PART 1:

CHAPTER 1:
INTRODUCTION

CHAPTER 2:
ACCESS CONTROL MODELS

CHAPTER 3:
OPTIMISTIC ACCESS CONTROL

CHAPTER 4:
USAGE CONTROL

CHAPTER 1:

INTRODUCTION

1.1 Introduction

With the advent of agile programming, lightweight software processes are being favoured over the highly formalised approaches of the 80s and 90s, where the emphasis is on people, not processes (Boehm, 2002). Likewise, access control may benefit from a less prescriptive approach with an increasing reliance on users to behave ethically. These ideals correlate with optimistic access controls. However, such controls alone may not be enough to ensure that users behave in a trustworthy manner. This research presents a model for enhancing optimistic access controls with usage control to ensure that users conduct themselves in a trustworthy manner. Usage control enables finer-grained control over the usage of digital objects than do traditional access control policies and models, as trust management concerns are also considered. It has become evident that the means by which software is designed and implemented can have a significant impact on software security (Devanbu and Stubblebine, 2000). The aspect-oriented paradigm can facilitate the implementation of additional security features to legacy systems without modifying existing code. This study therefore evaluates the aspect-oriented approach in terms of implementing security concerns such as usage control.

It is evidently difficult to implement access control and often in dynamic environments preconfigured access control policies may change dramatically depending on the context. Often in unpredicted circumstances users that are denied access could have prevented a catastrophe had they been allowed access. Consider as an example, a nurse – at a hospital that has been isolated during a tornado – who needs access to a patient's records but cannot access them as nurses are not authorised to access this information (Povey, 1999).

In this extreme case, it is possible that the patient's health and safety may be unnecessarily comprised due to the restrictions imposed by the access control system. The costs of implementing and maintaining complex preconfigured access control policies sometimes far outweigh the benefits. Optimistic access controls are retrospective and allow users to exceed their normal privileges. However, if a user accesses information unethically, the consequences could be disastrous. Hence this research proposes that optimistic access control be enhanced with some form of usage control that may prevent the user from engaging in risky behaviour.

Sandhu and Park (2003) who recognised the inadequacy of traditional access control models, proposed a new approach to access control called Usage Control (UCON). This model encompasses emerging applications such as trust management, in a unified framework. They claim that the missing components of traditional access control are the concepts of *obligations* and *conditions*. *Obligations* require some action by the subject so as to gain or sustain access, e.g. by clicking the ACCEPT button on a licence agreement. *Conditions* represent system-oriented factors such as time-of-day, where subjects are allowed access only within a specific time period. A family of models for usage control exists, involving pre-authorisation and ongoing authorisations.

The openness and flexibility of the optimistic access control approach has limited applicability. Hence this study investigates contextualising this approach within a mixed-initiative access control framework. According to Dewan et al. (2007), the mixed initiative access control approach is a means to resolve situations where users may wish to provide different controls for different objects or where users wish to have preferences in terms of their privacy settings. Such a control framework involves combing a minimum of two access control models where the request for information then is mediated by a mixture of access policy enforcement agents. In order for this type of integration to be successful, a software development approach was considered that would allow for the seamless augmentation of traditional access control with optimistic access control enhanced with usage control. Such an approach was found to be the aspect-oriented approach. The aspect-oriented paradigm can facilitate the implementation of additional security features to legacy systems without

modifying existing code. Consequently this study evaluates the aspect-oriented approach in terms of implementing security concerns.

Security is often extracted as a separable concern, due to its orthogonal nature in respect of the functional requirements of a system. Hence the separation-of-concerns principle of the aspect-oriented paradigm is well suited to addressing security concerns (Robinson et al., 2004). Aspect orientation has the potential to enhance the implementation of security concerns in terms of reusability and extensibility, thereby improving the robustness and maintainability of a system. Evidently, abstracting a security feature into a security aspect increases the possibility that it may be reused for other applications (Padayachee and Wakaba, 2007). Access control and encryption, for example, have similar requirements for most applications. Vanhaute and De Win (2001) demonstrated how to convert these security concerns into reusable generic aspects.

Several authors cite the benefits of using aspect-oriented programming for implementing security concerns (De Win, Vanhaute et al., 2002; Viega et al., 2001). According to Bodkin (2004), aspect-oriented programming is relevant for all major pillars of security: 'authentication, access control, integrity, non-repudiation, as well as for supporting the administration and monitoring disciplines required for effective security'. Even security-related bugs such as buffer overflows or race conditions can be considered security-related concerns (De Win et al., 2003). Security aspects can be used to modularise access control and authentication (see (De Win et al., 2003); (Shah and Hill, 2003) and (Slowikowski and Zielinski, 2003)). The primary argument supporting aspect-oriented programming is that the average programmer does not have the requisite skills in security (Viega et al., 2001). This can be attributed to a lack of expertise and few tertiary institutions offering tuition in software security (Viega and Evans, 2000). Programming tasks such as authentication, access control and integrity should be abstracted away from developers and allocated to security experts. Secondly, it is observed that security concerns such as encryption and access control tend to crosscut the code base. Thirdly, a security aspect can be reused for other applications since access control has the same requirements for most applications (De Win et al., 2001). Fourthly, aspect-oriented software design is flexible enough to

accommodate the implementation of additional security features after the functional system has been developed.

This study proposes using the aspect-oriented paradigm to facilitate the non-intrusive insertion of access control features such as usage control into a fully operational software system. (This is validated by a proof-of-concept prototype that will be presented in Chapter 8.) Chapters 2, 3 and 4 explore traditional access control models, usage control and present the concept of optimistic access control respectively. Chapters 5 and 6 discuss the concepts of aspect-oriented programming and their relationship with security. Chapters 7 and 8 present the model itself and evaluates the model concept. Chapter 9 concludes the thesis by assessing the model presented and the implementation technique applied.

1.2 Motivation for this study

Discretionary access control is an access policy that restricts access to files and other system objects such as directories and devices based on the identity of the users and/or the groups to which they belong (Russell and Gangemi, 1991). In the case of discretionary access control, no control is enforced on the use or dissemination of the information once this information has been released to an authorised user (Pfleeger, 1997). For example, a subject *Jane* may at her own discretion decide whether *Sam* may read the file entitled *Logistics*, assuming she owns the file. Discretionary access control is very flexible but highly vulnerable to Trojan Horses. As a result of this inadequacy, mandatory access policies were proposed.

Mandatory access control (Ramachandran et al., 2006) refers to access control policy decisions that are made beyond the control of the individual owner of the object. A central authority determines what information is to be accessible by whom, and the user cannot change access rights (Pfleeger, 1997). With mandatory access policies, every object and user in the system is assigned a sensitivity label that consists of a level of secrecy and a set of compartments (Bell and La Padula, 1976). Mandatory access control is deemed to be superior to discretionary access control as it is not vulnerable to illegal information flows. An

illegal flow arises when information is transmitted from one object to another in violation of the information flow security policy (Samarati et al., 1997). Even the most dominant model of recent times, the role-based access control model, is vulnerable to illegal information flows, as is demonstrated by Chon et al. (2004). Within role-based access control (RBAC), system administrators create roles according to the job functions performed in a company or organisation, grant permissions (access authorisation) to those roles, and then assign users to the roles on the basis of their specific job responsibilities and qualifications (Sandhu et al., 1996).

