INTELLIGENT INTERFACE AGENTS FOR BIOMETRIC APPLICATIONS

A Thesis Submitted to The University of
Kent at Canterbury
For the Degree of Doctor of Philosophy
In the subject of Electronic Engineering

By
Nick, Mavity
May, 2005

Abstract

This thesis investigates the benefits of applying the intelligent agent paradigm to biometric identity verification systems. Multimodal biometric systems, despite their additional complexity, hold the promise of providing a higher degree of accuracy and robustness. Multimodal biometric systems are examined in this work leading to the design and implementation of a novel distributed multi-modal identity verification system based on an intelligent agent framework. User interface design issues are also important in the domain of biometric systems and present an exceptional opportunity for employing adaptive interface agents. Through the use of such interface agents, system performance may be improved, leading to an increase in recognition rates over a non-adaptive system while producing a more robust and agreeable user experience. The investigation of such adaptive systems has been a focus of the work reported in this thesis.

The research presented in this thesis is divided into two main parts. Firstly, the design, development and testing of a novel distributed multi-modal authentication system employing intelligent agents is presented. The second part details design and implementation of an adaptive interface layer based on interface agent technology and demonstrates its integration with a commercial fingerprint recognition system. The performance of these systems is then evaluated using databases of biometric samples gathered during the research.

The results obtained from the experimental evaluation of the multi-modal system demonstrated a clear improvement in the accuracy of the system compared to a uni-modal biometric approach. The adoption of the intelligent agent architecture at the interface level resulted in a system where false reject rates were reduced when compared to a system that did not employ an intelligent interface. The results obtained from both systems clearly express the benefits of combining an intelligent agent framework with a biometric system to provide a more robust and flexible application.

Acknowledgements

I would like to thank my family for their support during these years I have been conducting this research. I also would like to thank Imy whose tireless understanding throughout this period has helped me stayed focused on the work during many a dark time.

I would like to express my sincerest thanks to my supervisor Dr. F. Deravi and also Professor M. Fairhurst for their assistance, without whom this research could not have been carried out.

I would also like to thank everyone in the Electronics Laboratory, staff and students. In particular I would like to thank J George, for her diligent assistance with certain phases of the IAMBIC framework.

Finally I would like to thank EPSRC for their financial support.

Contents

List of Figures

Chapter 1

Introduction and Overview of the Thesis

1.1 Introduction

It can be said that interest in biometric systems for identity authentication has been growing in recent years. The strengths of biometric systems are well known, however, the success of any biometric system is measured ultimately by its usability and user acceptance. Serious obstacles to wider adoption of biometric system include user interface aspects which can lead to poor performance. More attention is also being directed towards multimodal systems to enhance the robustness and security of the authentication process.

The term “software agents” is used to describe a wide range of systems with diverse properties. Software agents exhibit aspects such as autonomy and adaptivity, which were seen as some of the key properties that could be employed in order not only to develop a distributed multimodal biometric authentication system, but also an agent based framework for improving the performance of uni-modal biometric system.

The research in this thesis addresses the possible integration of software agents with biometric systems and the potential benefits that can result from this combination. The rationale behind this marriage of technologies is to provide an underlying technology that can manage the complex interactions that may be involved especially when multimodality is employed. Software agents can also be engaged at the user interface level to provide a robust interaction during sample donation even for uni-modal biometric systems. Both of these aspects of software agent integration will be examined during the research presented in this thesis.

In this chapter a number of subject areas will be introduced that are relevant to the research presented in this thesis, namely biometrics and software agents. These areas are investigated in more detail in their relevant chapter, however, a brief introduction into each major research area explored in this thesis is presented in this chapter. The organisation of the thesis is also outlined at the end of this chapter, providing a brief description of the following chapters and highlighting the main contributions for each chapter.

Both of the research fields introduced in this thesis are very broad and contain a diverse range of research interests. In the biometrics field, the scope of the thesis covers aspects of multimodality, specific modalities, interface characteristics as well as issues surrounding the implementation and deployment of biometric systems. More fundamental low-level issues such as algorithm development for feature extraction and matching are considered out of scope for this thesis.

A similar approach is taken to the field of software agents also presented in this thesis. An overview of agent research is presented, however, the focus is on a subset of the overall possible agent types. These types illustrate the main architectures employed in software agent systems, as well as highlighting some of the interesting characteristics that software agents can possess.

1.2 Software Agents

"The idea of an agent originated with John McCarthy in the mid-1950's, and the term was coined by Oliver G. Selfridge a few years later, when they were both at the Massachusetts Institute of Technology. They had in view a system that, when given a goal, could carry out the details of the appropriate computer operations and could ask for and receive advice, offered in human terms, when it was stuck. An agent would be a 'soft robot' living and doing its business within the computer's world." [Kay, 1984].

Since this tentative idea was formed, the field of agent research has spawned an expanding, diverse and vibrant community. Agent research itself can be split into two main strands, each covering two different time periods [Nwana, 1996]. The first, starting about 1977, is described as having "concentrated mainly on deliberative type agents with symbolic internal models." Work in this field is said to have contributed to an understanding of "*macro* issues such as the interaction and communication between agents, the decomposition and distribution of tasks, coordination and cooperation, conflict resolution via negotiation, etc." It was the field of Distributed Artificial Intelligence (DAI) that mainly provided the 'roots' for this strand of agent research. The second started around 1990, and is described as having led to the broadening of the range of agent types that were being investigated. In this strand there has been a perceptible shift in emphasis from the more theoretical aspects of research towards the practical realisation of software agents.

1.2.1 A Definition

The initial challenge encountered trying to provide a formal definition of an agent is that every developer and researcher in the agent field assumes their own definition of the term ‘agent’. This issue will be explored in more depth in Chapter 3. But for now an agent can be defined thus:

“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.” [Russell & Norvig, 1995]

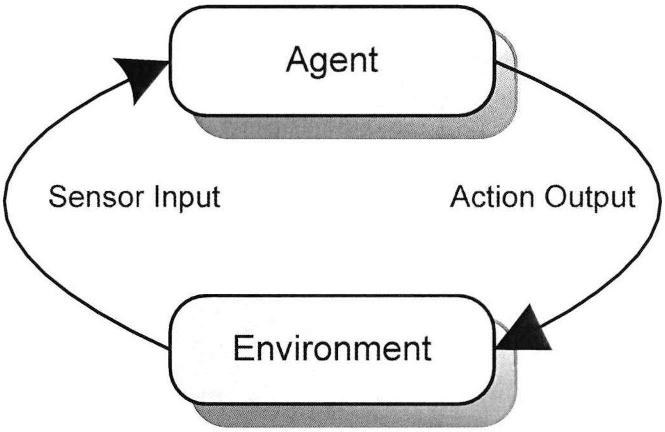


Figure 1.1 Basic agent

A software agent can be further defined:

"A software agent is a computational system which has goals, sensors, and effectors, and decides autonomously which actions to take, and when" [Maes, 1997]

Although there is much debate on what characteristics should be present in order for an object to be termed a software agent, it is generally agreed that autonomy is one of the core aspects. A software agent must be communicative, not only with the user but also other software agents or software processes. A software agent is also perceptive, it is able to perceive and respond to changes in its environment.

Software agents can possess varying degree of 'intelligence' in their selected application domain. Some agents may only need very basic competence to perform their desired goals and may be perceived as quite 'dumb'. 'Smarter' agents have been imbued with higher levels of intelligence in order to accomplish the given task. [Russell & Norvig, 1995]

1.2.2 Agent Research

The field of agent research can be split into three primary groups.

Agent Theories

This field of research is responsible for the definition of agency and the properties which agents should possess. Agent theorists are also responsible for devising a suitable formal representation for the properties of agents. [Moore, 1985][Seel, 1989][Rao and Georgeff, 1991]

Agent Architecture

In order to realise agents that satisfy the properties required of them by the specifying theory, agent architectures are used to identify hardware and or software structure that are appropriate. [Brooks, 1991][Ferguson, 1992]

Agent Languages

These are systems which enable hardware or software system to be programmed in the terms of some of the concepts developed by the agent theorists. [Shoham, 1993] [Nwana et al, 1999][Gutknecht and Ferber, 2000].

1.2.3 Types of agents

Agents can be classified in a large number of ways due to the number of attributes which the agent may possess. An agent typology can be used to differentiate between the various types of software agent.

- Reactive Agents
- Deliberative Agents
- Hybrid Agents
- Mobile Agents
- Interface Agents

1.2.3.1 Reactive Agents

The most important characteristic of this type of agent is that it does not possess an internal symbolic model of the environment. Actions are chosen by referencing a lookup table of situation-action pairs. The situation-action pairs [Suchman, 1987] match input conditions to pre-defined states in order for an output to be effected. In this manner the agent responds in a stimulus-response manner to the present state of the environment.

This type of agent has been found to be very effective for well defined problems. Incorporating them in a system that requires run-time flexibility or goal-directed behaviour is very difficult. This approach is also most appropriate for when the environment the agent is located in is static. Reactive systems have the advantages of being able to react quickly to incoming stimuli as there is no need to generate or select elaborate plans of action [Ferber, 1994][Nwana, 1993][Brooks, 1986].

Although these agents are relatively simple and the interaction with other agents is basic, complex patterns of behaviour emerge from the interactions. [Agre and Chapman, 1987].

1.2.3.2 Deliberative Agents

This type of agent has the ability to consider the alternative courses of action before an action is taken [Wooldridge, 1995]. Core to this ability is the symbolic reasoning and planning module. By employing symbolic reasoning the agent is able to decide on an appropriate sequence of actions or plans in order to realise its current goal.

Planning systems have some inherent problems associated with them, however, scalability is an issue as the complexity of the problem increases. Also planning systems can have problems reacting in real time. One of the reasons for this is during the often slow reasoning or execution phase, the environment in which the agent is situated may change. Another reason is that the plans generated by the agent rely on the ability to predict the outcome of a series of actions, but the environment may not behave as predicted. [Yamauchi, 1998][Kitano et al, 1997][Titmus et al, 1996][Wiegland and O'Brien, 1996].

1.2.3.3 Hybrid agents

Hybrid agents combine both reactive and deliberative methodologies to produce a system that is capable of forming and executing long term plans via the deliberative layer but also has the capability to react quickly to incoming stimuli by utilising the reactive component layer. Examples of such hybrid systems can be found in [Rousseau and Hayes-Roth, 1998][Muller, 1996][Ferguson, 1992][Georgeff and Ingrand, 1989]

1.2.3.4 Mobile Agents

Mobility of an agent could be construed as a property or attribute of an agent rather than a class of agents in its own right, however, for the purposes of this thesis, mobility will be considered as class of agent. Mobile agents are executing code that can move from host to host in a network when and where it chooses in order to meet it goals. There are varying degrees of mobility these will be investigated in Chapter 3. Mobile agents offer numerous benefits such as asynchronous computing, distributed computing and also can help reduce communication costs. [Bellifemine et al, 2001][Huber, 1999][Arnold et al, 1999].

1.2.3.5 Interface agents

"*Interface agents* are computer programs that employ Artificial Intelligence techniques to provide active assistance to a user with computer-based tasks" [Maes, 1994]

This type of agent is primarily designed to act as a personal assistant which can aid the user in a number of ways:

- Perform tasks on the user's behalf
- Can train or teach the user
- Help different users collaborate
- Monitor events and procedures.

The agent is able to observe and learn from the user as well as acquiring competence in the given task by also querying agents assisting other users. The application range for this particular type of agent is broad and can include, information retrieval, mail management, meeting scheduling etc.[Menczer, 2003][Billsus & Pazzani, 1999][Boon, 1998][Sen et al, 1997][Chen & Sycara, 1998]

1.3 Biometrics

The term Biometrics refers to the automatic identification or recognition of a person based on his/her physiological or behavioural characteristics. [Biometrics Consortium, 1995] Current authentication techniques involve the use of a token (such as a card) or secret knowledge (such as a PIN number), or a combination of both schemes. Biometric systems differ as it is a characteristic the user possesses that is used as the authentication medium.

