Universiteit Leiden

ICT in Business

The Rubik's cube model:
A metadata driven approach to access control
in modern ECM systems

Name: Orestis Gkikas
Student-no: s1218697

Date: 8/7/2014

1st supervisor: Dr. Hans Le Fever
2nd supervisor: Dr. Luuk J. P. Groenewegen

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Acknowledgements

Coming to an end of an effort like this always leaves you in mixed feelings.
The satisfaction for what you have achieved, the happiness for reaching the end of the journey, the excitement waiting for the presentation of your findings, the nostalgia of your student life and of course the stress that accompanies the whole project.

Acknowledging their efforts and their contribution in making this experience happen I would like to thank the following persons:

My academic supervisor Dr. Hans Le Fever who was the person that trusted me with this project and guided me through the research process with his valuable advice and the brainstorming sessions.

Dr. Luuk Groenewegen who supported this research by accepting the role of the second academic supervisor.

My supervisors from FileLinx Frank Bunck and Arjan de Goffau for the continuous support during the research and the vital help when things were literally stuck and also for providing me with all the necessary resources. I would also like to thank the rest of my colleagues from FileLinx for the time that they spend on my project and for their mental support.

My parents that gave me this opportunity to do this master at Leiden University and supported me with all their powers so that I can achieve my goals.

Last my friends and my girlfriend for their valuable help in the relax-time during this project and the understanding they showed regarding my restricted free time.

Contents

Glossary of terms

AC	Access Control
ACL	Access Control List
ACM	Access Control Matrix
AD	Active Directory
AIIM	Association for Information and Image Management
BPM	Business Process Modeling
CRM	Customer Relationship Management
DAC	Discretionary Access Control
DMS	Document Management Systems
ECM	Enterprise Content Management
ERP	Enterprise Resource Planning
HIPAA	Health Insurance Portability and Accountability Act
HR	Human Resource
IS	Information Systems
MAC	Mandatory Access Control
NIST	National Institute of Standards and Security
PBS	Project Breakdown Structure
RBAC	Role Based Access Control
SoD	Separation of Duty
SSO	Single Sign On
TRBAC	Temporal Role Based Access Control
WCM	Web Content Management

List of figures

List of tables

– Abstract –

The term Enterprise Content Management refers to the strategies and the software tools used by organizations to manage the content of their information assets. As ECM systems contain sensitive and confidential information the need to retain high security standards is critical. Based on research results, what is considered to be the most important issue regarding security in ECM systems is the access control management. While access control models tend to be rigid enough to ensure strict security, organizations become more and more dynamic. In the organizational context, this research provides with a new approach to a dynamic access control model based on the Role Based Access Control standard. The RBAC standard is extended with the use of the metadata of the information assets found in the ECM systems. The metadata is a state-of-the-art concept in ECM systems nowadays. The model uses the metaphor of the Rubik's cube to conceptualize its dynamic nature and provides with a proposed functional design to stimulate and guide its implementation.

Keywords: Access Control, ECM, RBAC, Metadata, Rubik's cube, Functional design.

1. Introduction

Enterprises of the Information Age produce loads of information striving to pioneer and to profit by creating a private knowledge-based ecosystem based on their own information assets. However, as the rise of the Big Data phenomenon indicates, organizations produce much more information than they need; according to a research from Capgemini 50-60% of the information stored in organizations is rather not-useful to them. They also produce much more information than they can manage; based on Gartner 80% of the information stored in organizations is unstructured. One of the biggest challenges associated with the growing quantity and importance of the information assets is security. Information is flowing in and out of organizations in an unprecedented rate and the challenge all organizations face is to balance the benefits gained from this information against the potential risks and to deliver a security framework that maintains high levels of protection whilst not stifling business processes [15]. Having this in mind and with the focal point being the application of security practices in ECM systems we are going to introduce an approach to manage the security of information resources using the metadata associated with this information stored in ECM systems.

The scope of this research is placed in the area of Information Systems (IS) and more specifically in the area of Enterprise Content Management (ECM) systems. According to AIIM, which is the global community of information professionals, those systems are used to capture, manage, store, preserve and deliver content and documents related to organizational processes. Because ECM relates to both technological and managerial challenges according to vom Brocke et al. in [7] it also constitutes a relevant Information Systems topic. The management of such systems bridges the gap between computer science and business as it enables information technology applications to support the major business functions and activities. Since information is one of the most valuable assets for organizations today, the security of these systems is critical. The focus of this research is the aspect of security that relates with the access control in ECM systems and the way this can be optimized with the use of metadata.

The content inside an ECM system can be related to Customer Relationship Management (CRM), Human Resource Management (HRM), internal processes, legal issues, managerial data and many more. This makes these systems an important value-adding service for the organizations. Additionally, an information security survey from AIIM [15] indicates that 30% of the organizations store in those information repositories data that are classified as secret and 14% of them data that are classified as top secret. Especially for larger organizations, this percentage extends to 40% and 17% respectively. As for the classification criteria, secret and top secret information is considered of highest level of confidentiality where disclosure of this data to unauthorized users can be even prevented by laws and regulations. It goes without saying that privacy under this context is a highlighted keyword.

However, according to Schoeman privacy is a state of condition of limited access to a person. In general, organizational privacy policies describe the practices used over the collection and the use of information, the period that information resources must be retained and the persons involved in the whole information lifecycle. The privacy mechanism

that determines which users are permitted access to information resources is known as access control mechanism. So, while information is a key resource for the development of organizations today, it seems that a major internal effort is needed in order to keep information away from the users. What sounds to be controversial is also for organizations the essence of doing the right trade-offs regarding the access control not only in one system but in the whole system landscape where information resides. Besides, security is always accompanied by some kind of trade-off; when it comes to information security the question to ask is not whether it makes the organizations feel secure but whether it's worth the trade-off. Following a strict access control process leads to a rigid security system not appropriate for dynamic and fast growing organizations but good enough for others like governmental and legal organizations. On the other hand, following more loose policies will result in critical security breaches. So, what could be in this context a silver bullet?

1.1. Problem description

In an ECM system a user can find a wide range of information resources. However, only a small part of this information is usually necessary for the users in order to be able to perform their core business tasks. While this is a small chunk of the overall information resources it is important that the users will have access to what they need when they need it. To ensure this requirement many security models have been deployed, most of which determining a static access control mechanism to decide who can access what. What appears to be a misfit in this practice is that in dynamic environments requirements change fast. Diverse project teams, flexible organizational structures, multiple and flexible business tasks, partnerships and synergies, mobility of the labor, are just few factors that can affect the needs of a user for information. In addition to this, cloud services and mobile devices offer possibilities to a new way of working and now information can be available practically everywhere. Setting the requirements for an access control mechanism of a system or of a system landscape seems to be a problem with multiple dimensions. The dynamic nature of the problem disregards static solutions to access control and favors solutions of similarly dynamic nature.

It is the goal of this study to develop a model that can be used as a flexible access control mechanism to manage the security of dynamic ECM systems. The conceptual basis of this model will be the Rubik's cube model which was introduced by the participating team to the Strictly for Business 2013 event representing the ICT in Business master class of Leiden University. The research objective is to construct and describe an access control model tailor-made for Enterprise Content Management systems that can serve dynamic enterprise organizations with a large number of users.

1.2. Research questions

Given the above mentioned problem description and goal of the research we can concisely restate hereafter a concrete set of questions so that we outline in this way the detailed

objectives of this research and give an indication to the reader over the expectations from this thesis.

Research question:

To what extent can we apply the Rubik's Cube conceptual model into a functional design of an access control system that allows dynamic and flexible permissions granting to users of an Enterprise Content Management system?

Performing some further analysis to the basic research question we can break it down to a set of four sub-questions.

Research sub-questions:

What are the main issues in ECM security?

ECM is quite a new trend for the enterprises and it is an umbrella term used to include and support multiple systems and user types, multiple data formats, multiple data repositories and multiple devices. In this research we will identify what are the main issues with respect to the security of the data stored in ECM systems.

Which flexibility requirements can be defined under the context of access control systems?

Flexibility is a broad term with various definitions available depending on the field of application that we define it. For the purpose of this research we need to provide with a suffice definition of the requirements of a flexible access control mechanism.

What is the Rubik's Cube Model?

The Rubik's cube model is a conception of a group of students from the master program ICT in Business of Leiden University, based on a metaphor using the original Rubik's cube to describe a flexible access control system. The way this metaphor is used in this model will be further described in the following chapters.

How can we test if the theoretical design can be applied to a functional design?

As a last part of this study we need to implement a functional design that will be derived from our Rubik's cube model. In order to do so, we first need to test if this model is possible to be converted in a functional design suitable for ECM systems and then see if this can be turned into a hands-on solution.

1.3. Motivation

The motivation for this study was big as it was derived from different sources. At first there was a major personal motivation for this research. This thesis was the concluding project for this master. The challenges of managing and implementing a research project are significant and in this case the personal motivation keeps you always on track. From the personal perspective also, this project was important as it allowed the continuation of a concept that was developed during another project of the master program. The Rubik's cube was a proposed solution for the business challenge assigned to a project team from ICT in Business program during the Strictly for Business 2013 event. Due to the strict schedule of the event this concept remained broad until I was given the opportunity to continue this research during an internship in FileLinx BV.

The academic motivation was also important. The concept of Rubik's cube was not based on academic knowledge. The challenge in this case was to see if there was any connection between the conceptual model and the existing grounded theories on access control that would enable a solid model description. That concept was also of great interest for FileLinx which was the company that supported this research. The company has its own ECM software and the problem that was described in the previous paragraphs was also a problem of the company. That gave me an extra motivation to conduct a research that also solves an existing hands-on case. Hence, the challenge for this research was to contribute to the academic knowledge and also provide a solution to a practical assignment.

1.4. Contribution

While numerous models were developed to address the issue of access control in Information Systems such as the Lattice-based access control models [20] or the Bell-LaPadula model [23], the most successful one, especially among enterprises, is the Role-Based Access Control model. The proposed standard already since 2001 [29] has been used in many variations such as the Task-role-based access control model [32] or the RBAC designed for intranet security [45].

ECM literature tends to favor constructive researches, conceptual models and qualitative empirical works. For example a research by vom Brocke et al. analyzes ECM drivers over a business-process-oriented perspective [7] while Chiu and Hung in [10] explore access control issues in Financial ECM systems using the Enterprise Privacy Authorization Language (EPAL) to describe a privacy-based access control architectural framework. In addition, it is rather common to encounter case studies discussing the issues that evolve around the implementation and customization of ECM systems. AIIM conducted a research of over 60 organizations that implemented an ECM solution exploring customization issues that occurred in each case. Of the same spirit was the exploratory case study of Nordheim and Päivärinta [12] that explored customization issues in the case of Statoil ECM implementation and also described the concept of ECM customization in detail.

ECM is a two-faceted concept [46] as we can distinguish between business issues and also technology issues. This study contributes in the existing literature with a constructive research which focuses on ECM systems and extends the standard RBAC framework by using the metadata of the information stored in these systems. Issues with respect to the security of ECM systems will be analyzed and a new conceptual model based on Rubik's cube and the RBAC model will be introduced. Although the research publications regarding RBAC are numerous, this study offers an aspect of the RBAC that has not been studied before by making use of a state-of-the-art concept of metadata.

1.5. Thesis outline

The rest of the thesis is structured as following:

Chapter two
In this chapter we are presenting the main literature findings that support and explain the main concepts of this study. This includes the basic terms and the related work over the ECM systems and the access control literature.

Chapter three
In the third chapter the basic methodology and the steps followed during this research are explained.

Chapter four
The fourth chapter contains the results of the research. This chapter starts with an analysis of the business and the system requirements regarding the access control mechanism and then continues with the description of the model and the other parts of the functional design. The chapter ends with the validation of the presented results.

Chapter five
The last chapter of the thesis concludes the research by checking the degree to which the research questions were answered and by discussing the main results and outcomes of this research.

2. Literature review

The challenge for the organizations and the information community today is to fully understand and exploit the information that is hidden in the huge amount of stored data. Given a huge boost by the adoption of new network technologies and innovations in internet technologies, the reduced prices of data storage means and the advanced tools for data exploitation like the Business Intelligence software data production increased the last years exponentially. An astonishing 90% of the data in the world today was produced only during the last 2 years a phenomenon that follows the hype of the Big Data era.

Before, documents, files, records, and databases were the most common means to organize information [17]. In the Big data era organizations experienced an uncontrolled growth of information resources as a result of the digitalization of documents and the vast use of websites, intranet, extranets and social media. Natural language text, augmented graphics, pictures, audio and video material forms a great portion of these information assets [16]. Thus the term Enterprise Content Management was adopted to refer to the technologies used to organize and manage those information assets. The major focus of the research in ECM from the Information System perspective lies in the systems themselves; where the content resides and through which the content becomes available to the users.

What is crucial in ECM systems and is also the core of this study is security. The significance of security of the ECM technology is accentuated since ECM may include sensitive information such as business strategies, business plans, customers, marketing, product specifications, financial situation and R&D developments [16]. This renders the access control in such systems vital for the enterprise. The research literature over access control is quite broad and covers thematic areas that vary from computer networks and databases to software and Information Systems while a few research initiatives scope down to access control and security of ECM systems.

What appears to be the most popular topic for information security technologies is the one that defines roles in the access control mechanism. Starting from 1992 and up until 2010, the amount of publications regarding role-based access control models results to at least 1361 known publications [13]. This great amount of publications includes a premature phase of the role-based access control, literature with a theoretical focus and also literature with a practical focus.

In the rest of this chapter we are going to present findings from the literature review with respect to the Enterprise Content management systems and also with respect to access control mechanisms that address the security issues in Information Systems. The part of the literature that scopes down access control in Enterprise Content Management systems comprises the last part of this chapter.

2.1. Introduction to the ECM industry

The acronym ECM used to briefly refer to the term Enterprise Content Management emerged in 2001 by AIIM (Association for Information and Image Management) in order to describe the systems used for the management of the information resources within the enterprises. According to Smith and McKeen (2003), the concept of Enterprise Content Management includes the strategies, the tools, the processes and the skills an organization needs in order to manage all its information assets over their lifecycle [4]. ECM according to Gartner [5] can be viewed both as a software toolset and as a strategic approach to managing information.

According to the same source, as a strategic approach, ECM can help enterprises take control of their content and, in so doing, boost effectiveness, encourage collaboration and make information easier to share. As a software toolset, ECM consists of a set of capabilities and applications for content life cycle management that interoperate, but that can also be sold and used separately. ECM is an umbrella term used to include and support multiple user types, multiple data formats, multiple data repositories and multiple devices. It is the goal of ECM systems to allow the business user easily and continuously create, update and enhance enterprise content.

Moreover ECM systems offer to enterprises a set of benefits as identified in [6]. Through the use of ECM systems it can be expected that:

- Internal and external collaboration is improved.
- Value is added to the business by adding new customer services and products.
- Reliability and quality of content are improved.
- Communication of knowledge is increased.
- Organizational memory is improved.
- Operating costs are reduced.
- Fulfillment of external regulation and standards is better met.
- Business processes tend to be more efficient, effective and flexible.

Due to its strong association with the enterprise information, the uptake of ECM offerings remains strong as organizations harness their content to drive key business applications. Gartner in the Enterprise Content Management version of Magic Quadrant [5], scopes down ECM industry to include systems that contain the following core components:

Workflow management tool
Workflow tools support the business processes and allow business process modeling, content routing, work tasks assignment, audit trails generation and other important automated procedures. Workflows in ECM systems are defined as automated business processes and the field of BPM (an acronym used for the term Business Process Modeling) is one of the most popular applications of such systems.

Document management
Document management tools are fundamental for ECM systems with their list of capabilities to include multiple edit, version control, library services and various security

settings.

Image processing

Image processing applications (e.g. Optical Character Recognition applications) evolved as an extend of document management due to the organizations constantly moving forward to digitize as much of their information stored in paper as possible. These tools allow capturing, transforming and managing images of paper documents. The whole process of digitizing paper documents is fully automated, starting from the point that a paper document is scanned and ending with the archiving of the document in the appropriate format.

Records management

This component is used in the case of long term retention of content to ensure legal, regulatory and industry compliance. Data sovereignty is defined as the consideration of laws and regulations with respect to storing digital content in a different location than the one that the content is created. But except from the application of local and international regulations over data protection, compliance with industry specifications and standards are also very important. The importance is reflected on the vertical solutions that many vendors of ECM systems offer to meet this demand for compliance. For instance, compliance with HIPAA regulation is important for the health industry companies and an ECM product that offers such solutions would include security policies to protect patients' data from non-purpose use; such as the commercial exploiting of patients data.

Web Content Management

This component includes content creation functions such as templating, content flow, change management and also content deployment functions that deliver prepackaged or on demand content to the web servers. WCM tools serve mainly for controlling the content and influencing the interactions of a web experience.