These models often assume that users want and are able to determine permissions before the actual access is made. These mechanisms require a priori setting of permissions that are difficult to specify and maintain in highly dynamic environments. In this thesis, this category of access controls is referred to as *traditional access controls*. These types of access controls are characteristically pessimistic. In other words, the models assume that human beings cannot behave in a trustworthy manner and the system has to prevent them from behaving in an undesirable way. Human trust is subjective and context specific and hence it is difficult to form a definition that incorporates all views and types of trusts (Grandison, 2003). Integrating trust/distrust into the computing world requires transforming a complex social concept into an easy-to-use technical product that embodies the basic principles of trust/distrust (English et al., 2002). Human beings make decisions based on the circumstances of a particular situation. For example, within a typical mandatory access control model, *doctors* may have the privilege to view sensitive information but *nurses* and *clerks* would not. In the case of role-based access control, the role could be based on job responsibilities; for instance, a patient's record can be written by any health professional assigned to the role of ward physician (Pudney, 2003). However, this does not guarantee that a valid user demonstrates integrity or acts professionally.

Access controls are difficult to implement and are evidently deficient under certain conditions. Traditional access controls offer no protection for unclassified information – such as a telephone list of employees that is unrestricted, yet available only to members of the company. On the opposing side of the continuum, organisations such as hospitals that

manage highly sensitive information demand stricter access control measures. Yet, traditional access control may well have inadvertent consequences in such a context. Often, in unpredictable circumstances, users that are denied access could have prevented a calamity had they been allowed access. It has been proposed that controls such as auditing and accountability policies be enforced to deter rather than prevent unauthorised usage. In dynamic environments, preconfigured access control policies may change dramatically, depending on the context. Moreover, the costs of implementing and maintaining complex preconfigured access control policies sometimes far outweigh the benefits. This research considers an adaptation of usage control as a proactive means of deterrence control to protect information that cannot be adequately or reasonably protected by access control. Deterrent controls are intended to discourage individuals from intentionally violating information security policies or procedures. Hence, if software systems could trust humans to decide how and when they can access information, this would be a more accurate assessment of trust. Trust on a humanistic level is highly complex and there are a variety of factors that influence trust. The emergence of trust-based access control frameworks is largely due to communications occurring among parties where each party is unknown. This communication is typically decentralised. There is now a need for a new type of access control where the access is not preconfigured and where, essentially, the user is trusted to behave ethically.

While pessimistic access controls such as DAC, MAC and RBAC maybe highly appropriate in certain contexts, optimistic access controls may be more appropriate in other circumstances. For instance, Stevens and Wulf (2002) considered an actual inter-organisational co-operation scenario where it was found that traditional access control did not comply with the organisation's requirements and that co-operation and competitive reasons motivated the use of interactive and optimistic access controls. Hong and Landany (2004) also established that there is a need for privacy-sensitive systems to have a range of control and feedback mechanisms for building pessimistic, optimistic and mixed-initiative applications.

The approach of deterrence control is an application of optimistic access control. Optimistic access control is useful in cases where openness and availability are more important than complete confidentiality (Povey, 1999). Optimistic access control also has the advantage that it is far easier to implement, since it is rather difficult for administrators to predict all of the possible usage scenarios and thus all of the necessary permissions. Optimistic access control is based on the assumption that most access control processes will be legitimate, and relies on controls external to the system to ensure that the organisation's security policy is maintained. The scheme allows users to exceed their normal privileges in a way that is constrained, so that it is securely audited and may be rolled back (Povey, 1999). Optimistic access control involves a combination of audit and accountability; and deterrent mechanisms to encourage trustworthy behaviour. This approach is characteristically more retrospective rather proactive. However, the application of usage control within an optimistic access control context may provide a proactive means of deterrent control. Within traditional access control models, usage control would offer an extra layer of restriction against unauthorised usage. However, under the optimistic access control paradigm it would not restrict users but rather deter them from accessing and misusing information. As is defined in terms of the optimistic access control paradigm, the user must ultimately be able to access the requisite information.

Optimistic access controls trust human beings to perform legitimate accesses and take retrospective action once such trust has been breached. The initial cost of implementing optimistic access control methods is minimal; however, the fall-out could be disastrous. If such a breach is discovered, it could involve prosecution or performing a roll-back procedure. The roll-back procedure may be able to restore the system to its original state. However, it is highly likely that it may not be able to undo the damage done.

1.3 Problem Statement

Access controls are difficult to implement and maintain due to the highly complex task of envisaging all the possible usage scenarios and thus all the necessary permissions. Notwithstanding this fact, the pre-configured access controls may not be appropriate to all contexts. Furthermore, a security policy specified by a single access control model may not applicable to all data in an information security system. This research considers an adaptation of usage control as a proactive means of deterrence control to protect information that cannot be adequately or reasonably protected by access control. This thesis therefore presents a model for reformulating usage control under the optimistic access control paradigm.

To accomplish the main goal identified above, the following sub-goals were identified:

- Providing a critical overview of access controls and optimistic access controls
- Providing an overview of aspect-oriented programming and its relevance to security
- Deriving a model that enforces **O**ptimistic **A**ccess **C**ontrol with **U**sage **Con**trol – designated the OAC(UCON) model – within a mixed-initiative access control framework
- Providing a proof-of-concept prototype to demonstrate the suitability of aspect-orientation in terms of implementing the model concept
- Providing evaluative prototypes of the model concept in a small-scale experiment using the design science methodology

1.4 Terminology used in this thesis

Access control is a fundamental part of computer security where every requested access must be governed by an access policy stating who is allowed access to what; i.e. the request must be mediated by an access policy enforcement agent (Pfleeger and Pfleeger, 2003).

Aspect-Oriented Programming provides explicit language support for modularising design decisions that cross-cut a functionally decomposed program (Walker et al., 1999), i.e. the developer is able to maintain the code (cross-cutting functionality) in a modularised form.

Mixed-initiative access control framework is an access control strategy that involves combining a minimum of two access control models where the request to information is mediated by a mixture of access policy enforcement agents.

Optimistic Access Control is a scheme that allows users to exceed their normal privileges in a way that is constrained, so that it is securely audited and may be rolled back.

Usage Control (UCON) is an access control model that encompasses emerging applications such as trust management in a unified framework.

1.5 Research Methodology

The research methodology involved the design of a proof-of-concept prototype to demonstrate a subset of the model concept and to evaluate the suitability of the aspect-oriented paradigm. Additionally, the model was evaluated in terms of the design science research method so as to test its scalability and efficacy as a security measure. As the value and utility of the model concept was evaluated, the design science method was selected to this end. It involves a two-step process of building and evaluating (March and Smith, 1995). During this process several evaluative prototypes were developed to verify the model concept for commercial systems. The small-scale experiment tested the theory that users' interaction with the prototype will be perceived as an effective countermeasure against data misuse. In order to test the hypothesis, two qualitative data collections were employed during the evaluation, namely participant observation and open-ended interviewing. Postgraduate students with an extensive knowledge of information systems were utilized to develop and evaluate the model concept. Since some of these students are already employed within the information systems sector, this profile of participants can often serve as representatives of systems developers.

1.6 Delimitations

It is important to distinguish between access control and information flow control. For example, an access policy might specify that user1 can read from file1 and write to file2, while a flow policy might specify that information in file1 is at most confidential and always less than the class of information in file2 (Andrews and Reitman, 1980). The present study is primarily concerned with access control, and more specifically with enhancing access control by means of usage control. Thus it will specifically consider the reformulation of usage control under the optimistic control access control paradigm.

Although the issue of trust is an important component of the mixed-initiative access control framework, it is beyond the scope of this thesis to provide details as to how trust can be measured and maintained. The issue of mutability of access rights based on trust is given due consideration – however, it is not explored in any detail.