The biometric approach towards authentication is gaining more acceptance in today's society for a number of reasons. Donating a biometric sample requires the user to be present at the point of identification decreasing the risk of non repudiation of a transaction. Also it eliminates the need for the user to remember a password or carry a token, both of which may be forgotten or misplaced by the user. The advantage of employing biometric systems is that it is expected to reduce unauthorised access or fraudulent use, as well as providing a more convenient form of identification to the user.

Table 1.1 illustrates some of the diverse range of biometric modalities that can be employed using either a subjects behavioural or physiological characteristics.

Anatomical	Behavioural
Face	Gait
Fingerprint	Signature
Iris	Voice
Retina	Keystroke dynamics
Hand vein	
Hand geometry	

Table 1.1 Biometric modality types

There are a number of features which can be used to determine the suitability of the particular modality for a target application. These are illustrated in Section 2.2.1. Choosing a suitable biometric modality for a target application consists of considering a large number of factors. Typically the performance and security as well as the cost of a modality are considered. Other factors include the user acceptance of the modality and also the environment in which the system will be located must be carefully considered. Ambient environmental conditions may influence the acquisition of some biometric samples e.g. a noisy environment may prevent voice recognition from operating correctly.

Employing a biometric introduces some procedural factors, these include aspects such as the procedures that are employed for user enrolment and whether interactions with the biometric systems are supervised to prevent the possibility of fraud. Information technology aspects cover areas such as the computer resources required to deploy the system, reliability of the hardware and the maintenance and backup costs.

1.3.1 Biometric system overview

Before a subject can use a biometric system they must enrol. This involves donating a number of samples from which a template is constructed. This template is used during the verification process during which the subject donates samples that are compared against the enrolment template in order to authenticate the identity of the user.

In every biometric system there are a number of clearly defined processes that occur within the system in order to perform the required biometric operation. Figure 1.2 provides a simplified overview of a biometric system.

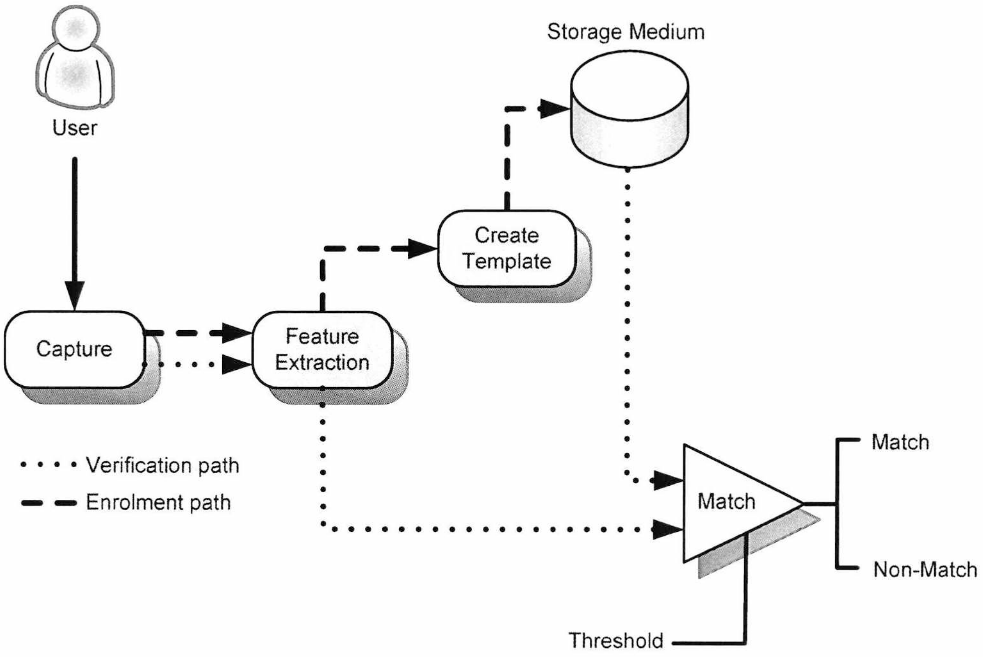


Figure 1.2 Simplified biometric system

1.3.1.1 Capture

This is the first stage in any biometric system. This stage involves the capture of the raw biometric sample. Obviously the nature of the capture will vary with the particular biometric modality employed. For example a facial biometric would need to acquire an image of the subjects face via a camera, a voice-based biometric would need a sample of the subjects voice captured a microphone. Quality of the acquisition sample is important as poor samples may have an impact on the overall performance of the system.

1.3.1.2 Feature Extraction

This stage involves the extraction of the relevant features from the raw sample in order to perform either the comparison phase or to create a template for the enrolment phase. The features extracted are dependant on the modality that the system uses. For example a fingerprint-based modality may extract minutiae points from the image of the finger.

1.3.1.3 Creation and Storage

This particular phase is only used during the enrolment process. A template contains the extracted features from a users sample or samples, this template is used in subsequent verification attempts where a current sample is compared a current sample to determine whether a user has passed or failed verification. Once the template has been created it can be stored for future use. Generally the size of this template is smaller than the captured sample but depends on the modality itself e.g. a template for a fingerprint biometric may be in the order or around 800 bytes, whilst a facial biometric system may have templates in the order of 50 kilobytes [Identix, 1982] The template can be stored in a number of locations such as the remote databases, the client machine itself or even a smart card which the subject possesses.

1.3.1.4 Matching

This stage is used when a user is attempting to verify. The current sample donated by the subject is compared against the previously created template. The threshold setting determines how closely the features need to match, or the threshold may specify a set of distance values from which the extracted features need to lie in order in order to pass verification. The setting of the threshold value is usually adjustable in software and also determines the overall usability of the system as it also dictates the proportion of users who may be falsely rejected or the number of impostors which may be accepted by the system. During the matching phase two different methods may be employed in order to authenticate the user.

Verification

This type of authentication can also be known as 1:1 matching. The user is making a *positive* claim to an identity. Authenticating the user consists of attempting to match live sample against the previously stored enrolment template.

Identification

This type of authentication can also be known as 1:N matching. The user is making no claim or an explicit *negative* claim to an enrolled identity. The matching process consists of searching a database of enrolment templates for a match against the given sample. This may result in a number of possible candidates being identified which may be the subject under investigation, if this occurs the system will offer a probability associated with each potential match.

1.3.2 Biometric Challenges

Although biometrics initially appear to provide an effective and convenient method of authentication there are still a number of key operational and implementation issues that have been identified that are tending to hinder the widespread deployment of biometric systems [Jain et al, 2004]. Theses issues are not related to any one biometric in particular but rather the domain as a whole. These issues include aspects of accuracy, scaling and security, these will be discussed in more detail in Section 2.9.

Biometric technologies are emerging as key components in the regulation of online information access and significant application areas exist such as electronic commerce, telemedicine and database access for example. Recognition based on one particular biometric alone may not prove to be sufficiently robust or acceptable by a particular user group. Multi-modal biometric systems are seen to provide as a viable approach for overcoming performance and acceptability barriers, however, by employing multimodality in such a distributed environment it increases the complexity of the overall system interaction not only with the user, but also between the distributed components.

Software agents were seen as critical in the management of this complexity involved when employing multimodal biometrics for remote access. Software agents themselves are capable of autonomous action and can respond in a flexible manner to their environment. It was such properties that stimulated interest in employing an agent based framework to providing a novel robust control structure for distributed multi-modal authentication systems. The tasks for such agents include the handling of multiple authorisation levels, managing the dynamic user interaction and also regulating data across a number of remote repositories.

1.3 Organisation of the Thesis

The remainder of the thesis has been divided into six chapters. The organisation of this research is as follows:

Chapter 2 Biometric systems

This chapter provides an overview of biometric systems. Properties of biometric systems are introduced and the methods that are used to quantify the performance of biometrics are discussed. A number of biometric modalities are also examined; the capture technologies are investigated along with the techniques for feature extraction and pattern matching. Some problems with current biometric systems are also explored in this chapter and an approach to overcome some of these shortfalls will be presented.

Chapter 3 Software agents

This chapter presents a review of software agents. Within this chapter the issue of attempting to classify what is an agent is discussed, and various taxonomies are introduced to illustrate the diversity of this field of research. After a number of classification scheme are introduced, each relevant agent typology is examined, key underlying theories and architectures are discussed. Also a number of examples of each particular agent type are presented in order to illustrate the work performed in these fields of agent research. At the end of this chapter a brief introduction is given to Agent Communication Languages (ACL's).

The main focus of this chapter is to establish an understanding of software agents and also to illustrate the potential benefits that the application of the agent paradigm can in certain application domains.

Chapter 4 The IAMBIC system

The novel IAMBIC system (Intelligent Agents for Multimodal Biometric Identification and Control) is introduced in this chapter. IAMBIC is a distributed multimodal biometric verification system which was designed for resource access over a network using a number of biometric modalities as the authentication medium. Software agents were employed to manage the complexity of the multimodal transaction in this remote environment. Within this chapter design methodology is discussed and functional components of the system are presented.

Chapter 5 Adaptive Interface Layer

A novel adaptive interface layer is the presented in this chapter. The motivation for the development of such a layer is discussed, and the behaviour of the layer encapsulating the interface agent paradigm explained. The components of the adaptive layer are examined and the overall behaviour of the system is defined and analysed.

Chapter 6 Experimental Setup and Testing

This chapter presents the experimental procedure used in order to evaluate both the IAMBIC system and also the adaptive interface layer. The performance of the adaptive layer is compared against a non-adaptive system, and user feedback on the system is presented and analysed.

Chapter 7 Conclusions and further research

This chapter provides a summary of the work along with some suggestions for future research in this particular field.

Chapter 2

Biometric Systems

In this chapter we will examine what a biometric system is and also how the biometric industry has developed to harness this new method of human identity recognition. The performance measures of biometric systems will be introduced and evaluated. Section 2.4 will look at a number of biometric modalities available today as well as the enabling underlying technology of each particular modality. The chapter concludes with a look at some of the shortcomings of current biometric systems and some of the possible techniques for overcoming these deficiencies.

Although this chapter cannot provide exhaustive coverage of this domain, the material covered is hoped to be comprehensive enough to enable a fundamental understanding of biometric systems.

2.1 Introduction

This chapter provides an overview of biometric systems. Properties of biometric systems are introduced and the methods that are used to quantify the performance of biometrics are discussed. A number of biometric modalities are also examined, the capture technology is investigated along with the techniques for feature extraction and pattern matching. There are some problems with current biometric systems these also will be explored in this chapter and an approach to combat some of these shortfalls will be presented.

2.2 Fundamental techniques of identity recognition

Traditional methods of identity recognition rely on the three conventional methods detailed below. These methods are in everyday use for a range of applications where authentication is required.

Secret Knowledge "What I know"

Here authentication takes the form of secret Personal Identification Numbers (PIN) and passwords, which the user has to remember. The authorised user has to share the secret knowledge with the authenticator.

Personal Possession "What I have"

Examples for authentication are having a key, ID card, or pass (with or without a chip), which allows entrance, for example, into a private room. Key to this approach is the existence of unique features whether they are covert or overt.

Combination Systems

For security reasons, often the above approaches are combined, e.g., a bank card with a PIN. Following the definition above, a password written down on a sheet of paper exclusively belongs to the group of "personal possession", it is not secret knowledge any more.

Biometrics "Who I am"

Biometrics uses nature's oldest system to identify people - via unforgettable and unchanging physical characteristics.