Social content

A very important driver for the use of ECM is that it enables interorganizational collaboration in various terms. ECM systems simplify file sharing and knowledge management and also support collaboration between project teams. Blogs, wikis and intranets are added to this category as a portal for online interaction between users in the same organization.

Beyond the core components, the Magic Quadrant specifies a set of extended components that include digital asset management, e-forms, search functions, analytics, email integration and management, information archiving and application integration.

ECM is about managing content over the entire lifecycle, starting from the creation phase and ending to the deletion phase [7]. In this way it distinguishes itself from other related IS concepts that typically focus on individual content lifecycle phases. This is a result of the various components ECM systems contain, the core of which is mentioned previously in this paper. For instance, Document Management (DM or DMS in abbreviation) is present in storing and retrieving content, Web Content Management (WCM) facilitates publishing content and Record Management refers to arching and retaining content [8].

Consequently we can infer hereafter, based on these observations, that ECM is a tool that applies over the whole lifecycle of content management. To create an understanding of the content lifecycle we need first to break down into the major phases it contains. IS literature offers a multitude of content lifecycle models. Päivärinta and Munkvold (2005) break the lifecycle down to capturing, creating, reviewing, editing, distributing, publishing, storing, archiving and deleting. McNay on the other hand in [9] more generally differentiates between creating, approving delivering, and managing content. For the context of this research though we use a simple classification based on the aggregated knowledge from existing literature and also from the ECM software of Filelinx over which this research was conducted. The result of this classification is the model depicted in *figure 1*.

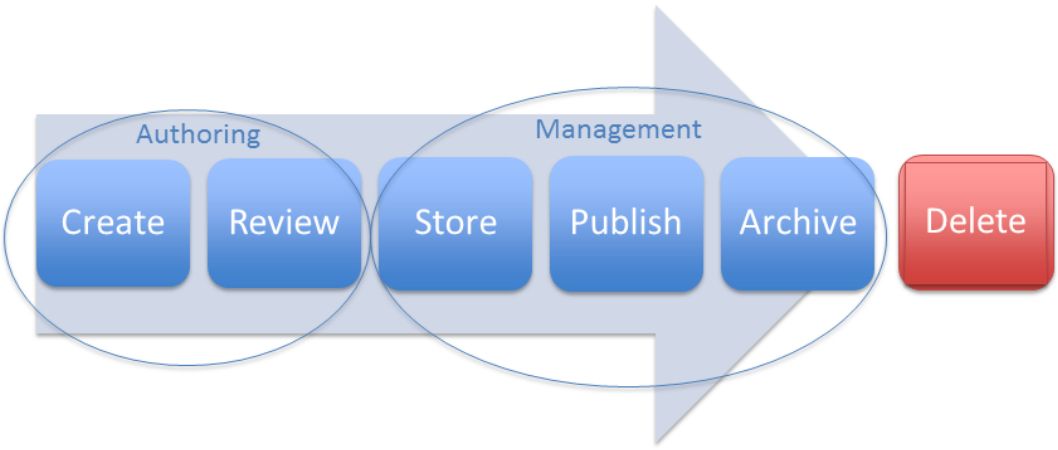

Figure 1. The content lifecycle

The *Authoring* phase is the start of every content lifecycle and it entails the content generation. This may be a result of content creation or capturing and it also includes existing content reviewing. The second phase refers to the *Management* of the content which is broad enough to nest storing functions and archiving, publishing and every other way of utilizing information. The last phase which signals the content's end is *Deletion.* Since we assume that a lifecycle can start also with reviewing existing content, deletion phase is not necessarily the end of an existing lifecycle as a lifecycle that starts with the creation of content can start all-over again with the reviewing of this content. What is interesting now that we explained the content lifecycle concept is to investigate how ECM fits herein. Due to their diverse portfolio of built-in tools, ECM systems fit seamlessly to the whole lifecycle, as we can see in *figure 2*.

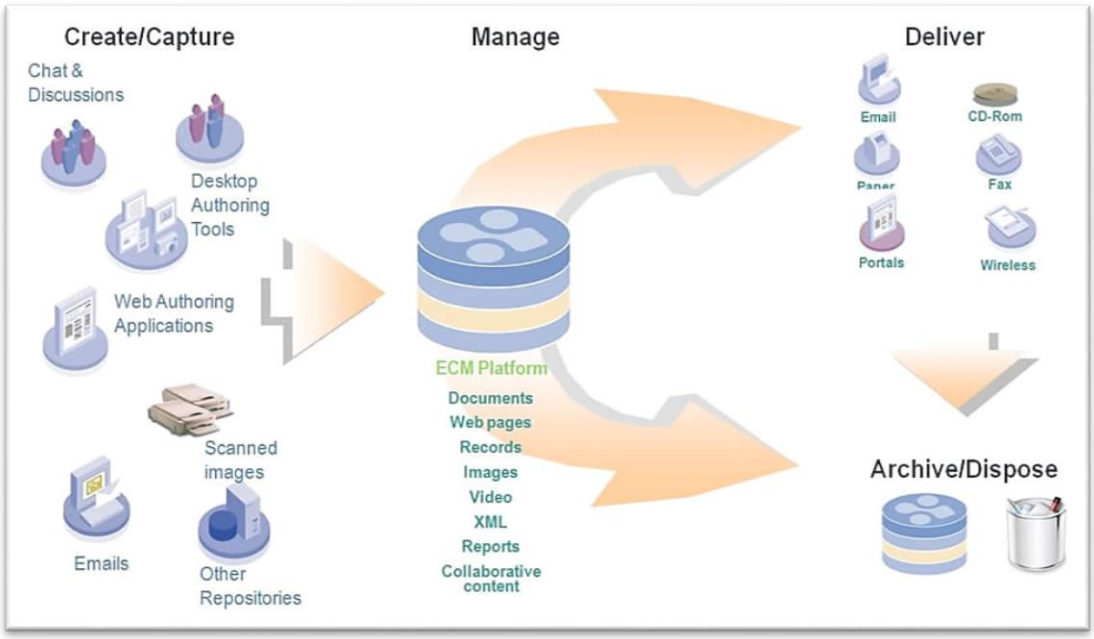

Figure 2. The content lifecycle in an ECM system

ECM systems are definitely not plug-and-play applications. And this is something deeply considered by organizations that invest in such systems. ECM solution is expected to last for many years, with an increasing functionality evolving over the years. The implementation of an ECM solution for an enterprise is oftentimes a tough case where a lot of parameters should be considered and a lot of resources must be committed. Hence, it is important that organizations follow an implementation plan supported by consultants specializing in the specific software. Based on the research by Nordheim and Päivärinta [11] and based on the implementation process followed by Filelinx consultants we can refer herein to the following model in *figure 3* that serves as a roadmap for the implementation of an ECM system.

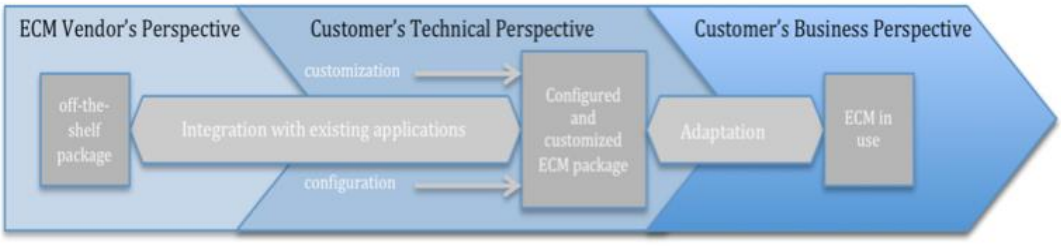

Figure 3. ECM implementation roadmap

After an organization selects an ECM vendor to provide with the most appropriate ECM solution a series of issues should be taken into consideration. The ECM system should be integrated with the existing applications and the infrastructure. Especially in organizations

today the application landscape is complex and integration is a hard task to achieve. An ECM system usually interacts with other applications to exchange data and execute automated processes. Those applications include Office tools, ERP systems, Web publication software, Portals, Collaborative platforms and legacy systems. ECM systems also interact with various hardware devises such as servers, printers, scanners and mobile devices.

Moreover ECM systems are highly customizable. Customization under this context is defined as an activity that modifies the properties of a software tool so that it better fits the organizational goals. It is two-folded term that can be perceived from a technical perspective and from a business perspective [12]. Configuration on the other hand is the selection from the available options during the installation process. Those processes combined lead to adaptation of an ECM system which is also a time consuming phase. Adaptation is bidirectional in this case and it leads to smooth utilization of ECM capabilities. On the one side the enterprise users of ECM learn to adapt to the use of the ECM system while on the other side the ECM adapts to the business processes and the business requirements of the enterprise.

It can be inferred from this introduction to the ECM systems that they offer a lot of benefits to the organizations and their importance grows alongside with the growing significance and bulk of the enterprise information that therein resides. Hence, it is very critical for organizations to protect their valuable content against unauthorized access. Chiu and Hung in [10] mention potentially dangerous consequences of a poorly designed access, including unauthorized disclosure, modification and destruction of information as well as unauthorized utilization and misuse of resources.

Information Security is the term used in this case by the authors to describe the means that are employed to protect information against these threats. The aim of Information Security is to minimize risks related to the three main security goals confidentiality, integrity, and availability [13]. The diversity in the way of accessing information and in the potential locations where it can be stored increases its importance. The most important features of Information Security in ECM systems are identified by the main off-the-shelf ECM vendors in [14] and are presented in *Table 1*.

Features of Information Security	Description
Audit trail	Monitor the activity of the users: Recent edits, new versions of files, read documents, etc.
Check in and check out	Control multiple and simultaneous editing of information by different users so that the information remains up to date.
Grant group and/or individual permissions	The access control model of an ECM system allows the authorization of access to individual users and/or groups of users
Automatic backup	The ECM system allows scheduled and automatic backups of the stored information.

Prevent accidental delete	This is a functionality that offers a temporary deleted resource location out which information can be retrieved back.

Table 1. ECM information security issues

Additionally and according to a research by AIIM [15] there are some core security risks that organizations should take under consideration about the security of their information resources. Starting from the risk that appears to be the one that the organizations are mostly aware of AIIM lists the potential security risks as follows in *figure 4*.

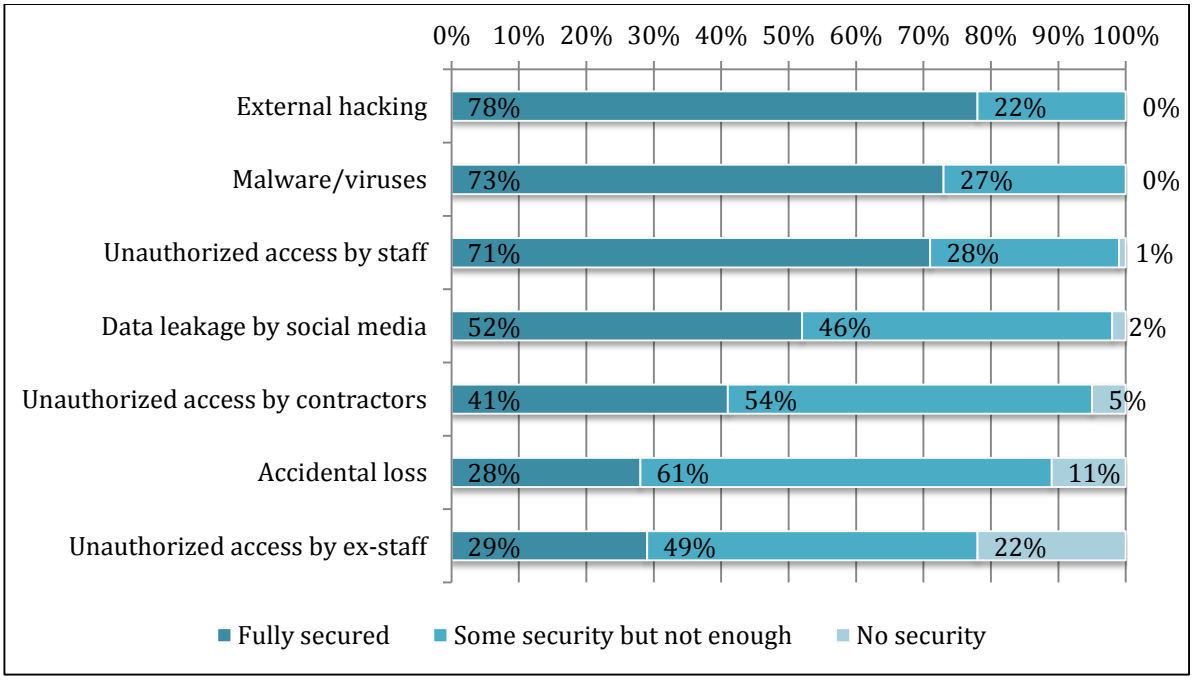

Figure 4. Levels of information security against security risks

Based on the findings of the same research by AIIM, unauthorized access by staff is the area of largest concern despite the fact that organizations believe they have protection in place against it. This indicates that the internal risk is perceived to be greater than the external risk. It also potentially highlights that the internal threat is not so straightforward to protect against. The same audience also believes that the risk of that type of internal breaches accidental to be is quite low. This indicates that in most of the times the internal risk for information violations from staff is rather deliberate.

Since the most highlighted security risk identified by the participating organizations is the unauthorized access by staff, the most prevalent deployed technique for information security in ECM systems is permissions system and access control. The rest most common techniques are used to address external threats. Ensuring the correct allocation and de-

allocation of access rights to content and information ideally by means of a role-based authentication and permission-based access control system would be a way to ensure security against unauthorized activities by internal users.

2.2. Access control models

Chiu and Hung (2005) perceive access control as the mechanism by which users are permitted access to resources according to the authentication of their identities and the associated privileges authorization. At an enterprise level the process to determine the appropriate privileges that delegate access to content is not simple. The work of Chiu and Hung is one of the few pieces of literature that scope down their research context over the access control in ECM systems. The most common tactic in academic literature is to deal with the access control of software systems and computer networks in general.

In the enterprise organizations information is produced within various business processes, which consist of business activities and is accessed by different users. In the enterprise environment business information has the characteristic of information sharing. Access to those information resources called objects is the action of reading, editing, deleting, executing or setting up policies over the access of this information. Hence, access control (or in short AC) aims to specify the users and their access over the information objects. Access control is the means to enhance control over these actions.

There are three primary abstractions used in access control system planning: AC policies, AC models, and AC mechanisms. Access control policies are high-level requirements that specify how access is managed and who may access information under specified circumstances. At a high level, AC policies are enforced through a mechanism that translates a user's access request, often in terms of a structure that a system provides [19]. Access control models bridge the gap in abstraction between policy and mechanism. Rather than attempting to evaluate and analyze AC systems exclusively at the mechanism level, AC models are usually written to describe the security properties of an AC system [19].

Access control is usually based on access permissions, also known as authorizations. Authorizations specify the subjects and the objects that can be accessible by these subjects and perform an action. There is a technical distinction between the subjects and the users when we talk about the use of an ECM system. A user is defined to be a human being recognized by the system with a unique identity. A subject can be a process or a program requesting access to the system and also can be associated with a user. In general a user may have many subjects concurrently running on the user's behalf in the system [20]. For the context of this research we are not distinguishing between users and subjects of the system and we will consider hereafter only users of ECM systems. Objects are information resources stored in the ECM system and can be documents, contacts, projects, employees, customers, organizations, workflow items and many more.

Authorization and access control have traditionally been the cornerstones of the IT security approaches [18]. Access control models come with a wide variety of features and

administrative capabilities, each with their individual attributes, functions, and methods for configuring a class of access control policies. Permissions are organized in terms of and derived from security policies that are applied under the context that the access control is also applied, providing a strategy for organizing, managing, and reviewing permission data, and controlling the access requests of the users [19]. As mentioned before, the literature over the topic is vast and the models derived are numerous. In this chapter we will refer to the most important ones:

Lattice-based access control

The lattice-based access control models [20] introduce levels of security on the information objects of the system. The information in the system flows from one object to another creating this way a lattice of interconnected objects. Every object is assigned a security level according to the information it contains and in order to be accessible by a user the security level of the user must be greater than or at least equal to the security level of the object. This model's overarching field of application is military and governmental systems where the demand for strict security mechanisms is high. The counterpart of this is the resulted rigidity of the model.

One of the well-known applications of lattice-based access control is the Chinese Wall policy which was identified by Brewer and Nash and described in [21]. The objective of this policy is to prevent information flows which cause conflict of interest as in the case of external consultants who deal with confidential company information for their clients. A consultant must not have in this case access to information about companies of the same type (E.g. two different banks) because such information creates conflicts of interests in the consultant's analysis and potentially abuse of this knowledge for personal benefits. The Chinese Wall has a dynamic aspect on this as it restricts the access of the consultant to any given bank B provided that the consultant has access to a bank A.