1.7 Thesis Layout

Chapter 1: Introduction: The problem is introduced in relation to access controls in that they are too restrictive and difficult to pre-configure.

Chapter 2: Access Control: This chapter introduces traditional access control models and presents the problem with the traditional approach to access control.

Chapter 3: Optimistic Access Control: This chapter introduces optimistic access control and presents its strengths and weaknesses. While Chapter 2 covers the more traditional methods of access control, Chapter 3 focuses on a non-conventional method of implementing access control.

Chapter 4: Usage Control: This chapter introduces usage control and how it may be used to address the weaknesses of optimistic access control.

Chapter 5: Aspect-Oriented Programming: The aspect-oriented programming paradigm is introduced as a mechanism to implement access control measures. The abstraction of this chapter allows for an adequate overview of the aspect-oriented paradigm as it is a fairly new programming technique and not yet ubiquitous within the South African context.

Chapter 6: Aspect-Oriented Security: This chapter demonstrates how the aspect-oriented paradigm may be used within the information security domain.

Chapter 7: The OAC(UCON) model: This chapter presents the model that is used to address the inadequacies of t traditional access control requirements.

Chapter 8: Prototyping and Model Evaluation: This chapter describes the implementation of the model proposed as a "proof-of-concept" by using an aspect-oriented programming language. It also provides an evaluation of the approach and the model concept using evaluative prototyping.

Chapter 9: Conclusion: This chapter concludes with directions for future research and evaluates the contribution made by this thesis.

Figure 1.1 below presents a schematic overview of the thesis.

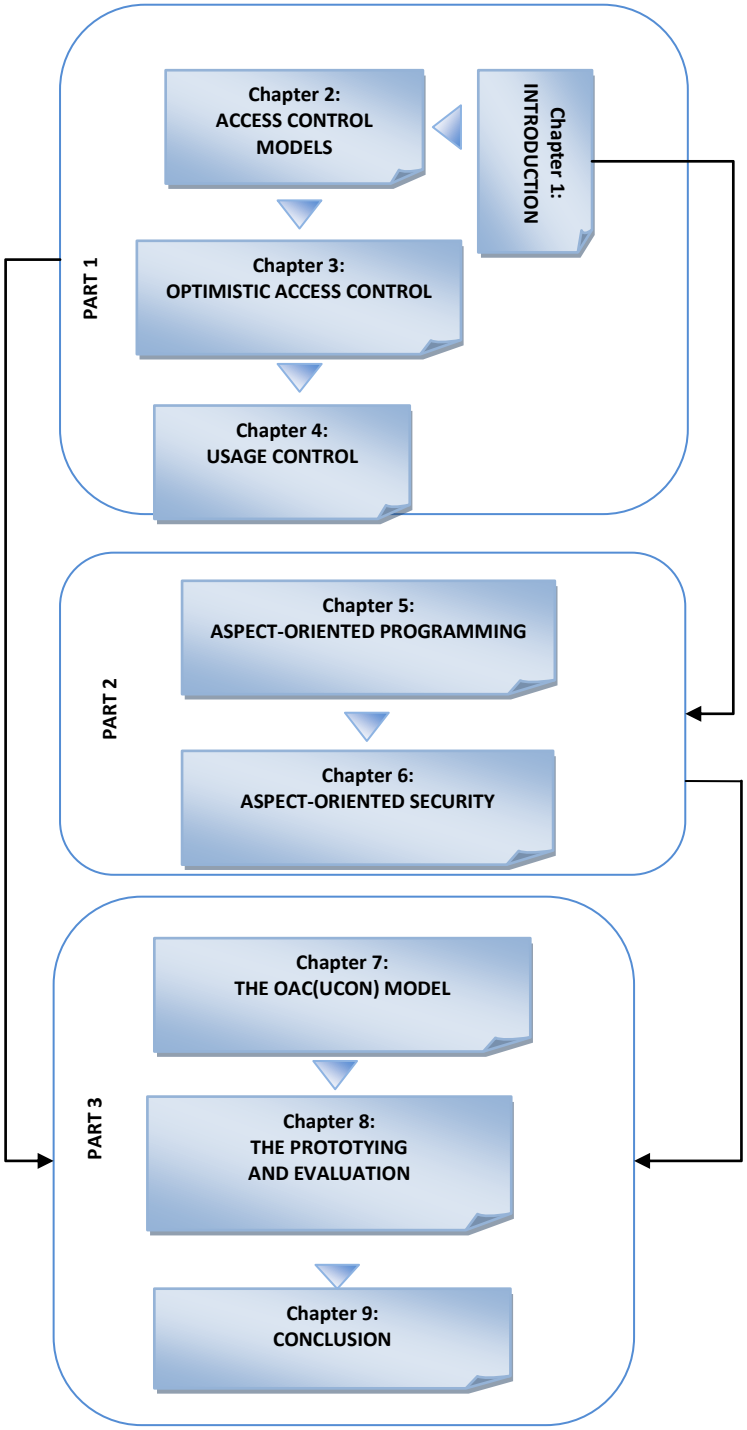

Figure 1.1: Overview of Thesis

1.8 Summary

The proposed solution offered by this study may ease the burden of system administrators significantly. It is rather difficult for administrators to predict all of the possible usage scenarios and thus all of the necessary permissions. With optimistic access control, it is ultimately left to users to make that judgement. The complexity of the implementation and maintenance of pre-configured access control policies is therefore relegated to the way the user interacts with the system. Adapting usage control as a deterrent mechanism provides a proactive mechanism that can be used in addition to the retroactive methods of auditing and accountability offered by optimistic access control. Through using a proactive means of deterrent control, a larger subset of information may be relegated into the public domain. This research does not obviate the need for traditional access control. For example, payment processing e-business applications demand stricter information controls (Haldar et al., 2005). Consequently the model presented here is intended to be incorporated into a mixed-initiative access control framework.

CHAPTER 2:

ACCESS CONTROL

2.1 Introduction

Access control is a fundamental part of computer security where every requested access must be governed by an access policy stating who is allowed access to what; the request must then be mediated by an access policy enforcement agent (Pfleeger and Pfleeger, 2003). There are three basic approaches to access control. The first approach requires pre-configured access control policies (explored in this chapter), while the second involves temporal access control policies (explored in Chapter 4). Chapter 2 provides an overview of traditional access control models such as discretionary access control (DAC), mandatory access control (MAC) and role-based access control (RBAC). The inadequacies of these models are explored, thus providing the rationale for investigating the third approach to access control – optimistic access control. This approach is characteristically retrospective.

2.2 Discretionary Access Control

Discretionary access control (DAC) is an access policy that restricts access to files and other system objects such as directories and devices based on the identity of the users and/or the groups to which they belong (Russell and Gangemi, 1991). With discretionary access control, no control is enforced on the use or dissemination of the information once this information has been released to an authorised user (Pfleeger, 1997). For example, a subject *Jane* may at her own discretion decide whether *Sam* may read the file entitled *Logistics*, assuming she owns the file (see Figure 2-1).