2.3 What is a biometric?

The terms 'Biometrics' and 'Biometry' have been used since early in the 20th century to refer to the field of development of statistical and mathematical methods applicable to data analysis problems in the biological sciences. More recently the term 'Biometrics' has also been used to refer to the emerging field of technology devoted to identification of individuals using biological traits, such as those based on retinal or iris scanning, fingerprints, or face recognition. A more formal definition of a biometric is given below.

"Biometrics are automated methods of recognising a person based on a physiological or behavioural characteristic." [Biometrics Consortium, 1995]

Biometric authentication systems have some inherent strengths which makes them attractive as a tool for identity verification and identification, these aspects are discussed in some detail in Section 2.8.

2.3.1 What features make a good biometric?

There are a number of properties that must be considered before the decision is made to employ a physiological or behavioural characteristic as a biometric. It is the variance in the degrees in which these properties are fulfilled that determines what sort of application the particular biometric can be employed in.

- **Universality:** Every person should have the characteristics. All cases must be handled such that no proportion of society is discriminated against
- **Uniqueness:** No two persons should be the same in terms of the biometric characteristics. Whilst identical twins would be identified using fingerprint

recognition for example, face recognition might not distinguish one twin from the other

- **Permanence:** The characteristics should be invariant over time. Some features are more invariant over time than others. The retina for instance is highly invariant, whilst a person's facial features are likely to change slightly with time
- **Collectability:** The characteristics must be measured quantitatively and obtaining the characteristics should be easy
- **Performance:** This refers to the achievable identification/verification accuracy and the resources and working or environmental conditions needed to achieve an acceptable accuracy
- **Acceptability:** This indicates to what extent people are willing to accept the biometric system. This is also a measure of the invasiveness of the system to the user, traditionally retinal based biometrics are deemed to be highly invasive
- **Circumvention:** This refers to how difficult it is to fool the system by fraudulent techniques. This is a subject which will be examined in more depth in Section 2.10.6

Although not explicitly stated in the above list there is another property which should be mentioned and that is intrusiveness. Closely related to both acceptability and collectability of a particular biometric it describes how intrusive the particular biometric is to acquire.

A number of primary biometric modalities have emerged as what could be termed as 'mainstream' biometrics. This includes modalities such as face, finger, voice, signature, iris and retina.

There is a smaller subset of biometrics which although not as well known as the ones mentioned above but can be used such as gait, keystroke, hand geometry and hand vein

2.3.2 The 'Ideal' Biometric

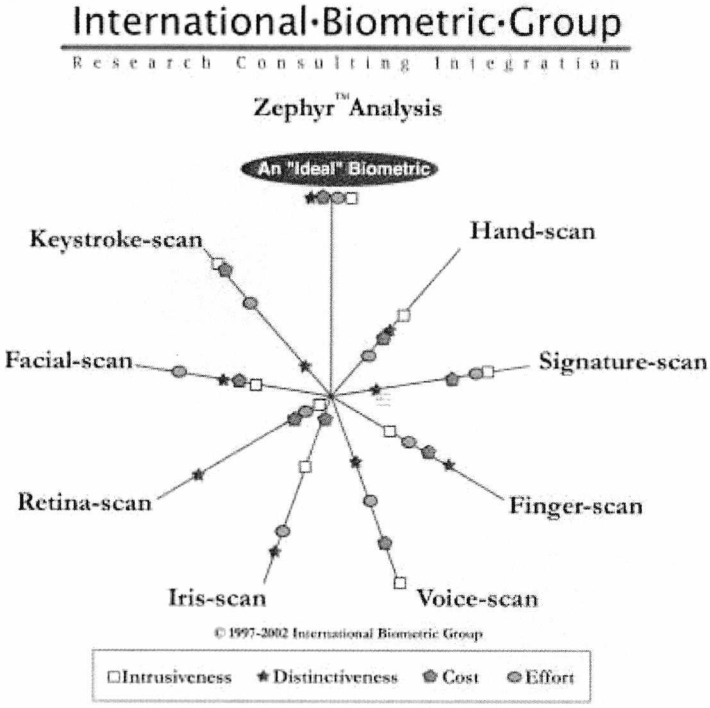


Figure 2.1 Zephyr Analysis ©IBG

Figure 2.1 was produced by the International Biometric Group (IBG) in order to answer the question "What is the best biometric system?" It analyses the most popular biometric technologies in terms of ease of use, cost, accuracy and perceived intrusiveness, these being seen as the most important characteristics of a biometric system. From the figure it can be observed that iris scan is seen as the most distinctive of the biometrics analysed, however the cost of deployment of these systems is seen as a major disadvantage.

It can be seen quite clearly that each modality offers different parameter values and choosing a particular biometric for an application depends on many criteria and unfortunately there is no such thing as an 'ideal' biometric.

2.4 Biometric System Fundamentals

Figure 2.2 illustrates the logical operational structure of a generalised biometric system.

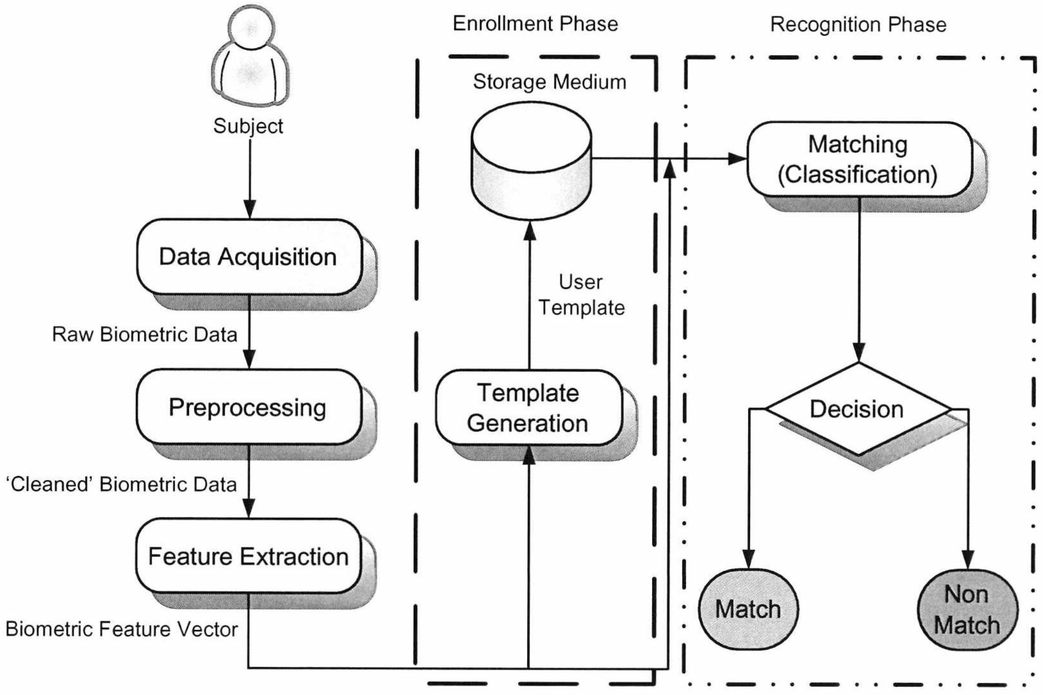


Figure 2.2 Biometric system building blocks

Data Acquisition

This is the process by which the initial raw data is captured from the relevant sensing device (i.e. a camera, scanner etc). This initial data forms the basis for the following computations. Obviously the quality of this data will have an impact on the performance of the complete system. This implies that high quality capture data should enhance performance of the biometric system. It is known that subsequent samples from the same subject can be highly variable as noise or human-machine interface issues may affect the donation of samples.

Pre-processing

This stage serves to further enhance the quality of the initial sample. Steps such as noise removal or changing the format of the acquired data into a more suitable representation for feature extraction.

Feature extraction

This is the process by which the relevant features are extracted by an algorithm in order to perform verification or enrolment tasks.

Matching or Classification

This is the sequence of matching and decision. The matching process calculates a similarity or dissimilarity measure based on the current feature set and the reference data set (template). The matching process must also compensate for the variability introduced during the process of sample acquisition.

2.5 Biometric System Interactions

There are three main types of biometric system interaction. Enrolment is mandatory for all biometric systems, although whether the system employs verification or identification at the recognition phase is application dependant.

2.5.1 Biometric Templates

A biometric template is generally a small file derived from the features of a user's biometric sample or samples and is used when biometric matching occurs. The template is created after the algorithm locates the relevant features from the sample the user has donated, during the enrolment phase. The format of user templates vary from vendor to vendor even for the same modality and in some cases the templates are stored encrypted in order to increase security. Steps are being taken in order to produce standards that will harmonise these template files across vendors, these are exposed in section 2.10.4.

2.5.2 Enrolment

This is the process by which a user's initial biometric sample or samples are collected and analysed to extract the necessary features in order to generate a template for the user. Depending on the modality and vendor the amount of samples acquired during the enrolment procedure in order to create a valid template may vary. I.e. for a finger modality the user may have to donate two separate finger images from the same finger in order for successful enrolment.

2.5.3 Recognition (Verification)

This is more commonly known as 1:1 matching or authentication. This is the process by which the identity of the individual is checked against their own template which has been previously created.

2.5.4 Recognition (Identification)

This is more commonly known as a 1:N, one-to-many or recognition. This is the process by which a person's identity is determined by performing matches against multiple biometric templates. Generally in this particular mode of operation there may be a number of candidates identified which could be the subject under investigation, in this case usually the system will indicate the probability that each of the identified individuals is likely to be the subject.

2.6 Biometric Performance Metrics

For all biometric systems there exist a number of established measures that serve to evaluate and compare performance. These are detailed below.

2.6.1 False Accept Rate (FAR) or Type II Error

The FAR is the frequency that a non authorised person is accepted as authorised by the biometric system. Because an instance of false acceptance may lead to system penetration, FAR is generally considered to be a security relevant measure. FAR is a statistical quantity which does not only show a personal correlation, it can even be determined for each individual feature (called personal FAR) [2.1]

$$FAR(n) = \frac{Number\ of\ successful\ fraud\ attempts\ against\ a\ person\ (or\ feature)\ n}{Number\ of\ all\ fraud\ attempts\ against\ a\ person\ (or\ feature)\ n} \qquad (2.1)$$

The overall FAR for N participants is defined as the average of FAR(n):

$$FAR = \frac{1}{N}\sum_{n=1}^{N} FAR(n) \qquad (2.2)$$

2.6.2 False Rejection Rate (FRR) or Type I Error

The FRR is the frequency that an authorised person is rejected access to the system. FRR is generally thought of as a comfort criteria, because a false reject condition is most of all annoying, rather than a serious security concern. FRR can even be determined for each individual feature (called personal FRR). The majority of biometric systems will reject a biometric sample if it is of poor quality, such as a voice sample polluted with a large amount of external noise. Although the system will still reject the user in this case it is not through faulty operation, however the user still perceives the failure as a false reject. Personal FRR can be calculated using the equation 2.3.

$$FRR(n) = \frac{Number\ of\ rejected\ verfication\ attempts\ for\ a\ qualified\ person\ (or\ feature)\ n}{Number\ of\ all\ verfication\ attempts\ for\ a\ qualified\ person\ (or\ feature)\ n} \qquad (2.3)$$

Overall FRR for N participants is defined as the average of FRR(n)

$$FRR = \frac{1}{N}\sum_{n=1}^{N} FRR(n) \qquad (2.4)$$

2.6.3 Total Error Rate (TER)

This is a unique measure that is calculated by the addition of FRR and FAR. From this figure another measure can be defined, the Total Success Rate (TSR)

$$TER(\%) = \frac{(FAR + FRR)}{(Total\ number\ of\ accesses)} \times 100 \qquad (2.5)$$

$$TSR = 100\% - TER \qquad (2.6)$$

2.6.4 Failure to Enrol (FTE)

This is the probability that a given user will be unable to enrol in a biometric system due to insufficiently distinctive biometric sample or samples. Failure to enrol is an important parameter as repeated failed enrolment attempts with a particular modality will lead to poor user perception of the biometric system. There will always be a proportion of the population for which enrolment in a specific modality will not be possible, in this case the failure is permanent. This occurs where an individual cannot present the required feature i.e. cataracts which render the use of a retinal system impossible. This is opposed to temporary failure which may occur with worn down or sticky fingertips for a fingerprint system.