Mandatory access control

In the mandatory access control model only the administrator of the system manages the access control in it. The administrator defines the usage and access policies, which cannot be modified or changed by users, and the policies will indicate who has access to which objects. Occasionally MAC is called rule-based access control [22]. MAC is based on security labeling in order to specify access rights for the users. In this case each object is assigned to a security label and in order to be accessible by the user the security policies defined by the security label are tested to be checked if the access will be granted. MAC is based on the lattice-based access control model in the essence of the enforcement of the information flows in a lattice of security labels [13]. Two of the most important and primal applications of mandatory access control are the Bell-Lapadula [23] and the Biba [24] model.

Discretionary access control

Discretionary access control (or DAC in abbreviation) is a type of access control in which a user has complete control over all the objects he owns and also determines the permissions other users have on those objects [20]. The access control is discretionary in the sense that a user with certain permissions is capable of passing these permissions to other users unless a superior MAC model prohibits this action. DAC is not effective in ensuring security inside a system, as it is inadequate to control information flow among

users. If a user delegates another user with read rights over his objects then the delegated user may simply copy the content and create owned objects of the same content which in turn can delegate to any other user in the system. DAC allows users to copy data from object to object and as explained the result is that users who do not have access rights over the original data to gain eventually access to a copy of these data. The essence of DAC is that the owner of an object is the one who ultimately determines who is allowed to access it [13].

Access control models such as the aforementioned ones are usually accompanied by effective access control mechanisms that enable the operability and the realization of the model. Two commonly used access control mechanisms and also an optimization approach are presented here in the following paragraphs:

Access Control List

Access Control List (ACL) is a very commonly used access control mechanism of high granularity that associates users with permissions over objects. In this approach, permission to access resources or services is moderated by checking for membership in the access control list associated with each of these objects [25]. This list specifies the users who are granted access to the object as well as what operation they are allowed to do. Every entrance in the list is a set of a subject and an operation. For instance an entry like (Anna, delete) regarding an information object would give Anna the rights to delete it.

Access Control Matrix

Access Control Matrix (ACM) mechanism extends the ACL mechanism in a way that it includes a table in which each row represents a user, each column represents an object and each entry is the set of access rights of a user to an object. ACM and ACL based models though, are not suitable for enterprises as in large systems the complexity of the model increases rendering it unsustainable, hard to maintain and inefficient. However, in this chapter we will describe also two optimized versions of Access Control Matrix as introduced in [26] by Peisert and Bishop.

ACM: Standard Access Control

This model is based on an access control matrix mechanism, which serves as a mapping of users with objects. If we name this table A and every entry on this table a, then every entry will contain the allowed rights of a user over an object. This can be the result of a binary decision function, which will indicate which rights are allowed and which are not. If for example *u* is a user and *o* is an object then the expression a[u,o]={r} means that the user *u* is allowed to exert the rights included in the set {r} over the object *o*.

ACM: Conditional Access Control

This model is based also on an access control matrix mechanism and is augmenting the standard access control model by including external factors that are affecting the access rights stored in the table A. These external factors are given as an input to the system via the set M that describes them. The resulted rights in this case will be the outcome of a function *f()* that also takes into consideration a set of information stored in M input. Hence, the given expression now for the access rights is: a[u,o]=f(u,o,{r},m), where {r}

represents the rights under consideration and m the input variable. If for instance a system allows a maximum of three login attempts to a user then this can be expressed as follows:

$$A[u,o] = f(u,o,\{l\},m) = \begin{cases} \{l\}, \text{if } m \leq 3 & \text{, where } \{l\} \text{ the login right.} \\ \emptyset, \text{ else.} \end{cases}$$

An alternative to those models which became fast the epicenter of attention not only for academic studies but also for enterprise use is the role-based access control (RBAC) model. RBAC is deemed to be an attractive solution for access control in enterprises' applications and systems as it promises simplified and flexible user management, reduced administrative costs, improved security, as well as the integration of employees' business functions into the IT administration [13]. For instance, it can be shown that the cost of administrating RBAC is proportional to the sum of users assigned to a role plus the permissions assigned to the same role. If the permissions where assigned directly to the users then in this case the administrations costs would be given by the product of those two [25]. These advantages combined with the ability of RBAC to implement the security policies of the enterprise within an access control model that represents naturally the organizational structure were critical for the model's omnipotence.

The role-based access control model is based on granting privileges to roles inside the system before the users of the system are assigned to one or more of these roles. The fundamental idea behind RBAC is the removal of the direct linkage between the user and his permissions. Following this paradigm, roles are created for the various job functions and users are assigned to roles based on their responsibilities and qualifications [13]. After a user is authenticated in the system he is authorized to access resources according to the associated privileges of his assigned roles.

Roles represent functions within organizations and authorizations are granted to roles instead of users [27]. A role is a higher-level representation of access control. It can relate to either a single user or a group of users and is associated with different permissions. All users assigned a given role share the same privileges as the permissions associated with that role allow. The essence of RBAC is that it is a pragmatic tool that helps security architects to design a mechanism that satisfies a number of security policies. Sandhu et al. in [28,29] define a framework for RBAC that contains a family of models as follows:

- $RBAC_0$: the basic model where users are associated with roles and roles are associated with permissions.
- $RBAC_1$: the same model extended with role hierarchies.
- $RBAC_2$: the same as the previous model with restrictions on user/role, role/role and role/permission associations which are either static or dynamic (Separation of Duty).

Due to their importance and popularity in the academic and the industry world, the basic

RBAC models became the basis of a series of models that extended the functionality of RBAC in order to make the model more dynamic in some cases or more functional in some other. Zhang and Parashar in [25] for instance, extended RBAC so that it takes into account context information of the user to authorize access during an active session. If, for example, users are mobile and they access information using mobile devices, the context of the user could contain information about the location, the time, the system resources, the network state and the network security configuration to name but a few. This model dynamically adjusts role assignments and permission assignments based on context information adding extra functionality to the RBAC but not focusing on optimizing the model itself.

A model that dives into RBAC model attempting to optimize its usability in dynamic enterprises is the Temporal-RBAC (TRBAC) which was introduced by Bertino et al. in [30]. In TRBAC roles can be enabled in some periods and can be disabled in some others. Time intervals are used to better describe the periods of time that roles can be enabled or not. Moreover, TRBAC provides triggers that allow one role to specify enabling or disabling dependencies between roles. Although TRBAC provides a quite rich access control framework the efficient implementation and the design methodologies are considered to be open issues.

Another access control model based on RBAC was introduced in [31] and extended RBAC with a group concept. The group-based RBAC was actually an administrative model of access control that allowed decentralized administration management of the information system based on diverse groups which were self-administered. A group in this case is defined as a collection of users one of which is also assigned the role of the group administrator. The users are assigned specific privileges in the system based on their roles but they can also inherit the roles assigned to the groups that they belong to. The model is quite dynamic and enables a collaborative system but the degree of administration decentralization can always induce security risks. The association of groups with permissions in this model contributes also to the complexity of the administration model as the set of assigned roles to a user during the time that the user is activated in the system includes: 1) roles that are directly assigned to the user, 2) the roles that are directly assigned to the groups that the user belongs to, which are inherited to the user and 3) the roles assigned by the group administrator. Taking into account that every role is assigned to a set of permissions and permissions define operations over information objects we can infer that complexity of this access control mode will be an important obstacle to its implementation.

Furthermore, another extended model based also on roles and named the Task-RBAC, relates permissions and roles with business tasks [32]. In this way a role will have the privileges that are related with the tasks that this role has according to its business function. The role then is a mapping of a business role and the tasks are business tasks entailed from this role. This is a purely business enterprise oriented approach to access control as it is the case also with this paper. The Task-RBAC introduces an additional level of access control in the model which is the level of the business tasks. Every role of the system is analyzed to a set of business tasks and thereafter every task is associated to a set of permissions over the information objects. The tasks are then classified in order to avoid conflicts between the permissions that are assigned to roles. This is an important approach to access control in enterprise organizations because access privileges in the information system are assigned to users according to the actual tasks of their business role and also because the application of

the enterprise specific security policies are directly related to the tasks assigned to a role. That makes the application of the security policies in an enterprise environment easier. However, the model introduces an extra level of control as a consistent set of tasks must be related to the roles of the system and also to a set of permissions. This makes administration management more complicated and cost sensitive. Additionally, in enterprise environments with loosely defined business functions and business tasks this model misses most of its dynamics.

Last, one of the few published ad-hoc researches investigating access control issues in ECM systems was conducted by Chiu and Hung in [10] and is based on a case study in an international financial institute. The authors based on their findings present an architectural framework based on RBAC and also an implementation and integration framework suitable for ECM systems. The granularity though of this model is quite low as the focus of the research is on the various web services interoperating during the whole content lifecycle and on the architectural point of view of the implementation framework and not on the description of how an access control model would be specified in this framework. To the best of our knowledge, no further research study approaches the access control application in the context of modern Enterprise Content Management systems.

3. Methodology

The purpose of this research is the construction of a model based on the RBAC model and extended with the use of metadata. This is thus a constructive research that is conducted following the GQM process and it is realized by the Project Breakdown Structure (PBS) attached on the Appendix A. The PBS was used as a management tool of the whole project and it is in turn based on the Scrum software development framework, which is used in FileLinx BV enterprise organization where the research was conducted. In Appendix A the steps, the deliverables and the activities for every step are stated. Järvinen (2001) defines constructive research as typically involving the building of a new innovation based on existing fundamental knowledge. According to the same source, it is possible to accept a prototype or a plan instead of a full product when doing constructive research.

The realization of a constructive research project as it is the case of this master thesis project is a process that leads from theory to practice or more philosophically stated from intangible concepts to tangible ones. A research project demands devotion, time, organizing and discipline. And even though the first two sound pretty straightforward, the rest two are critical for the success. The research methodology for this project is based on the GQM research process [1], which is actually a framework for research projects. This framework constitutes a thorough end-to-end description of the research process as it is depicted in *figure 5*. The steps that are described in the GQM process were the basis for the steps and the activities of the PBS analysis found in Appendix A.

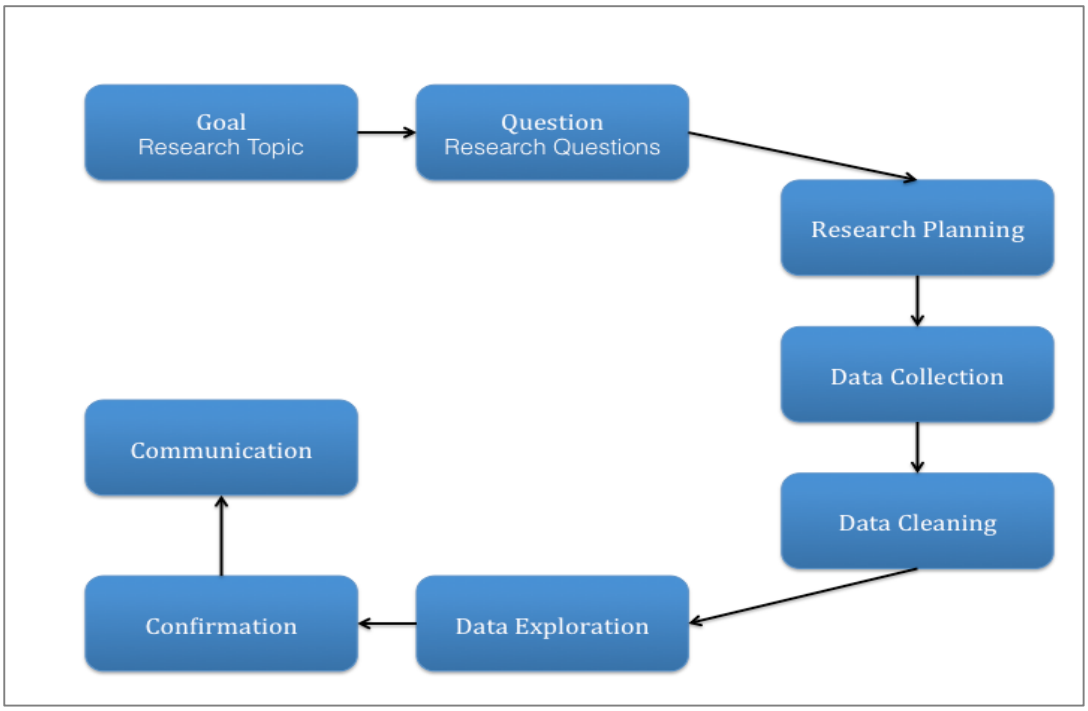

Figure 5. The GQM research process

3.1. Immersion and crystallization approach

According to Crabtree and Miller [3] the immersion and crystallization approach entails in broad terms, a prolonged immersion into and experience of the data and then emergence, after concerned reflection, with an intuitive crystallization of the data. To be more specific, the immersion phase started with exposing the researcher to the organization and to information vis-à-vis the ECM industry and afterwards the ECM software of FileLinx. In this way and during the immersion phase, a good realization of the research context and a clear description of the problem and the research goal are achieved. We have already provided a problem and goal description in the introduction of this thesis document. Moreover, during the immersion phase, a relative literature review was conducted, after collecting literature material regarding access control and its application on information systems.

Three main sources were used in order to collect the necessary set of literature material. The basic search engine used was Google and the referenced items of the search results were thereafter available in the main source of literature material of this thesis which is Google Scholar. The second important source used was the Digital Catalogue of Leiden University. This second source made the searching of material a bit more lengthy but provided access to big databases and publishing portals such as Elsevier [34] and Science Direct [35]. Last, an important collection of literature material used was obtained by the Research Gate [33] network of published articles. Research Gate is actually a social media for academic writers including also academics from Leiden University. The requested materials to these sources were mainly papers and articles but also scientific journals and books.

The combination of words that was used during the searching queries in these sources includes the words: “ECM” or “Enterprise Content Management” or “Information Systems” or “Information Management Systems” in combination with “security” or “access control” or “access rights”. Oftentimes, an extra filter was applied in order to scope down the results containing the words: “flexibility” or “conceptual model”. However, only the set of published studies over RBAC lists more than 1300 items. Hence, the definition of certain inclusion and exclusion criteria was necessary in order to narrow the results down to a reasonable number of documents.

The following inclusion criteria were used:

- Publications focusing on the application of access control in ECM systems or in Information systems in general.
- Publications that optimized or extended access control model such as RBAC so that it offers more flexibility or make it more dynamic.
- Publications regarding modelling techniques and tools for access control models.
- Publications regarding ECM drivers and trends.
- Classification studies regarding access control broad research area.

Accordingly, the following excluding criteria were used:

- Publications with a purely technological focus on areas like: Operating Systems, Databases, Web Networks, Middleware and architectures and others.
- Publications focusing only on one industry.
- Publications that were not related somehow to Information Systems.

The resulted set of published material contained about 70 studies over the research topic of this thesis not all of which were used in the purpose of this though. The extracted data were mainly grounded theories and tools that could be used to found the Rubik's cube model. Using theories broadly accepted by the scientific community allowed this research to better describe a new model of access control with a primary application on ECM systems which was based on an existing well-grounded model and was extending it in a way that it becomes more flexible and dynamic with the use of the metadata technology.

And this is where the crystallization phase starts. During the crystallization phase the goal is to find the fit between the requirements of the ECM platform owned by FileLinx and the conceptual Rubik's cube model. Crystallization is the phase when data analysis performed during immersion temporarily stops in order for the researcher to reflect on the extracted information and attempt to identify patterns or themes. Immersion and crystallization is not a one-shot process. This dual process continues until all the data have been examined and the patterns emerged are meaningful and can be well articulated [36]. The *Table 2* shows synoptically how the process was realized during this research.

Phase	Description	Steps
Immersion	Experience and realization of the research context and the data	Problem description Research objectives ECM market research Product analysis Literature study Interviews
Crystallization	Reflect on the analysis and identify patterns	Business requirements Data analysis Model description Functional design Validation

Table 2. Immersion and crystallization process

3.2. Interviews

Another source of data collection that was used during this research is the qualitative interviews. While the related literature review is a secondary source of data, the interviews performed for this thesis are considered a primary source. The basic goal of the conducted interviews is to collect data that will be used to define the overall business requirements of the new access control model. Those business requirements then were translated into the system requirement regarding the ECM platform of FileLinx upon which the access control model would be applied.

Two types of interviews were conducted. The first type includes formal interviews with the basic customers of FileLinx software and the second type includes the informal interviews with the selected employees from FileLinx organization itself. In total 6 different formal interviews were contacted each one with a person from a different client organization of FileLinx. The reason for this was basically to offer a bigger diversity in the data without the need of a big number of interviews. The persons that were selected for the interviews were mainly administrator users of FileLinx in order to guarantee a deep knowledge of the system. Additionally, interviews with employees of FileLinx were also very important. We can highlight here two types of interviewees inside FileLinx: 1) users of FileLinx and 2) users of FileLinx who are also involved in the FileLinx software development. Informality of those interviews lies in the fact that no standard questionnaire was used for them. The information in these cases was retrieved from notes that were generated during meetings, conversations and emails.

