binary Optimized Packaging (XOP) which enables XML to directly include binary data in special XOP Infoset containers. Hence no encoding or decoding is needed and the files are directly deserialized by the receiver. This process is much faster and also more memory efficient. MTOM can be enable in Java by using the according "@MTOM" annotation. Binary files are then wrapped with the "DataHandler" class which manages the transferring of the data and also improves the performance of the upload process.

functional programming

Functional programming is a programming paradigm which processes code as the evaluation of mathematical functions which do not induced state changes in contrast to imperative paradigms such as procedural or object-oriented programming languages. Functional programming is also free of side effects which makes it easier to predict the programs output and enables processing of collections in parallel. These functional calls are based on lambda expressions which describe how functions are evaluated. Normally, object-oriented languages do not support functional paradigms because here the program is understood as a set of interacting objects and not as a succession of stateless mathematical functions. However, functional programming has the advantage that the order of the execution of the commands is not important and therefore performance benefits of concurrent functions can be exploited and that the code can be more concise since the programmer only has to describe what he or she wants and not how to get it. Thus,

[20] http://www.w3.org/TR/soap12-mtom

many object oriented programming languages such as F# or Scala make use of functional structures.

In Java 8 functional programming was introduced which extends now Javas object-oriented programming paradigm with functional lambda expressions which are similar as know from Haskell.

Since mahut strongly depends on the processing of collections using functional programming paradigms is a convenient way to implement queries for the database. For this, it was possible to make use of Java 8 lambda expressions. Examples of functional calls can be found on various places in the code; mostly in the manager classes in the webapp-api module. An example of such a functional method is shown in Figure 5.6 where a functional call is used to verify if a user does have a write permission for a set.

```
public boolean isWrite(@Nullable final User aUser, @Nullable final Set aSet) {
    if (aUser != null && aSet != null)
        return m_aEntries.values().stream().filter(aGroup -> aGroup.contains(aUser)
                                                  && aGroup.contains(aSet)
                                                  && aGroup.getPermissionFor(aSet).isWrite())
                                          .findFirst().orElse(null) != null;

    return false;
}
```

Figure 5.6: The depicted method verifies if a user has got write permissions assigned for a given set. It streams all groups in the database and filters only those groups where the searched user and the searched set are contained and a write permission is set. Thus, it aborts the execution when the first group with such parameters is found since finding one group with the needed permissions is sufficient. Afterwards it returns the found group or null if no group fits the search criteria. The return value is then checked and evaluated to a boolean value.

ontology provider

In mahut an ontology is used to find related terms for certain search keywords. In the current version of the implementation only keywords which are taken from already existing hashtags can be searched. Hence, it is only possible to search for and find sets which are tagged with according hashtags whereas free text descriptions are ignored.

In order to find related terms a simple ontology manager (the "OntologyManager" class in the webapp-api module) is implemented. The class is based on the OWL API project and is able to search through an OWL ontology to find related terms. For this, it traverses the ontology tree to find the searched keyword (node or class in the tree) and if the keyword is present it loads all keywords which have the same parent as the search term, as shown in Figure 5.7. This collection of keywords is then filtered to keep only existing hashtags from the database so that for all displayed related keywords at least one set can indeed be found.

For test purposes an already existing ontology which describes animal species is used. Since the used ontology is not complete and because other, i.e. non-animal related keywords are not included not every possible hashtag can be found within this ontology.

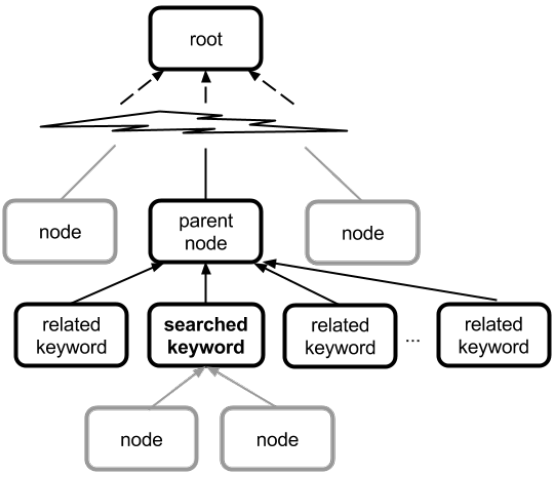


Figure 5.7: If a keyword is found within the ontology all other nodes on the same level which have the same parent are loaded (nodes with thick lines). The found keywords are then filtered to display only those of them for which corresponding hashtags can be found in the database.

Hence, this method does not always show related items even if there is a logical related hashtag in the database.

Since providing a complete ontology which covers every aspect of the world is certainly impossible the search will always be limited to the powerfulness of the provided ontology. Thus, it is necessary to find or create an ontology which fits the field of operation of the media company best beforehand. To cover also special cases, in future implementations of the system the ontology could be generated within the system on the fly by using semantic extraction technologies.

CSS

Since usability is a major concern for the proposed MAM system the website is designed to be fully responsive, i.e. so that the layout automatically detects and scales to all kinds of different device sizes. Several CSS frameworks, such as Bootstrap, exist which can be readily included to enable RWD. Since mahut does not implement much of the functionality provided by such frameworks own CSS classes are written to implement responsive features for the website. All of the layout relevant dimensions, such as height, size and width specifications are denoted in relative units, i.e. in percent or em. CSS3 also supports maximum and minimum properties which help to define flexible layouts. An additional major advantage of using CSS3 is its ability to discriminate between different screen sizes by implementing Media Queries. Furthermore CSS3 Flexbox layout can then be used to reorder elements on the page when the screen width is below a certain threshold. Using a combining of both techniques makes it possible for the site to change its layout and structure for each kind of different device. An example of how the layout scales to different screen sizes is depicted in Figure 5.8.

Figure 5.8: The website as shown on a Nexus 7 tablet. When the tablet is rotated to one side, the page is automatically re-rendered to fit the screen better. Here, the main components of the view are displayed beneath each other and the order of the elements is changed to better adjust the structure to the new width of the page.

Additionally, the website is designed to support multiple browsers and different version of them. For that reason, it is necessary to write CSS classes in a way which compensates for browser inconsistencies. Therefore, it was tried to reduce using CSS3 modules which are not yet (completely) supported by all browsers. Unfortunately this was not always possible such as with the Flexbox layout. Here, and for others, it was necessary to implement certain cross browser hacks such as the "-webkit-" prefix for Safari. Still, the website works and looks the same on most modern browsers. For this, it was also required to include Andrew Krespanis's CSS[21] hard reset in the implementation too. The CSS files can be found in the webapp folder of the webapp module.

performance

Besides usability another major goal was to keep the implementation as performant as possible. Since the website is designed to be used to distribute sizeable collection of large files across the internet loading such files via limited connections can take longer and thus the backend as well as the frontend code should not increase the execution time of the website any further.

Special attention was also paid to use only performant algorithms and functions right from the beginning. This includes basic principles such as performing inexpensive checks first to avoid expensive ones later on or to break from loops as soon as possible.

[21] `http://leftjustified.net/journal/2004/10/07/css-negotiation`

Furthermore, Java 8 also introduced many improvements to conventional methods such as a new and faster Base64 decoder[22]. The new functional paradigms for collections provide additional performance improvements and therefore using a database which is tightly integrated in the collections API proves to be a good decision. Further improvements were made by looking closely to the execution times of the implemented methods with VisualVM.

VisualVM is a freely available Java profiling tool from Oracle which gives detailed information about the execution time and memory usage of Java processes. Once a Java process is started VisualVM can measure the exact execution time of each method and give a detailed picture of the code's performance. Thus, it is possible to find bottlenecks (so called Hot Spots) and based on this information look closer into the implementation and hence revise the methods in question. This helped to find further weaknesses of the code. To give an example, it was possible to avoid the usage of frequent non performant regular expression checks and revise the database to use a more performant transaction model. Moreover, this also helped to discard slow implementations from third party libraries which were replace with faster ones wherever possible. So far, most of the execution time (up to 99%) is spent by the Servlet container - which is out of scope for the current implementation.

Nevertheless, some methods cannot not be further improved. An example is shown in Figure 5.9 which depicts the CPU time of a typical login process. For the measurements I used a 64bit Ubuntu 14.04 system with 4GB RAM and a 2.53GHz Quad-Core Intel i5 CPU.

As Figure 5.9 shows, the slowest methods are the "_streamFile" method in the "StreamAssetServlet" and the encrypt method for the password encryption and the "allFor" method from the "SetManger" which loads all sets of a user to show them on the start page.

The "_streamFile" method needs so much time since it strongly depends on the underlying internet connection used to stream files over the network and can only be speed up by using a faster connection. The encrypt method is slow on purpose since it hashes the password for 20.000 iterations which is used as an protection against brute force attacks. For that purpose, there do even hash algorithms exist which are designed to be extra slow[23]. However, an execution time of nearly 100ms is regarded to be in the right range. Since in this example it was the first call of the "SetManger" the "allFor" method has to load the set collection from the database and therefore takes more execution time (the database loads the individual tables lazily). Once the website is fully loaded and the thumbnail files are cached by the browser the website is much more performant. An example of the resource usage for general browsing the website is shown in Figure 5.10.

As exemplified in Figure 5.10 for most parts the website virtually does not cause any overhead. As indicated, VisualVM only measures methods which are called frequently such as the "Authentication-" and "RewriteUrlFilter" which intercepts each HTTP request and the "StreamAssetServlet" which is used to stream the files which are now cached

[22] http://java-performance.info/base64-encoding-and-decoding-performance

[23] https://crackstation.net/hashing-security.htm

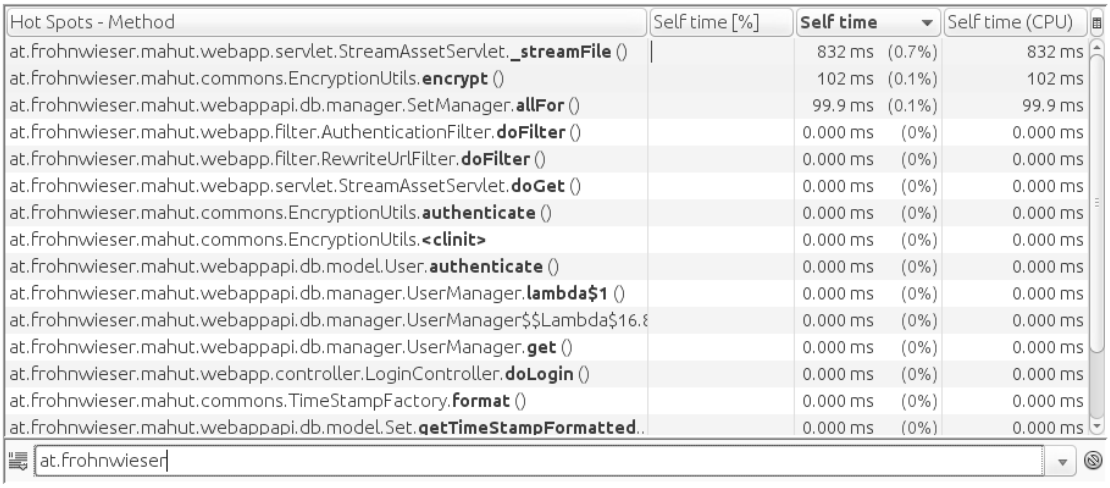


Figure 5.9: The execution time of a login process viewed with VisualVM. Only those methods were filtered which are implemented by the author. Additionally, VisualVM also filters only those methods which cause at least some measurable delay (despite most methods are reported to have an execution time of 0.000ms). The column "Self time" states the execution time of each method in milliseconds and indicates the share this method has of the whole execution time in percentage. This includes all executed methods including also the Servlet container and third party libraries.

by the user agent. Still, none of them causes any concerning delay. This is especially remarkable since for each operation a lot of different database queries have to be executed to check the user's permissions and to retrieve the sets and assets from the database tables. Performing typical administrative tasks is also nearly as fast, as is shown in Figure 5.11.