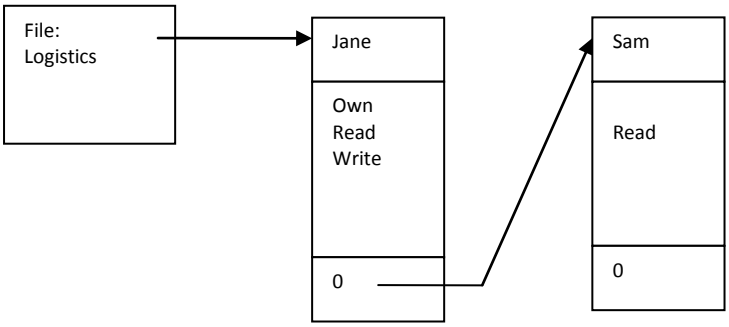

Figure 2-1: Discretionary Access Control based on an Access Control List (adapted from(Tolone et al., 2005))

According to Pieprzyk et al. (2003), there are several deficiencies with this type of enforcement:

- If a permission ά is transferred from one subject to another, then the second subject can propagate the permission ά with no agreement from the first subject.
- The read permission allows a reader to copy the object and to grant friendly subjects read access to the copy.
- If two or more untrustworthy processes conspire, they may exercise their permissions collectively.

Discretionary access control is particularly flexible, yet highly vulnerable to Trojan Horses (Downs et al., 1985). According to Li et al. (2009), this is because it is assumed that all programs are benign and will not be exploited by malicious inputs. McCollum (1990) contends that discretionary access control is inappropriate for the enforcement of an 'integrated, global access policy based on a comparison of explicit markings on data to attributes of the user seeking access', as such controls are designed to relate individual users to specific data objects. (Li et al., 2009) go on to add that the problem with discretionary access controls is that the enforcement mechanism cannot correctly identify the true origins of a request made by multiple principles. To this end, there has been an inclination to complement DAC mechanisms with some form of mandatory access control (Mao et al., (2009).

2.3 Mandatory Access Control

Mandatory access control refers to access control policy decisions that are made beyond the control of the individual owner of the object. A central authority determines what information is to be accessible by whom, and the user cannot change their access rights (Pfleeger, 1997) (see Figure 2-2). With mandatory access policies, every object and user in the system is assigned a sensitivity label that consists of a level of secrecy and a set of compartments (Bell and La Padula, 1976). For example, as depicted in Figure 2-2 below, the sensitivity label of the Logistics file is SECRET [ALPHA, VENUS], where SECRET indicates the level and [ALPHA, VENUS] the compartments. The security level is an element of a totally ordered set. The levels generally considered are: TOPSECRET, SECRET, CONFIDENTIAL and UNCLASSIFIED, where TOPSECRET > SECRET > CONFIDENTIAL > UNCLASSIFIED (Russell and Gangemi, 1991). The set of compartments is unordered. An access class c_i dominates (≥) an access class c_j if and only if the security level of c_i is greater than or equal to c_j and the compartments of c_i include that of c_j. Access control is based on the following two principles formulated by Bell and LaPadula (1976), which are adopted by all models enforcing mandatory access security policies:

- No read-up: A subject can read only those objects whose access class is dominated by the access class of the subject, namely the Simple Security Property.
- No write-down: A subject can write only to those objects whose access class dominates the access class of the subject, namely the *-Property.

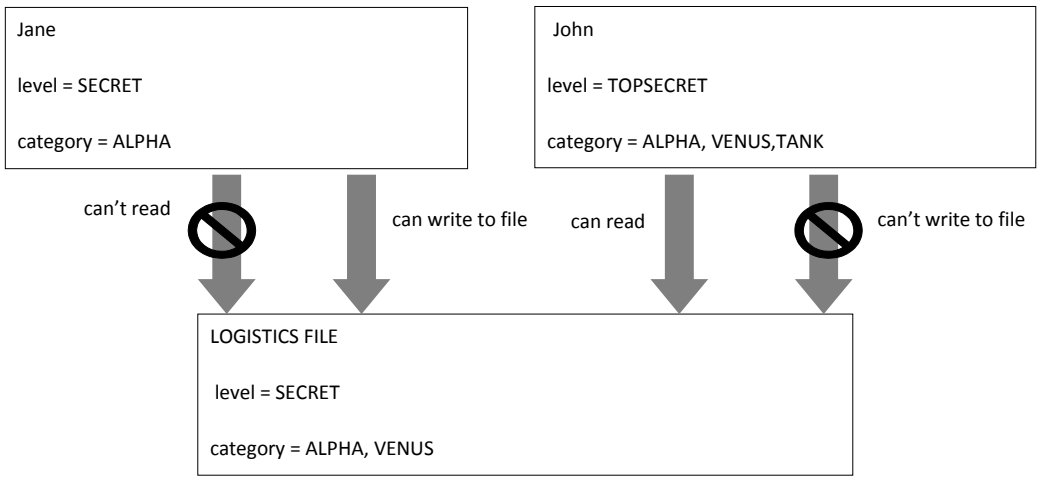

Figure 2-2: Mandatory access control (MAC) (adapted from (Russell and Gangemi, 1991))

It is important to note that the term '*object*' does not imply 'object' in the typical object-oriented sense. In fact, the term 'subject' is the active process that requests access to the 'object', which are passive entities such as files or records.

Although mandatory access control is considered to be superior to discretionary access control, it is difficult to implement in reality and the model has a number of deficiencies as indicated below (Anderson, 2001):

- 'Blind write-up' – The inability to inform low-security data whether a write to high-security data has happened correctly
- 'Downgrading'– Moving information from a high-security level to a lower level is sometimes desirable
- 'TCB bloat' – A large subset of the operating system may end up in the Trusted Computing Base (TCB)

Mandatory access control (MAC) was once thought to be relevant to the military only. These days, however, it is gradually being incorporated into commodity open operating systems such as BSD and Linux. As Zakrzewski and Haddad (2002) put it, 'mandatory access control mechanisms are efficient for supporting complex relationships between different entities in the computing environment'. Systems such as payment processing, e-business applications and medical data applications may also require similar stringent controls. Mandatory access control is highly applicable in areas such as privacy, as access to privacy-sensitive data can be regarded as analogous to access to multilevel security data (Rjaibi and Bird, 2004). Maintaining the privacy of individuals is one of the most compelling reasons for implementing strong access controls in an organisation. Weippl and Essaymr (2003) also demonstrate that besides its applicability to the military, mandatory access control has the efficacy to protect personal digital assistants (PDAs). Another reason why mandatory access control is deemed to be superior to discretionary access control, is because it is not vulnerable to illegal information flows. An illegal flow arises when information is transmitted from one object to another object in violation of the information flow security policy (Samarati et al., 1997). Even the most dominant model of recent times, the role-based

access control model, is vulnerable to illegal information flows, as demonstrated by Chon et al. (2004).

Mandatory access control can be easily unified within the role-based access framework, since role-based access control (Sandhu, 2001) is a means of articulating policy rather than embodying a particular security policy (Osborn et al., 2000). Mandatory access controls generally cannot prevent implicit flows arising from the control paths not taken at run time (Zheng and Myers, 2004). To prevent an information leak like this, Denning and Denning (1977) proposed a mechanism to certify that a program does not violate information flow policy. Information flow is concerned with the control path of information as a software system executes. There exists a semantic gap between access controls of operating systems and programming languages as languages such as the Java Virtual Machine lack mechanisms to enforce mandatory access controls, programming languages have been created (Haldar et al., 2005).

2.4 Role-based Access Control

Within role-based access control (RBAC), system administrators create roles according to the job functions performed in a company or organisation. They grant permissions (access authorisation) to those roles, and then assign users to the roles on the basis of their specific job responsibilities and qualifications (Sandhu et al., 1996).