For the formal interviews with the customers of FileLinx a formal questionnaire was used. The questionnaire that was used for these interviews can be found in Appendix B. Four main areas are covered with the questions that were included in the interviews. The first area and the last one serve as the introduction and the epilogue of the interview and the goal is to gather general information about the interviewee and the reflection of the interviewee over the interview correspondingly. The rest two categories target to collect information about the use of FileLinx ECM platform and about security issues. It is also requested from the interviewees to propose solutions on how the issues under discussion could be better with respect to the security of the system. Those two categories were expected to offer the most meaningful data and thus in the process of creating these questions information was included from the market analysis and from the literature review.

An important point that occurs with interviews as a source of data for research purpose is bias. Bias in the conducted interviews is derived from the interviewer as well as from the interviewees. On the one hand, interviewees were experiencing FileLinx under the context of their organization so the issues and the suggestions they could provide with was certainly affected by this experience. It was also expected that during the interviews their proposed changes and solutions that could be applied to the new access control model would be regarding their organizational needs. Another type of bias from the side of the interviewee has to do with level of understanding of the questions. A question that is not correctly articulated or is not understood correctly can bring a misleading answer. In order to overcome these types of bias, questions were included regarding the overall experience of the interviewee and not only the present one in this organization. It was also asked that they

elaborate the answers by thinking out-of-the-box as much as possible. Additionally, the questions were conducted so that they are as simple as possible and also directions and examples were given in case a question was too broad.

On the other hand, bias deriving from the interviewer was also something to be considered. The new access control model was primarily something that could be applied directly to the ECM system of FileLinx. In this case unavoidably most of the questions were related to FileLinx system itself. This bias was present not only during the interviews but also during the rest of the research. In order to overcome this bias, the content of the interviews and of the whole research was based primarily on academic sources while FileLinx system was considered to be a system that the new model could be applied to. Hence, FileLinx was the enabler of this research and also the means to check its feasibility eventually.

3.3. Model and validation

The last part of this research was to generate a detailed description of the Rubik's cube access control model and also examine the validity of the model. After all, these are the most critical parts of this thesis and they are presented in the following chapter which is titled *Results*.

The model description included graphic representations of the model and text. No formal annotation or mathematical language was used for this purpose. Complementary to this, a combination of UML diagrams was used for the visualization of the model. A UML class diagram was used to visualize the main constraints that could be applied in the access control mechanism while a UML object diagram was used for the role inheritance description. Moreover, in order to describe the way the model works in practice we preferred to visualize it using a User Story model that implements a real case scenario. The User Story is combined with a data model that shows the values that the various components receive during the implementation of the scenario. The User Story should not be confused hereafter with the Use Case as the second is describing the interaction between a user and the system. The User Story instead is a realization of scenario and it was preferred as it makes the functionality of the model easier to understand.

As last part of the next chapter which presents the results of the research and as following of the model description, we provide a validation of the introduced access control model. According to the definition of Wikipedia [37] for software development, validation ensures that all the wanted functionality, as defined in the requirements, is delivered and the model satisfies its intended use. Additionally, in this thesis, the validation of the model will be extended to include also its feasibility implications.

4. Results

In this chapter we are going to present what is actually the result of the crystallization phases of this research. As we saw in the preceding chapter, the approach of this thesis follows an iterative process the second part of which delineates the descriptions of the patterns and of the concrete results from the observations and theories. Our basic model describes the application of an access control model on Enterprise Content Management systems and is using the metaphor of Rubik's cube to reflect its dynamic nature. The overarching use of the Rubik's cube model under this context is as a tool with which we can determine the golden section between the static and the dynamic application of access control in Enterprise Content Management systems.

One of the most important steps in describing a new system is to define first its requirements. We did so considering this access control system after analysing the input data mainly form the conducted interviews and also from the corresponding literature review. The background of the interview questions was technical and focused on the system context. Hence, the requirements extracted were primarily requirements of the system. Nevertheless, the business aspect of the requirements was extracted from the academic literature and especially from sources like Gartner and AIIM. In this way we managed to do a mapping between the business requirements and the system requirements of our Rubik's cube model, which are presented in the following paragraphs.

4.1. Business and system requirements

In a business environment there are several factors that relate to the requirements for an access control mechanism. The organizational structure and the job functions associated with job tasks, the business rules and the business processes, the security policies and the content to which they are applied, the application landscape and the enterprise architecture of the organization, are some of the basic factors that relate to the interaction of users and information objects [32]. A user can have several business roles from time to time in an organization so this leads to different access requirements in different systems. The user acquires specific privileges to access information and those privileges must comply with the business rules and the security policies of the organization.

Security policies should be concise, unambiguous and easy to understand and they are reflected in security rules [38]. Security policies can be the result of internal regulations of an enterprise or of external regulations that apply to the industry or the organization itself. For example if an owner of a bank account cannot withdraw at once more than 1000 euros from his bank account this is a security policy of the bank. The security rule in this case would be expressed more detailed with the statement:

[Bank account B, Withdraw amount W, then W< Balance(B) && W<1000].

The main responsibility of the access control model is to protect data from unauthorized access. Users want to access information resources in order to perform their business activities. It is the main goal of the access control mechanism to decide whether an access request of a user is valid or not. The access control model must also reflect the organizational structure where roles in the system can be mapped to roles in the organization. This is also important for the sustainability of the system in large scales. In big organizations with a lot of users and big amounts of data, sustainability and maintainability are critical issues. A model that maps roles in its inert mechanism and also utilizes the data inside the data in order to delegate access privileges can be a very flexible solution for big organizations.

Moreover, it is an overarching driver of the use of ECM systems that they allow better collaboration and information flow. Hence, along with retaining security, information sharing must be also enhanced. Although users are allowed to perform a set of operations over information resources, they are not allowed to define the access of objects in the system like it would be the case in a DAC model. Thereafter, the administration management must be centralized and authorized to define the structure of the access control mechanism in the system by assigning roles to users and by defining permission sets.

Concluding, in *Table 3* we present the requirements of our access control model accompanied by the functionalities of the system that implement them. It is important to highlight here that the focus of this study is one ECM application system where the access control model is applied to and not the whole application landscape of the organization. The collaboration of this model with integrated security systems like the Single-Sign-On (SSO) and Active Directory (AD) is not part of this study. Extending the Rubik's cube access control model so that it can be applied holistically as an integrated security solution to an enterprise application network could be the topic of further research.

Business Requirements	System Requirements	System Functionality
Protect data from unauthorized access, disclosure and utilization	Define roles with specific authorizations over information resources. Monitor users' activity. Protect from data loss. Allow group and individual permission granting.	System roles reflect job functions and business roles. One or more roles can be assigned to users. Each role is granted permissions. Audit trail functionality. Information objects versioning. Active rights overview. Automatic backup. Prevent accidental delete and restore functionality.
Enhance secure information sharing	Restrict information flow outside the system. Information encryption. Secure download.	Fileshare with external recipients enabled by specific rights delegation. Encode information with security keys or protect it with passwords. Allow only authorized resource download.
Allow flexibility on the access of information resources by the users	Use the metadata of the information objects. Map users to roles and not directly to permissions.	Fine-grain permissions at the object level. Allow temporary access rights. Allow user accounts deactivation. Allow conditional permissions.
Reflect the organizational structure	System roles are mapping business roles. Retain sustainability in large systems with many users.	Allow role and permission inheritance. Create standard set of permissions for specific roles. Separation of duty.
Conform to internal security policies and also to external regulations	Translate internal security policies into access rules. Translate external regulations & requirements based on the industry into access rules.	Translate the access rules into sets of permissions and access operations to the information objects.
Allow small degree of decentralization of access control management	Centralized administration management	Small number of administrator users (optimally one if possible). Administrator user aggregates all possible access rights.

Table 3. Mapping the requirements for the access control model

4.2. Rubik's Cube model description

The concept of the Rubik's cube model is based on a metaphor derived from the popular homonymous 3D puzzle game developed by the Hungarian sculptor Ernö Rubik. A player that tries to solve the Rubik's cube aims to match the colors in each of its six sides. Twisting the cube in its three dimensions allows an astronomic number of more than $4x10^9$ possible permutations. Every move generates a new combination of boxes in each of the cube's sides rendering the solution of the puzzle very arduous. The challenge of the Rubik's cube was alluring for many researchers who used the Rubik's cube metaphorically to express complex and dynamic concepts. McCumber in [39] described a model for information security assurance that considered all different factors that affected the model plotted in the three dimensions of the cube. A more recent application was from IBM [40] where the cube was used as a metaphor to describe the need for aligning all possible dimensions of a Business Process Management review process in the same way that all colors of a Rubik's cube must be aligned in the same side of the cube.

Under the organizational context and driven also by the metaphor of the Rubik's cube, we are going to introduce a new approach to a flexible access control model, which is based on the metadata of the information resources that is suitable for Enterprise Content Management systems. The Rubik's cube serves here as a visualization tool of the possible combinations of users, information objects and metadata rules that may occur in an access control model until a perfect combination is reached. This combination changes dynamically even if a small change occurs in one of its dimensions, as it would happen with the Rubik's cube, representing this way the dynamic nature of security in enterprise information systems today.

4.2.1. Extending the RBAC model

The basis of our approach towards a flexible access control model for enterprise organizations is the RBAC model as it was described by Ferraiolo et al. in the Proposed NIST standard for the Role Based Access Control [29]. According to the RBAC model, permissions are granted to roles inside a system and then the roles are assigned to the user. The fundamental idea behind RBAC is the removal of the link between users and access privileges [13]. Following this paradigm, roles are created for the different job functions within an organization so that they reflect their responsibilities and qualifications. In this way, there is a direct mapping between the roles of the system and the business roles of the enterprise organization. Hence, the model reflects naturally the organizational structure.

The basic RBAC model as we can see it in *Figure 6* consists of a set of elements and the relationships between those elements. The basic model includes six core data elements, namely *Roles*, *Users*, *Sessions*, *Permissions*, *Operations* and *Objects*. It also includes two assignment relations, namely *User assignment* and *Permission assignment* and two mapping functions, namely *User-Sessions* and *Session-Roles*. The last components of the basic RBAC

model are the Role Hierarchy and the Separation of Duty that we will analyze later in this paper.

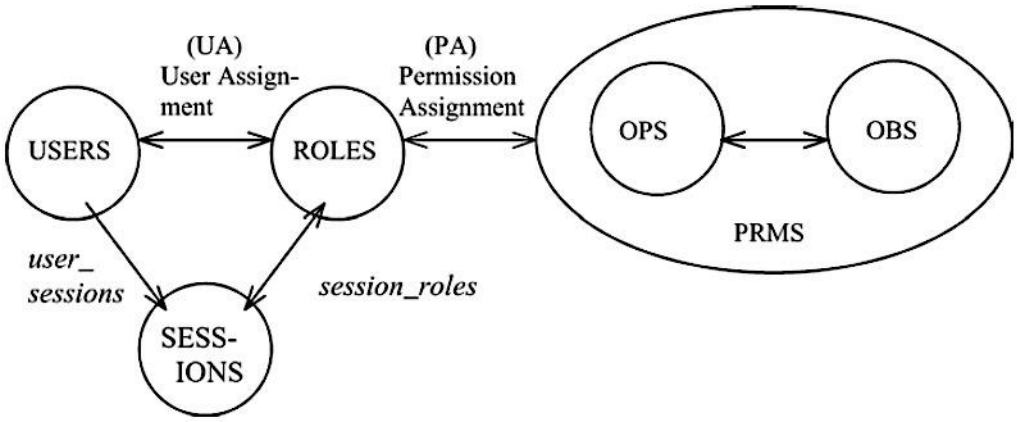

Figure 6. The Basic RBAC model

The basic elements of RBAC are described in this section. The RBAC model as a whole is fundamentally defined in terms of permissions assigned to roles and roles assigned to users. The RBAC model contains six basic elements.

Users

A user is defined as a human being. Although the concept of the user in information systems is usually extended to include processes, network entities, or intelligent autonomous agents, for simplicity reasons we specify an ECM system user as a person. Typically users of ECM systems are employees or managers working for the enterprise organization, or external users like customers, partners or contractors. In any case a user will own an account in the system accompanied by unique credentials that will individualize this user from the rest users of the system. Users are able to generate information, which is stored in the ECM system, and they can also use information generated by other users.

Roles

A role is a job function within the context of an organization with some associated semantics regarding the authority and the responsibility conferred on the user that the role is assigned to. Access privileges are assigned to roles instead of users rendering roles a high level representation of access control in the system. Roles can relate to a single user or a group of users and all the users that are assigned the same role share the same access privileges. A user may be assigned more than one roles and a role may be assigned to more than one users thus making the user-role a many-to-many relationship.

Sessions

A session is a mapping between a user and an activated set of roles that are assigned to the user. When a user logs-in in a system, a new session is activated. The user may retain more than one active sessions in the system. The set of active roles of the user is always a subset of the set of the authorized roles the user may be assigned. The session can be

terminated by the user or by the system itself, as in the case where a user is idle for a long time.

Objects

An object is an information resource that is stored in an ECM system and can be a document, a project, a contact, a workflow step, an employee, an email, a customer or even a company information object. Hence, an object can be an information container, such as a PDF file, a more complex system entity, such as a project, or even an exhaustible system resource, such as a printer. Following our metadata-driven approach to the RBAC model, attached to every object of the system is a set of fields that describe the properties of this object. Some examples of these property fields can be: The name of the object, the date when the object is created, the creator of the object, a brief description of its content, the type of information contained in the object, the status of the object regarding ongoing workflows, keywords that facilitate search results and many others. The information stored in these fields will serve as the unique metadata information for this object. The metadata fields and the information stored in it and also files of any format that can be attached comprise the entity of the ECM systems that is called information object and will be the main component of the access control in this model

Operations

An operation Is an executable image of a program which upon invocation executes some function for the user. Within the context of ECM systems, operations are defined as the actions that the users are allowed to perform over the information objects. Some of the most basic operations suitable for ECM systems can be defined hereafter:

- *READ* or in abbreviation '*R*': allows a user to view the information contained in an object. If the object is a document for example, then the user can read this document.
- *ADD* or in abbreviation '*A*': allows a user to create an instance of the object.
- *EDIT* or in abbreviation '*E*': allows a user to access the content of an object and also edit it generating this way a new version of the object.
- *DELETE* or in abbreviation '*D*': allows a user to delete an object.
- *CONFIGURE* or in abbreviation *'C'*: allows a user to modify the permissions of an object.

Permissions

Permissions relate the objects with the operations in an ECM system. It is an approval to perform an operation on one or more protected objects. The set of permissions in a system may be as big as the set that includes all the possible combinations of objects and operations. Permissions are assigned to roles authorizing them with specific access rights inside the system. A role may have more than one permissions and also one permission may be assigned to more than one roles, rendering it a many-to-many relation. If a role R_1 and a role R_2 have the same permissions the two roles are equated: R_1=R_2. In an ECM system standard permissions could be defined such as:

- *Allow Full Control:* It is suitable for administrators, gives full access rights inside the system.

- *Manage Permissions:* This allows objects' permission management inside the system.
- *Limited Access:* This could be especially assigned to external users of the system.

Central to the basic RBAC model is the concept of the role relations and the concept of the mapping functions. Those concepts describe the links between the basic elements and reflect the functionality and the flexibility of the model. They also facilitate the application of the *principle of the least privileges*, which in the context of ECM systems can be translated as the limitation of the access only to the information that a user needs. This principle is in accordance with the demand for authorized access, in the sense that the user is granted access only to the necessary information objects without violating internal security policies.

User assignment

A user assignment is a many-to-many relationship where a user can be assigned to one or more roles and a role can be assigned to more than one users. In this way the same role can be assigned to more than one users in the same session.

Permission assignment

A permission assignment is also a many-to-many relationship where permission can be assigned to one or more roles and a role can be assigned to one or more permissions. In this way the system role is enriched with the business context, as it is now representative of the business needs and the tasks that the business role fulfills. In the case of permission assignment it is not relevant for the business to have more than one roles with exactly the same permissions.

User – Sessions mapping

A user-session is a function that performs a mapping of one user onto a set of sessions. A user then establishes a session during which a subset of the roles that are assigned to this user is activated. Each session is associated with a single user but a user may be associated with one or more sessions.

Session – Roles mapping

A session-role function performs a mapping between sessions and roles. The result of this function is a set of roles that are activated during a session. This is usually a subset of the complete set of roles that exist in the ECM system.