Again, as seen in Figure 5.11 the two filters cause some overhead but which is still not of much consequence for the overall execution time. Since now there are also write operations performed by the database these methods also appear on the list but do not cause a measurable delay.

As demonstrated above, the website is as highly performant as intended which is mostly due to the fast backend system and the thorough usage of up to date libraries and algorithms. Of course, when using faster hardware in the cloud system the application is further speed up.

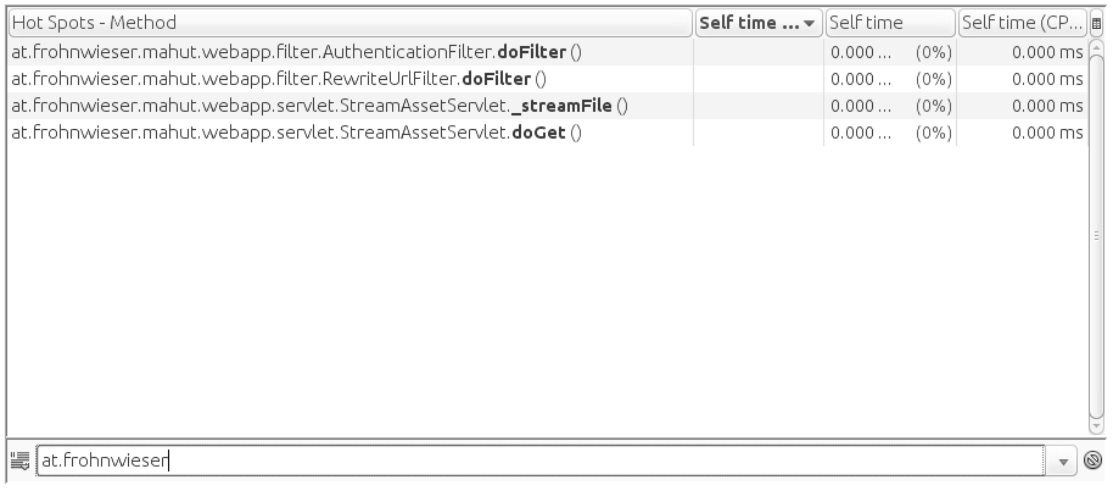


Figure 5.10: The execution time when browsing the website at random. Doing so, the database has to only execute read queries which are very fast.

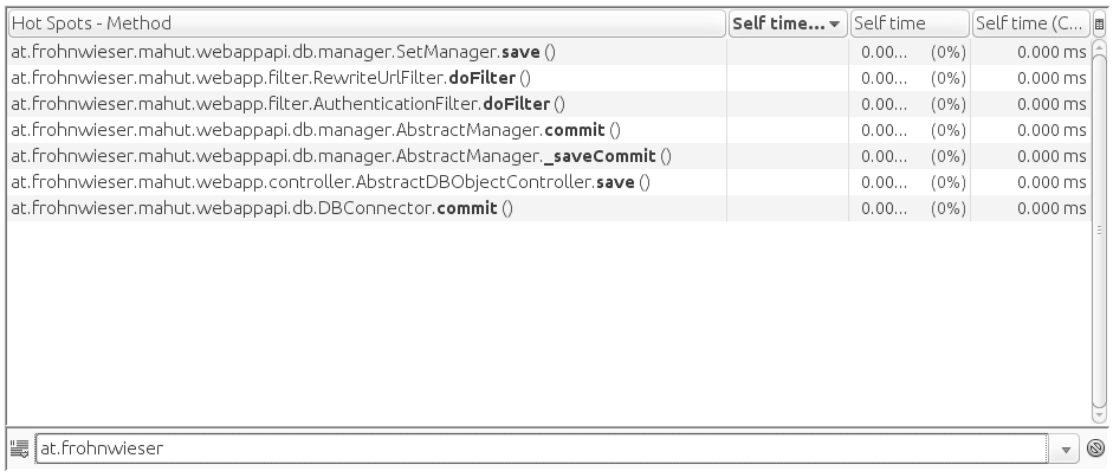


Figure 5.11: Administrative tasks in the set view visualised with VisualVM. Here the database also has to perform write queries which causes MapDB to serialize the whole dataset to the filesystem to persist the changes.

CHAPTER 6

User tests

After the prototype was implemented experts from the preliminary interviews and from the Vienna University of Technology which were willing to evaluate the application were interviewed. The user tests were carried out in February and March 2015. For the user tests, the prototype is set up on a public server in the internet and the client installed on a test computer. The system was then tested with the experts directly at their workplace. For that, test files were prepared and a short test session planned where the probands had to perform some of out of eight predefined representative tasks in the system. After a short introduction where the system was introduced and a quick overview over the systems' architecture given the subjects had to carry the tasks out. While the experts conducted these tasks they were encouraged to talk about their thoughts and ideas for improvements while rich notes were taken. To keep them continuously talking the interviewees were asked direct questions about the problems they experienced when using the software. This technique is called Thinking Aloud [LR08]. It is a powerful usability design tool which helps to recognize what users are thinking when they use an application. The techniques is also helpful to identify misconceptions of the users and to clarify why these misunderstandings happen[1]. Afterwards, the outcomes of this process were used to redesign the system in order to eliminate the problems found and to make further enhancements to the systems' usability.

Test I

The first participant started with the client to upload files. He quickly noticed that many input and control fields in the client lack of clear explanations. For instance, for him it is not clear what the fields "copy to" and "upload to" really mean, i.e. what the client does with the files when these checkboxes are checked. Furthermore it was also not clear to him how he can display the file lists (by clicking into one of the drop zone boxes) or the progress bars (when clicking the status bar on the bottom). Here, he also

[1] http://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool

wishes that the progress bars are always displayed and that there is an overall progress indicator which states how many of the files are already processed and how many are due. Other improvements he suggests include that he would like to see more feedback from the application, such as a popup with a warning message if a file cannot be accepted by the client as a material.

He also disliked the strong distinction between materials and meta-content. In his opinion pictures and documents can also be regarded as materials, since in his company they often use static images for their productions too. Hence, he would like to be able to also upload other types of files as primary material files too. Furthermore, he suggest to enable the uploading of audio files since in his company they also produce their own background music for the productions. For a musician, he thinks, such a system could also be valuable, especially when it can be integrated into the whole production workflow. Further on, he also misses a reset button after the upload has finished to faster continue with the next task. Nevertheless, he finds the separation between materials and meta-content useful and would keep it. The possibility to add textual descriptions and keywords with hashtags to the uploaded sets is viable for him too.

For the webapp he had further suggestions. First it was hard for him to comprehend the differences between materials and meta-content there, especially when there are also child sets displayed within the uploaded set. He clearly advises to make the distinctions between those segments more apparent. The assets within the materials or meta-content section should be at least sorted alphabetically by their name or in a chronological order and must be searchable.

He also would like to view more assets at once. The system should display all, or at least up to a reasonable number of, for instance 30, assets of each set in the view. Then he would be able to see all contents of a set, even if he has to scroll then more. For that, fold and expandable tables should be used to prevent a too long site and make the set easily overseeable.

He also indicated that the set hierarchy structure must be more visible. The bread-crumbs at the top should be revised to contain also neighbour sets and should be displayed more prominent. Additionally, in the asset view he would like to have the direct, i.e. download, link to the assets below the asset itself to make it more easily to find.

Overall, he thinks that the current prototype would not be a suitable archiving solution for him or his company. He especially misses the option to also search for hashtags of assets and not only for sets. Of course, this functionality is planned but was not implemented at the time of the user tests. He also suggests to make it possible in the search field to distinguish between assets and sets. According to him, currently there is too much focus on the sets and too less attention paid to the assets. To find specific resources he needs a better integration of the assets and hashtags into the system. He also confessed that the set hierarchy (folder structure) is yet not quite clear to him.

On the contrary he finds the possibility to share assets and sets with external clients via this platform helpful but said that he does not need such functionality much. Still the distinction between the share modes of "private", "published" and "public" is sound for him. According to him, such solutions as this one must also be understandable for

non-technical users as well. After the tests were completed he showed a similar archiving system which is tightly integrated into their work environment as an example of an already matured system.

Test II

After the first test was completed it was also possible to briefly query a colleague of the first subject. For this, only a shortened subset of the test cases was conducted which covered only the most important tasks.

The proband also has difficulties to identify certain input and control fields in the client, which were almost the same as for the first experimentee. He also had problems to comprehend the set and child set hierarchy of the uploads. Again, according to him, a more clear site structure is needed to depict the specific types (materials, meta-content and child sets) better. He also would like to have the possibility to directly upload assets into an existing set. On the positive side he finds the keywords ("hashtags") very useful and thinks that the application can be used in a real-world environment as an archiving system. He also reckons the possibility to share resources online with this system.

After the first two tests it was clear where the most critical parts of the client and the webapp were. For the upcoming test sessions the most necessary changes to make the application more usable were implemented. These changes were drawn from the suggestions of the first two subjects. Improvements included are an overall progress bar and better explanations of the input and control fields in the client. In the webapp, "sets" are renamed to "project" and "childsets" to "imports" to clarify their function better. Additionally, the set view in the webapp is revised, so that the difference between the materials, meta-content and child sets are more apparent. In this view, the sets and assets tables are now also sorted in a decreasing chronological order.

Additionally, also he test data was revise. Now it includes a more a clear project structure, so that the probands can orientate themselves more easily. This included now only one large project with about 15 childsets. The subjects then only have to upload their assets directly into this project set. The subjects are now alos provided with a camera with which they can make live videos and images during the test session and then upload the created assets to the central service in the cloud.

In the following weeks I had the chance to conduct further user tests. Two Phd students from the University of Technology were interviewed which both do not work with media productions but could provide as non-involved users useful suggestions to ease the systems complexity and improve the overall usability.