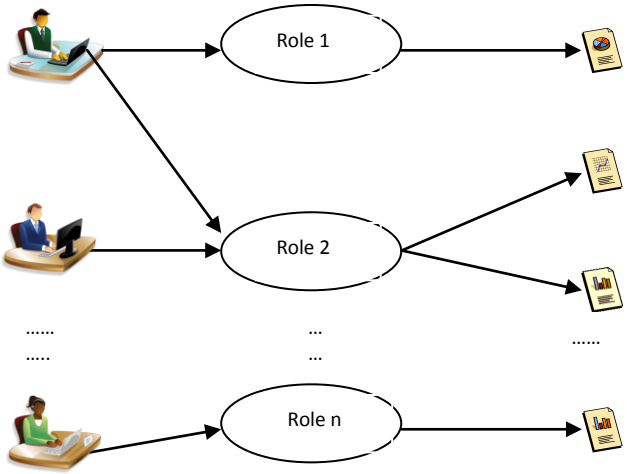

Figure 2-3: Role-based Access Control (adapted from (Samarati and de Capitani di Vimercati, 2001))

According to Tolone et al.(2005), the shortcomings of role-based access control can be summarised as follows:

- The nature of the roles is static and they lack flexibility and responsiveness to the environment in which they are being used.
- It lacks the ability to specify fine-grained control of individual users in certain roles and individual object instances.
- Constraints are an important aspect of role-based access control and a powerful mechanism for stipulating high-level organisational policy, the specification of which is not expanded on in the model.

Izaki et al. (2001) demonstrate how illegal information flow may occur among objects within the role-based access control model. As with discretionary access controls, role-based access control can only restrict the access of objects in a system – hence information flows among variables cannot be controlled (Chou, 2003).

Role-based access control does not scale up with access control model issues. The core is for the most part unchanged and based largely on the access matrix model (Zhao et al., 2007). This inflexibility is to be addressed by the model concept presented in this thesis. The aim is to provide a model that is not entirely dependent on subject-object attributes (as with the models presented in this chapter), but rather a flexible model that is dependent on temporal factors.

2.5 Conclusion

Traditional access control is based on static authorisations that depend exclusively on a subject's permissions with regard to target objects (Zhao et al., 2007). Usage control seeks to address these inadequacies as it considers other factors that may influence a subject's rights to a target object. Moreover, it considers the mutability of attributes during access. Usage control will be explored in more detail in Chapter 4. Traditional access control models are based entirely on denial of access. They do not consider contextual factors or extenuating circumstances that may warrant overriding these controls. Access control

models such as DAC, MAC and RBAC often assume what users want and are able to determine permissions before the actual access is made. They require permissions to be pre-configured, which is difficult to specify and maintain in highly dynamic environments where access policies may fluctuate on a regular basis. In the next chapter optimistic access control is considered. It is based on the assumption that most accesses will be legitimate and the control is retrospective.

CHAPTER 3:

OPTIMISTIC ACCESS CONTROL

3.1 Introduction

Industry surveys prove that a substantial share of computer security incidents are due to the intentional actions of legitimate users. – the consequences of which include negative publicity, competitive disadvantage and loss of consumer confidence (D' Arcy and Hovav, 2007). Traditional access control models are evidently deficient under certain conditions. For instance, a particular organisation may necessitate access controls to be less prescriptive for the purposes of intra-organisational cooperation (Etalle and Winsborough, 2007; Stevens and Wulf, 2002). Traditional access controls such as mandatory, discretionary or role-based access control offer no protection for information that is unclassified and freely available in the public domain. In the recruitment industry, for example, information such as client lists and candidate lists has to be shared freely for the purposes of collaborative job matching. As there are no controls over this information, an employee may well download it and distribute it to competitors.

On the opposing side of the continuum, organisations (e.g. hospitals) that manage highly sensitive information stipulate stricter access control measures. Yet traditional access controls may sometimes have an undesired effect in these circumstances as well, for instance the denial of access based on the attributes of the users rather than the context of the access. The meaningful implementation of access control remains a difficult task and preconfigured access control policies may at times change dramatically in dynamic environments, depending on the context. Moreover, the costs of implementing and maintaining complex preconfigured access control policies sometimes far outweigh their benefits. It has been proposed that auditing and accountability measures be enforced to

deter unauthorised users rather than to completely prevent them from gaining access (Etalle and Winsborough, 2007). While pessimistic access controls such as DAC, MAC and RBAC may be highly appropriate in certain contexts, optimistic access controls may be more appropriate in other circumstances. This issue is investigated in Section 3.4.

3.2 Optimistic Access Control

Optimistic access control is useful in cases where openness and availability are more important than complete confidentiality. Optimistic access control also has the advantage that it is far easier to implement, since it is difficult for database administrators to predict all of the possible usage scenarios and thus all of the necessary permissions. Optimistic access control is based on the assumption that most access control processes will be legitimate, and relies on controls external to the system to ensure that the organisation's security policy is maintained. The scheme allows users to exceed their normal privileges in a way which is constrained, so that it is securely audited and may be rolled back (Povey, 1999).

According to Povey (1999) the optimistic enforcement of security policies are retrospective and rely on administrators to detect unreasonable access and take steps to compensate for the action. Such steps might include:

- Undoing illegitimate modifications
- Taking punitive action (e.g. firing or prosecuting individuals)
- Removing privileges

Optimistic access controls trust human beings to perform legitimate accesses and take retrospective action after such trust has been breached. This approach is characteristically more retrospective rather proactive. However, the application of usage control within an optimistic access control context may provide a proactive means of deterrent control. Povey (1999) suggests using integrity to complement optimistic access control, where the user is unable to manipulate data arbitrarily. However, this thesis is not concerned with maintaining the integrity of the data – rather, the model presented here involves protecting the access of information.

3.3 Requirements for Optimistic Security

Since the seminal article on optimistic access control was written by Povey (1999), the next three sections are based largely on his work. Providing an optimistic security system requires mechanisms to ensure that the likelihood and consequences of a user maliciously using or plainly misusing the system are minimised. In order to satisfy this objective, the following controls should be considered:

- **Constrained entry points**: Users should not in general be allowed to exceed their privileges. Users should be warned when they exceed their privileges and be reminded of their obligations towards their organisation.
- **Accountability:** The system must have strong enforcement of authentication so that users are associated with their actions.
- **Audit:** The system must log the actions of users in detail, so that a post-mortem analysis can determine whether an access has been legitimate or not.
- **Recoverability**: There should a mechanism for the system to be rolled back to ensure that a user cannot damage a system irreparably.
- **Deterrents**: One effective way of reducing risks in an optimistic system is by using punitive measures to deter misuse. The punitive measures themselves can be either optimistic (with the system administrator enforcing the measure on the detection of misuse) or pessimistic (with the punitive measure implemented immediately and reversed if the action is determined to be legitimate).

In the model presented in Chapter 7, the constrained entry points stipulation is satisfied by pre-obligations and pre-conditions offered by usage control. The accountability notion is addressed by the fact that only authorised users are allowed access. The model system retrospectively provides mechanisms for audit and recoverability. The issue of deterrents is enforced by the obligations and conditions offered by usage control.

3.4 Applicability of optimistic security

According to Povey (1999), optimistic access control may be applied in the following contexts:

Emergency "break-glass" tool

The software equivalent of the "break-glass" container would be a program that is suitably constrained using an optimistic security system and that gives stern warnings about misuse before it is activated. This mechanism is incorporated in the model presented in Chapter 7.

Retrospective content filtering

One of the negative aspects of systems that provide filtering of material which is deemed harmful or inappropriate is that the algorithms used to determine which content to filter can often result in false matches. The result of this is that users can be denied access to legitimate content, forcing them to search for ways to circumvent the system. By applying the principles of optimistic security, users would be able to access any material they desired, and an administrator would log all material accessed and run the content-filtering algorithm retrospectively.