The probability for the lack of success for a certain person to enrol (FTE(*n*)) is given by the equation.

$$FTE(n) = \frac{Number\ of\ unsuccessful\ enrolment\ attempts\ for\ a\ person\ (or\ feature)\ n}{Number\ of\ all\ enrolment\ attempts\ for\ a\ person\ (or\ feature)\ n} \quad (2.7)$$

A problem facing all authentication systems is that a fall back process is required in case the primary method of authentication cannot be performed. These procedures are commonplace in current authentication systems so it is expected that biometric systems would employ some form of non biometric fall back system such as a conventional PIN or token scheme.

2.6.5 False Match Rate (FMR)

This measure is employed to avoid confusion in applications that reject the claimant if their biometric data matched that of the enrolee. In these applications the concepts of acceptance and rejection are reversed. This is the rate at which non-authorised people or impostors are falsely recognised during the feature comparison stage on a single template to sample comparison. Samples that have been rejected due to poor image quality (FTA) are not accounted for using this measure [Mansfield and Wayman, 2000].

2.6.6 False Non Match Rate (FNMR)

This is the probability that a sample will be falsely declared not to match a template of the same measure from the same user supplying the sample. Samples that have been rejected due to poor image quality (FTA) are not accounted for using this measure [Mansfield and Wayman, 2000].

Both FMR and FNMR represent measures that characterise the performance of the system at the matching algorithm level and are generally defined as the result of a single comparison of a submitted sample against a single enrolled template. FAR and FRR can be thought of as decision error rates. Decision errors are due to matching errors or image acquisition errors (or, with some systems, binning errors). How these errors combine to form decision errors depends on (a) whether one-to-one or one-to-many matching is required; (b) whether there is a positive or negative claim of identity; and (c) the decision policy, e.g. whether the system allows multiple attempts.

2.6.7 Threshold

Both FAR and FRR depend on the setting of the decision threshold set on any biometric system. This threshold is an adjustable decision threshold for the similarity of a scanned feature to a saved reference feature. The value of threshold is dependant on the target application, and also goes some way to determining the usability of the biometric system. Since FAR and FRR are related depending on the value of threshold chosen, it means that if a low FAR is required then this will lead to a higher proportion of FRR errors. The converse also holds true.

2.7 Additional Performance Measures

Apart from these performance metrics, there are derived metrics generated from the analysis and comparison of FAR/FRR and FTE.

2.7.1 Equal Error Rate (EER)

This is the value of the threshold at which FAR and FRR are equal. If a biometric systems threshold is set to this point then the same number of people will be falsely rejected as falsely accepted. This is also referred to as the crossover rate or crossover equal rate (CER).

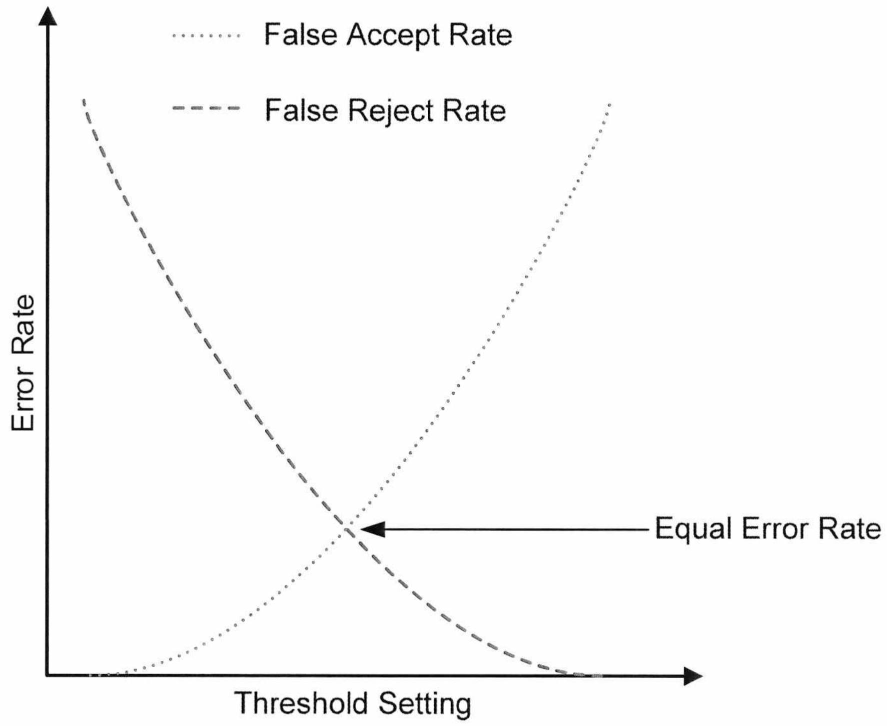


Figure 2.3 Equal error rate graph

2.7.2 Receiver Operating Characteristic (ROC) graphs

This type of graph is a method to summarise the performance of pattern matching systems. The ROC curve plots the FMR on the x axis and FNMR on the y axis as a function of the decision threshold. This type of graph is useful because it allows performance comparison of different systems under similar conditions, or of a single system under differing conditions.

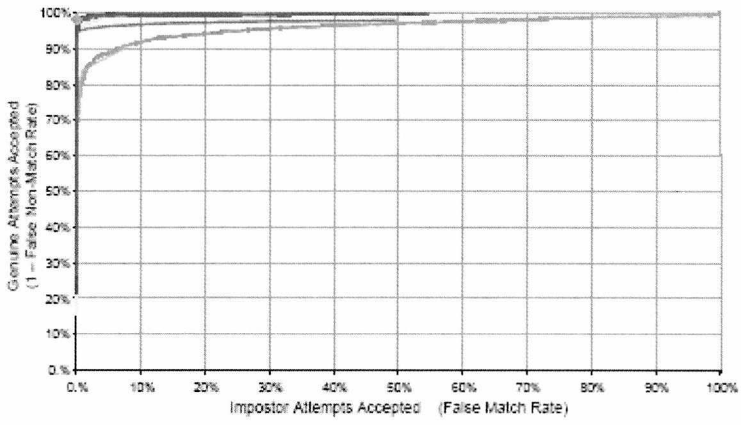


Figure 2.4 ROC curve [Mansfield and Wayman, 2000]

2.7.3 Detection error trade-off (DET)

A DET curve plots error rates on both axes, giving uniform treatment to both types of error, and represent a modified ROC curve. By using logarithmic axis the plot becomes spread out and can help distinguish different well-performing systems more clearly. DET curves can be used to plot matching error rates (false non-match rate against false match rate) as well as decision error rates (false reject rate against false accept rate).

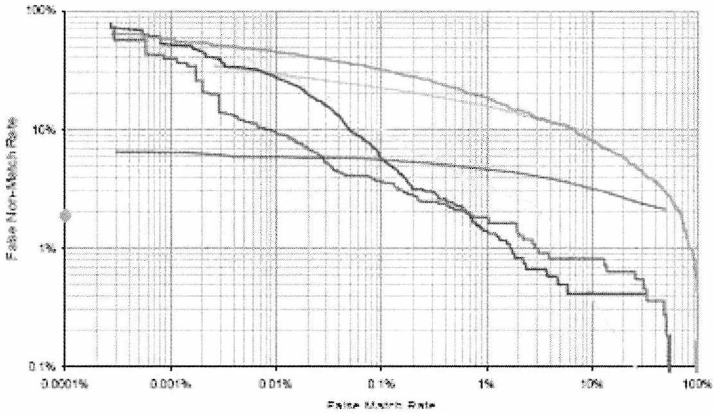


Figure 2.5 DET curve [Mansfield and Wayman, 2000]

Matching algorithm performance for each biometric system, over a range of decision criteria, is shown in Figure 2.5. The lower and further left on the graph a curve lies, the better the performance. The node on each curve shows performance at the default decision threshold. The leftmost point on each curve represents a single false match in the total number of cross-comparisons made.

2.7.4 Failure to Acquire (FTA)

This is when the attempt of the user to donate an image of sufficient quality fails. This can occur for a multitude of reasons, poor lighting in the case of a facial biometric system, insufficient pressure being applied to a finger print sensor etc. This figure may be included in the figure of FRR and may not be stated explicitly by the biometric system.

2.7.5 Throughput

This is an important measure as it determines how quickly the system can be used. It is defined as the average transaction time for all transactions. This begins from the time of initial contact with the system to the time of acceptance or rejection, this also includes FTA conditions and repeat attempts for example.

2.8 Uses of Biometrics

Biometric technologies are becoming the foundation of an extensive array of highly secure identification and personal verification solutions. As the level of security breaches and transaction fraud increases, the need for highly secure identification and personal verification technologies is becoming clear. Biometric technologies have become much more prominent since the tragic events of September 11 2001, and biometric systems are being seen as a more attractive as a viable and effective security tool.

Biometric-based solutions are able to provide for confidential financial transactions and personal data privacy. The need for biometrics can be found in local governments, in the military, and in commercial applications. Enterprise-wide network security infrastructures, government IDs, secure electronic banking, investing and other financial transactions, retail sales, law enforcement, and health and social services are already benefiting from these technologies.

Biometric-based authentication applications include workstation, network, and domain access, single sign-on, application logon, data protection, remote access to resources and transaction security Trust in these electronic transactions is essential to the healthy growth of the global economy. Utilised alone or integrated with other technologies such as smart cards, encryption keys and digital signatures, biometrics are set to pervade nearly all aspects of the economy and our daily lives. Utilising biometrics for personal authentication is becoming convenient and considerably more accurate than current methods (such as the utilisation of passwords or PIN's). This is because biometrics links the event to a particular individual (a password or token may be used by someone other than the authorised user), is convenient (nothing to carry or remember), accurate (it provides for positive authentication), can provide an audit trail and is becoming socially acceptable and inexpensive.

Biometric technology has made the jump from the big screen to the desktop, although as yet not quite into our daily lives. The range of biometric devices is becoming staggering as well as the level of biometric device integration, laptop computers are being built with incorporated fingerprint sensors to replace the more common log on procedures. Even the mobile phone is seen as a potential biometric platform, as recently it has been demonstrated that multimodal biometric technologies can be integrated into such a device to enable secure payment for services amongst other things [SSM-HESY, 2003].

The biometric industry is predicted to grow substantially over the next couple of years as major investment by the United States and other governments into the procurement and deployment of biometric systems for use in border control, driver's licenses, airport and transport worker security, traveller authentication to name but a few. Whether this predicted growth in the biometrics sector materialises remains to be seen, although the future is looking bright for biometrics.