Test III

The first student started by searching through the provided project in the webapp. After a brief introduction he readily figured the systems characteristics out. He swiftly noticed the distinctions between the materials and meta-content as well as the functions of the child sets. He does not have problems with the set hierarchy structure and can orientate himself quickly. This is partly due to the improvements made in the set view but since

he also is an experienced computer scientist he possibly has advantages when coping with new technical systems. Only, he has got problems with the wording of some fields. It is not quite clear to him why a set can be regarded as a project or an import and thus the organization of the test project with its child sets was problematic for him at the beginning. These renamings were done after the first user tests were conducted and therefore were not tested at this time.

Yet, he also said that he understands the problems of the conflicting poles of having a flexible enough set structure and providing a simple project view for the users. Nevertheless, he can imagine that a professional user with more experience with similar systems than him might be faster to learn the systems structure. Hence, he thinks that the set structure is appropriate.

However, he suggested to depict the set structure more clear and less fuzzy. To make the set structure more apparent he suggests to move the title or name of the set to the top of the page and provide a tree view like in a folder tree of a file browser to enable the viewer to easily overview the set hierarchy and jump to other sets fast. This was already suggested in the previous tests by other users. Withal, he finds the breadcrumbs at the top of the page helpful and can use them easily.

At the beginning, he also thought that meta-content can only be textual content but not images or videos like in the current implementation. However, this was fast clarified. Other improvements such as the toggle-able meta-content section are very useful to him. Hence, he can use the system very effectively and was also very fond of the provided search mechanism. Nevertheless, he wishes to have finer graded search capabilities such as to limit the search to specific sets or to only search after certain file types or dates. Furthermore he misses also a full-text search and the not yet implemented hashtags for the assets.

After a quick introduction he could also use the client without problems. Nevertheless, he identified some usability problems and also readily posed solutions for them. First, he had problems to distinguish between the material and the meta-content section. Here he suggested to depose the meta-content section a little bit more or draw a border around it to clearly distinguish it from the other control fields. Additionally, he also remarked that the select buttons should be named differently such as "select materials" and "select meta-content". He was also the first one to point out that the drop-down list for the writeable projects is not a good solution since it is hard to use if it got many items in it. Unfortunately, he does not have an idea with what type of control it could be substituted. Even so he found the improved interface clear and could easily find the file list or expand the progress bars.

After an execution finishes he wishes to be better informed about it. This could happen in a popup which also displays a link to the just uploaded set and also a reset button to clear all input fields to comfortably reuse the client for the next upload. Overall he finds the system useful and could imagine that it can be used in a productive environment.

Test IV

The second proband from the University had at first slight problems to comprehend

the set hierarchy of the system. However, after a more exact explanation the structure was clear to him. Nevertheless, he would like to be able to change sets once they are created and upload additional assets to them too. The distinction between materials and meta-content was not quite clear to him but he said that he thinks that this is helpful in media production companies.

He especially likes the idea of using hashtags to organize the resources and clearly wants to be able to add hashtags for the assets too. Moreover, he also wants to have extended search functionalities. For one, he likes the possibility to concatenate search terms such as "elephant india" to search for an Indian elephant but he would also like to have more special search methods. For instance, if a user wants to search for a set with a certain hashtag and a specific creation date one should be able to enter: "elephant 18.03.2015". The search engine then should automatically detect and recognize the date and use it as a filter for date entries. According to him, this could also happen with special makers such as: "elephant date:18.03.2015". Similar search tokens could be defined for file sizes or video durations too. He also would like to have a finer graded search for specific sets or files but likes to have the search compact in one search field without any additional checkboxes or dropdowns.

When he tested the upload with the client he could find all the needed fields and controls fast. There he suggested to also better exemplify the possibility to enter hashtags in the description text for the meta-content. As he said, he would not have thought for a long time about entering hashtags there. Also, he mentioned the problem with the dropdown menu for the upload set selection if there are a lot of writeable sets contained. There, he came up with the idea to replace it with a search field where one can search after sets. Of course, this field should make suggestions with available sets, nevertheless he also ascertained that it is then harder for a user if he or she does not know or remember the name of a searched set. Again, he also finds it not clear enough to see when an execution has finished. As he said, he misses a clear indication when a change of state occurs and when the execution successfully finishes. Here, he would like to have a clear description in the footer of the main window or a popup with an optional reset button. The idea to include a link to the just uploaded set is also very appealing to him.

In general, he advises to lay more hindsight on fast workflows and on extended search functionalities but nevertheless finds the administrative features of the system very good. He would also use the system as a private asset store, for instance for travel videos and images. He also likes the possibility to easily share the assets with others via the "private", "publish" or "public" setting. Only for the asset sharing modes the term "publish" is not distinctive enough in comparison to "public" for him. Here, he suggest to either rename the setting mode to "accessible" or to "shared".

Test V

For the last user test a production assistant from a public broadcaster in Austria as interviewed. The interviewee works in a team with several other peoples for a weekly aired news magazine. For that, they often have to travel, mostly in Austria but also in Europe and sometimes worldwide. Since the participant was not interviewed before, this

user test turned out to be also like an expert interview. During the test he could tell a lot about his work field and the technologies used there.

First he started to explore the set view online. There, he could remarkably fast orientate himself since he uses a very similar, yet more powerful, system in his company for a year. He quickly grasped the set hierarchy which is analogous to so-called "baskets" in their system. Yet, he too wishes to have a more clearer overview over the set hierarchy and wishes to have it displayed similarly to a file tree in a file explorer. Then it would be clearer to him how he can go back and forth in the system from each node. Nevertheless, he could make use of the project and imports structure and could also easily understand the difference between materials and meta-content although for the latter he could not readily comprehend the meaning behind the word and thus questioned what the files in this list stand for. Hence, he advised to rename "meta-content" to "Zusätzliche Inhalte" ("additional contents") since in his company most people only speak German and would have problems with technical English terms otherwise. "One cannot expect that all people from the media business are technical experts," as he stated. Most of his colleagues are already over fifty years of age and thus sometimes have problems with new technical solutions. He also likes to view all files in the set view at once to get a better overview over the contained files and thus does not mind to scroll more.

Only for the landing page of the user accounts, i.e. the page shown when a user logs in, he would like to have a more detailed overview of the uploaded sets and assets. Here, all sets for which he does have read or write permissions should be shown in a hierarchical view, so that he can easily see all of his assets at once to get a fast entry point into the system. According to him, older sets which were not used for a long time should be shielded there to straighten the view up.

Surprisingly, he also reported that his companies' in-house storage system implements a very similar technique as the share modes in mahut. In their system they have a traffic light model with three colors: green, yellow and red, whereas green is similar to "public", yellow to "publish" (only entitled users can see and edit files) and red to "private" (only a file owner can see and edit files). Thus, he likes the share modes in the proposed system but again would translate them to German.

His companies' storage system is also able to share assets over the internet but since this function is very complicated to use (they have to coordinate with the technical team in order to get a clearance) he and his colleagues almost never use it although they could very much make use of such a functionality. Instead of this, they often share video files with USB sticks or sound files with CDs and bring raw files on XDCAM discs to the company. There, the material files (so-called "hi-res" files in comparison to the downscaled "lo-res" files) are then imported into their system by a specialised technical team. Before that, they relied on magnetic tapes only for which they have still a vast archive which is currently getting digitized. However, the once digitized files are searchable in the system, but since he does not known how to use the search properly he does not use it often. Mostly, he only lets all of his files display at once and then searches manually for a specific file. The system is also only accessible with a Microsoft Internet Explorer and thus the company has to heavily rely on products from Microsoft.

Additionally, each department of his company, and sometimes even single productions, have got their own server where every member has got his or her own folder to archive and share files. There, the file structure is only known to the owner of the folder and searching for specific files and sharing them is therefore restricted.

Hence, he likes the possibility to assign hashtags for the resources and the capability to search after them. Since they use a text-based search mechanism in their system too, he can image that using such a method to find files can be workable and fast. In his companies' system, the texts for files contain rich descriptions which are written by special archivists and hence are very detailed. Unfortunately, these archivists are more and more pared down and hence they need a new way to capture meaningful descriptions faster. Using simple hashtags could solve these problems. For him, finding and retrieving searched assets fast is one of the most important features.

In the system used in his company, once files are imported the owner gets the rights to view the file and is so able to work with it tapeless. Hence, if he wants to bring files to the company from far-away shooting locations fast he has to use expensive satellite connections for this. Since this is highly cost-intensive, especially for short-term scheduled transactions, his company is hesitant to use it. Thus, an easy and user-friendly way to share assets over the internet is very useful for him, even if the files then have to be downscaled which makes them not usable for the production any more. Even so, as he said, he usually only needs low-res files for his work since he uses them mostly to analyse interviews and to find the most viable parts in them. Only later on, the cutters need the materials to create the TV report. For this, he could also make use of only audio files for this but nevertheless wants to have the video files anyhow to get a better impression of the recorded scene. Additionally, his companies' program also offers means to view all clips of a disc at once in one long video sequence with all individual subparts merged. Accordingly, he likes to have a similar function in the tested system as well.

He also liked the client and can easily use it. He finds the added archiving ("copy to") function useful since they have a likewise function in their system, although there, the materials are imported by a member of the technical staff. Again, he would like to have the fields for "copy to" and "upload to" renamed to "archive" and "low-res to" to better fit them into their current naming and terms schema. He also wishes that these two functions are dependent so that the files are automatically archived and uploaded since he cannot use an uploaded file which is not also present in the archive. Like other interviewees before, he also advised to make the discrimination between the material and the meta-content control fields more apparent and for this, set the meta-content section more apart.

In summary, he finds the system very useful. He strongly likes the implemented features and can make use of them. Thus, he can imagine to use the implemented solution with some adaptations for professional purposes such as in his work.

CHAPTER 7

Critical reflection

With this thesis many things are accomplished. A novel solution for a MAM system is proposed and the subsequent conducted evaluation shows that this system is workable in a real world environment in media production companies. One of the reasons why this work is successful lies in the strict use of scientific research methods.

The outcomes of this thesis are based on systematic research methods which were used in an iterative process. This process is made out of three consecutive phases, namely a data collection, an implementation and an evaluation stage. The first two phases were then further divided into several substeps to obtain more detailed results. Hence, five different methods were used as shown in Figure 7.1. These steps include a literature research, expert interviews, modelling, prototyping and concluding user tests. These individual steps are described in detail in Chapter 3. While the work on the project still continues, additional subsequent steps can and will be added to carry on the work.

Beginning with the analysis of state-of-the-art techniques in Chapter 2 MAM systems were closely examined. Many of the in this thesis proposed concepts and solutions are based on the findings made there. Together with the outcomes from the expert interviews the collected literature was used to design the here presented novel solution for a MAM system in Chapter 4. The system is mainly based on Schneiders' [Sch11] cloud based hybrid approach in a SaaS solution but incorporates techniques from various other areas such as CSCW [BS91] concepts and knowledge based systems [SS09]. Consequently, the implementation (as described in Chapter 5) is based on the system model found there and uses state-of-the-art software engineering techniques as discussed during the data collection phase in Chapter 2. Afterwards, the application was tested with users from the media industry which' outcomes are summarized in Chapter 6. For this, the user evaluation technique Thinking Aloud [LR08] was used to gather qualitative data about the usability of the system. This helped to understand the users' behaviour and their conceptions about the application better.