Sandboxing "somewhat-trusted" applications

Traditionally, the focus of "sandboxing" (or constraining the access privileges of programs) has been on untrusted code that is downloaded from the Internet. For example: an optimistic sandbox could track the changes made to the file system by a word-processing program, and allow the user to undo these changes in the event of a crash or malicious macro virus. This would improve the security and safety of these applications without the loss of functionality or expensive certification of the programs.

Watching your system administrator

The system administrator can be constrained in that the user is informed whenever the administrator accesses files that may involve a breach of the user's privacy.

With regard to the context of optimistic access control, it relies entirely on the user being accountable for its own compliance to access control policies, rather the system enforcing access control policies on the user and controlling the user's action. In terms of the compliance mindset subscribes to what might be called a deterrence theory of motivation, which employs mandates, procedural controls and threats of punishment to manage and motivate people (Herath and Rao, 2009). Deterrence theory is based on certainty, severity and celerity of punishment that affect people's decisions on whether or not to commit a crime or not (Higgins et al., 2005). In an information systems security context, these may be visualised in terms of an employee's assessment of the consequences of a security threat and the probability of exposure to a substantial security threat (Herath and Rao, 2009).

3.5 The extensibility of the Optimistic Access Control Model

Optimistic access controls address this niche where access control is not preconfigured and the user is essentially trusted to behave ethically. While traditional access controls such as DAC, MAC and RBAC may be highly appropriate in certain contexts, optimistic access controls may be more appropriate in other circumstances. A field study conducted by Stevens and Wulf (2002) who considered the cooperation between two engineering offices and a steel mill is a case in point. Within this real-world inter-organisational co-operation scenario, it was found that traditional access controls did not comply with the organisation's requirements and that co-operation and competitive reasons motivate the use of interactive and optimistic access controls (Stevens and Wulf, 2002).

A posterior policy enforcement offers interoperability, flexibility and scalability, which are crucial in collaborative environments (Etalle and Winsborough, 2007). Cederquist et al. (2006) also considered enforcing usage control policies a-posterior. This notion is similar to the optimistic access control. The ideas expressed by Cederquist et al.(2006) are different from the model presented in Chapter 7, as they focused on the auditing aspect of enforcing and updating usage control policies retrospectively. The OAC(UCON) model uses usage control as a deterrent mechanism to proactively prevent users from committing data misuse. Thus the idea of refining a policy should be the exception rather than the norm.

Additionally, the OAC(UCON) model focuses more on the pragmatic level of implementation at the application level. The OAC(UCON) model includes the optimistic access control requirements of roll-backs and the issue of continuity in terms of usage control. The model also considers how to leverage optimistic access control within the wider context of traditional access control. The issue of how access control policies will be enforced after a threat has been discovered are beyond the scope of this research. Typically such enforcement will be audit-based (Cederquist et al., 2006). However, a neural network or an expert system could also be applied to decrease a user's privilege to information in the public domain. This approach may prove to be more effective as the updates to access control policies could be more synchronous. This matter will be expanded on in Chapter 9 as a direction for future research.

The mixed-initiative approach is also gaining recognition, where traditional access control is combined with optimistic access control approaches. Imine et al. (2009) consider the domain of distributed collaborative editors that provide support for modifying simultaneously shared documents – such as sharing programming code among dispersed users. Controlling access in such systems is challenging due to dynamic access changes and low latency access to shared documents. They complemented mandatory access control with optimistic access control to solve this problem. To deal with the latency and dynamic access changes, optimistic access control was applied where the enforcement was retrospective. Briscoe et al. (2000) presented a multi-service packet network which involved a mixed-initiative approach where optimistic access control existed within the wider context of the pessimistic access control. For example, the customer may be given an Internet account after providing verifiable identification. Once past this pessimistic hurdle, the optimistic access to more specific parts of the system can be allowed, yet is enforced by punishment. The rationality for such enforcement is that it leaves the network structure clear to simply classify, route, schedule and forward.

The call for privacy-sensitive systems to have a range of control and feedback mechanisms for building pessimistic, optimistic and mixed-initiative applications has also been recognised (Hong and Landay, 2004). Esquivel et al. (2007) also employed optimistic access

control in terms of privacy in pervasive environments. There are environments capable of sensing personal information anywhere and at any time. Based on the ”fair-trade” metaphor, they presented a privacy solution dealing with a user’s privacy as a tradable good for obtaining services in an environment. Thus, users gain access to more valuable services as they share more personal information. This strategy combined with optimistic access control and logging mechanisms, enhances user confidence. Zhao and Johnson (2008) propose access governance with both flexibility and security of information systems. They combine information access, audit, violation penalties and rewards so as to enable self-interested employees to access information in a timely manner and seize business opportunities for the organisation, while managing security risks.

While protecting information from misuse, managers strive to ensure that employees can actually access the information they need to create value (Zhao and Johnson). In terms of the models presented above, optimistic access control is applied retrospectively, whereas the OAC(UCON) model, a flexible system with usage control to prevent misuse, is applied proactively.

The flexibility of the optimistic access control paradigm may be used to address circumstances where traditional access controls prove to be inadequate. However, optimistic access control must be qualified with enforcements to prevent data abuse. In terms of the OAC(UCON) model, it formalises the requirements for optimistic security such as constrained entry points, accountability, audit, recoverability and deterrents. Furthermore, the extensibility offered by this paradigm is complementary to a mixed-initiative access control framework into which the OAC(UCON) model is intended to be incorporated.

3.6 Conclusion

The implementation of optimistic access control requires minimal effort and is cost efficient, because it does not involve specifying and maintaining access control rights. However, the flexibility offered by optimistic access control is a security risk and subject to misuse. Hence it is proposed that this type of access control should be augmented with some sort of control to ensure that humans behave ethically. It is proposed that optimistic access control be complemented with usage control. Within traditional access control models, usage control would offer an extra layer of restriction from unauthorised usage. However, under the optimistic access control paradigm it would not restrict users, but rather deter and constrain them from accessing and misusing information. As was stated earlier, the user should ultimately be able to access the required information.

CHAPTER 4:

USAGE CONTROL

4.1 Introduction

Sandhu and Park (2003), recognising the inadequacy of traditional access control models, proposed a new approach to access control called usage control (UCON). The usage control model is highly appropriate in dynamic and distributed environments where other decision factors such as context should be included to offer stricter enforcement on the rights to digital objects. Typically, access controls consider enforcements that are made prior to access; the UCON model extends pre-authorisation by re-evaluating usage requirements throughout usages. This property is called “continuity” and has to be captured in modern access control for the control of relatively long-lived usage or for immediate revocation of usage (Sandhu and Park, 2003). The other unique property of the UCON model is "attribute mutability". In modern information systems, the decision policies may change as a consequence of certain actions that may result in modifications to the object or subject attributes (Park et al., 2004). The UCON model has been largely inspired from digital rights management and is a general purpose, unified framework that encompasses traditional access control, trust management and digital rights management (Park et al., 2004). This chapter provides an overview of the UCON model and elaborates on its applicability and extensibility.

4.2 The continuity and mutability of the UCON model

The UCON model encompasses emerging applications such as trust management in a unified framework. It is claimed that the missing components of traditional access control are the concepts of *obligations* and *conditions*. *Obligations* require some action by the subject so as to gain or sustain access, e.g. by clicking the ACCEPT button on a licence agreement or agreeing not to distribute the document. *Conditions* represent system-oriented factors such as time of day, where subjects are allowed access only within a specific time period.