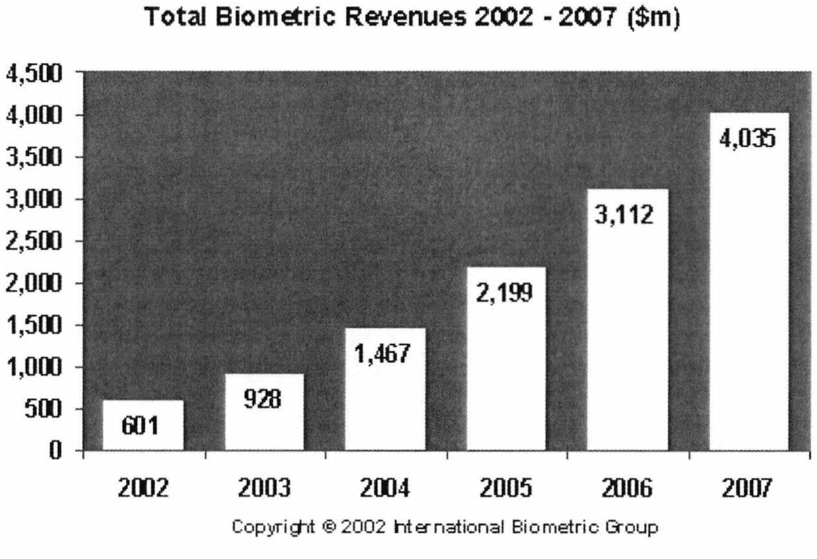


Figure 2.6 Biometric Revenues

2.9 Biometric modalities

This section will investigate three mainstream modalities, these modalities were employed in the IAMBIC system detailed in Chapter 4.

2.9.1 Fingerprint

Fingerprints have been used for verification and identification purposes since the dawn of civilisation. It is the oldest and most commonly accepted form of biometric technology. Over a hundred years ago, both the United States and Europe began documenting the use of fingerprints for identification and verification purposes. After all this time, and millions of fingerprints later, no two identical fingerprints have ever been found. Based on this kind of hard physical evidence, it is safe to say that fingerprints are truly a unique human characteristic. No other biometrics technology can boast this level of scientific history and evidentiary support. Accordingly, its advantage over other biometric solutions lies in its historically and scientifically proven accuracy, reliability, convenience, user acceptance and familiarity.

Any fingerprint may contain a number of distinctive features some of which are shown in Figure 2.7. From top left to bottom right: loop, double loop, central pocket loop, plain whorl, plain arch, and tented arch.

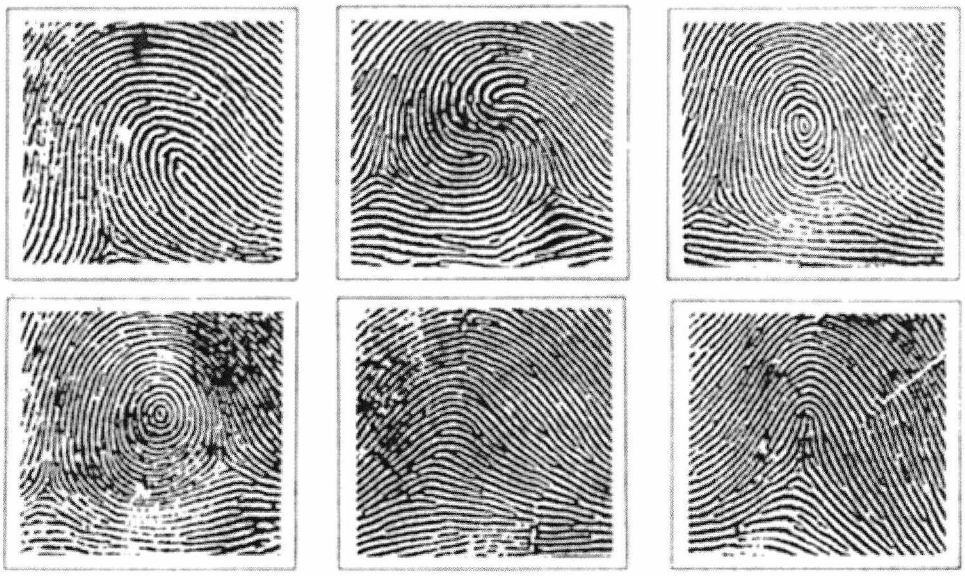


Figure 2.7 Finger patterns

The human fingerprint contains various types of ridge patterns, traditionally classified according to the Henry system [Henry, 1900]. Loops make up nearly 2/3 of all fingerprints, whorls are nearly 1/3, and 5-10% arches. This represents exclusively global features of the fingerprint [Hong & Jain, 1999].

Basic and composite ridge characteristics (minutiae)

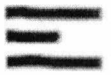
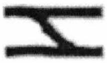
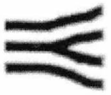
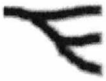
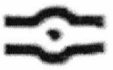
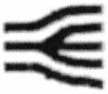
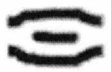
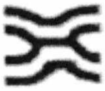
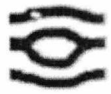
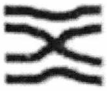
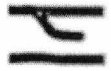
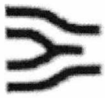
Minutiae	Example	Minutiae	Example
ridge ending		bridge	
bifurcation		double bifurcation	
dot		trifurcation	
island (short ridge)		opposed bifurcations	
lake (enclosure)		ridge crossing	
hook (spur)		opposed bifurcation/ridge ending	

Figure 2.8 Galton features

The bases for most finger scan authentication systems are minutiae (Figure 2.8), and consist of predominantly local features. They comprise of discontinuities that interrupt the otherwise smooth flow of ridges. Codified in the late 1800's as Galton features [Galton, 1888], minutiae are at their most rudimentary ridge endings, the points at which a ridge stops, and bifurcations, the point at which one ridge divides into two. Many types of minutiae exist, including dots (very small ridges), islands (ridges slightly longer than dots, occupying a middle space between two temporarily divergent ridges), ponds or lakes (empty spaces between two temporarily divergent ridges), spurs (a notch protruding from a ridge), bridges (small ridges joining two longer adjacent ridges), and crossovers (two ridges which cross each other).

Other features are essential to finger-scan authentication. The core is the inner point, normally in the middle of the print, around which swirls, loops, or arches centre. It is frequently characterised by a ridge ending and several acutely curved ridges. Deltas are the points, normally at the lower left and right hand of the fingerprint, around which a triangular series of ridges centre.

The most rudimentary minutiae are the point at which the ridge ends and the point at which a bifurcation begins, these are the points used in most applications. In compliment to the placement of the minutiae the angle of the minutiae is usually used. Where a ridge ends, its direction at the point of termination establishes the angle (curved ending require a more complicated set of rules). This angle is taken from a horizontal line extending rightward from the core, and can be up to 359°. Some vendors classify minutia by type and quality. This method has an advantage in that the searches can be quicker as notable minutia can be distinctive enough to lead a match. A vendor can also rank high versus low quality minutia and discard the latter. Some vendors do not employ this technique because of the wide variation from print to print, even on successive submissions and also the measuring of quality may introduce an unnecessary level of complication.

Approximately 80 % of biometric vendors utilise minutiae in some fashion. Those who do not utilise minutia use pattern matching, which extrapolates data from a particular series of ridges. This series of ridges used in enrolment is the basis of comparison, and verification requires that a segment of the same area be found and compared. The use of multiple ridges reduces dependence on minutiae points, which tend to be affected by wear and tear.

Minutiae-based automatic identification techniques first locate the minutiae points and then match their relative placement in a given finger and the stored template. A graph-based representation [Eshera & Fu, 1984][Gold & Rangarajan, 1996], constructs a nearest neighbor graph from the minutiae patterns. The point pattern based representation [Ranade &.Rosenfeld, 1993][Jain et al, 1997] considers the minutiae points as a two-dimensional pattern of points. Correlation-based techniques [Driscoll et al, 1991] consider the gray level information in the fingerprint as features and match the global patterns of ridges and valleys to determine if the ridges align.

The global representation schemes of the fingerprint used for classification can be broadly categorised into three main categories:

- Knowledge-based
- Structure-based
- Frequency-based

The knowledge-based fingerprint representation technique uses the locations of singular points (core and delta) to classify a fingerprint into five major classes [Jain et al, 1999]. A knowledge-based approach tries to capture the knowledge of a human expert by deriving rules for each category by hand-constructing the models and therefore, does not require training. Structure-based approach uses the estimated orientation field in a fingerprint image [Chong et al, 1997]. Frequency-based approaches use the frequency spectrum of the fingerprints for representation [Fitz & Green, 1996].

There are a number of differing technologies employed to acquire fingerprint images, depending on the vendor a different technique may be employed. Optical technology is the oldest and most commonly used. The finger is usually placed on a hard plastic window then a charge coupled device (CCD) is used to digitise the image of the finger comprising of dark ridges and light valleys. Optical devices are popular with vendors as it is a mature technology and fairly resistant to environmental temperature changes. They are cheap to manufacture and can provide sufficient resolution for the imaging of the finger. There are some problems with this technology though, latent prints (images of previous users fingerprint images) left on the plastic sensor can cause acquisition problems as they interfere with the imaging process. Another issue is with the size of the sensor required to acquire a reasonable image.

Silicon technology was introduced in the late 1990's and since then has rapidly gained acceptance. The way in which this method works is based upon DC capacitance. The finger acts as one side of the capacitor and the sensor acts as the other. The capacitance between the finger and the sensor is digitised and converted into a greyscale image. It has been proven that silicon can provide more detailed images with a smaller surface area of sensor when compared to optical methods. The reduced size of the sensor has enabled these silicon devices to be integrated into devices that were too small for optical technology to be employed. Devices such as credit card sized smart cards that have been developed by Infineon known as FINGERCARD [Infineon, 2001] which incorporate the sensor and associated electronics enabling registration and verification to be performed on the card itself. Although with this decrease in sensor size, finger placement during enrolment and subsequent verification attempts becomes paramount. One of the main drawbacks of this technology is the durability of the silicon sensor itself, it is the responsibility of the vendors to produce a viable solution to this problem before the use of silicon sensors becomes widespread.

Another technology starting to be used for fingerprint imaging is that of ultrasound. This technique is considered to be the most accurate. It works by transmitting acoustic waves and then measuring the distance between the platen and the finger. One of the advantages of this technique is that the acoustic waves are capable of penetrating dirt and residue on the finger and also the platen, one of the main drawbacks of optical based systems. Although this technology is still in its infancy it has several strengths, the sensing area is large and easy to use and it negates the problems silicon sensors have acquiring images under less than ideal conditions.

2.9.2 Face

As early as 1878 Francis Galton [Galton, 1878] proposed techniques for facial recognition, This work and subsequently the majority of subsequent work concentrated on detecting important facial features or key points. Through the measurement of the relative distance between these facial key points a feature vector can be built to describe each face. Facial recognition is a fairly young technology when compared to some of the other biometric modalities. Although research into this field has been going on for decades it has been during the last 10 to 15 years that the greatest advances have taken place.

The process by which this technique works is as follows, the first step is to acquire an image of the subjects face. This can be achieved through the use of a camera attached to a Personal Computer (PC) the resolution of the image can be as low as 320 * 240 at 8 bits per pixel (greyscale). The performance of the facial recognition system is highly dependant on the quality of the image captured, and also the particular algorithm employed. Images captured using high quality cameras are far more likely to lead to enhanced system performance, poor quality images will tend to result in problems with enrolment and matching. The software then analyses certain facial features such as:

- Distance between eyes
- Width of nose
- Depth of eye sockets
- Cheekbones
- Jaw line
- Chin

Verification of the individual is then performed by matching the extracted features to the template stored in the database. There are currently four main techniques being used for feature extraction from facial images, these are:

- Eigenfaces
- Feature analysis sometimes known as Local Feature Analysis (LFA)
- Neural Network
- Automatic Face Processing

2.9.2.1 Eigenfaces

This work was pioneered by Turk and Penttland who in 1991 [Turk and Penttland, 1991] used Principal Component Analysis (PCA) for face recognition and detection. The resulting principal components of a facial image were termed an eigenface. The work concentrated on a two dimensional approach to facial recognition. The eigenface approach does not try to model a face (e.g. 2 eyes, nose, and mouth) as other work has done, instead images are reduced to find the principle components that characterise the face. In mathematical terms, the eigenface method finds the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images, treating an image as a point (or vector) in a very high dimensional space. The eigenvectors are ordered in such a way that, each one accounts for a different amount of variation among the facial images.