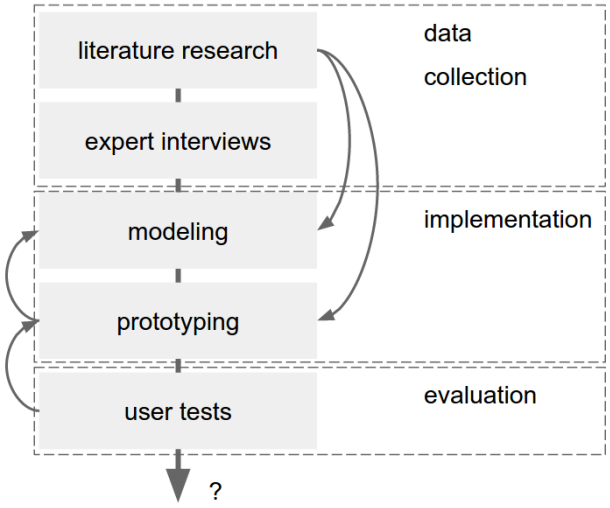


Figure 7.1: The research processes of this thesis was divided into three phases as shown in the dashed boxes. Some phases were then further divided into substeps. Each step was planned to be carried out in a strict order, however, some findings had impact on other phases, which were then incorporated into the step in question later on. Since the system is yet not complete, other, subsequent steps, can be added in the future too.

Thus, each of the five research steps was designed to be carried out in a strict order so that the findings of one step could be used as a basis for a succeeding step to build further investigations upon. Hence, each research step provided insights which were then verified in the next research step. Arising questions could so be clarified afterwards in the next step. However, several steps were interconnected since the results of some steps were used to further refine other research steps. For instance, the literature research was done continually, not only during the data collection but also in the implementation phase since some arising questions needed to be further researched and several assumptions verified. Also, the implementation was carried out in an agile process, where in each implementation cycle a preliminary modelling phase was conducted. Furthermore, as the user tests were carried out, some, especially urgent, design flaws needed to be corrected fast. For this, extra implementation work needed to be done during the evaluation phase. This helped to test the founded corrections shortly after during the subsequent user tests which brought further insights.

Many pending questions could be answered during this research process but some open issues still prevail. The system is yet not complete and many improvements can be made. It showed that inferences for the system design should also have been drawn from a broader range of fields, including e-learning and content-management systems. During the user tests, users often compared the system with what they already know and liked to see familiar approaches. Still a narrow focus on MAM systems was kept due to the

limited implementation and testing time. For future releases it is therefore planned to start with the user tests earlier and continue them in an iterative implementation cycle.

Before that, during the experts interviews, it also showed that many users do not use specialized tools but rely on standard tools which are readily available on their computers and smartphones. Such tools like Office programs, Evernote or WhatsApp can be used in a broad range of fields and hence the users often use them in private too. Whilst it showed that these tools are often capable of only limited functions for specific work tasks, on the contrary, the proposed system can maybe also be used in other fields too. These fields could include schools, museums, in-house document archiving systems for companies or even storage systems for private data collections. This is an interesting aspect which should be evaluated in further user studies. If the system is also usable in other areas the system architecture can be extended to expand its currently narrow focus on media production businesses to other areas. For this, it is also envisaged to port the client program to other platforms such as smartphones.

Designing the central server installation of the system to be runnable in a cloud computing infrastructure proved to be a good decision. As shown, by using cloud computing companies can save money [Dur10] while increasing the systems' reliability and performance [BYV+09]. Hence, the application can be used without much need for maintenance [Cos11] and flexibly react to unforeseen events, such as sudden increases in user traffic or rising demands for storage space.

Since the system is run in the public internet it can also be extended to be able to communicate with other instances of itself in order to exchange data between different tenants. Users from different companies could then share data easily in a standardized way as depicted in Figure 7.2. This data exchange can not only include digital assets but also information about the collected data or also knowledge about a certain domain.

This would help companies to easily sell their assets to other competitors but would also help them to buy needed production materials faster. When knowledge databases are shared, the information about the productions could be used in different contexts which would make the data more versatile. However, this is object for further inquiries and evaluations. Of course, the established security mechanisms then have to be re-evaluated and, if needed, enhanced for the new functions to secure not share-enabled assets and information from unwanted accesses.

While the current implementation (as described in this thesis) is only regarded as a prototype for the evaluation certain functionalities such as an automatic ontology creation were intentionally left out. Providing such a function seems to be indispensable for future applications since most of the existing ontologies found are either too complex - and therefore too large - or do not contain all of the needed terms. As shown, using knowledge based systems does have many benefits, also for small and medium-sized companies [ED13]. Building upon such intelligent indexing, cataloguing and search engines a structure to the data set is brought and semantic meaning to the stored information added [BLHL01]. This will lay the ground for the third generation of internet services, often referred to as the Web 3.0[1]. These promising technologies will help the

[1] `https://lifeboat.com/ex/web.3.0`

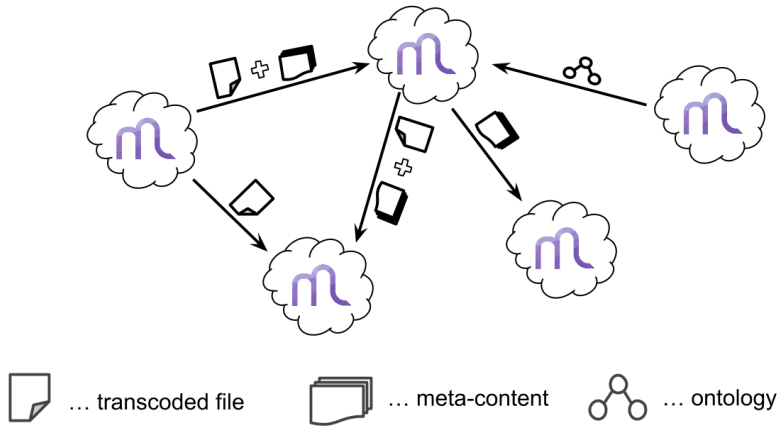


Figure 7.2: In upcoming releases the system could be used to communicate with other instances of itself. This would make the gathered data available for other companies. As depicted, these data exchanges need not be limited to contain only digital artefacts but could also include meta-content entries and knowledge about specific domains.

users to work with their data faster and also enables them to find entries only with vague search terms. Hence, it will ease the use of the system and increase the effectiveness of it. This work will be therefore a small part of this revolution.

Other simplifications of the prototype included using only basic security mechanisms such as leaving out time- and cost expensive configurations in the server like a SSL encrypted internet connection. Since security is certainly one of the most important aspects of an online service (c.f. [ID12]) in upcoming releases these parts must be added based on the current solution. For this, an overall security strategy must be elaborated and specific security evaluations, such as penetration tests, conducted. Because this part of the system is in the current implementation on many sides neglected, it is one of the most pressing requirements for future releases. Keeping the assets safe and secure is certainly one of the most important goals of this application. Hence, before this system can be used in a productive environment many security issues have the be addressed.

For productive uses it is also planned to provide the videos in more than one video format with different resolutions to make the video optimally accessible for all kinds of browsers and hardware devices. This is especially important since the number of internet-enabled devices is still increasing[2] and users more and more often use the internet with mobile devices which have now become the most frequently used internet clients[3]. As at the time of writing, users are spending more and more time on their smartphones

[2] https://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=5609

[3] http://searchenginewatch.com/sew/opinion/2353616/mobile-now-exceeds-pc-the-biggest-shift-since-the-internet-began

and tablets[4] and even Google now favours mobile friendly websites in their search results[5]. Hence, the usability - and with this the success - of a website more and more often depends on its optimization for mobile clients. This is mostly achieved in the current implementation by using up-to-date HTML5 and CSS3 techniques such as HTML5 videos and semantic elements and Responsive Web Design (RWD) for the websites layout. Consequently, the website is fluently scalable to also smaller screens and usable by a board range of devices which is another key factor for the success of the system. By using such an approach also additional programming efforts were spared since no dedicated mobile clients, such as for instance Android or iOS apps, had to be implemented. Of course, this concept will be kept and further enhanced in upcoming implementation phases.

Despite that, many additional technical improvements for upcoming implementation cycles were identified during the user tests. As the tests showed, useful enhancements include providing more functionality to administer and search resources but also to use assets in a different way. For instance, it should be possible to only search after assets alone or use date fields in the search query. As described in Chapter 6, these entries should be detected automatically or could be denoted with special markers. Further on, this mechanism can then be extended to search for other fields, such as file sizes or time intervals, as well.

Moreover, images could be used as materials or audio files included as assets too. A user suggested to limit the depth of the child sets for a given parent set. On one hand this would certainly decrease the systems' flexibility but would also help to reduce complex project structures. In that regard, it might also be suitable to let the users decided if they want to limit the depth of a set or not. Nevertheless, such changes have to be closely aligned to the users needs and therefore studied in upcoming inquiries. After this re-evaluated it can then be decided if a new functionality is usable in a real world environment and should be incorporated into the system or not.

Since performance is regarded as a critical usability success factor[6] one important goal was to keep response times to a minimum. This is also necessary to reduce cognitive stress of the users[7] and decrease waiting times and thus keep losses for the company due to slow computing low[8]. For this, the central server installation uses a performance enhanced NoSQL database in the backend. This proofed to be a good design decision, since, as the performance evaluation in Chapter 5 showed, the website does not cause much overhead and hence the searching and retrieving of assets is very fast and happens

[4] http://marketingland.com/nielsen-time-accessing-internet-smartphones-pcs-73683

[5] http://googlewebmastercentral.blogspot.co.uk/2015/02/finding-more-mobile-friendly-search.html

[6] http://www.nngroup.com/articles/response-times-3-important-limits/

[7] http://blog.trymyui.com/2015/03/new-horizons-bringing-ux-to-the-survey-design-field

[8] e.g. http://www.telegraph.co.uk/technology/news/10361881/Britons-lose-five-and-a-half-days-a-year-from-slow-computers.html

almost instantaneously, only limited by the underlying network connection used.

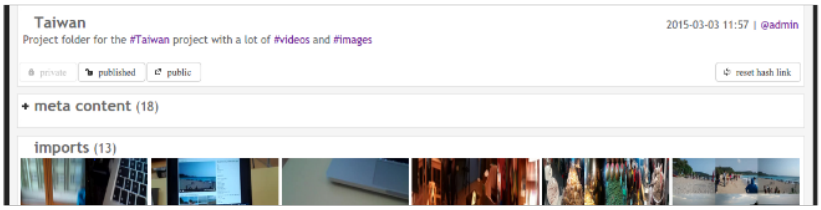
Furthermore, the client makes use of several threads to simultaneously execute tasks. This speeds up the upload and backup processes and thus decreases the waiting time for the users but also releases computing resources of the users' machine earlier. Thus, during the user tests some users were astonished how fast a recently made video was transcoded and uploaded to the external server and then available readily over the internet. This is also due to the preliminary transcoding and thereby downsizing of the videos in the client. This does indeed add additional processing costs beforehand but decreases the uploading times afterwards. In that regard, the uploading times are the more critical factor since transferring large, up to several GB of size, files over the internet is very slow, as exemplified in Chapter 2. However, since this two-step approach, i.e. the files are transcoded first and uploaded afterwards, can be executed parallelly this does not cause much delay because while one file is transcoded another file can be already uploaded at the same time.