In addition to these three decision factors (namely conditions, obligations and authorisations) decision factors, there are two important properties called *continuity* and *mutability*. Continuity is useful when there is sustained usage over a long period of time. Hence usage requirements would be evaluated throughout the usage – known as ongoing authorisation. The other property is mutability, which is useful in Digital Rights Management systems where attributes have to be updated as a side-effect of a subject's actions. These updates may be before (pre), during (ongoing) or after (post) usages (see Figure 4-1). Typically, attribute management can be either admin-controlled or system-controlled. The admin-controlled system is immutable in that the attribute modification is at the administrator's discretion, whereas mutable attributes are automatically modified by the system at the time of usage (Park et al., 2004).

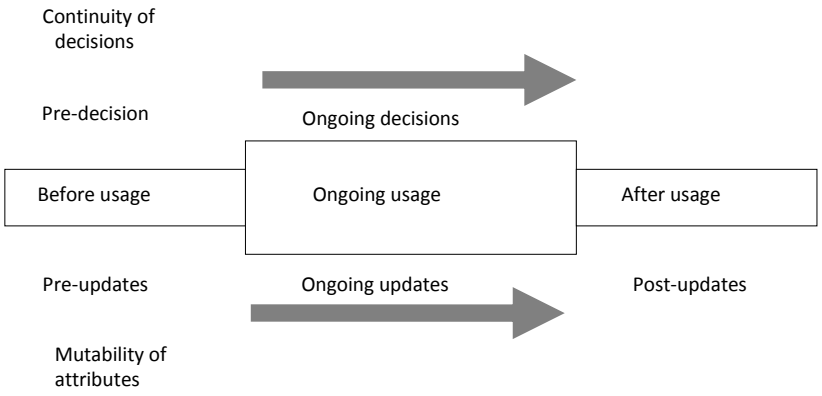

Figure 4- 1: Continuity and Mutability Properties (Park et al.).

4.3 The ABC Model for Usage Control (UCON model)

Sandhu and Park (2003) have expanded usage control into a family of models for usage control involving pre-authorisations and ongoing-authorisations. The implementation of pre-authorisation is relatively simple as it warrants checking the conditions and obligations before the user may proceed. However, the implementation of ongoing authorisation is non-trivial. Sandhu and Park (2003) do not offer a proposition towards how ongoing authorisations may be implemented.

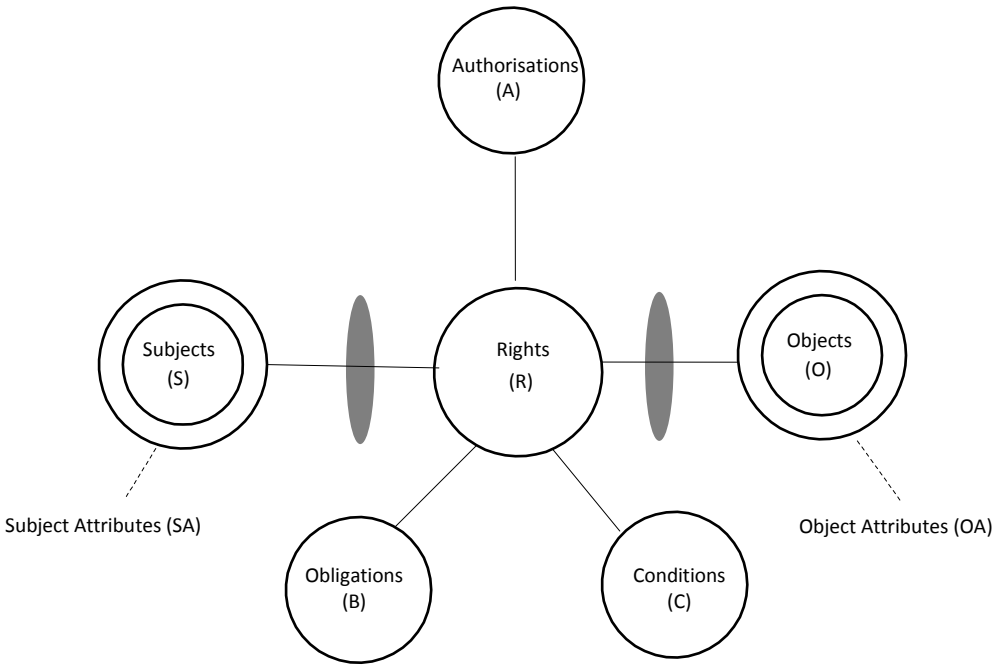

Figure 4-2: ABC Model Components(Sandhu and Park, 2003)

The ABC module (Figure 4-2) consists of eight components: subjects, subject attributes, objects, object attributes, rights, authorisations, obligations and conditions.

Each of these terms (adapted from Sandhu and Park (2003) and Park et al. (2004)) is explained briefly below:

- **Subjects**: represent users.
- **Objects**: target resources in respect of which subjects hold rights.
- **Right**: enables access of a subject to an object in a particular mode, such as a read or write access.
- **Subject and Object attributes**: properties that can be used during the access decision process. In practice, one of the most important subject attributes is subject identity.
- **Authorisation:** based on subject and object attributes.
- **Obligations:** requirements that a subject must perform before (pre), after (post) or during (ongoing) access.
- **Conditions**: environmental or system-oriented factors.

In terms of traditional access control, authorisation is assumed to be done before access is allowed (pre). However, the UCON model extends this to continuous enforcement by re-evaluating usage requirements throughout usages (ongoing). This type of enforcement, known as 'continuity', implies that access may be revoked instantaneously. Ongoing authorisation is active throughout the usage of the requested right and is repeatedly checked for sustaining access. Technically, these checks are performed periodically, based on time or events. For instance, suppose an ongoing obligation condition stipulates that a window declaring the *'Terms and Conditions of Use'* remains open during access. Thus, if the user ignores this stipulation and closes the window during access, the usage is revoked immediately (Sandhu and Park, 2003).

Usage control is relevant in many contexts, including privacy, digital rights management, management of Internet protocols and protocols for trade and administration secrets (Pretschner and Walter, 2008). A key motivation for using usage control is that it considers ongoing controls for extended access or for revocation. For example, Zhang and Nakae (2006) employed the UCON model for collaborative systems by leveraging the features of decision continuity and attribute mutability. This was ratified by the notion that traditional access control approaches do not consider the usage status of a shared object in authorisation. They developed a prototype and found that the main overhead of the system introduced by usage control involved mutable attribute acquisitions; policy interpretations and evaluations; and the updates of mutable attributes. Wang et al. (2006) also motivated using the UCON model for extended access, as it would be useful in ubiquitous environments where the information can be accessed anywhere and at any time, which is potentially unsafe. The ongoing continuity for authorisations, obligations and conditions found in the UCON model can be used to control objects in a dynamic environment since they provide more robust access control for ubiquitous computing environments and can protect sensitive messages from being disseminated.

4.4 The Usage Control Model architecture

From an architectural point of view, one of the most critical issues in enforcing the UCON model is the reference monitor. The reference monitor (Figure 4-3) associates decision policies and rules for the control of access to digital objects.