Extensive further work has been performed in this field of eigenfaces, which seek to address some of the initial problems reported by Turk and Penttland. These problems include the decrease in recognition rate when dealing with facial deformities, such as scarring. Also recognition rates can be affected by different light levels and the pose of the subject. In order to achieve optimum recognition the subject must present a frontal view.

2.9.2.2 Local Feature Analysis (LFA)

This technique is perhaps the most widely used face recognition method. It is derived from eigenfaces but it is able to cope with changes in facial aspect or appearance. LFA can best be described as an "irreducible set of building elements" [Penav and Atick, 1996]. LFA uses a multitude of features extracted from different facial features and also utilises the relative location of these features. The features that are extracted form the building blocks and their type and arrangement are used for identification and verification purposes. Since movement of one feature is likely to influence nearby features it is this mechanism that can accommodate the changes in facial aspect. An advantage of LFA over eigenfaces is that LFA can manage to work with facial images that are not square to the camera, and in fact it is quoted that LFA will still work with up to 25° of horizontal plane movement and about 15° in the vertical plane. LFA is employed by Visionics [Identix, 2001] a market leader in face recognition software, and was employed as part of the IAMBIC system, introduced in Chapter 4.

2.9.2.3 Neural Networks

In this method an algorithm is employed to determine the likeness of the unique global features of live 'donated' faces versus the enrolled faces, this is performed using as much of the facial image as feasibly possible. By using these features from both images a voting technique is engaged by the neural network to determine whether there is a match. If an incorrect vote is recorded resulting in a false match, the matching algorithm will adjust the weight it gives to certain facial features. This particular method can theoretically improve the ability to identify faces in difficult conditions.

2.9.2.4 Automatic Face Processing

This particular technique uses a simple technology, it involves calculating distances and the ratios of distance between some of the more easily obtained facial features such as eyes, corners of the mouth and the end of the nose. This particular technique is the least robust of the methods presented here.

2.9.3 Voice

Voice recognition works by utilising the distinctive aspects of the voice to verify the identity of individuals. The human voice is generated by the resonance in the vocal tract, the length of this tract and the shape of the mouth and nasal cavities affect the voice. It is this variability that enables the voice to be used as a biometric.

A profile of a subject's speech is digitised to produce a model voice print similar to a template. Each spoken word is broken down in segments, either sub-word word like syllables, phonemes, triphones or similar units of sound, composed of several dominant frequencies called formants, these remain relatively constant over that particular segment. Each segment has three or four dominant tones that can be captured in digital form and plotted on a table or spectrum. It is this table of tones that forms the speaker's voice print.
Speaker verification can be divided into text dependant and text independent methods. Text dependant methods require the user to repronounce specified utterances, usually containing the same phrases as in the training data, in text independent systems the user is free to speak as they choose.

It has emerged over the last twenty years that probabilistic methods have materialised as the method of choice for speaker verification tasks. Text-dependent methods are usually based on template-matching techniques. In this approach, the input utterance is represented by a sequence of feature vectors, generally short-term spectral feature vectors. The time axes of the input utterance and each reference template or reference model of the registered speakers are aligned using a Dynamic Time Warping (DTW) algorithm and the degree of similarity between them, accumulated from the beginning to the end of the utterance, is calculated. The Hidden Markov Model (HMM) can efficiently model statistical variation in spectral features. Therefore, HMM-based methods were introduced as extensions of the DTW-based methods, and have achieved significantly better recognition accuracies [Naik et al, 1989].

One of the most successful text-independent recognition methods is based on Vector Quantization (VQ). In this method, VQ codebooks consisting of a small number of representative feature vectors are used as an efficient means of characterising speaker-specific features. A speaker-specific codebook is generated by clustering the training feature vectors of each speaker. In the recognition stage, an input utterance is vector-quantized using the codebook of each reference speaker and the VQ distortion accumulated over the entire input utterance is used to make the recognition decision.

A method using statistical dynamic features has recently been proposed. In this method, a multivariate auto-regression (MAR) model is applied to the time series of cepstral vectors and used to characterize speakers. It was reported that identification and verification rates were almost the same as obtained by an HMM-based method [Griffen et al, 1994].

Voice recognition can utilise any audio capture device, including mobile and land telephones and PC microphones. The performance of voice recognition systems can vary according to the quality of the audio signal as well as variation between enrolment and verification devices, so acquisition normally takes place on a device likely to be used for future verification.

2.9.4 Market share of current biometric systems

Figure 2.9 illustrates the state of the biometrics market in 2003. It illustrates the market share for each particular modality.

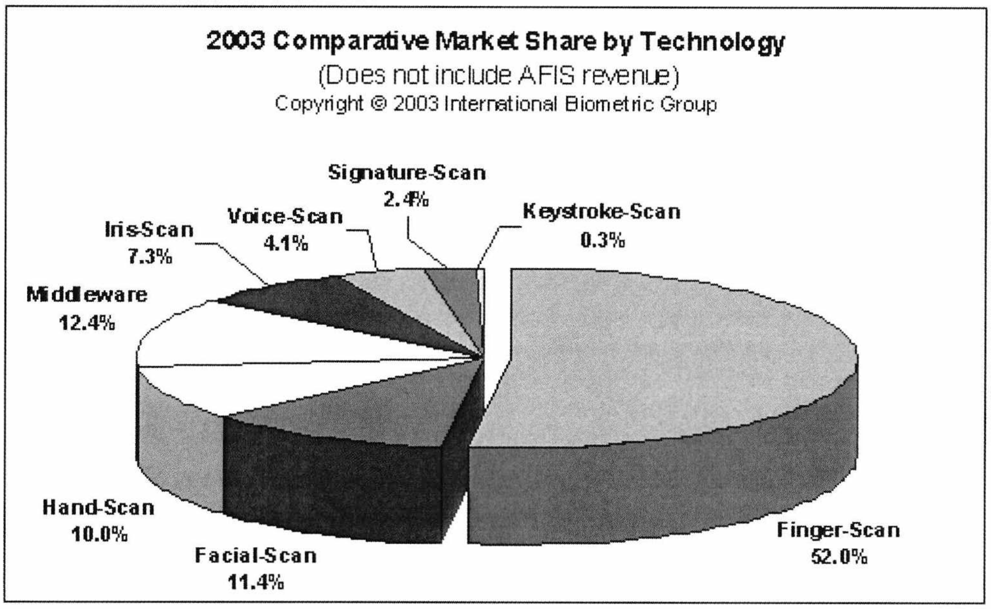


Figure 2.9 Market share by technology ©IBG

2.10 The problem with biometrics

It would appear from first glance that the power of biometrics is an ideal solution to the problem of identity recognition. From the point of view of the user, their use is more convenient that the current types of authentication i.e. password or PIN's. In the United Kingdom alone it is estimated that the cost of a lost or forgotten password to a company is approximately £15, and constitutes the largest component in helpdesk IT calls [RSA, 2004]. Biometric systems could eliminate this issue whilst still providing a similar security level. From the service providers viewpoint biometrics provide an irrefutable audit trail as it places the user at the point of authentication.

Unfortunately there are a number of issues with today's biometric solutions. Some of these issues are usage based and technology based and others are privacy and trust based issues. In the following section a number of these issues will be examined in order to illustrate some of the problems that exist in the biometric industry as a whole that are delaying the widespread adoption of biometric technology.

2.10.1 Privacy and Trust Issues

There have been concerns raised over the uses and misuses of biometrics and this has led to the discussion on whether biometric technology is privacy enhancing or privacy threatening. In [Woodward, 1999] the fundamental question is whether a user has full control over his data, knowing when, where, and why a submitted biometric feature is used. The issue here is that the highly personal nature of the biometric data heightens fears in users of possible compromise and reuse. Wirtz [Wirtz, 2000] summarises the privacy concerns facing biometrics systems.

- Unauthorised access to biometric data
- Unauthorised disclosure of biometric data to third parties
- Use of biometric data for other than intended purpose
- Collection of biometric data without the knowledge of the individual

Obviously these concerns are pivotal to the success of biometric systems and in order to combat these issues the IBIA (International Biometric Industry Association) [IBIA, 1988] was founded in Washington, D.C in 1988. This organisation is concerned with data protection issues and ID systems used in biometrics in particular from the viewpoint of the consumer, and are actively seeking to address these issues.

2.10.2 Template Aging

One problem with biometrics is the degree by which particular physiological traits are temporally invariant. Some modalities show slower variance over time than others. The problem with variance in the biometric feature is that template management must take into account this aging process. There is a term used in biometric systems to describe this variation of physiological traits with age and is termed ‘template aging’. This is the condition where the biometric feature being measured has changed sufficiently from the template which the user has recorded. This can lead to verification failures and necessitates the need for the user templates to be updated regularly in order to alleviate this problem. Also the choice of particular modality is important as a relatively invariant trait would be more advantageous than one which was highly variant. The aspect of template aging is usually glossed over by the manufacturer of a particular device so it is the responsibility of the biometric system developer to take into account this phenomenon.

Although there have been no widespread investigation into the phenomenon of template aging a number of studies are underway into this aspect [Kitchel and Elliot, 2004] and underlying techniques for updating templates [Uludag et al, 2003]. A heuristic has been proposed for the value of these periods that should be employed when testing biometric devices. These figures are generally based on the time it would take for the body part to heal after sustaining an injury [Mansfield and Wayman, 2000] and are illustrated in Table 2.1.

Modality	ΔT Suggested
Fingerprints	14 – 21 days
Facial	30 – 60 days
Retina/Iris	2 – 3 days

Table 2.1Template aging heuristic

2.10.3 Usage Issues

The aspect of intrusiveness of a particular biometric is an important issue to its usability and general acceptance. There are certain modalities that have inherent negative user perception for various reasons. Fingerprints for example are perceived by many people to have criminal connotations as they are used by the law enforcement agencies to identify criminals, eye based biometrics may have the problem where users are reluctant to put their eye near the sensing device for fear of optical damage. Obviously some of these matters can be alleviated through user education in the particular modality.

2.10.4 Application Programming Interface (API) Issues

Another problem facing the biometric industry as a whole is the propriety nature of many of the devices and associated software on the market. There exists a clear and present need for standardisation for application programming interfaces and also image and template formats. A number of these standards have been ratified and can be found in use in some of the current biometric products. E.g. BioAPI [BioAPI, 2003] exists to ensure that compliant biometric devices have an all encompassing interface to standard methods to acquire and process biometric samples. Another standard the Common Biometric Exchange File Format (CBEFF) [Podio et al, 2001] is proposed as a container format for biometric information to provide portability between vendors. X9.84 [ANSI, 2003] is concerned with the security and management of biometric data, including secure transmission and storage, and security of the surrounding hardware.

Work in this area is still ongoing and new standards are in development. Also compliance is not mandatory to any of these standards as a whole and it at the discretion of the vendor as to whether they support a particular standard or not. Without adherence to these unifying standards the biometrics industry will remain fragmented.

2.10.5 Biometric system performance

Performance of biometric systems is dependant on the modality itself and also the user. Unfortunately when choosing a biometric modality for a target application one of the most compelling features is the quoted performance of the biometric device or software. Obviously the manufacturer is trying to sell the device, so when performance figures are quoted they may be 'optimistic' in a real world setting to say the least. This is a problem that is accepted by the biometric industry and the need for standardised testing of biometric products is well recognised. In work conducted for this thesis [Fairhurst et al, 2002] it was observed that the performance of particular biometrics differed greatly from what one might expect to achieve given the performance figures from the manufacturer.