Other performance related improvements included using performance optimised algorithms and libraries, such as for instance json-smart, which is the fastest JSON parser for Java available[9]. By using a Java profiling tool, as described in Chapter 5, many performance leaks were found and eliminated. Thus, the system is highly performant and can be used without much measurable delay.

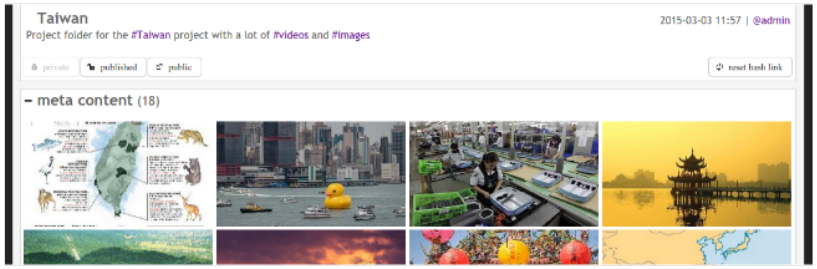
Still the main problems lie in the presentation of the underlying archiving concept. The prototype is implemented in regards of state-of-the-art techniques and makes use of up to date programming patterns and UI design principles. Nevertheless, it showed that a closer inclusion of the users right from the beginning would have helped to verify the proposed concepts and clarify misconceptions and poor designs right from the beginning. Input fields and controls in the client and the webapp were often too vaguely described and did not provide enough feedback for the users. More explicit annotations and color schemas would have helped to solve these confusions earlier on. Although the most pressing problems were corrected after the first user tests with Nielsen' [MN90] and Normans' [Nor02] design principles in mind these problems could have be eliminated beforehand. Other necessary improvements include that the client should also display more status messages during the user interactions and after the upload has finished.

Often, very simple design elements and already well known principles led to great results. For instance, since during the first two user tests the set view in the webapp was considered to be too confusing and too complex the meta-content list was redesigned to be toggleable, as shown in Figure 7.3. Now the meta-content list is by default hidden which helps to hide not particularly important segments of the website so that the user is not visually overwhelmed when viewing a set. This simple but yet powerful feature was one of the best adopted improvements during the subsequent user tests. A similar principle was already implemented in the client with the toggleable meta-content input fields.

[9] `https://code.google.com/p/json-smart/wiki/Benchmark`



(a) not toggled



(b) toggled

Figure 7.3: After the first two user tests a toggleable meta-content list was introduced. This simple technique helps the users to faster orientate themselves in the set view and is thus one of the most favoured improvements of the website. (a) shows the initial view of the meta-content list where the toggleable list is collapsed. In (b) the list is toggled and thus all contained elements are shown.

Another major improvement made after the first user tests concerned the processing status of the client. In the first tested version the status of each single execution process, i.e. copying, transcoding and uploading of assets, was only shown in a foldable popup at the bottom of the main window of the client. After the users requested to have an overall progress bar which is always present in the footer of the main window this feature was added in the next testing round. This request was harder to implement since every task runs in an own thread and thus the overall systems' status needs to be calculated in a complex way. However, the overall status bar, as shown in Figure 5.3, was quickly accepted and proved to be very viable. Nevertheless, if a user wants to, he or she can still fold the individual bars beneath the main window to see the exact status of each operation.

Still, while programming, keeping the users view in mind is not always easily possible for a technician but minding the users experience is the most important part for any technical solution. Hence, the client as well as the webapp should be once more revised in regards to established design principles to make the usage of them more transparent for the users. Additionally, it showed that functions should also be left out if they cannot be implemented fully with sufficient feedback so that they do not cause confusions. In the end, it proved to be true that the design should center around the user and not align to the technical requirements.

CHAPTER 8

Summary and future work

This thesis presents a novel solution for a MAM system which improves the functionality of current applications and solves pending issues of them. It compiles a comprehensive investigation and describes a novel system architecture of a MAM system in detail. For the investigation, a profound scientific research based on a preliminary literature review was conducted. Here, state-of-the-art designs for MAM systems and deepened information about the media industry and the herein used workflows, concepts and software applications were investigated. To get further qualified data seven experts from the media industry were interview which provided invaluable insights into their field of work. This showed how they work, what their greatest challenges are and what kinds of technical or also non-technical devices they use - or would like to use - to perform their everyday work tasks. Based on theses findings from the preliminary research phases and the conclusions drawn from the interviews a new system architecture for a MAM system is designed.

The proposed solution consists of a central MAM application in the public internet and individual local clients for each user installed at the users' own premises. This hybrid approach makes it possible to overcome the limitations of each technology used such as low network bandwidths or not scalable computing resources. It enables the users to share their work over the internet but also secures the stored assets in a performant and cost-efficient way. Furthermore, the client program can also be extended to be usable on mobile devices such as smartphones and tablets.

To test the assumptions made a prototype with many of the described features included was implemented and later on evaluated. The prototype is implemented in regard to state-of-the-art techniques and mechanisms and thus, all of the included features are fully functional and can be used in a real-world environment.

Afterwards subsequent user tests with five experts were carried out which helped to verify the proposed concepts and find weaknesses therein. Hereby, the current implementation was improved and new ideas for functionalities for upcoming releases generated.

The results show that there is still much work to do before the system can be used in a productive environment. Improvements have to be made especially regarding the usability of the system and to clarify certain misconceptions about the internal storage structure.

In future implementations of the work it is planned to further increase the ease of use of the application. Additional administration capabilities to control and edit the stored assets online will be added. Furthermore, the search engine will be improved to enable fine graded searches for all kinds of assets. During this process additional, not yet decided, features can be readily added to eliminate other weaknesses of the current implementation. It is also necessary to enhance the overall systems security further and port the client to other - for instance mobile - platforms.

The work must then be tested with more experts and users to ascertain the market maturity of the product. If the tests succeed the system can also be used in other fields of application, such as e-learning platforms or public media libraries. Hence the work on this project is yet not completed and will not end with this thesis.

CHAPTER 9

Appendix

To get deeper insights into the media production business six interviews with experts from the media industry were conducted. During the interviews rich notes were taken for further analysis from which these records were drawn. All interviews were conducted throughout the year 2014. In the following the interviews are summarized in detail.

Interview I
I started the interviews with a producer of a season of webisodes which was later on also broadcasted on a public television network. The interviewee was interviewed freely in accordance to his respective writing focus and lifestyle. For this the main focus lied especially on the factors that helped to make the project successful.

The initial webisodes started in November 2011. The serial consists of one season with ten episodes, each about 15 minutes long. At the beginning it was released exclusively on the projects website in predefined time slots during ten weeks on consecutive Tuesdays at 8:15 PM. The single episodes are hosted on YouTube and freely available.

The producer conducted the project with help of friends of his. The whole project was funded with their private money and all actors and project assistants worked for free which helped to keep costs to a minimum. This approach made it possible that no further advertisements on the website or in the series were needed to finance the project. The series were shot in a private flat during the summer months in 2011, whereas each episode was filmed in a single day.

The main idea of the project is to make a low-budget series for an internet-based audience. Therefore the whole series was designed to be broadcasted on the internet right from the beginning. Nevertheless, the concept for the series were driven by constraints such as time and budgeting. Furthermore, they also assumed that most people will not keep watching long episodes when sitting upright in front of a computer monitor. Thus, the script is short but creates a dense atmosphere with subtle dialogues. Since in most of the scenes only two people are present the narrative was considered most important.

The ideas for the dialogues were derived from everyday life events and discussions with friends.

To promote the series the producer gave several interviews to various austrian newspapers where the webisodes received positive reviews. Contacts to journalists were already established through friends of him. Otherwise it would have been hard or impossible to get reviews on major newspapers as the producer stated. In one interview with a newspaper the producer said that he wanted to arouse traditional television series with his concept. Shortly after that interview the public broadcaster drew its attention to the project and agreed to air the series. This coincidence was one of the main reasons which got the project on air.

For the television broadcasting the series was revised and each of two consecutive webisodes were cut to one with a total length of 25 minutes each. This was necessary in order to fit the available timeslots in the broadcasters program. As the producer stated, they did not plan to make any money with the project. Nevertheless, given only a little startup capital he was able to sell the transmission rights for five times more than they had invested. However, it is not planned to create a sequel for television since it would not fit the concept of the story. For the future the interviewee, who also wrote the script, writes on new series for other production companies. The low budget webisodes, the positive review on the newspapers and the public air-time helped to open that door for him. Earning money only with internet series is still not possible in small markets like Austria as he points out. Although there are crowd funding projects such as Kickstarter which are platforms to raise funds for new productions still only help already well know artists like Zach Braff. A newcomer like he was would not have been able to collect enough money for a full time movie there according to him.

Still, the screenwriter and producer is well aware that there were also other internet based series before that did not raise that much attention. In his opinion one major reason why he managed to be successful was that he was able to get a professional crew for his project. Contacts to other people made it possible to cast professional actors, some already well known in Austria and Germany, who volunteered for his project. The technical crew was also well experienced and helped to arrange a professional surrounding. Asked what he would improve for the next time he stated that he would release two webisodes a week (possible scheduled on Tuesday and Thursday) because people tend to lose their interest in a series fast especially when the duration of one webisode is only around 15 minutes.

Interview II

For the second interview a cutter from a relatively small film production company near Vienna was interviewed. There, they mainly produce documentaries for television, advertising clips or extended promotional films, e.g. for cities or private companies. They also have a digitised movie sample archive where they can store recordings and sell them via a web store. In order to do so, they store especially recordings of the Austrian nature which are often used for documentaries but are also viable for commercial adverts. This helps them to get an additional income but is also is one of their unique selling points in Austria.

My interviewee was one of their staff who is working as a cutter but sometimes also in the post-production as well as in the pre-production planning. Hence, he knew a lot about the whole production process. At the time of the interview he was working on a documentary about rural life and its special wine cellars in Lower Austria. While they are still shooting (they are filming throughout a year) he is already working in the post-production. Since they are filming on various places in the country this takes some further planning as he showed me. For the planning they are using a personalized Google Map where certain waypoints (filming spots or points of interest) together with additional information about the locations and contact persons are stored. Furthermore he managed to also include information about the light conditions (position of the sun, time of sunrise, etc.) and ways of filming schedules of chronologically dependent sequences. They also include pictures of the filming locations from preliminary photo shootings to check the on-site conditions. To store the geographically dependent information it is also necessary for the filming team to navigate and plan the shooting in advance. Hence, one of the biggest advantages of this approach is also that every member of the crew can check on information at every time when they need to with their mobile devices. Nevertheless, for special filming event such as, for instance, additional interviews, or for an external cameraman the camera crew still gets a daily call sheet with further information as a handout. Still, a lot of information is only verbally forwarded.