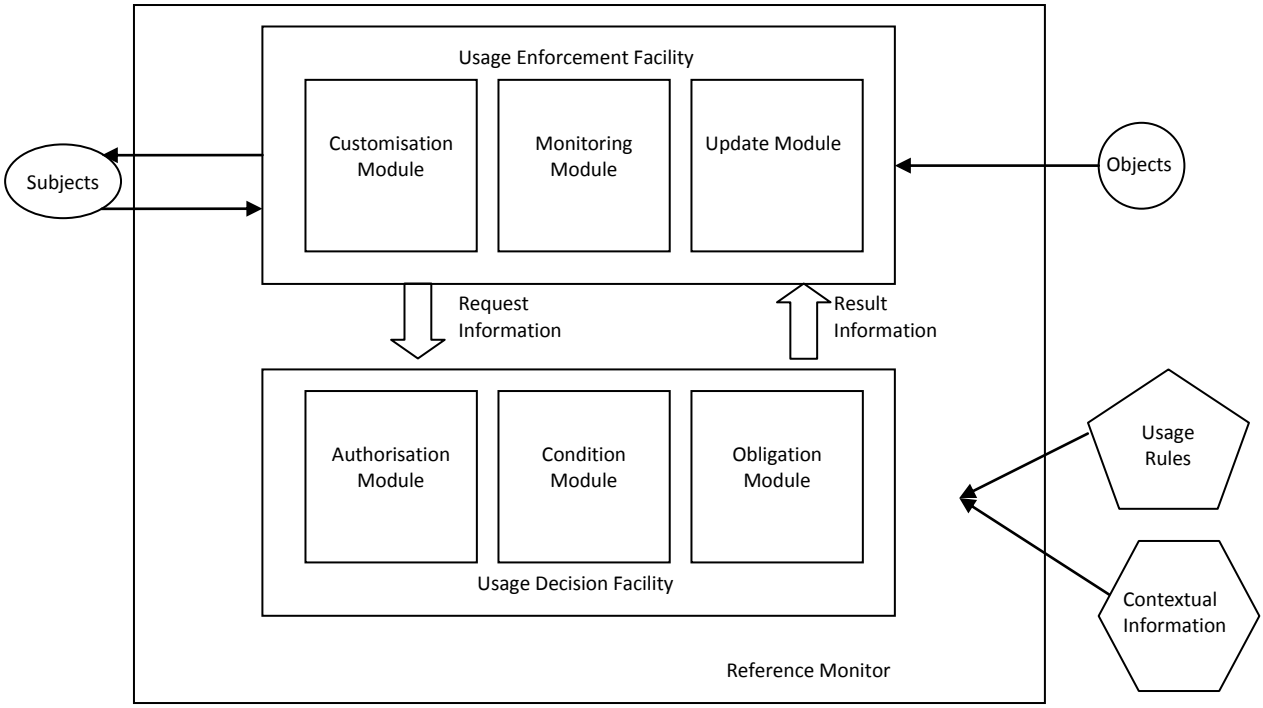

Figure 4-3: Conceptual Structure for the UCON Reference Monitor (Sandhu and Park, 2003)

Usage decisions are based on subject attributes, object attributes, authorisations, obligations and conditions at the time of the usage requests. The UCON reference monitor consists of a **Usage Decision Facility** and a **Usage Enforcement Facility**. The Usage Decision Facility includes modules for controlling the conditions, obligations and authorisations:

- The Authorisation Module deploys a process similar to traditional access control and utilises subject information, object information and usage rules to check whether the request is allowed or not.
- The Authorisation Module may return metadata to the Customisation Model indicating how the data may be presented or customised.
- The Condition Module uses the contextual information and usage rules to decide whether the conditional requirements have been met.
- The Obligation Module decides whether certain obligations need to be performed before, after or during access.
- If there is an obligation that must be performed, this is monitored by the Monitoring Module.
- The result will be resolved by the Update Module, which may change the attributes of the subject and/or object.

4.5 The Applicability and Extensibility of the UCON model

The UCON model is unique in that it can be applied in several contexts with differing access control policy strategies. The model encompasses other temporal and contextual factors aside from considering access rights on subject-object attributes. Consequently, there has been a trend towards complementing access control methods such as role-based access control with usage control (see Li et al. (2005) and Xu et al. (2003)). As indicated, usage control is relevant in many areas, including in privacy and digital rights management.

Although the UCON model is comprehensive, it has been extended in several ways. For instance, according to Lee et al. (2004) this framework lacks an important component in terms of access control. They maintain that the element of '*consent*' should also be included in an access control system, thereby increasing society's trust of a software system. In this scenario, consent is considered to be diametric to the 'concept of obligation' within the usage control model. 'While the obligation is obeyed by the customer, *consent* is observed by the provider' (Lee et al., 2004). The proposed method can extend the coverage of the UCON model in terms of security and enhance the right of both provider and customer. It also provides a solution for trust relationships in e-Commerce and for the protection of an individual's privacy. In a position paper, Pretschner and Walter (2008) considered usage control in the context of distributed systems that are composed of different actors assuming the role of data providers (who give data away) and data consumers (who request and receive data). In their position paper, they considered the element of negotiation for usage control. The term negotiation suggests that multi-step bidirectional communication takes place. Shin and Yoo (2007) extended the UCON model by incorporating an additional component, *delegation*, for effective modelling of the delegation of access rights in ubiquitous computing.

Syalim et al. (2005) proposed an enforcement to support data confidentiality in a database service provider by contextualising the usage control model and architecture for the aforementioned database service provider. In this context, the UCON model extended the access matrix by utilising either a server-side or client-side reference monitor, or both server-side and client-side reference monitors. Syalim et al. (2005) exploited the flexibility

of the UCON and separated the control domain in a database service provider into two parts: a database provider domain and a database user domain. In the database provider domain, the access control system controls users' access to the database services, while in the database user domain, the access control system controls other users' access to a user's database.

Due to the dynamic authorisation requirements in collaborative systems, Zhang and Nakae (2006) proposed a generalised authorisation framework for such systems based on the UCON model. In collaborative systems, organisations share their computing resources to establish virtual organisations. By leveraging the flexible policy specifications and attribute mutability of the UCON model, Zhang and Nakae's (2006) model supports virtual organisations-level authorisation policies, but also usage constraints defined by each resource provider. In their generalised authorisation framework, conditions are used to support context-based authorisations in ad hoc collaborations.

Due to the extensibility and expressibility of the UCON model and its all-encompassing nature, it may be extended to optimistic access control as well. The current research considers adapting usage control as a proactive means of deterrence control to protect information that cannot be protected adequately or reasonably by access control.

4.6 Conclusion

It has been suggested that where usage control on data consumers may not always be practical, feasible or sensible, optimistic access control may be more appropriate where the 'observation' or monitoring of user behaviour could be used as a deterrent (Pretschner et al., 2008). As discussed in Section 4.5, the UCON model is applicable in many areas due to its strong expressive power and policy specification flexibility (Zhang et al., 2006). The application of the UCON model to optimistic access control will be explored in detail in Chapter 7.

Usage control involves pre-authorisations and ongoing authorisations. The implementation of pre-authorisation is relatively simple, as it warrants checking the conditions and obligations before the user may proceed. In contrast, the implementation of ongoing authorisation is non-trivial, and Sandhu and Park (2003) have not proposed any stipulations as to how ongoing authorisations may be implemented. It has in the meantime been suggested by Padayachee and Eloff (2007) that multithreading should be applied to implement ongoing authorisations. However, in general, there is no notion of how usage control as a whole may be implemented in the real-world context to complement existing access control approaches. The aspect-oriented paradigm is now considered to be suitable for this context as it facilitates the abstraction of usage control decisions from other access control mechanisms and application logic. This approach would result in an implementation that is easier to augment to existing access control implementations. Before the OAC(UCON) model and the prototype are presented in Chapters 7 and 8 respectively, Chapter 5 unpacks aspect-oriented programming and the evolution that has led to this new programming paradigm.