A user's biometric performance can improve over a period of time, this process known as habituation is expected to occur with the user towards the biometric modality over a certain period of time in which the device is used. This habituation process represents the fact that the user is becoming experienced using the system and will generally give better samples, this will lead to a decrease in the rate of FRR.

2.10.7 Current biometric system performance

The following tables and graphs are from a biometric vendor testing program conducted by the National Physics Laboratory in 2001 [Mansfield et al, 2001]. Face, Fingerprint, Hand Geometry, Iris, Vein and Voice recognition systems were tested for a scenario of positive identification in a normal office environment, with cooperative non-habituated users. The evaluation was conducted in accordance with the "Best Practices in Testing and Reporting Performance of Biometric Devices" [Mansfield and Wayman, 2000].

System	Failure to enrol rate (%)
Face	0.0
Fingerprint – Chip	1.0
Fingerprint – Optical	2.0
Hand	0.0
Iris	0.5
Vein	0.0
Voice	0.0

Table 2.2 Failure to enrol rates [Mansfield and Wayman, 2001]

System	Failure to acquire rate (%)
Face	0.0
Fingerprint – Chip	2.8
Fingerprint – Optical	0.8
Hand	0.0
Iris	0.5
Vein	0.0
Voice	2.5

Table 2.3 Failure to acquire rate [Mansfield and Wayman, 2001]

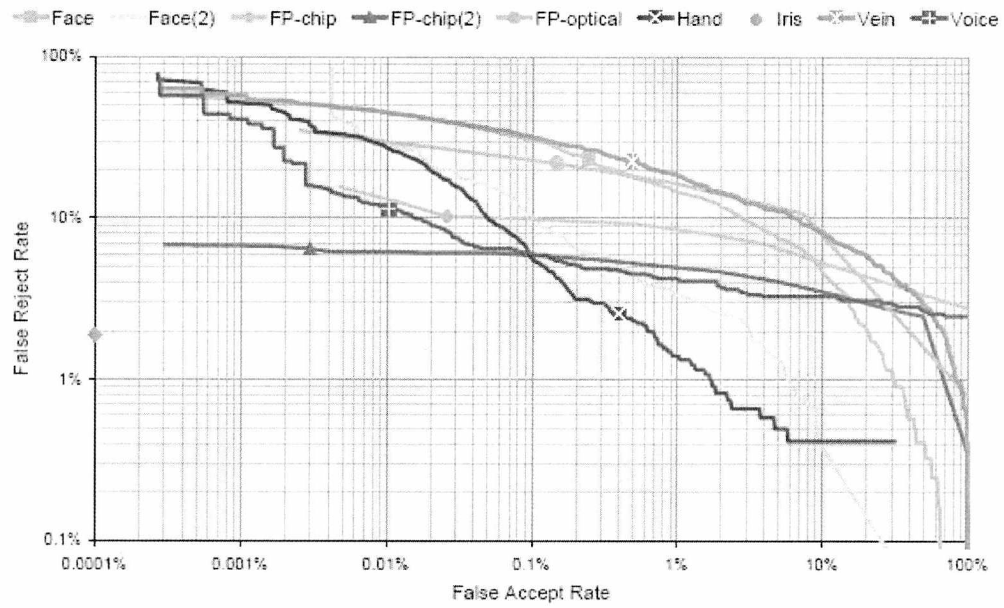


Figure 2.10 DET curve single attempt [Mansfield and Wayman, 2001]

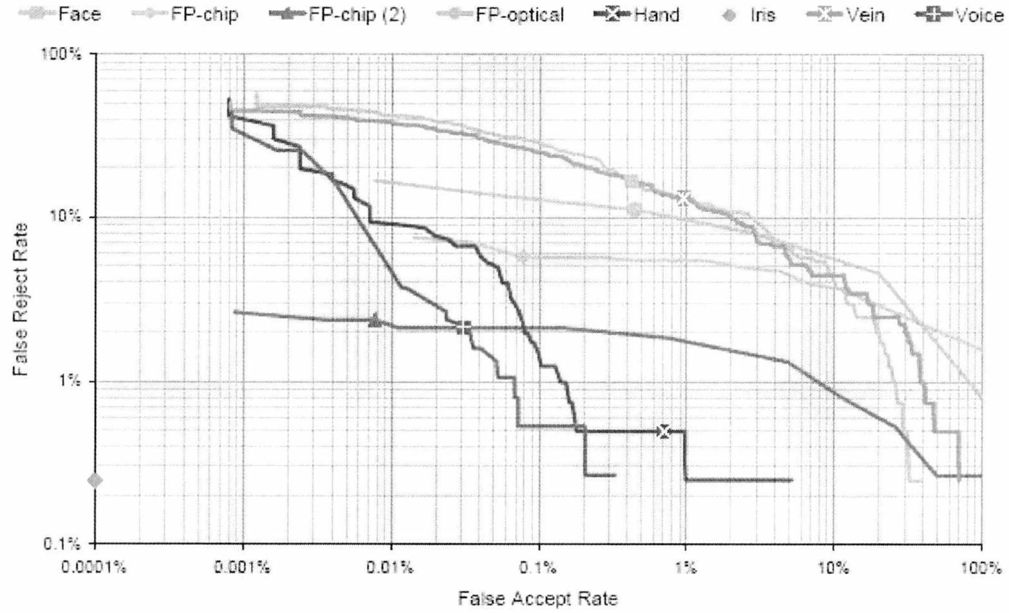


Figure 2.11 DET curve "best of three" attempts [Mansfield and Wayman, 2001]

2.10.6 "Spoofing" biometrics

Spoofing a biometric involves using artificial attempts, fakes and mimicry to fool a biometric device into believing a real sample has been presented to the biometric sensor.

A number of tests both academic and from the media have been performed on a number of modalities to test this aspect of biometric spoofing [Matsumoto et al, 2002][Thalheim et al, 2002]. In these tests it has been shown that fake fingerprints, facial images and also iris images can be used as valid biometric samples. These samples were then used to generate valid enrolment templates and also to verify the enrolled subjects. The severity of effort needed to perpetrate these attacks ranged from very simple, in the case of one fingerprint modality all that was required was to breathe on the sensor to reveal the latent image, to a more complex approach of manufacturing false fingerprints from gelatine. In the case of iris technology it was shown that false iris images could be superimposed upon images of human eyes and then used to spoof the biometric system.

Liveness testing is the process by which a biometric system checks to ensure the sample that is being donated to the system is from a live human being. It has long been thought that the majority of biometric systems are capable of detecting liveness in the given donated samples. Requiring a live biometric sample to be donated to a system ensures that repudiation of a transaction becomes difficult, an important feature for security and accountability. For some modalities a challenge response protocol can be used in order to ascertain liveness. E.g. In the case of a facial biometric the subject could be asked to blink or smile.

This aspect of "spoofing" (using fake samples to circumvent biometric systems) is of very serious consequences to a biometric system and some of the issues surrounding this spoofing include the following points.

- An attacker could attempt to penetrate existing biometric security by using a 'fake finger' in order to access resources.

- By using a fake biometric sample to enrol in the system, the same enrolment template could feasibly be used by many people, totally negating the strong security aspects of a biometric system.

Figures 2.12 and 2.13 illustrate results from a trial conducted at The Biomedical Signal Analysis Laboratory at West Virginia University [Parthnasardhi et al, 2002] to investigate this aspect of spoofing fingerprint devices. The results are disturbing as they indicate that spoofing is not limited to any one particular technology process but is a widespread issue across all the fingerprint devices under test. The problem of this 'fake finger' approach to fooling fingerprint biometric devices has led to calls for the limitation of the use of possible applications of this particular biometric [van der Putte, 2001]. This is especially prevalent considering how large the finger based biometric market share is (Figure 2.9).

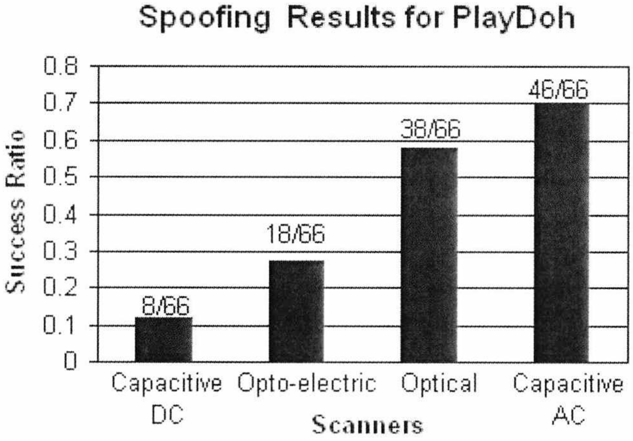


Figure 2.12 Spoofing results for PlayDoh [Parthnasardhi et al, 2002]

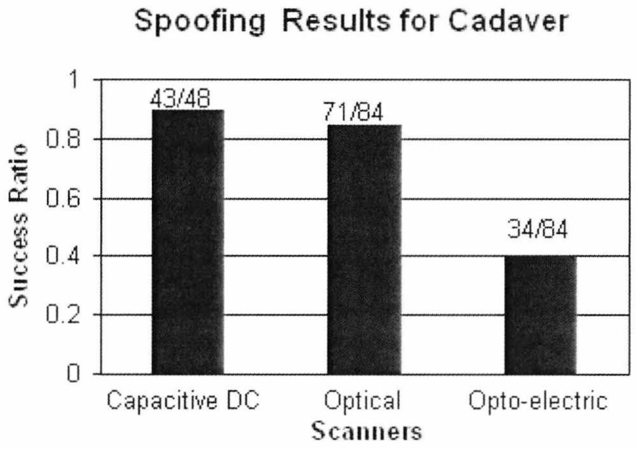


Figure 2.13 Spoofing results from cadaver [Parthnasardhi et al, 2002]

The issue of liveness testing is a key factor to the adoption of a number of biometric systems, if biometrics are to be seen as the solution for a large number of application domains, such as border control, civil identification and network security to name a few. Then if these systems can be circumvented so easily by in some cases extremely rudimentary techniques then really biometric systems cannot be seen as a valid security solution until this issue had been addressed.

2.10.8 Security and Trust

There is also a security aspect to this problem of employing devices attached to a host computer. There needs to be a mechanism of trust between the system and the input device. If the input device itself can be tampered with then there can be no trust throughout the rest of the system. Smart cards which can capture and match on board go some way to alleviate this problem of ensuring that the system has not been tampered with.

Aspects of trust are also relevant for the location of the authentication software as well. There must be adequate security throughout the system and communication channel if authentication is being performed server side in a client server application. Also if a client is untrustworthy then there is no point attempting to authenticate a user at that computer.

2.10.9 Multi-Modal Biometrics

Since the performance of many of the biometric modalities is less than ideal there has been a case proposed to incorporate a number of modalities in a biometric system to further enhance the security of the system and also the flexibility the system offers to the user. These systems are known as multi modal biometric systems. There are broadly three types of multi modal biometric systems.

1. **Either/Or Multimodality**

   This type of system supports a number of modalities, however only requires verification to be performed through a single modality. In order to use this system the user must enrol in all the modalities that are employed by the application, even though only one will be utilised to authenticate the user.

2. **Asynchronous Multimodality**

   This system requires that the user perform verification over a number of modalities in sequence, i.e. a user is only successfully verified after passing all the modalities in turn. The key benefit of this approach is that the risk of a successful impostor attack greatly reduced as the chance of the attacker managing to fool *n* separate modalities is small. This benefit is offset by the reduction in convenience to the user as the time it may take to verify an individual is now the time it takes to verify *n* modalities.

3. **Synchronous Multimodality**

   This involves using a number of biometrics in one authentication transaction, such as using a face and voice system concurrently, this can lead to a reduction in verification time and also increases the resistance to impostor attack.