At the beginning of each production the information about the shooting locations is compiled during an inspection where the producers and the director go through each site, gather information, take pictures and look for good camera views. For this, they also use information folders which they typically delegate to some other company to write at the beginning to sell the production to the client. For the whole information gathering process they take a lot of time to get as much information as possible beforehand. At the end they aggregate this information in a document where they prioritize each entry. This is also helpful in order to have a backup plan in case something does not work out as planned. However, for documentaries like they were shooting at the time of the interview they do not create any other script or formal description of how the completed movie should look at the end. The outcome mostly depends on the appointed regiseur and cameraman.

When the filming is completed the cameraman hands his recordings to the interviewee along with information about the filming and the shooting location. This is done in a tapeless workflow. For each filming day he expects to have about three terabyte of data which is backuped on two disks and archived in a predefined folder structure for each project. Most of the work is then done in the post-production where they lay special attention to small details. For instance, they mask cars out and warp the picture to hide people going by. He also uses special 3D effects to bring static images to life. Most of the techniques he uses there are self-induced by him since he got no specific training in special effects during his studies. Altogether these works can take up to 180 hours per film. After the first cut they add the music which is composed by an in-house musician for each movie. When the movie is accepted by the client they get special narrators to read the texts which are again written by external specialists (in accordance to the

cutted scenes). Afterwards they finish the movie by making the final changes such as color corrections and the last fine cuts.

For the filming they also use special devices such as a Quatrocopter for flyover recordings. This relatively cheap technology enables them to film from various different angles which would be very expensive otherwise. This is also important because they lost some orders during the financial crisis in 2008 and now do not have the monetary possibilities to use expensive equipment. Still, since the professional equipment got cheaper during the last years they manage to uphold the very high quality level for which they are known for.

Thus, often other companies are willing to use their material for their own projects. Therefore, they installed a movie sample database which is hosted on an in-house server and accessible through the internet. Here other companies can select from a great range of film samples and watch them in a lower quality. If they plan to use one of these samples they can order a copy of the tape directly at the company. For this, they have a vast archive in the house with all physical tapes. The tapes are identified with codes which are also stored along with a short description (which is usually taken from the daily dispositions) of the scenes in a self-built archiving system. This is also very helpful for intern searches where the organization of the artifacts saves a lot of search time and prevents the company from unnecessary duplicate filming. Moreover, this system was also sold to other production companies before. Unfortunately, since the system is a collection of various different programs such as a transcoding software and a database backend and written in different coding languages it is no longer maintained since 2008. Due to this, it is only usable with great effort and often does not fit current needs. Archiving new films is therefore just done tentative because only one person in the company really knows how to operate the system. Although this system is seen as a valuable part of the companies successes there is currently no money to develop it further or to even create a new one to supersede the old one.

Interview III

Afterwards another expert from a medium sized production company in Vienna was interviewed. Her company produces various documentaries, television series as well as advertising films. There she is a project leader and at the time of the interview was currently working on the third season of a documentation series about an animal farm in Namibia. Each of the seasons consists of about 40 single episodes where each episodes has a length of exactly 48 minutes. The series is produced on behalf of a German public broadcaster and is aired therefore in Germany only but it also viewable in the online media library of the TV station. Since the production turned out to be quite successful they had at the time of the interview already 85 episodes aired and did plan to go on with a fourth season as well.

The main job of the interviewee is to plan and supervise the whole production process. Since the filming crew consists of about 35 people who are partially in Namibia and Austria, whereas their client is located in Germany this is difficult job. Mainly, she uses her email program which has a calendar tool integrated with which she can also

schedule work tasks. Most of the communication with her production crew, as well as with her partners from Germany is done therefore via email. As she pointed out this is also necessary because there is only a very bad and unreliable internet connection in Namibia which does not work with synchronous messaging programs.

For the filming and mastering each series takes about 14 months to produce. Since they already are filming the third season they have yet a well attuned team where she can leave most of the work unattended to the individual colleagues. To built up such a good team took a long time. At the beginning they had various problems which cost them more than two months of work. This was also due to cultural differences with the Namibian animal keepers and working with untrained animals in general. Nevertheless, now most of the work goes on without a need to intervene from her side. Only new workers may need to be introduced and cared of. Thus, she also does not need to flight to Namibia or Germany regularly but can do most of her work from Vienna. As she pointed out, still the personal relations between the members proves to be the hardest but also the most valuable part of the production. Unfitting members such as an inapt director can ruin everything.

The actual filming is done by four different teams where each team is filming for nine weeks in Namibia and then gets replaced by the next filming crew. Since the series is conducted as a documentary there is no script or general filming instruction. They are filming on their own whenever something interesting happens. The filming is relatively open and each new production keeps to be surprising. The only thing they got at hand is a detailed daily call sheet where the most important information is aggregated. Each day they have to create a report where they state what they have done, describe the stories behind it and then email the reports to her. After a short check she includes the report along with certain other information (such as the date, filmed protagonists, tape numbers, etc.) in an online spreadsheet on Google Drive. This report is also used as a basis for the text writers and the narrator. After their time is done, the filming teams return with the tapes to Austria and hand them to the cutters and the post production crew where the series gets cut and a first draft version is created. This version is then sent to Germany where the orderer can revise it and make requests to change certain parts of the film. After the changes are incorporated the episode is finished and mailed as a physical tape to Germany where they will air it. Since there are strict deadlines given by the Germans they are only able to change small parts of the episodes once the film has been cut. However, since they work together for almost three years they already know well what the others expect and can therefore align to it.

Data exchange over the internet is done via an in-house media server where all tapes are stored as digital copies in a lower quality. The German clients can then download the needed files via FTP where they got special system rights to do this. Any other form sharing such as via a One-Click-Hoster (OCH), did not work out as she said. Furthermore, she stores her personal project files in an order hierarchy on her laptop which is sporadically backed-up on an external HDD. Accordingly, she does not use any media management or archiving software. Once movie sample is backuped it is therefore hard to find. Often only a cutter or a long time serving mastermind knows where a

certain film sample might be stored and so things might get filmed twice as she admits.

One remarkable aspect of the production is that since the series is produced in Austria but aired in Germany they have to regard certain national differences although both countries are German speaking. For that reason, now all texts are written by Germans based on the journal entries provided by the filming crews and afterwards a German "native" speaker records them. As the interviewee pointed out, for the first season they tried to write the texts by themselves in Austria, but shortly after discovered that specific dialects and a slightly different wording makes it impossible for Austrian texts to be used in a German series.

Another interesting aspect of her way of working is that she hardly uses any social media platforms to connect to or communicate with other people. She only uses Facebook infrequently and never private, and did not even consider to use other platforms such as Xing or Skype. The only online tool she uses are Google Drive spreadsheets which are shared among less than ten people. This is mostly due to the fact, that she does not have any access to the in-house media server from home (she only comes to the office twice a week). She also uses only one laptop (which is remarkable in contrast to the vast device collections other interviewees have) and makes only strictly limited use of her mobile phone. Although she has to coordinate many people in three different countries she says that she does not have to be available all of the time. For the planning she does not use any wall calendar or flip charts which is often seen in other companies. But as she told us right at the beginning, in the future she considers to use a Lego wall where each worker in Austria such as the cutters, can state his or hers working task on it with a Lego brick. As she mentioned, she does not need any other planning tools or additional collaboration software.

The only thing she would like to have is an automated archive software where each movie sample is identified, tagged and store in a database with no further need to interfere. This could work like Facebooks facial recognition software. For the German clients she wishes to have an easier to use data exchange tool since most of them are technically unadept.

Interview IV

In the fourth interview the head of a small film and motion-design company in Vienna was interviewed. Approximately five to six people there in an open space office together with several other companies. These companies all work for the same Austrian department of an international print and online magazine. The magazine includes other companies too which are separated in three different locations in Vienna and employs about 80 people. The interviewee and his company mainly produce branded content and short documentaries for the magazine but furthermore also produce advertisements and visuals for events and various well-know national and international companies such as mobile phone companies or a brewery. His company is therefore funded by branded content only. Hence, their main scope lies on online based videos and animation motion design. Beside that he also works as a designer. One of his favourites works is the trophy for an annual music award ceremony which is a very well known in Austria.

In their main field, i.e. branded content for the online part of the magazine, they do everything beginning with creating concepts and project planning up to the producing and broadcasting on the magazines website. For this, they are able to use the Content Management System (CMS) of the magazine. Since the magazine is became an almost entire online medium only, the coordination and editing is done exclusively online. Ideas and topics are encountered directly by the editors and then are discussed with the executives of the six other companies which work for the magazine in Austria. In the topic selection they are entirely free since the international executive board of the magazine just want to see their business numbers, as the interviewee pointed out, but not deal with specific topics. According to him the business model of the magazine could be described as a franchising (they have to pay licenses fees) rather than a coherent media company. Hence, they are able to work completely autonomous from other national subsidiaries of the magazine. The international office also does not revise or edit their content or dictate them any directions. Although this leads to various advantages on their side the negative effect of this model is that they do not have much contact to other national branches of the magazines such as in Germany and do not have access to the rich monetary production possibilities of the magazine in America. This limits them to publish only local stories from Austria since they do not have the extensive funding as in America where they are able to produce stories and documentaries from all around the world. This leads also to the peculiar effect that stories about other (eastern) European countries are mostly produced by the Americans.

Nevertheless, once they have decided on a topic (mostly after several meetings) they commit the topic to an editor (a so called "Head") who is then responsible for the story. The Head is then completely free to finish the story however he or she wants. This can take from several hours up to several months of research according to the breadth and needed currentness of the story.

Videos and documentaries are often also released on their Vimeo channel where they pay for a professional account. They also use this channel to upload and show unfinished or work in progress parts of a film to their customers. The interviewee values most the user friendliness and adjustably of this platform, although he is not yet sure why they decided to use Vimeo instead of other competitive hosts. Special features like the possibility to exchange videos once they got uploaded may have contributed to this decision. Nevertheless, since the storage on this account is limited they delete videos once they get outdated - which is usually often not until they need more file storage space. Nonetheless, the raw files are always only used on the cutting machines and never sent elsewhere.

Once the story is finished they load all their presorted data to a house intern RAID system. While they are usually producing only low-fi content for the magazine full documentaries created by them do often not contain more than one TB of data although nothing gets delete as the interviewee pointed out. Bigger productions such as visuals for commercial companies usually consist of two to three TB of data. Hence, they do not need an extensive storage system to backup their files. To organize the system they use a special folder structure where every project is stored in a folder beginning with

the name of the customer. Once a story or project is finished after half a year it will be backuped to an other offline RAID system to save space in the work RAID. Since this is their only archiving system they are not able to search actively for a specific file or sample. Thus, the success of an individual search partly depends on if someone knows where the file is stored. This is also a problem because they are not able to search for a specific topic or person.