4.2.2. The Rubik's cube model

The proposed Rubik's cube model is an augmented access control model based on the core $RBAC_0$ model, including the following additional characteristics: Role Inheritance, Object Classification, Security Levels and Metadata Permissions. The mechanism that enables those extended characteristics is depicted in *figure 7*.

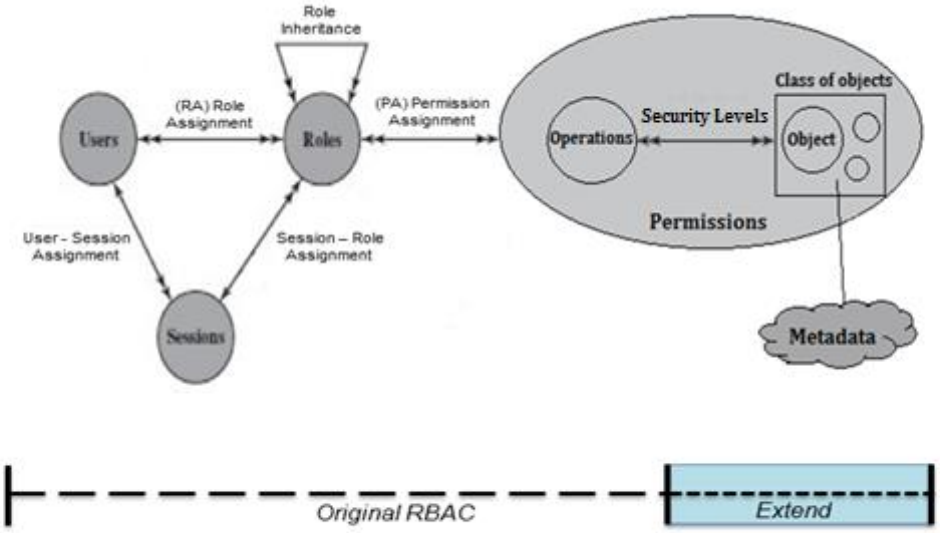

Figure 7. The Extended RBAC model

Proceeding with the description of the model we are going to analyze in the following section those additional characteristics of the Rubik's cube model:

Role hierarchy

Hierarchy in system roles is in accordance with the organizational structure reflected in the business roles hierarchy. Employees, line managers, senior managers and directors compose hierarchical levels in the most traditional organizational structures. Hierarchies are a natural means of declaring the scalability of responsibilities and authorities of the business roles. Equivalently, it is a means of structuring system roles following the same rationale based on the fact that roles share permissions to a certain extent. As a result, superior roles may be defined by extending the permissions of subordinate roles.

Role hierarchy in this perspective is mainly an administrative tool that facilitates the role engineering process by allowing the definition of roles based on existing ones. A superior role inherits a subordinate role and is also privileged with an extended set of permissions that justifies enhanced access rights in the system. While the $RBAC_1$ model incorporates role hierarchy, it is rather strict to be applied in dynamic ECM systems. In our model, role hierarchy is mainly an administrative tool used to simplify the role engineering process without imposing additional constraints besides the ones defined already by the separation of duty and the principle of least privilege. Hence, a role can be created based on a specified role of the system by extending its permissions as it is depicted in the UML

object diagram in *figure 8*. However, it is important that a role cannot inherit permissions from a constrained role or from a set of roles where constraints apply so that we comply in this way with the security policy restrictions.

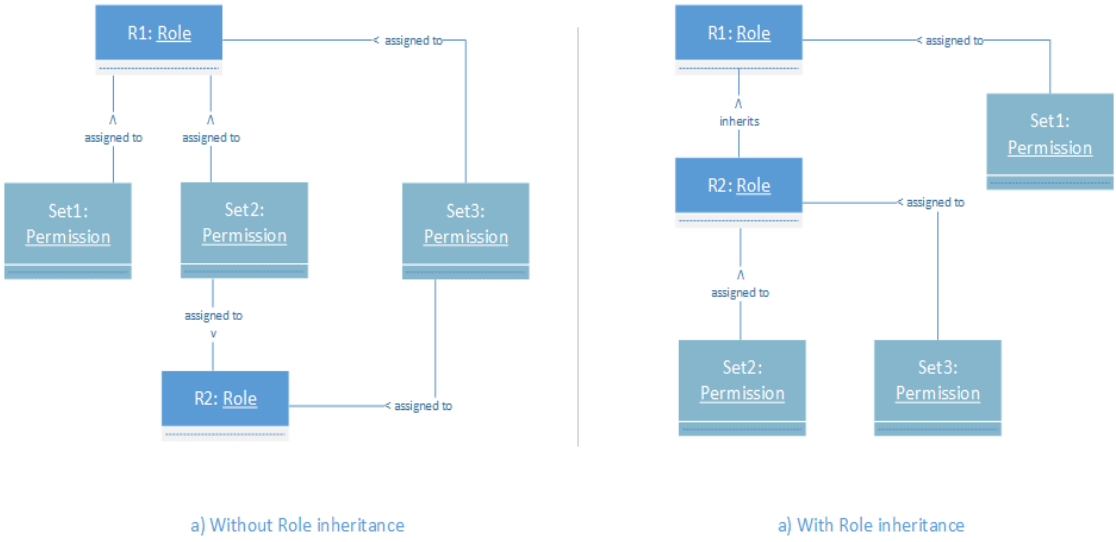

Figure 8. Role inheritance object diagram

Classes of Objects

An ECM system offers extended capabilities for information management in organizations and this is also enhanced by the existence of information objects in the system. Objects increase granularity in ECM systems but also add complexity. In order to deal with complexity in large systems with many types of objects we introduce in our model a classification system based on the organizational structure. We use classes of objects to categorize the information and reflect the types of information used in each organizational function. A deriving functionality from the use of classes is that we can introduce role permissions at a class level, so that they determine access over the objects of the class in a bundled way. The classes that are defined in this model are related to the organizational structure reflecting organizational functions and objects with similar content. So for instance we can create an HR class or a Finance class for a specific department in an enterprise organization containing information objects of relevant content.

Metadata

The basic feature of our Rubik's Cube access control model is the use of metadata that we derive from the property fields of each object. The information from the metadata will be used in order to fine-grain the permissions which are already assigned to the roles of the system. The roles represent a static view of the access control in the ECM system while the metadata are the realization of the dynamic view of the access control. The system administrator can specify in the permissions defined at the metadata level a set of access control rules that check if any roles, users and conditions are applied. In this case the access to the object which was based on the preexisting rules defined at the roles' level will be modified.

The metadata, also known as data about data, include information that better describes the data properties. This information may be with respect to the lifecycle of an object, the object owner, users that edited the object or versions of the object that are still available, timestamps for each version, conditions related to steps of ongoing workflows or related to the status of projects that this object is part of, and as mentioned already, information with respect to the allowed actions over the object. According to Gartner the metadata management is a point that requires attention as metadata can unlock the value of the data.

Already metadata capabilities have been part of ECM systems as a content tracking tool. As such, metadata offer pieces of information that can be traceable during a search function in which the user may search for the name of the object, the owner of the object, the date that it was created or for any other property of the object stored in its metadata. Based on research findings, AIIM identifies some more ways to use metadata in ECM systems:

- *Retrieve content:* Helping users to find and retrieve content is the function of metadata that is most closely connected with taxonomies. In this function, descriptive metadata include things like taxonomy topics, subject keywords and document descriptions.
- *Track usage of content*: A more sophisticated use of metadata is to track the usage of an object and to connect content to other content. A well-known example of this is the Amazon message *"people who bought this book also bought…"* This information is collected automatically by the system and it associates content based on tracking user behaviors upon the content. In this function, examples of usage metadata might be user ratings, downloads, data sharing and data links to the content.

Moreover, based on the same source, metadata can play an important role in the content management. Information stored in metadata can specify a set of allowed permissions to the object, conditions that must be fulfilled for the permissions to be valid and even time constraints with the respect to the duration that this access will be allowed. For this function administrative and structural metadata capture information such as version number, archiving date, security rules, file format and many others. In this way, it becomes possible, for example, to make objects available for the time that a project is active or according to the step of an ongoing workflow. Hence, metadata could be used not only as a tracking mechanism but also as a control mechanism in the context of ECM systems. This control mechanism will be supplementary to the access control based on the roles of the system, increasing its granularity and its flexibility.

The Rubik's cube model incorporates the main RBAC concept with the extensions described in the previous section into a new access control model tailor-made for Enterprise Content Management systems. The model is described using the Rubik's cube as a metaphor to visualize the functionality and the flexibility in assigning permissions to users under dynamic conditions.

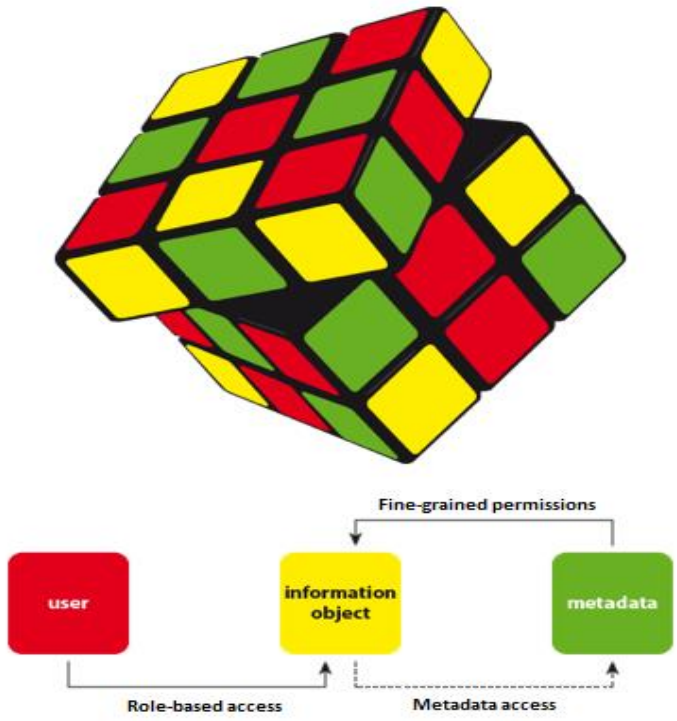

Figure 9. The Rubik's cube model

In *figure 9* we can see a graphic representation of the cube with its three dimensions. The three-dimensionality of the Rubik's cube and the multiplicity of the combinations between the colors of the cube as we twist around the dimensions led to the adoption of the Rubik's Cube as a means to describe this access control mechanism. At the basis of the model, users are assigned to preconfigured roles. Every role is determined with a set of permissions to access the objects of the classes. A role assigned to a Human Resources employee may allow the user to read the content of the objects in the HR class, but a role assigned to a Human Resource manager may be augmented with edit rights over the contents. Permissions of the employee role can be inherited to the role of the manager. Roles defined in the system with a small set of privileges, offer more flexibility to the administrator of the system as they are more specific on their allowed set of actions and they can be easily managed in order to avoid conflicts between roles.

One of the main administrative responsibilities when assigning roles to users is to ensure that a user is not granted the ability to incarnate two or more conflicting roles. During the configuration of roles, certain constraints must be applied in order to ensure the fulfillment of security policies in the organization. The UML Class Diagram of *figure 10* based on the research of Ahn et al. [43] shows an overview of how constraints fit to the rest modules of our model. In addition, RBAC is considered to be policy neutral as organizational security policies are included in the permissions specified for each role. A security policy in this case may act like a restriction on the permissions allowed to this role. What is important though is to ensure that there is no conflict between policies and between roles and policies in the extension of RBAC.

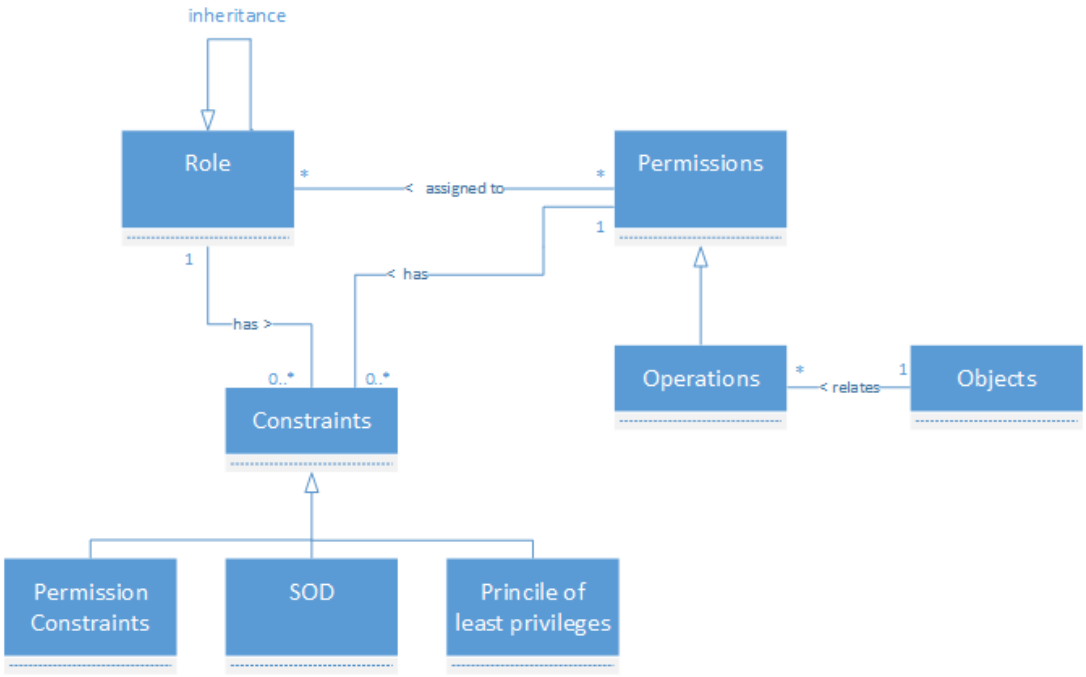

Figure 10. Roles and Constraints class diagram

An example of constraints between permissions is when employee who creates a purchase order cannot be the one who also approves it. To enable this functionality we must create sets of non-combined roles, a security mechanism that is known as the Separation of Duty. Separation of duty means that if a user is assigned to one role, this user must be prohibited from being a member of a second role that belongs to a predetermined set of conflicting roles. In the same example, one security policy in an enterprise organization states that the cash handling activities must be separated from the record keeping activities in order to reduce the likelihood of fraud. Compliance to that policy is reflected in our model by prohibiting a specific user to be assigned the role of the purchases manager and the role of the accounts payable manager simultaneously and during the same session. In a more straightforward example, a user cannot be assigned the role of the sales assistant and the role of the sales manager during the same session. Using the UML object diagram and based on the approach of Ray et al. in [44] we visualized these constraints and present them in *figure 11*.

Another important security mechanism that allows us to enable security policies is the principle of least privileges. At its core, this principle limits the number of data objects that user has access to by ensuring that the users can only access only what is necessary when it is necessary in order to perform their job better and in a safe manner in the system. A user with a marketing employee role for instance, does not need access to objects containing payroll records. To achieve a flawless implementation of the principle a deep analysis of the system is needed involving all the roles and all the objects that this contains. A review process is also necessary to ensure from time to time that the business requirements are still met and the security policies are still enabled [41].

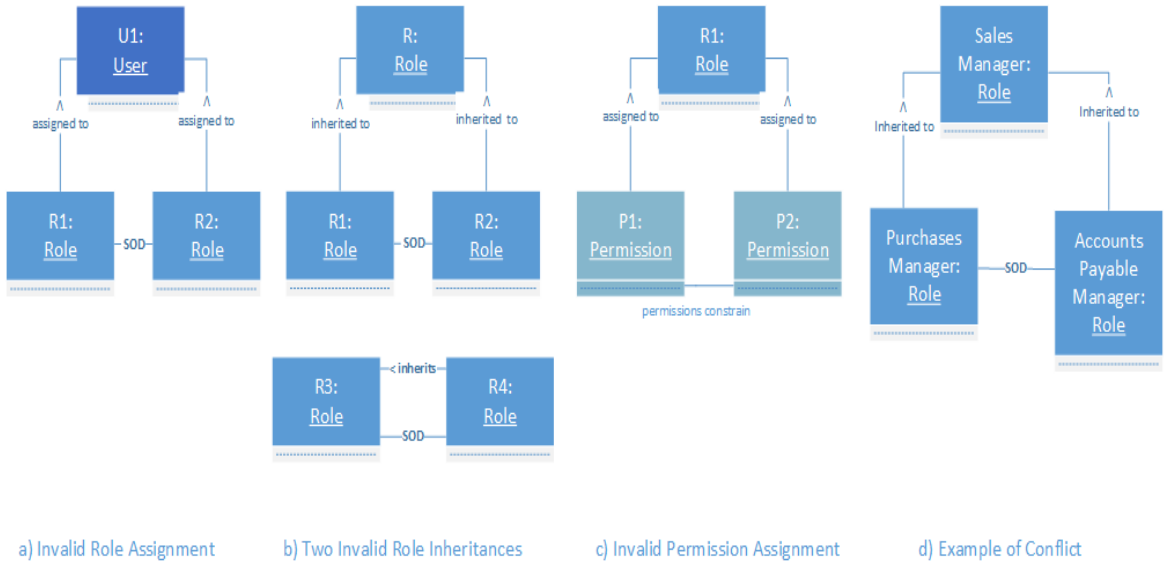

Figure 11. The Separation of Duty object diagram

The metadata is the third dimension of the cube that adds flexibility to the model and makes the access control dynamic as permissions can be fine-grained at the object level. Initially permissions are set at the role-object level. In this way the object types are classified based on the functions of the organization and the access power of each role is defined accordingly. When objects are created, the information from the metadata of the object can be used to fine-grain the permissions assigned to roles over the accessibility of the object. In this way we can extend the access over the object, we can deny access to roles or users or even define unique access rights for the specific object.