Altogether they work in a very free and relatively unbureaucratic way. Producing a two-parted 30 minute documentary takes for them only about two months. If they have to air a documentary shortly after a certain event they are also able to work ahead and then merge the actual footage together in only a few days. To keep their viewers coming again they often cut a documentation of a specific topic into more parts and release single parts of it within a few days on undefined time slots.

Managing these production processes is mostly done with shared Google Docs and via email. They write most documents in their meetings and often use them over several months. To keep this process coordinated only one person is determined to be the chief of the document which is often the Head of the reportage. For further planning they use calendars, spreadsheets, file hosting services such as Dropbox, Google Drive or WeTransfer and a coordinating program which was programmed by an external company. Although the international office of the magazine provides them with a sophisticated CSCW system, they do not use it at the moment for any planning, but only for bookkeeping since they are obliged to use it. Although they use social media platforms such as Facebook or WhatsApp for communication they do not really depend on it but instead use it as an alternative communication channel to email or SMS.

Asked about what kind of software he would like to use in the future the interview partner pointed out that they could really make use of a program which combines all products they use into a single program or an alternative archiving system. This would be especially useful for them because they often have parallel structures and duplicate files. According to him such a software should work automatically in the background without the need for any further editing or tagging. Since they already often write rich notes to each recording before the cutting they could provide these texts for the archiving as well. Based on this previously created notes an effective search in the archiving system should be possible. Another thing he misses is a usable process managing tool which is based on the interaction between people and not on the work tasks which need to be accomplished. Since this would be more related to his practical experiences this would help him to introduce a structure to the whole production of a film and keep the processes which take place there manageable.

Interview V

In the next interview an independent film-maker who was at the time of the interview in the process of creating a professional fantasy series which takes place in an apocalyptic post-war time was queried. Since up to today there was no such project in Austria the creator wanted to start a production by his own. For this, the interviewee wants to have as much liberties as possible and therefore does not want to work with a big television or

production company. As he pointed out there is also no actual support for such projects from the national television broadcaster or a possibility to get state funding. Thus, he and his team plan to host the series in the web exclusively and then release it afterwards on BlueRay. For this, they plan to shoot six episodes with ten to 15 minutes length each. The whole first season should then be available for free on their website or on other videos hosting platforms such as YouTube. If the first season succeeds they will make a second season which must be paid then.

One of the most remarkable aspects of his project is that they plan to shoot the whole series in English and just maybe want to make a German synchronised version afterwards. The whole script is written in English and only English native speakers are casted for the roles. Although the series got a German title he does not want to have any reference to Austria or Vienna in particular in the series. This is done to also reach international viewers more easily and be able broadcast it worldwide.

Since this series is produced only by a small number of private people the financing is crucial. As the interviewee told us they plan to invest about EUR 200.000 for the first season. In order to work with such a small budget they rely on voluntary workers and are only able to pay small wages for certain workers. Since most of the members already work in the movie business they already got most of the equipment such as cameras and a post production studio at hand. Nevertheless, if they are able to make money he will pay better wages, because the interviewee does not want to exploit his workers who often do not work full time for the series but rely on other incomes too. Since up to today they got no real sponsoring they also plan to draw from multiple revenue streams. They plan not only to bind sponsors and sell merchandising but they also found other, more creative, ways to get the needed production goods. For instance, they are able to use the filming location - an old industrial site - in exchange for filming for the owner for free. It is also being considered to shoot advertising on the film set during breaks to get more financing and attract further attention to the project. These countertrades are only one part of the financing model. For the merchandising they already planned a great range of different articles beginning with simple movie posters up to action figures and other film-related specials.

He and his team began with their work for the series about four years ago and to get started they already produced a short teaser which was shot in November 2013 and at the time of the interview was nearly finished. The postproduction of this took longer because the trailer contains massive special effects which is also uncommon for most Austrian productions. It was released at the end of April 2014 during a trailer release party which should help them to get the financing ensured and should also draw attention to the project. The series will then be filmed in the autumn 2014 and finished and released in 2015.

Although the main idea came from another person (a friend of the interviewee) they both work together with equal rights. They got help of another female who works as a production assistant. The whole project is coordinated via a Facebook group. This makes the whole planning process very transparent for the single members. The group also helps to advertise the project and to find voluntary workers. As he said the feedback was

overwhelming. A few dozen people from all different kinds of disciplines such as actors, makeup artists or music composers readily agreed on their own to take part and there are still new people continuing to come. So it was easy for them to find the best crew during five consecutive castings. They could also draw from other Facebook groups such as Cosplay fans and filming enthusiasts and connect to other communities. The producer is 24 hours, seven days a week online as he stated. One of the biggest advantages of this is for him that almost all people are nowadays on Facebook (or join the network for that purpose) and can connect to the network via mobile apps and therefore are always reachable. The ability to send messages which will pop-up on the phones of the others immediately is also a crucial aspect which he finds most important.

The group also uses Google Drive to share files but most of the time relies on emails and other lo-fi technologies. Usually they simply only use USB sticks to share files among them. However, since most of the team members are relatively young (20 - 30 years of age) they do not have any problems with new technologies. Still, further tools such as a collaboration software or other web-based communication platforms are not used since they are just in an early stage of their filming and 90% of the coordination work is done by only two people. Nevertheless, if the workload increases they will maybe use one but do not have yet decided to do so or which one to use.

The filming crew consists of about 30 members and almost all of them have a professional background. As the interviewee continued to point out, the reason why these people want to work with them is that they are bored to make only advertisements, work for limited theatres or other commercial projects where they do not have much personal interest in it. The scene is dried-out and desperate for new projects as he mentioned. These people want to show what they are capable of and therefore try to achieve the best they can. They also built the film set and created the characters, costumes and items such as weapons on their own and plan to use a lot of advanced special effects. Generally they pay great attention on little details. For instance, since the series takes place in a post war time, they do not use any form of synthetic material or modern clothing with zippers. This whole project is therefore considered as more than just a hobby of a few enthusiasts. In order to find them the interviewee made strongly use of modern communication platforms in the internet.

Interview VI

I also interviewed the assistant of a movie production. His job is to manage all workflows of the production directly at the set. His main tasks are therefore to plan, check and enforce the daily dispositions. He is responsible that all needed persons, goods and technical equipment are present. Furthermore he also has to push through the time plan and is liable to meet the days schedule. Thus, he is often one of the first ones on the set where he builds up the power supply, installs tents for the makeup artists and catering booths and organizes all other needed infrastructure before anyone can start to shoot. Throughout the day he then manages the arrivals of the actors and other personal or goods like rented cars and or special equipment. He has to ensure that everything is in place when it is needed. Thus, he also has to be prepared to quickly react to unforeseen

events, like changes in the weather or failures of team members. Additionally, he also has to manage other tasks such as talking with the local police or farmers to get shooting permissions or needed equipment.

He said that the loves the flat hierarchies on the set but to keep a strict emphasis on the time plan is also required because actors get paid for the whole day they stay on the set, regardless if they actually shoot or not, as soon as they are in the mask. Since they can be quite expansive he has to plan carefully not to send an actor to the makeup before he or she is needed, but soon enough to be ready in time. Today it is more and more important to keep the movies budget and pay attention to monetary constraints. Mostly everybody is stressed during the filming weeks. After the filming is done he then has to disassemble everything again and plan the schedule for the next day. Thus, he has to stay up all day and nearly all night to keep everything running but must be aware to not forget anything. This very demanding job is also one reason why he only works about six months a year.

His main and often only artefact to work with is his daily call sheet or disposition. This is usually an A4 sheet, printed on both sides, which contains all the necessary information for current day like the exact schedule, the scenes to shoot and additional names and telephone numbers of actors, the filming crew, locals, makeup artists, drivers and so on. If needed, additional information, such as maps or contracts can be included on further sites. Nevertheless, he does not use any other devices such as a laptop or tablet and uses his cellphone only to write SMSs to organize people. He also does not use Facebook and uses WhatsApp only as a convenient way to send pictures to other people.

At the time of the interview his current project was a TV movie which is shot in a small village on the country site. For such a 90 minute movie like this he plans to shoot about 22 days. Although they made a major plan beforehand when to shoot what at the very beginning of the filming, the dispositions for the next day are always made in the last hour before the set is closed in order to be able to flexibly incorporate new information but also legal facts such as mandatory periods of rest. If the directors assistant accepts the dispo for tomorrow he hands out a copy to everyone on the set or sends them via email. One reason to keep such a close schedule is that todays directors and filming crews are becoming more and more careless as he states. But it is also necessary because due to possible longer periods of bad weather or missed time plans they sometimes have to reschedule everything or even rewrite the screenplay. Consequently, they also have some plan B "bad weather dispos" ready.

The dispos are written by an assistant in a spreadsheet program. The basic template for it had not change for six years, hence they neither need nor use up-to-date equipment to create them. He himself does not use any special program but could tell us what he would like to use in the future if he had one wish for a perfect production assistance program. In his eyes the perfect program would support human information workflows throughout the whole production. This program should be able to generate dispositions with only a few clicks (maybe drag and drop individual scenes) based on preliminary created database entries. These should include all needed information about the actors, the director, all assistants and workers, the shooting sites, external supplier and many

more. Since all of these information is already entered into a spreadsheet he thinks it would be no problem to migrate it into a database.

After the dispo is generated the program should print it out on a paper sheet like he is used to work with. He does not need a mobile access to the program via a smartphone App or an other internet based service.

While he does not make much use of modern communication technologies he still thinks that in the future more and more traditional TV shows, especially documentaries, will be broadcasted via online streaming platforms where they are always available in on-demand and prepaid services. Additionally, since the average viewer of traditional public broadcasters is quite old (about 60 years and above) he thinks that shift will happen within the next few years. He thinks that this will only be interesting for younger filmmakers because today the public broadcasters only shoot movies if they have them financed beforehand where they can expect a good monetary outcome. New ways of thinking must be pursued here as he stated. Nevertheless, he also emphasises that still a minimum level of quality must be kept and in contrast to most of nowadays small productions the actors and filming crew must be well paid for their work in order to not exploit younger and inexperienced filmmakers. He thinks there will still be a use for people like him who manage the workflows behind the sets and create dispositions . Even major Hollywood productions such as Mission Impossible only shoot movies like they do, as he pointed out. Only the biggest part of their budget is spent on famous actors and in the postproduction for excessive special effects.

Interview VII

Finally a production manager who works for numerous TV shows and documentaries about nature in Austria was interviewed. The interviewee is responsible for the whole organization and administration of the production from the very beginning up to the completed film. He manages the filming crews, gets the needed shooting permissions and keeps an eye on the finances. This is all done in accordance with the principals, directors and various other stakeholders of the individual projects. Hence, he has to manage and coordinate a lot of people and functions as the central contact person for all of them.