A single cube from the whole cubic structure combines all three dimensions and is a realization of the active permissions of a user with a specific role over an object at a given time. For example, if an object is an HR contract of an employee in an enterprise organization, then normally this object can be accessible by roles related to the HR functions. This standard access can be fine-grained though by metadata values of this object such as the name of the employee. Using the name of the employee we can match this information object with the user that has this name and then dynamically assign rights only to this user even if the roles that this user has do not allow access to HR related objects. By following this dynamic process iteratively, we can use the metadata of every object to specify permissions that apply uniquely for an information object. This process is depicted in *figure 12* which visualizes the conceptual model behind the Rubik's cube.

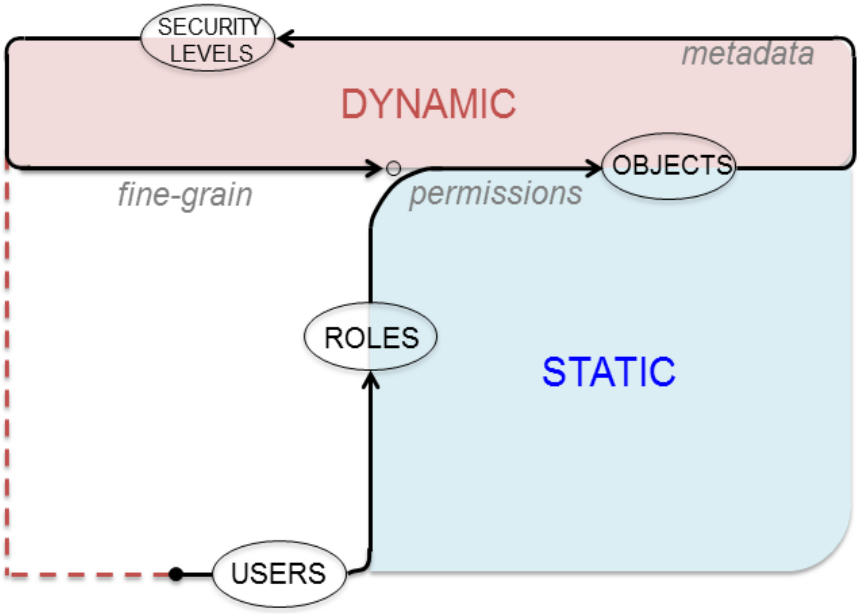

Figure 12. The concept behind the Rubik's cube

As we can see also in the conceptual model, there are two levels of implementation of access control. The first is the static level, where specified permissions are authorizing roles to access objects in the ECM system. A user that activates a session with the ECM application is assigned a set of roles that deliver a static set of permissions. Then, based on the metadata of the objects the dynamic part of the model is enabled and fine-grains the permissions at the object level. Based on the information stored in the metadata we can apply security rules that deny or extend the access that the roles of the user allow as we are going to see in the user story.

In order to simplify this mechanism and allow better administration management we introduced an additional mechanism that allows some users with administrative rights to select the security level of an object. This decentralizes the administration management but in a small scale of decentralization it offers simplified administration management. The way security levels are used is explained in the following paragraph.

Security Levels

The security levels allow the administrators to select some objects in a system where no metadata permissions will apply. In this case they can indicate another object from which the access rights will be inherited. This object can be characterized as object container and the object that inherits the rights will be called internal-object. We name this security level *Internal*. In this way we extend the essence of inheritance in RBAC model so that it describes also permissions inheritance at the object level and not only between roles. The users can also choose to make an object accessible only by them or shareable between a small number of users. In this case the applied security level is called *Private*. In any case, if there are no metadata permissions defined at the object level, the access to the object will be defined by the roles of the system. In terms of consistency we will call this security

level *Public*. To obtain a better understanding of how these mechanisms are implemented, we created a set of rules which are presented in the Appendix C. Those rules allow a better realization of the model.

Moreover, along with the security levels, a second administration tool is proposed that allows the administrators to obtain an overview of the access control structure in the system.

Access Rights Overview

Since our model allows high granularity and flexibility on the way access control is determined, it is critical for every ECM system that adopts a mechanism like this to provide also an access rights overview solution that depicts the active rights at a given moment in the system. This is mainly an administration tool that allows the administrator to view at any given moment the active rights of a specific object and also the active rights of a user.

4.3. Role Engineering

Role engineering is the process of defining roles and related information, such as permissions, constraints and hierarchies as they pertain to the use of systems, applications, processes and others [42]. Organizations must invest time to define roles in sufficient detail. High-level roles do not reflect actual organizational functions. Permissions mapped to high-level roles are usually generic in nature. This results in systems not delivering the expected business value.

The importance of roles is critical to our model. The bedrock of our access control model is the RBAC model and therefore well-defined roles serve a series of benefits. Roles defined in our ECM system serve as the ideal mapping between the business roles (i.e. job functions) and the permissions they need to have in the system. This mapping between the business roles and system roles with specified permissions makes administration management easier since permissions are assigned to roles and then roles can be assigned to users. In this way we can have flexibility when changes occur on the business side inside the organization as it is the case when a user changes a job. Well-defined system roles that reflect with precision the business tasks of a business role are very important for the sustainability and the operability of the ECM system.

The process of defining roles should be based on the analysis of how the organization functions and should include input from a wide range of users including business line managers and human resources. Following a top down approach roles are defined based on responsibilities of a given job function. This is a typical business driven approach where strong alignment between business and IT objectives is achieved. The steps for such an approach are as follows:

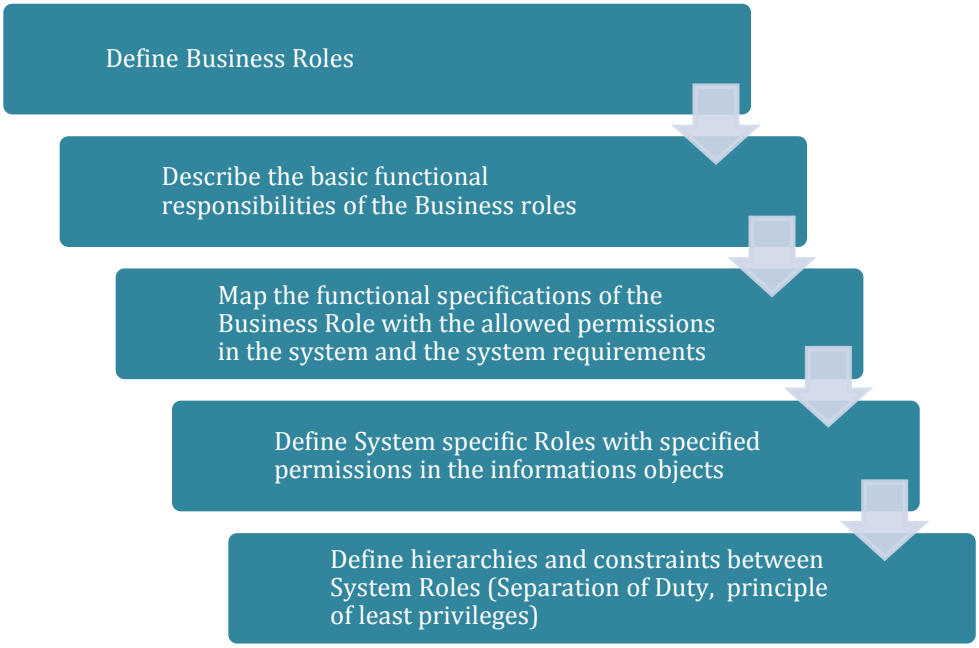

Figure 13. Top-Down Role Engineering process

The System Roles in this case are system specific and it is responsibility of the business and IT alignment process to make a mapping between those two role types. A role domain engineer is then responsible to organize the permissions sets, construct the roles hierarchies and specify the constraints in the system. Finally, the system administrator will assign permissions to the roles, roles to the users and maintain the system. The process described above is top down starting from the Business Roles and ending to the System specific Roles. It is based on the role engineering approaches that are presented by Vanamali in [42].

4.4. User Story

In this part of our paper we are going to introduce a user story to describe the functionality of our Rubik's Cube model in a more specific way. In this way we can examine in practice how the Rubik's Cube access control mechanism works.

In the following user story we have included two main object containers, the company object and the project object. The user-story diagram is based on the object diagram of UML, where the underlined text shows the object type and the text before the colon is the name of the instantiated object. In ECM software like FileLinx, a user can create instances of an object type. In this user story for instance, the underlined values (like the Company) determine an object type (or just object), while the value before the colon indicate an instance (also called item of the ECM system). Additionally, we used the role engineering process that we described in the previous paragraphs in order to define a set of illustrative roles to use in our scenario.

There are 8 users, 8 business roles and 13 system roles in this use case scenario. Each user has a unique business role and each business role is mapped to at least one system role. Every system role is assigned a specific set of permissions, which allow certain access to the information objects. In total, there are 20 objects distributed in three different classes. The allowed operations on this objects for each user is determined by the roles assigned. The static instance of the right structure is referred to as a *Public* security level, where the access control is only specified by the roles. Moving to the dynamic aspect of the model, every object of the use case scenario appears with a selected security level. This security level has an effect over the active rights of the specific object. In the *Table 4* following, we present the basic components of the user story which in turn is depicted in *figure 14*.

Users	Business Roles	System Roles
Frank	Sales Director	**Sales Manager** **Project Controller** **Workflow Controller**
James	Account Manager	Sales Manager **Project Manager** **Workflow Manager**
Jan	Sales Assistant	**Sales Support** **Mobile Role**
Anna	HR Manager	**HR Manager** Project Manager Workflow Manager
Sandra	HR Assistant	**HR Assistant**
Eric	System Support	**Role Engineer** **Test Role**
Paul	IT specialist	**Administrator**
Jane	Partner	**Guest Role**

Classes	Objects
Sales	Company, Contact person, Sales contract, Order, Invoice, Expenses
HR	Project, Project document, Project activity, Workflow, Workflow step, Role, Document, Idea, Email
General	Employee, HR contract, Timesheet, Absence request, Time booking

Table 4. Modules of the User Story

In this case there is one user with an Administrator role. The system administrator aggregates *Full-Rights* by default over the information objects. There also some more roles with a practical significance in an ECM system. The Role Engineer will be responsible for the configuration of the roles, the permissions and the constraints that are applied due to the security policies of the organization. It is the user that will execute the role engineering process in order to specify the system roles. The Test Role accompanies the Role Engineer as it allows the testing of the rights that a system role will eventually have in the system. Hence, a Test Role would have variable permissions. We also included a Guest Role in the user story that is used in case we need to allow temporary access to an external user. In this

case a partner or a client could obtain restricted access to information in the ECM system for a specified time.

The rest of the roles are oriented towards serving the tasks that the business roles need to execute in the ECM system. The HR Manager will have full access over the information objects related to the HR function while the HR Assistant will be restricted to viewing and editing of this information. Likewise, a Sales Support and a Sales Manager would be granted with access privileges over the information that is related to the sales function of the organization that is store in the ECM application. The aforementioned four system roles are also granted permissions to the general view. This is a view of general purpose that includes objects like documents, projects and emails. Last and especially for users that work frequently away from the office, we included in the use a Mobile Role that will be activated when a session is started from a mobile device or from a remote location. This role will be assigned restricted access to ensure information security.

The last four roles of the use are related to the general class of objects and usually supplementary to other system roles. The roles of the Project Manager and the Project Controller will be granted access to the objects that are related to the project management of the organization while the roles of the Workflow Manager and the Workflow Controller will be related to the management of the workflows that implement the business processes of the organization. The construction of the roles and of the permissions complies with the constraints applied by the Separation of Duty and principle of least privileges and the resulted rights are presented in *Table 5*.

System Roles	Role Permissions
Administrator	Full-Rights
Test Role	Permissions depend on the configured role that is tested
Guest Role	Permissions depends on the external user that will make use of this role
Mobile Role	*READ* rights to the objects of Sales Class in this case
Role Engineer	Full configuration rights (E.g. Role object)
Project Manager	R - E to Projects R - A - E - D to Project Documents and Activities
Project Controller	R - A - E - D - C to all Project related objects
Workflow Manager	R - E to Workflow and Workflow Steps
Workflow Controller	R - A - E - D - C to Workflow and Workflow Steps
Sales Support	R - E to Sales Class objects R - A - E - D - C to Documents, Emails and Idea objects
Sales Manager	R - A - E - D - C to Sales Class objects R - A - E - D - C to Documents, Emails and Idea objects
HR Assistant	R - E to HR Class objects R - A - E - D - C to Documents, Emails and Idea objects

HR Manager	R - A - E - D - C to HR Class objects R - A - E - D - C to Documents, Emails and Idea objects

Table 5. The permissions of the System Roles

However, as we already saw, a business role can combine permissions from more than one system roles. This functionality results in the following *Table 6* where the role permissions per user are assigned based on the system roles that every user aggregates. The roles were defined in such a way so that conflicts between the combined permissions are avoided when system roles are aggregated to the same user.

Users	System Roles	Permissions
Frank	Sales Manager Project Controller Workflow Controller	R-A-E-D-C to Sales Class objects R-A-E-D-C to General Class objects (except from the Role object)
James	Sales Manager Project Manager Workflow Manager	R-A-E-D-C to Sales Class objects R-A-E-D to Project Documents and Activities, R-A-E-D-C to Documents, Ideas and Emails R-E to Projects, Workflows and Workflow Steps
Jan	Sales Support Mobile Role	R-E **OR** R to Sales Class objects R-A-E-D-C to Documents, Ideas and Emails
Anna	HR Manager Project Manager Workflow Manager	R-A-E-D-C to HR Class objects R-A-E-D to Project Documents and Activities, R-A-E-D-C to Documents, Ideas and Emails R-E to Projects, Workflows and Workflow Steps
Sandra	HR Assistant	R-E to HR Class objects R-A-E-D-C to Documents, Ideas and Emails
Paul	Administrator	Full Access Rights

Table 6. The user permissions

The objects and the permissions that were mentioned before refer to the static instance of the access rights in our ECM system. Every time items are created in the system this instance will be fine-grained based on the Rubik's cube mechanism. In the following use case, the *Planning* document and the *Stage Contract* of the employee are both tagged as internal objects. In this case they both inherit their access rights from their containers, the *Sales*

Project A project and the *Orestis* employee respectively. The object containers and their internal objects will always have the same rights.