Usually he works for two to five projects at once whereas each single project may need a very different preparation and filming time. Normally this is needs one year of preparation and then about two years of shooting (to get, for instance, two winters of shooting) time. Of course, they do not shoot at every day during this time but often only shoot at a few days in each season. This is done according to a script which is written at the beginning of the project and then used as a guideline throughout the production. In this script they have outlined the whole idea behind the film and written all scenes in detail. When he gets the finished script he then can begin his work to plan each day of shooting and book the filming teams, equipment and all other things necessary. Since the script can, and often will, change due to unforeseen problems with, for instance, the weather, needed permissions or availability of team members they often have to rewrite the script and keep a plan B ready. Normally, he also does not plan specific days in advance, but tells his filming crew to keep themselves ready for one or more weeks where

they must be available to shoot. This is required since he has to plan the shooting months in advance where he cannot anticipate all circumstances at the day of shooting.

The filming crews are always teams from external companies. To simplify his tasks he often only decides on a certain director which in the following then nominates his filming crew. Once these teams got their instructions set up and their full equipment they can last for days on their own and shoot in the wilderness where they stay in mountain cabins or tents. Needless to say that these teams also need unusual skills and equipment such as trainings in mountain climbing or avalanche transceivers.

Over the years he got to know most of the filming crews better and knows all their personal strengths and weaknesses. This enables him to easily compile the right group for each project. He does not have to intervene into the filming often and usually only gets the records at the end of the filming period or additionally certain pictures during the shootings. Thus, he ordinarily manages the project from his office in Vienna and does not show up at the filming sites.

After the shooting is done the tapes are transferred to the post-production where they are archived and the best sequences are cut. This can also happen during longer shooting breaks and they do not need much planning for this. Afterwards, the director or the cutters decide which scenes to take according to the script. Following that, they present the raw version to their clients and decide on further details.

For his work he does not use any special software but relies on documents, spreadsheets, email and his telephone alone. Sometimes he uses Dropbox to exchange photographs. He can also make use of a company intern FTP server or Vimeo to upload and show video sequences to his partners. The finished records from the shootings are transferred back to their company always after the shooting where they are then archived in their own storage system. They have a separate technical crew consisting of about five people which is the heart of the company as he calls them. Here they store all the productions of the company and provide accesses to archived video material. If he needs some historic material from them they are able to search it depending on the production name, director or tape number. For time critical productions they are also available on weekends but usually are able to prepare certain things in advance.

Other things he relies on are the dispositions where all required information for the filming team are written. He writes them in Word and sends them via email to the technical department where they print it out and book all the needed equipment. For the filming the director has everything he needs "in his mind" where the interviewee can trust him to do the right things.

In addition, the company bought the planning software Farmerswife to manage their projects and resources. This software also includes a central booking system. He does not make much use of this program but likes to see the progress of a project and get a picture over the work processes. For instance, he can always check back if a certain cutting station is already booked or if a needed resource is available. Since he only writes the dispositions in Word the dispatcher has to record all of his bookings in the booking software again. Thus, this work is done twice. But once the booking is covered he likes the possibility to check back if everything is in place. On the other hand he confesses

that they do not use much functionality of the program and hence leave a lot of the programs possibilities unused. Since not all of their days are similar it would be hard to plan everything ahead as he mentioned. Often they need to reschedule and rely therefore on the dispatcher to overview everything and know which resources are available and where to get them. For this, the dispatcher usually also writes certain dates on a printed calendar and searchers solutions on his own. This is also not covered by the system but often wrote back into the program afterwards.

Asked for his dream software program to use he said that he does not need any special tools beside the ones he is already using. One thing he could think of would be a resource planning tool which is similar to the one they are using but which should focus more on usability. In addition, it should also have the possibility to automatically generate dispositions. This could be hard to automatise since dispositions can be quite different at a time and therefore they would need to enter a lot of information into the system they usually "just know" or assume somebody else is aware of. The other tool he would like to use is an archiving tool which is able to also include other information sources such as GPS data and notes from the filming crews. This should make it then possible to search for certain sequences and after filming sites. But as he mentioned it is unlikely that he could compel his filming crews to enter data into a system while they are in the nature filming. Nowadays they often have to review sequences very often and rely on people who know where something can be found. Sometimes they just say out loud during the recording where they are and what they are filming in order to make it easier for the cutter to know what he has to do. Even not the best technology can replace human relations as he emphasized.

Bibliography

[AFG+10] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, April 2010.

[AS94] K.R. Abbott and S.K. Sarin. Experiences with workflow management: Issues for the next generation. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 113–120, New York, NY, USA, 1994. ACM.

[BE01] R.A. Botha and J.H.P. Eloff. Separation of duties for access control enforcement in workflow environments. *IBM Systems Journal*, 40(3):666–682, 2001.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5):1–5, 2001.

[BN89] D.F.C. Brewer and M.J. Nash. The chinese wall security policy. In *Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on*, pages 206–214, May 1989.

[Bor97] P. Bordia. Face-to-face versus computer-mediated communication: A synthesis of the experimental literature. *Journal of Business Communication*, 34(1):99–118, 1997.

[BS91] L.J. Bannon and K. Schmidt. Studies in computer supported cooperative work. chapter CSCW: Four Characters in Search of a Context, pages 3–16. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 1991.

[BYV+09] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25(6):599–616, June 2009.

[CB05] T.M. Connolly and C.E. Begg. *Database systems: a practical approach to design, implementation, and management*. Pearson Education, 2005.

[Cos11] T. Costello. It without hardware. *IT Professional*, 13(2):63–64, March 2011.

[Dur10] D. Durkee. Why cloud computing will never be free. *Commun. ACM*, 53(5):62–69, May 2010.

[DWN99] G.M. Donahue, S. Weinschenk, and J. Nowicki. Usability is good business. *Available at http://www.yucentrik.ca/usability.pdf [24.4.2003]*, 1999.

[ED13] I.R. Edvardsson and S. Durst. The benefits of knowledge management in small and medium-sized enterprises. *Procedia - Social and Behavioral Sciences*, 81(0):351 – 354, 2013. World Congress on Administrative and Political Sciences.

[FCK95] D. Ferraiolo, J. Cugini, and D.R. Kuhn. Role-based access control (rbac): Features and motivations. In *Proceedings of 11th annual computer security application conference*, pages 241–48, 1995.

[FTD+12] A. Floratou, N. Teletia, D.J. DeWitt, J.M. Patel, and D. Zhang. Can the elephants handle the nosql onslaught? *Proc. VLDB Endow.*, 5(12):1712–1723, August 2012.

[GALSH04] J.A. García Avilés, B. León, K. Sanders, and J. Harrison. Journalists at digital television newsrooms in britain and spain: workflow and multi-skilling in a competitive environment. *Journalism Studies*, 5(1):87–100, 2004.

[Gar10] J.J. Garrett. *Elements of User Experience, The: User-Centered Design for the Web and Beyond*. Pearson Education, 2010.

[GHJV94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994.

[Gon10] M. Gonnering. Dam, youtube is great for video, but it is not digital asset management. *Journal of Digital Asset Management*, 6(6):99–118, 2010.

[ID12] I. Iankoulova and M. Daneva. Cloud computing security requirements: A systematic review. In *Research Challenges in Information Science (RCIS), 2012 Sixth International Conference on*, pages 1–7, May 2012.

[JCG+05] A. Jaimes, M. Christel, S. Gilles, R. Sarukkai, and W.-Y. Ma. Multimedia information retrieval: What is it, and why isn't anyone using it? In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '05, pages 3–8, New York, NY, USA, 2005. ACM.

[KH10] A.M. Kaplan and M. Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59 – 68, 2010.

[Lea10] N. Leavitt. Will nosql databases live up to their promise? *Computer*, 43(2):12–14, 2010.

[LR08] C. Lewis and J. Rieman. Task-centered user interface design - a practical introduction, 1993. *University of Colorado, Boulder.(This shareware book is available at ftp. cs. colorado. edu)*, 2008.

[LW91] L.J. LaPadula and J.G. Williams. Toward a universal integrity model. In *Computer Security Foundations Workshop IV, 1991. Proceedings*, pages 216–218, Jun 1991.

[Mar03] R.C. Martin. *Agile software development: principles, patterns, and practices*. Prentice Hall PTR, 2003.

[MB09] E. Mitchelstein and P.J. Boczkowski. Between tradition and change: A review of recent research on online news production. *Journalism*, 10(5):562–586, 2009.

[MN90] R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3):338–348, 1990.

[Nie94] J. Nielsen. *Usability engineering*. Elsevier, 1994.

[NKCM90] C.M. Neuwirth, D.S. Kaufer, R. Chandhok, and J.H. Morris. Issues in the design of computer support for co-authoring and commenting. In *Proceedings of the 1990 ACM Conference on Computer-supported Cooperative Work*, CSCW '90, pages 183–195, New York, NY, USA, 1990. ACM.

[NM03] S. Natu and J. Mendonca. Digital asset management using a native xml database implementation. In *Proceedings of the 4th Conference on Information Technology Curriculum*, CITC4 '03, pages 237–241, New York, NY, USA, 2003. ACM.

[Nor02] D.A. Norman. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002.

[Pal04] M. Palmer. Newsroom workflow. *IBE: International Broadcast Engineer*, 40:58–59, 2004.

[PRM+04] W. Pratt, M.C. Reddy, D.W. McDonald, P. Tarczy-Hornoch, and J.H. Gennari. Incorporating ideas from computer-supported cooperative work. *Journal of Biomedical Informatics*, 37(2):128 – 137, 2004.

[Saw01] A. Sawarkar. Digital asset management. *Digital Media Solutions Center, XP-002198865*, 2001.

[SCFY96] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[Sch91] K. Schmidt. Riding a tiger, or computer supported cooperative work. In Liam Bannon, Mike Robinson, and Kjeld Schmidt, editors, *Proceedings of the Second European Conference on Computer-Supported Cooperative Work ECSCW '91*, pages 1–16. Springer Netherlands, 1991.

[Sch11] S. Schneider. Hybrid software as a service (saas) solution for media asset management. In *2011 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–4, June 2011.

[SR03] A.P. Sheth and C. Ramakrishnan. Semantic (web) technology in action: Ontology driven information systems for search, integration, and analysis. *IEEE Data Engineering Bulletin*, 26(4):40, 2003.

[SS75] J.H. Saltzer and M.D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[SS09] S. Staab and R. Studer. *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

[Sta08] A. Stanek. Media asset management in der filmproduktion: Ein entwurf eines media asset managementsystems für die gesamte filmproduktion. Master's thesis, FH St. Pölten, 2008.

[SW03] A.M. Sokusky and R. Wang. Media asset management for broadcast news: Integrated workflow management. *Broadcastpapers.com*, 2003.

[SWL11] L. Shan, L. Wei, and Z. Ling. Research on novel solutions to existing problems of mam based on logical function framework. In *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*, pages 1–4, May 2011.

[TMAS11] A. Taivalsaari, T. Mikkonen, M. Anttonen, and A. Salminen. The death of binary software: End user software moves to the web. In *Creating, Connecting and Collaborating through Computing (C5), 2011 Ninth International Conference on*, pages 17–23, Jan 2011.