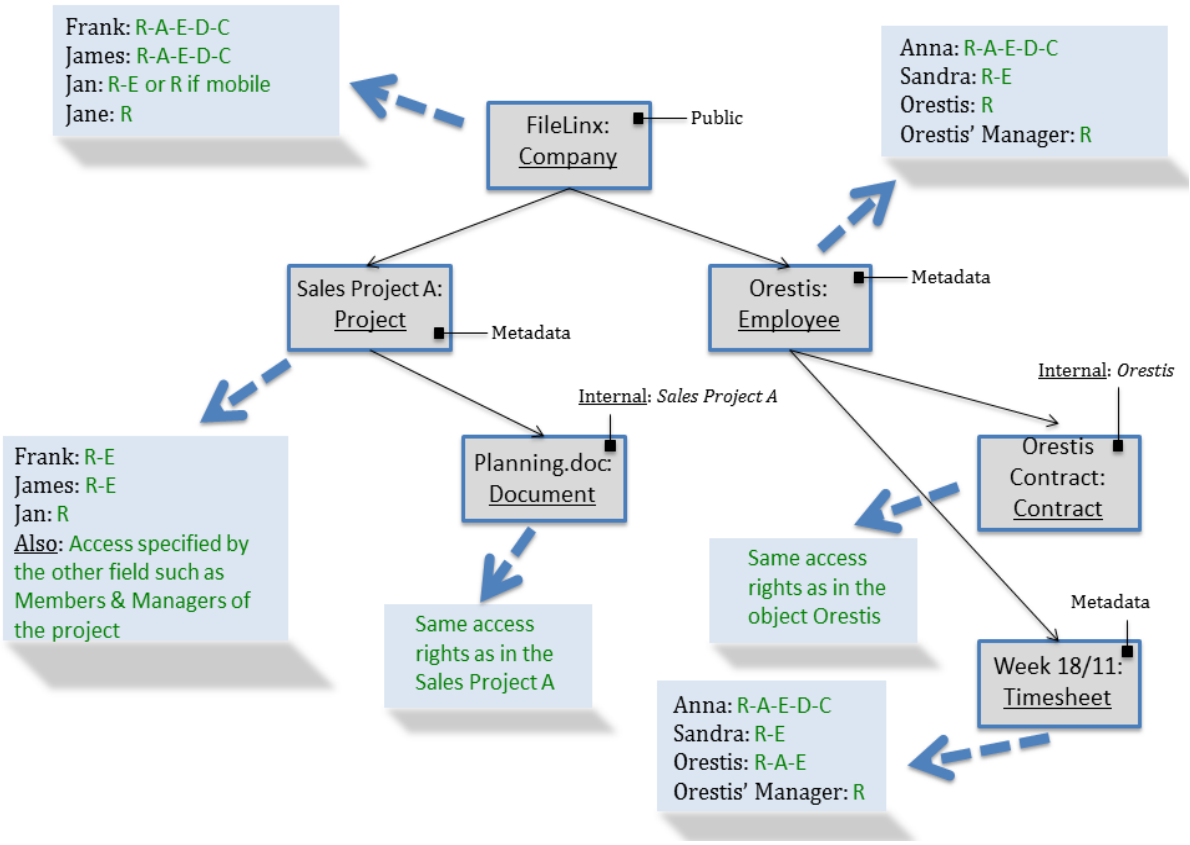

Figure 14. A use case scenario of the Rubik's cube model

The container items in this case fine-grain the role permissions applied to them using the information stored in their metadata. The *Sales Project A* item filters the access of the users who are assigned the system roles Project Manager and the Project Controller so that it allows access only to those related to the Sales function. This kind of information can be easily stored in a metadata field in the user object where the job function of the user is mentioned. Hence, the access to this item will be prohibited to the user Anna even though she is assigned the role Project Manager.

Moreover advanced access privileges are granted to the members and the manager of the project. Those are users that take part in the *Sales Project A* and the access will be allowed even if they are not granted with permissions from their assigned roles. The access to the project team members will be allowed as long as the status of the project is *Active*. The project members and the project manager, the status of the project, the start date and the expected end date are characteristic metadata information for project objects in ECM systems. The way permissions are fine-grained with the metadata is by setting up on the object level a set of access control statements like the ones presented below. When the statements are True, the fine-grain mechanism is activated.

```
While( this.Status == Active)
{
If (Username == this.Project Manager)
ALLOW: R-E-D-C
If (Username == this.Project Member)
ALLOW: R-E
If (User.Function == this.Project Type && User.Role == Project Controller)
ALLOW: R-E-D
Else
DENY access to Project Manager and Controller roles
}
```

What stands for the accessibility of the Sales Project A is exactly the same for the objects contained to the project. Hence, the access to the *Planning* document will be allowed to the same users. The employee item *Orestis* is also fine-graining the role permissions using the metadata information. In this case the role permissions are extended so that they also allow only to the employee and the manager of the employee to read the digital content of the contract. An item in the ECM system that contains information about the employees of the company will be accessible in this way by the HR personnel and also by the employee and the manager of the employee. In the same way as described before, the contract item *Orestis Contract* will share the same rights with its container. Hence, it will allow access to the HR personnel, the employee specified by the metadata and the manager of the employee.

The timesheet object is used by user in order to book their working hours in the system. The *Week 18/11* timesheet item belongs to the employee item as we can see in the use case diagram, but the rights of the item are defined by its metadata nevertheless. In the case of the timesheet item we need to extend the access so that it also allows to an employee to also edit and add new timesheets. In order to do so, the security level of the timesheet object is set to *Metadata* and not to *Internal*.

Last, the item *FileLinx* is an instance of the object company and it contains information about the company FileLinx. The access to this item is allowed only to users with a business role related to Sales and CRM functions. Although this item is related to project and employee items of the ECM object structure, the access is only affected by the roles of the system. Hence, it is possible that users who can have access to the *Sales Project A* project, will not be able to access the information item *FileLinx.* The active rights of the items presented in the use case and their security levels, are depicted in the use case diagram.

4.5 Validation

The validation of the Rubik's cube access control model will be based on checking if specific criteria are met. The first step is to see if the requirements set during the research, based on the clients' perspective and from an academic perspective, are fulfilled. The whole concept behind the Rubik's cube model was to serve dynamic environments and offer flexibility on the way access is controlled. The use of the metadata which is unique information stored in every information object ensures that the flexibility is delivered to the access control model. Metadata access rules are configured once per object type and they are realized in every item uniquely based on its metadata values. That makes our model very dynamic and flexible.

Moreover, the Rubik's cube is based on the RBAC security model which is broadly accepted as one of the most suitable models for enterprise organizations. Hence, with the basic RBAC we ensure that our model, in any organizational context it is implemented it will be representative of the organizational structure. This structure will be reflected in the model where the systems roles will be mapping the business roles and the permissions will be enabling the business tasks of the users in the system. Another requirement was set by the need to conform to internal (organizational) security policies and also to external (governmental or legal) regulations.

This requirement was addressed by the existence of the security rules that are dynamically enabled by the metadata. Those statements can be used for quality management and also for compliance purposes. Additionally, the metadata values like the category and the status of the information object can be used in management control processes to ensure these compliance. Moreover, one of the basic attributes of the RBAC is that it is considered policy-neutral as the main security policies of any organization are considerably included in the roles configuration. Hence, during the role engineering process, security policies and regulations must be taken into account. A security policy in this case may act like a restriction on the permissions allowed to a role. What makes the existence of policies also important in the security model is that a role in different organizations may have the same set of privileges except if the organizational security policies applied to the roles enable variations in each case.

The proposed model allows a small degree of decentralization of the administration management with the introduction of the security levels. To the extent that this is allowed on a wide range this may have serious consequences in the access rights managements as the model will turn to discretionary system (DAC). The proposed implementation is that in the ECM system it is possible to have more users with administrative privileges that can change the security levels of the objects. The basic goal of an access control system is that the data are protected from unauthorized access. This is our basic requirement and therefore it is the core of the Rubik's cube functionality that the users can access only what they are authorized to.

A requirement that is not met in the described model is the enhancing of secure information sharing in the ECM application environment. In practice this can be achieved in a number of ways. The focus of many ECM applications nowadays is to empower and to

simplify the collaboration between users. The information that is exchanged between the users must remain secure within the organizational environment and not be disclosed to external participants without authorization. However, the focus of this research was on the development of a mechanism that can manage the access control in these systems. Hence, this requirement was not met by the described model.

The second step of the validation process is the relation of the Rubik's cube with the academic literature and the applications in the industry. The proposed access control mechanism is based on a well-established theoretical framework whose appropriateness for the enterprise environment is highlighted in numerous studies [13]. Beyond its theoretical value the RBAC model demonstrates also hands-on significance as it is the base of many applications in information systems used by organizations nowadays. Its significance expands from ECM systems to any other security application in computer science. One of the main findings of the market research performed as the first part of this research was that many ECM vendors are implementing security mechanisms which are based on the definition of roles in the system. In some cases also, this mechanism is expanded so that it covers the whole application landscape of an organization. In this way system roles are defined on the organizational level and their permissions are specified locally on the application level. In another example, the RBAC integrates with Active Directory in Microsoft Network Domains so that the user is assigned roles from the first time that connects to the domain. Hence, the RBAC framework offers a solid base for our model that is both theoretically and practically accepted.

The last part of this validation process analysis is the feasibility of this model. As part of the functional design of the model we created a detailed description of the Rubik's cube concept and of the overall functionality. We also generated the top-down role engineering process suitable for organizational use and also a scenario and a data model for the user story description. Additionally a set of access control rules (Appendix C) was created as an implementation guide towards the development team. Although the model was not implemented in the ECM software of FileLinx, during the communication of the results the expert's opinion was asked over the feasibility of the model. A number of adjustments were introduced so that the model can be fitted to the existing software and a database model was created using the concepts of the Rubik's cube. This database model is the first hands-on proof of the feasibility of this model and can be found in Appendix D. According to the product development team of FileLinx software it is possible to implement the model in one of the new versions of the software. However, a migration roadmap needs to be constructed first to ensure that the model will be fully integrated in the system without causing consistency issues with the current information object structure. In this way a decision can be made about which release will be based on the Rubik's cube access control model.

5. Discussion and conclusion

This thesis has investigated the use of metadata in an access control model for modern Enterprise Content Management systems. The goal of the study was to develop an access control model tailor-made for ECM systems that would allow a dynamic and flexible management of the access rights of the users of the system. This study has shown that it is possible to use the information stored in the metadata of the information objects in order to determine dynamically and real time the active rights of a user. This was achieved in the model by fine-graining the rights that a user was granted based on the assigned roles with the conditions set by the metadata at the level of the information object. This continuous process allowed a dynamic combination of user, object and rights at any given time in the system.

Looking back at the four research sub-questions set at the introduction of the thesis we can now examine if a sufficient answer was provided to each of them. The study has gone some way towards enhancing our understanding of security issues in ECM system by identifying the most important features of security and also the most common threats. The most important threat based on the research by AIIM was the unauthorized access by staff. Although the literature regarding access control is significant the focus on ECM system is minor with exceptions like the research by Chiu et al. in [10] were security issues in Financial ECM systems is investigated. Our study makes a noteworthy contribution to the current literature regarding security and ECM information systems.

Moving on, flexibility was critical regarding the requirements of the access control model. While we can define flexibility based on the context that we are referring to, for the purpose of this research this was insufficient. The fact that our proposed model must be flexible was translated to a set of business and system requirements. This actually is the first part of the functional specification of the access control model. In order to determine the set of the requirements we conducted interviews with core clients of FileLinx and also supportive input was extracted from the literature review. In this way, flexibility, which was the basic property of the model, was reflected in the requirements of the system.

The main contribution of this research is the Rubik's cube model, already a nascent concept before this research began. This concept was enriched with supporting literature evidences derived from the RBAC framework developed by Ferraiolo et al. in [29].Thereafter, its implementation as an access control model applied to ECM systems was investigated. A detail description of the model including its components and its functionality was provided and a process on how to define roles was described. The last part included a detailed use case scenario visualized using the UML object diagram where the functionality of the model was described in detail.

The implications about the design of the model were also critical. Since everything started with a conceptual model, in the scope of the research it was important to decide which will be its limits. Due to the practical interest of this research, it was decided not just to generate a framework of the functional design that describes what should be included in it and to what level of detail but instead to develop one. Based on the Scrum software development

process followed by FileLinx, before the deployment of a new software a functional design is generated which in turn is converted to a technical design that includes the implementation details. Hence, it was decided for the scope of the research that the functional design would be the end point. The biggest part of the fourth chapter with the results of the research comprises the functional specification of the project which mainly includes the requirements, the functional description with the components of the model and the data model, the graphics and the use case scenario. The data model includes the basic entities of the model with their interactions and also a set of exemplary values presented in the tables of the user story. The transformation to a Technical Design was beyond of the scope of this research.

In the description of the functional design no formal framework was applied. This was left in our discretion to decide. Additionally as far as the functional specification is concerned, it seems that it is more of a practical issue than a point of academic interest. To further explain this, it is easy to find ways on how to make a functional design but there is not a standard framework that indicates what should be in the functional design and to what level of detail. In this case a simplified, hands-on approach was followed that included the description of the model, the business and the system requirements and the data model with the use case scenario. Technical implementation details were in most of the cases not included in the functional design. This was not the case in chapter four of the thesis where it was explained how we can create metadata conditions in order to enable access rules based on the metadata. For this purpose, we used a pseudo code part to describe an algorithm on how this could be achieved. A second exception was the relational database model that was included in the Appendix D as a proof of the feasibility of the concept based on the experts' opinion.

Finally, a number of limitations need to be considered in this study. While in other extends of the RBAC model (Group based RBAC [31], Task-RBAC [32],) and also the RBAC framework itself (Ferraiolo et al., 2001) a formal mathematical annotation language was used to describe the models, in the description of the Rubik's cube model no formal language was used except for the UML language in some cases. Further research needs to be done in order to establish a formal specification of the model using a standard language. Due to time constraints this study is also limited by the lack of the application of a formal validity framework. It would be interesting to assess the proposed access control model based on a standard framework or based on specific quality metrics for access control. Last, in the process of describing the model, a certain influence was exerted from the ECM software provided by FileLinx like the use case scenario where certain attributes of FileLinx were used. Despite the influence from the context provided for this research by FileLinx, it can be inferred that this model can be applied in most ECM systems in the market. This statement is based on the market analysis that was performed during this research and the common characteristics among the ECM systems.

Returning to the basic research question posed at the beginning of this study, we can now state that it is possible to generate a functional design based on the concept of Rubik's cube model which is an access control model tailor-made for ECM applications. This study sets the base of an access control prototype that can be applied to a software market of continuous growth the last couple of years. Major trends in this market like cloud and mobile

applications pose new challenges in the security of the ECM systems that could trigger further research in this model. Another extend could examine also the application of the model so that it covers a broader system landscape and removes the focus from one system only. In each case the added value would be significant not only for the industry but also for the academic world. It remains to be seen if the metadata will add a significant value in this information era by offering solutions like this to the information security and also to other aspects related with the information management.

6. References

[1] Basili V. R., Caldiera G., Rombach D. H., *The Goal Question Metric approach*, Encyclopedia of Software Engineering by John Wiley & Son, pp. 528-532, (1994).

[2] Archibald J.A., *Computer science education for majors of other disciplines*, AFIPS Joint Computer Conferences: 903-906, (May 1975).

[3] Miller W.L., Crabtree B.F., Primary care research: *A multi-method typology and qualitative roadmap*, in BF Crabtree and WL Miller (Eds.) Doing Qualitative Research (1st edition). Newbury Park, CA. Sage Publication, (1992).

[4] Smith H. A., McKeen J. D., *Developments in Practice VIII: Enterprise Content Management,* Communications of the Association for Information Systems, (2003).

[5] Gilbert M. R., Shegda K. M., Chin K., Tay G., Kruener H. K., *Gartner Magic Quadrant for Enterprise Content Management*, (October 2012).

[6] Päivärinta T., Munkvold B. E., *Enterprise Content Management: An Integrated Perspective on Information Management*, In Proceedings of the 38th Hawaii International Conference on System Sciences, Big Island, (2005).

[7] vom Brocke J., Derungs R., Herbst A., Novotny S., Alexander S., *The drivers behind enterprise content management: A process oriented perspective,* (2011).

[8] vom Brocke J., Seidel S., Simons A., *Bridging the gap between Enterprise Content Management and Creativity: A research Framework*, In proceedings of the 43d Hawai International Conference on SyStem Sciences, Kauai, (2010).

[9] McNay H. E., *Enterprise Content Management: An Overview*, In proceedings of the 2002 International Professional Communication Conference, Portland, (2002).

[10] Chiu D. K. W., Hung P. C. K., *Privacy and access control issues in Financial Enterprise Content Management*, In proceedings of the 38th Hawaii International Conference on System Sciences, Big Island, (2005).

[11] Miles D., *State of the ECM industry: How well is it meeting business needs*, AIIM www.aiim.org, (2011).

[12] Nordheim S., Päivärinta T., *Customization of Enterprise Content Management systems: An exploratory case study*, Proceedings of the 37th Hawaii International Conference on System Sciences, (2004).

[13] Fuchs L., Pernul G., Sandhu R., *Roles in information security – A survey and classification of the research area*, Elsevier Ltd., (2011).

[14] Toptenreviews.com, *Enterprise Content Management system review*, (2013).

[15] Jones D., *Information security for the modern enterprise: How safe is too safe?*

Information lock-down vs sharing and collaboration, AIIM www.aiim.eu / Opent Text www.OpenText.com, (2012).

[16] Tyrväinen P., Päivärinta T., Salminen A., Juhani L., *Characterizing the evolving research on Enterprise Content Management*, (2006).

[17] Sprague R. H., *Electronic document management: Challenges and opportunities for information systems manager*, MIS Quarterly pp. 19, 29-49, (1995).

[18] Purser S., *Why access control is difficult*, Journal Computers and Security, (2002).

[19] Hu V. C., Scarfone K., *Guidelines for Access Control system evaluation metrics,* National Institute of Standards and Technology, NISTIR 7874, (2012).

[20] Sandhu R.S., *Lattice-based access control models*, IEEE Computer, (1993).

[21] Brewer D. F. C., Nash M. J., *The Chinese Wall security policy, Proceedings IEEE Symposium on security and privacy*, 215-228, (1989).

[22] Lindqvist H., *Mandatory Access Control*, Umea University, Department of Computer Science SE 901-87, (2006).

[23] Bell D. E., LaPadula L. J., *Secure computer systems: Mathematical foundations and model*, Mitre corporation, Bedford-Massachusetts, (1975).

[24] Biba K. J., *Integrity considerations for secure computer systems*, Mitre TR-3153 Mitre Corportation, Bedford-Massachusetts, (1977).

[25] Zhang G., Parashar M., *Context-aware dynamic access control for pervasive applications*, Rutgers University, (2004).

[26] Peisert S., Bishop M., *Dynamic, flexible and optimistic access control*, (2013).

[27] Bertino E., *RBAC models-concepts and trends*, Elsevier, (2003).

[28] Feinstein H., Sandhu R., Coyne E., Youman C., *Role-based access control models*, IEEE Computer, 29(2):38-47, (1996).

[29] Gavrila S., Kuhn D. R.,Ferraiolo D. F., Sandhu R., Chandramouli R., *Proposed NIST standard for role-based access control*, ACM Transactions on information and System Security, 4(3):224-274, (2001).

[30] Bertino E., Bonatti P., Ferrari E., *TRBAC: A Temporal Role-based Access control Model,* ACM transactions on information and systems security, (2001).

[31] Li Q., Zhang X., Xu M., Wu J., *Towards secure dynamic collaborations with group-based RBAC model*, Elsevier, (2008).

[32] Park S., Oh S., *Task-role-based access control*, Elsevier, (2002).

[33] Research gate for scientist, link.

[34] Elsevier BV academic publishing, link.

[35] Science direct full-text scientific database, link.

[36] Qualitative Research guidelines project, Robert Wood Johnson Foundation, Immersion and Crystallization approach, link.

[37] Software validation definition, Wikipedia, link.

[38] Lin A., Brown R., *The application of security policy to role-base access control and the common data security architecture*, Elsevier Science BV, (2000).

[39] McCumber J., *Assessing and managing security risk in IT systems*. (2004).

[40] Simmons S., *Evaluating BPM applications*, (2013).

[41] Vogel D., *How to successfully implement the principle of least privilege*, IT Security link, (2013).

[42] Vanamali S., *Role Engineering and RBAC*, White Paper CA technologies, (February 2011).

[43] Ahn G. J., Hong S. P., Shin M. E., *Reconstructing a formal security model*, Information and Software Technology 44: 649-657, (2002).

[44] Ray I., Li N., France R., Kim D. K., *Using UML to visualize Role-Based Access constraints*, ACM 1-58113-872-5/04/0006, (June 2004).

[45] Tari Z., Chan S. W., *A role-based access control for intranet security*, (1997).

[46] O'Callaghan R., Smits M., *A Strategy Development Process for Enterprise Content Management*, In Proceedings of the 13th European Conference on Information Systems, (2005).

7. APPENDIX

7.1. APPENDIX A *The Project Breakdown Structure*

During the first phase of the research, a project break down structure was generated in order to obtain an overview of the whole project. This overview comprises the main steps of the research. Each step was further analyzed in the deliverables that should be expected and activities that are needed in order to execute the step. For every activity, an estimation of the time needed to complete was done but it is omitted here as it was part of the overall Scrum implementation framework of FileLinx BV.

A. Perform Market Analysis

Deliverable: *A document with information about the ECM market and the basic competitors.*

Activities:
- A1. Identify facts about ECM market.
- A2. Identify major trends and future potentials about ECM systems.
- A3. Identify the main drivers of ECM systems
- A4. Describe main security issues and risks.
- A5. Identify major competitors
- A6. Identify main features of ECM systems.
- A7. Identify main features of the access control system.
- A8. Find strong points among competitors**.**
- A9. Find weak points among competitors.

B. Make an analysis of FileLinx software

Deliverable: *A detailed problem description.*

Activities:
- B1. Acquaint with the system:
  - B1.1. Experience FileLinx Platform to understand how the application works.
  - B1.2. Collect Information about the access-right system.
- B2. Describe current situation:
  - B2.1. Identify KPI's of the current access-rights system.
  - B2.2. Identify main features.
  - B2.3. Find strong points.
  - B2.4. Find weak points.
- B3. Realize the structure.
- B4. Describe main functionalities.

`

C. Benchmarking

Deliverable: *A benchmarking grid that compares all the basic features that ECM systems optimally must have based on the market research with the features that the studied ECM systems have.*

Activities:
C1. Identify the main features that ECM systems are expected to have.
C2. Examine per case if these features are implemented.

D. Explore related Literature

Deliverable: *A set of papers/articles/magazines/books that is appropriate for our study.*

Activities:
D1. Identify a number of sources that allow access to papers, articles, books, magazines and books.
D2. Specify the scientific domain and the research area of the study.
D3. Define the keywords that will be used for the literature search.
D4. Use these keywords to create a set of papers/books/ articles to be studied.
D5. Define inclusion and exclusion criteria:
- D5.1. Create a set of statements to be used as the criteria to include a paper in the study list.
- D5.2 Create a set of statements to be used as the criteria which exclude a paper from the study list.

D6. Apply these criteria to narrow the set from D4 down to the set to be studied:
- D6.1. Use document skimming technique to the abstract, the introduction and the conclusion of the document and the document scanning technique for the rest of the document to compare with these criteria.

E. Write Research Proposal

Deliverable: *A document with the complete research proposal for my thesis.*

Activities:
E1. Follow the research proposal form to conduct the proposal document:
- E1.1. Draft the main research question.
- E1.2. Mention what the literature have already contributed to the study.

E1.3. Explain what will be the contribution of this study to the work that already exists.
E1.4. Explain what is the intended result of the project.

F. Explore the related Literature

Deliverable: A conceptual framework of the research

Activities: F1: Study the literature.
F2: Extract the related theory to be the basic of the conceptual Rubik's cube model.

G. Create Questionnaires

Deliverable: *A document with a list of main questions and sub-questions targeting FileLinx users.*

Activities: G1. Identify the target audience of the interviews.
G2. Create a number of thematic categories that should be covered during the interview.
G3. For each type of interviewee create a list of question to be included in these categories.

H. Perform Interviews

Deliverable: *A document with the transcribed interviews and hash-tags.*

Activities: H1. Create a list of potential interviewees based on the target audience.
H2. Contact those persons to arrange an interview.
H3. Perform the interview.
H4. Transcribe the interview data.

I. Analyze Data

Deliverable: *A document with the gains and the information from the competitive benchmarking, the literature, and the transcribed interviews.*

Activities: I1. Develop concepts from the literature that lead to a better understanding of the problem.
I2. Find facts and patterns that answer the research question.
I3. Develop hypothesis and test.
I4. Find methods to implement the new concepts.

I5. Build a conceptual model from the interviews that reflects users' needs.
I6. Extract new features and functionalities modules from the benchmarking.

J. Create Functional Design

Deliverable: *A Functional Design for a new conceptual access right system.*

Activities:
J1. Describe the new functionalities
J2. Describe new requirements.
J3. Describe new properties.
J4. Describe use cases.

K. Communicate and conclude the project

Deliverable: *A presentation of the new concepts and of the new model, a feasibility check and model validation*

Activities:
L1. Examine if and which requirements are met.
L2. Validate the model
L3. Examine the feasibility of the model
L3. Wrap-up and draw the conclusions of the study.
L4. Recommend further research.

7.2. APPENDIX B *Interview Questions*

The interviews were the primary source of data of this research and the input from the interviews was mainly used to describe the business and the system requirements for the new access control model and also to obtain a better understanding of the problem from the point of view of the customers of FileLinx. The questionnaire used for the formal interviews with the customers is presented below.

Background Questions

1) What is your organization about and in which industry does it belong?
2) What is your background?
3) What is your role in the organization?
4) What are your main tasks?

Facts about ECM and FileLinx

5) How many years of experience do you have working on ECM or related software in any organization?
6) How many years of experience do you have working with FileLinx?
7) What percentage of the enterprise content is stored in FileLinx approximately?
8) Is your organization also using other applications where information is stored?
   To name a few: ERP, CRM, HR applications, Finance applications.

9) How many users does FileLinx have in your organization?
10) Is FileLinx integrated with any other application in your organization?
11) What are the main objectives of using FileLinx within your organization?
12) What are the main features and/or functionalities of FileLinx you are using?
    12a) Are they supporting adequately your main tasks of your job?

Information Security

13) How is information security ensured in your organization? Which are the main issues and risks with information security? What kinds of security techniques are applied over the enterprise information assets?
    To name a few: firewall, anti-virus/malware software, VPN security, Active Directory, audit trails, encrypted information or password encrypted objects.

    13a) Is FileLinx integrated with some of these techniques?
14) What types of information do you store and use mostly in FileLinx?
    Options for an answer: name object types, document types, emails, etc.

    14a) Are there levels of data confidentiality in your organization?
    IF Yes
    14b) Information of what security level do you store in FX?
    14c) Accordingly, what do you choose not to store in FX & why?

15) How do you classify the information inside FileLinx? Which object Views do you use to achieve that?
16) ***(To non-administrative users)***
Are you aware of your access rights in FileLinx system?
17) Does FileLinx access control structure reflect the organizational structure and the organizational roles?
18) What kind of roles can you identify in an access control system so that it better reflects the organizational structure?
19) What kind of access rights over the information stored in FileLinx do you think are important for each role?
20) Have you encountered problems in accessing the information you want?
IF Yes
21a) What kind of problems have you encountered?
21b) How did you deal with that problem?
21c) What do you think is the cause of this problem?
21) Do you feel confident on storing your own data on FileLinx, knowing that this information will be available only on those it should be?
The concern of this question is with respect to the access control mechanism of FileLinx.

22) What would you change in FileLinx access control mechanism? What kind of characteristics and functionalities would you add to make it more secure? Examples?

Closing

23) Would you like to provide any other information related to the topic of the interview?
24) Do you have any remarks to add over the interview or any follow-up questions?

7.3. APPENDIX C *Access Control Rules*

As a way to better realize the proposed model we created a set of rules that give directions on how the specific access control model should be implemented. Those access control rules allow also a better understanding of the concepts contained in the model

- Objects may belong to more than one Classes.

- If an object belongs to more than one Classes then it is also transferable from one Class to another. In this case it should be specified in which Class it will be included. This can be dependent on the access rights of the user-owner of the object.

- Classes may be also configured to allow specific access to the objects that belong to the Class. This kind of access rights will be active only for the objects that are set as internal to the Class. This can be achieved by the metadata of the Class.

- Roles and metadata permissions should be specified at the object configuration phase of the system and not during the creation of an object. This ensures that organization wide policies are applied at any time in the system. Roles may be related to Classes with respect to their access privileges.

- Permissions will be assigned to roles in order to reflect their business tasks on the access to information inside the system. Hence, every role will have a pre-defined set of permissions that allows specific access to objects and by-default this access is applicable to every new object created. This is the static representation of access control in our model.

- In addition there are going to be 2 security levels next to the existence of roles in the system. Those security levels will operate as a mechanism, which will fine-grain the permissions set at the role level allowing a dynamic access control of each object created.

- The Configuration rights allow the users to select one of these 2 security levels when they create a new object.

- According to the power of the role assigned to a user, the user may be offered as an option all or none of these security levels to select from.

- The security levels are: *internal* and *private*.

- In order to make the system more responsive and better assist the user to select a security level, at the stage of the creation of the object, the user (according to his/her role) will be prompted to select between these security levels the one that will be the effective fine-graining mechanism. By doing so he/she will be previewed the results of this selection on the access allowed to the object.

- If the security level selected is **private** then the user-owner of the object is the only one that has access to it. Additionally, the user may select from a prompted box (if necessary) who else except from him/her has rights over the object.

	R	E	D	C	…
User A	+	+	-	-	
Role X	+	-	-	-	
…					

- If the security level selected is **internal** then the user will be asked to select a container (IF ANY).

- A container may be another object, like a project object or a company object, or may be a class of the system. In this case the active rights of the container are inherited to the contained object, which is thus classified as internal.

- If no security object is set and at the object level there are specified metadata permissions then then the fine-grain mechanism set to be the **metadata** properties of the object.

- Using the property fields of the metadata of the object, during the configuration phase, specific access statements are formulated for the object and they are activated when the conditions set in the statements are fulfilled. Those statements are determined under the schema:

[*PROPERTY_FIELD*] <**operator**> [*VALUE*] <**allow** OR **deny**> [*PERMISSION (SET*)]

- It is important for the security mechanism to be able to fine-grain the permissions which are set at the role level. This means that in some cases we need to add permissions to an object, additionally to what is set by the roles. While in some other cases it is important to revoke permissions allowed to the roles.

- The above statement is true not only in the case that the metadata permissions come genuinely from the object itself, but also in the case that the statement is set up at the container object. The active rights that are the result of the statement are inherited to the contained object.

- In order for the mechanism to be functional then for each object there must be a list that shows users and roles (or just all the users) with access rights over the object.

7.4. APPENDIX D *The Rubik's cube Database model*

The database model presented here was developed by the development team in FileLinx as a proof of concept for the Rubik's cube access control model. In the relational database schema below we can see the basic tables of the database which correspond to the modules of the Rubik's cube and also the relationships between them.

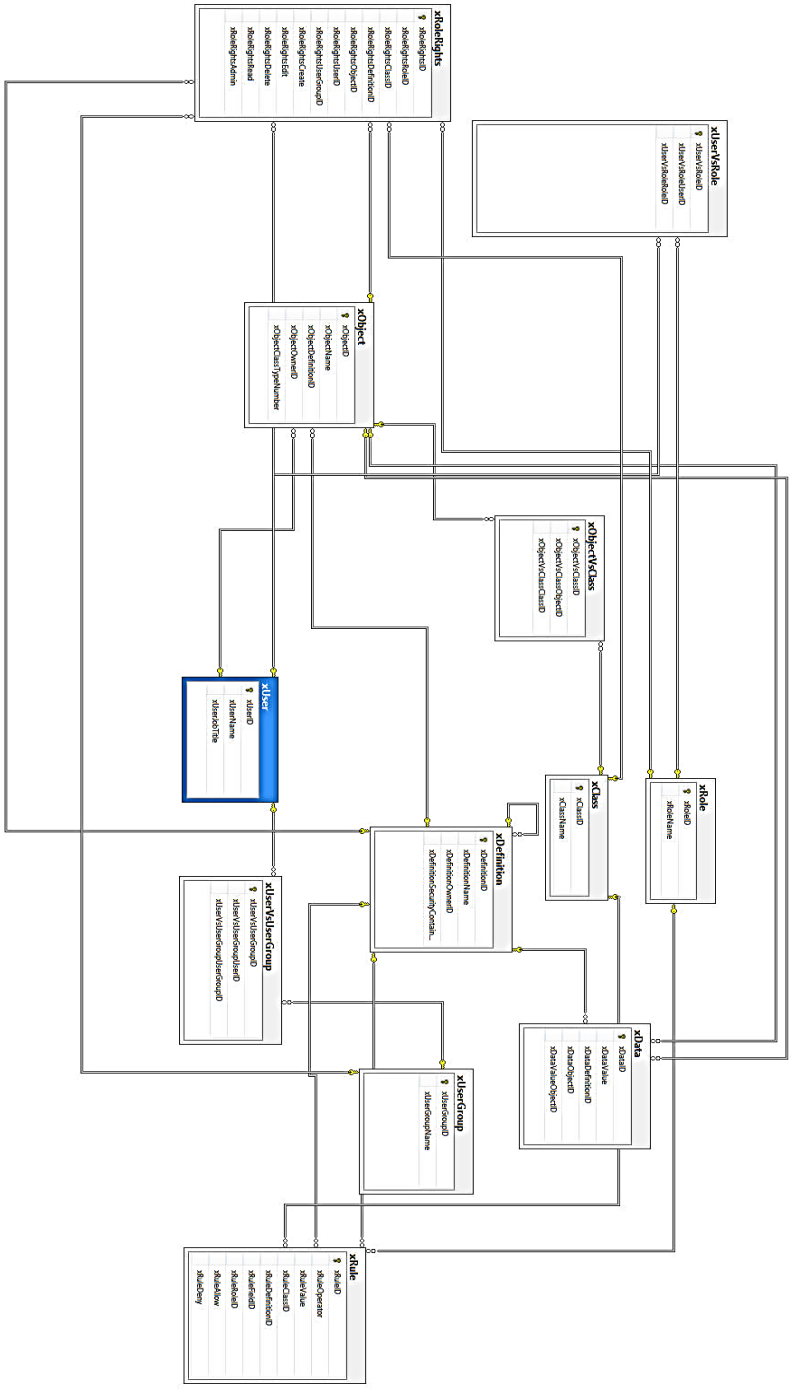

Figure 15. Relational database model