There are some usage issues inherent with all these multi-modal techniques, primarily it is the problems which some users may find familiarising themselves with the various modalities employed in the system. The intrinsic cost increase of employing a multimodal system is seen as a prohibitive factor and the management of the further complexity of additional modalities contributes to the implementation issues for multimodal system integrators.

Also the types of scenario where these multimodal systems can be deployed are limited. Physical access application requiring a certain degree of throughput may be hindered by an asynchronous multimodal approach. Daugman has voiced concerns over the merits of the combination of biometric modalities versus uni-modal systems.

Daugman argues that "There is a common and intuitive assumption that the combination of different tests must improve performance, because "surely more information is better than less information." On the other hand, a different intuition suggests that if a strong test is combined with a weaker test, the resulting decision environment is in a sense averaged, and the combined performance will lie somewhere between that of the two tests conducted individually (and hence will be degraded from the performance that would be obtained by relying solely on the stronger test)." [Daugman, 2001].

2.10.10 Biometrics at the interface level

Perhaps one of the overlooked areas in the biometric domain is that of the biometric application user interface itself and the underlying mechanisms required for a robust authentication system. A large amount of research and development is ongoing in the fields of algorithm and hardware design for current and future biometric systems. The aspect of the user interface may have a significant impact on improving the robustness and usability of a biometric system. It is seen that by the natural process of habituation that the user can be expected to provide consistent samples of sufficient quality to ensure satisfactory operation of the particular device.

By employing the interface agent paradigm, in which the agent is portraying a tutor and is aiding the user through the donation of biometric samples, it is envisaged that this habituation process can be accelerated. It is also hoped to demonstrate that a biometric system endowed with such an adaptive user interface can be successfully used unsupervised. This is an important feature to illustrate as the deployment of any biometric system that would require a supervisor for any length of time in order to initially train users would have cost implications.

This notion of an 'adaptive interface layer' will be presented used in conjunction with a commercial optical fingerprint device in Chapter 6.

2.11 Conclusions

This Chapter has presented the notion of a biometric and also what a biometric system is. The features that can be employed as a biometric are established and measures that are employed to characterise the performance of biometric systems are introduced. A small subset of the possible biometric modalities are examined in order to establish the underlying mechanisms that enable the particular characteristic to be engaged as a tool for identity authentication. Although biometric systems are seen as a powerful tool in the domain of identity authentication there are a number of issues that are impeding the wider deployment of biometric systems. The issues are documented and some proposals are made in order to address some of the issues raised.

Chapter 3

Software Agents

This chapter presents a review of software agents. Within this chapter the issue of attempting to classify what is an agent is discussed, and various taxonomies are introduced to illustrate the diversity of this field of research. After a classification scheme is introduced each relevant agent typology is examined, key underlying theories and architectures are discussed. Also a number of examples of each particular agent type are presented in order to illustrate the work performed in these fields of agent research. At the end of this chapter a brief introduction is given to Agent Communication Languages (ACL's).

3.1 Introduction

The birth of the notion of an intelligent agent has come from work in the Artificial Intelligence (AI) community. Agents are starting to fill certain application niches, for example 'Clippy' the help system used in Microsoft® Word was developed from work performed in the field of interface agents [Horvitz et al, 1998]. More recently the film Lord of the Rings employed an agent system termed 'Massive' [Regelous, 2000] which has been able to portray large battle sequences using agents to model individual fighters in a virtual battlefield. The prospect of mobile agents presents an opportunity to develop migratory applications, which can provide real benefits over their static counterparts [Lange and Oshima, 1999]. One branch of research is concerned with the development of what is called believable agents, these are being actively used in computer games to control the behaviour of non player characters which hoped to lead to a more immersive experience for the player [Laird, 2001].

Although the spread of agents into everyday life has not been quite as prolific as some early agent researches have predicted, agent research is a vibrant and diverse community which is seeking to drive agency into more application domains as well as consolidating the hold agents have in specific applications.

In this chapter the notion of a software agent will be introduced and the properties and attributes exhibited by these agents will be presented. A brief overview of the research areas within this domain will be examined and some agent topologies will be offered to illustrate some of the issues this field exhibits with the classification of agents. A number of specific agent types are inspected in order to discover the motivation for the development of such an agent and implementation specific examples are cited.

3.2 What is an Agent?

Unfortunately the term 'agent' has no agreed formal definition at the current time. The problem is similar to that in the Artificial Intelligence community where researchers cannot agree on a consensus definition for the term Artificial Intelligence (AI). There are a number of reasons why it has been so difficult to define precisely what an agent is. Firstly, agents researchers do not 'own' this term the same way that fuzzy logicians/AI technicians, for example, own the term ‘fuzzy’ logic. The word agent is used widely in everyday use as in travel agents, estate agents etc. The other problem with the term ‘agent’, is that within the software community the word is really an encompassing term for a diverse body of research and development.

This lack of definition has led some agent researchers to invent more synonyms including knowbots (knowledge based robots), softbots (software robot), taskbots (task based robots), personal agents, autonomous agents, personal assistants to name a few. There are some valid reasons for these synonyms, firstly the fact that agents can exist in different physical guises, those that exist in the physical world such as a factory are called robots, those which exist to perform specific task are called taskbots. The term “autonomous agents” typically refers to robots or mobile agents that operate in uncertain and dynamic environments. Secondly, those agents can play many roles, therefore personal agents or knowbots can have expert knowledge in some specific domain.

It is generally accepted that autonomy is the core notion of agency. One definition of an agent is given thus:

“An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment to meet its design objectives.” [Wooldridge, 1999]

Autonomy can be defined “as the ability that agents are able to act without the intervention of humans or other systems; they will have control both over their own internal state and over their behaviour.” [Casterfranchi, 1995]

3.2.1 Anatomy of an agent

A very basic view of an agent is given in Figure 3.1.

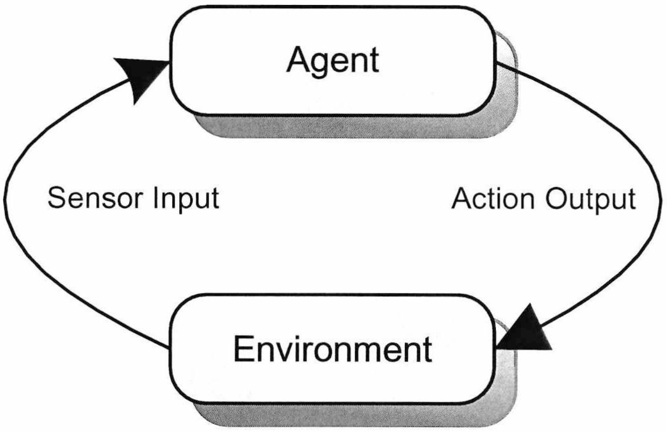


Figure 3.1 Basic agent

Figure 3.1 illustrates the action output generated by the agent in order to effect in which it is situated. It is reasonable to assume that in most domains of reasonable complexity that the agent will not have complete control over its environment. This means that from the point of view of the agent the same action performed twice is apparently identical circumstances might appear to have entirely different effects and in particular may fail to have the desired effect. Therefore, agents must be prepared for the possibility of failure.

An agent will usually have a set of actions available to it. This set of possible actions represents the agent's ability to modify the environment in which it is situated. Not all actions can be performed in all situations, for example an action "lift brick" is only applicable in the situation where the agent can actually lift the brick. Actions must have a pre-condition associated with them that define the possible situations in which they can be applied. One of the main problems facing an agent is deciding which of its actions it should perform in order to satisfy its design objectives. The complexity of the decision making process can be affected by the type of environment the agent is situated in. Comprehensive descriptions of the types of environment an agent may exist in are presented in [Russell & Norvig, 1996].

3.2.2 Agent Theories

Agent theories are regarded as specifications of an agent, the field of research is also responsible for the development of formalisms to represent the properties of agents.

In [Seel, 1989] an agent is described as an entity "which appears to be the subject of beliefs, desires etc", this type of system has been termed an *intentional system* by the philosopher Dennet [Dennet, 1987]. An intentional system describes entities 'whose behaviour can be predicted by the method of attributing belief, desires and rational acumen'. Questions have been raised about validity of attributing these attitudes to artificial agents. McCarthy [McCarthy, 1978] presents a paper that argues that the intentional stance "is most useful when applied to entities whose structure is incompletely known", and in this manner could be suitable for describing agents. In work conducted by Seel [Seel, 1989] and also Rosenschein and Kaelbling [Rosenschein and Kaelbling, 1986] it was demonstrated that simple automata-like objects can be ascribed using these intentional descriptions. The use of these intentional notions provides a set of abstraction tools which can be used to describe, explain, and predict the behaviour of complex systems.

There are two main categories of attitudes which are most suitable for representing agents, these are shown in Table 3.1 [Wooldridge and Jennings, 1995].

Pro-Attitudes	**Information Attitudes**
Desire	Belief
Intention	Knowledge
Obligation	
Commitment	
Choice	

Table 3.1 Agent attitudes

Pro-attitudes are used in some way to guide the actions of the agent, whilst information attitudes are related to the information the agent knows about the world in which it is situated. In order to begin to formalise these notions it is important to develop methods for representing and reasoning about intentional notions.

Early investigation into the use of classical (first order) logics [Benthem, 1983] for reasoning about intentional notions found that in their standard form they are not suitable. The issues with developing a logical formalism for the intentional stance are two fold, the syntactic problem and the semantic problem. There are two approaches used in order to alleviate the syntactic problem. Firstly a modal language can be employed which contains non-truth modal operators, the need for non truth operators is because intentional notions such as belief are not truth functional. Secondly, meta language can be used in order to represent intentional notions using a meta language predicate.

The semantic problem also has two basic approaches. Probably the most widely used approach is to employ possible world semantics proposed originally by Hintikka [Hintikka, 1962]. This technique characterises an agent's beliefs, goals and knowledge into a set of possible worlds with what is termed an accessibility relation between these worlds. One of the main problems with the possible world semantics is that it implies that agents are capable of perfect reasoning. This condition is known as the logical omniscience problem. The alternative to this approach is to use a sentential or interpreted structures approach. In this technique, beliefs are viewed as symbolic formulae, represented in a data structure associated with an agent. An agent can believe in Φ if Φ is present in its belief data structure [Konolige, 1983].

Theories of agency strive to bring together not only the formalisms for knowledge and belief but also must represent the temporal variant aspects of agency and the environment in which they are situated. Also there exists a need to formalise some representation of action in order to describe the capability of the agent to effect its environment. For a complete agent theory not only do these properties require a suitable logic but also a description of how the properties are related. The theory must be robust enough in order to explain how the information and pro attitudes are related, also how the cognitive state of the agent is changing over time and how the environment will affect its 'mental' state. The theory also needs to describe how pro and information attitudes direct the agent to perform actions. It is the formalism of these relationships that is said to be the main obstacle in producing an all encompassing theory of agency.

It can be said that Moore [Moore, 1985] was responsible for pioneering the use of logic for acquiring the aspects of agency. His main focus was the question of what an agent needs to know in order to perform some action. The formalism presented allows for the possibility of an agent having incomplete information about how to achieve some goal and performing actions in order to determine how to achieve it.

One of the most influential agent formalisms was proposed by Rao and Georgeff [Rao and Georgeff, 1991][Rao and Georgeff, 1993][Rao and Georgeff, 1995]. This framework for agent theory was based upon three primitive modalities, beliefs, desires and intentions more commonly known as BDI. This was related to previous work by Cohen and Levesque [Cohen and Levesque, 1990] in which only two attitudes were used, beliefs and goals. Cohen is also known for an influential contribution to the field of agent theory with the formalism that was used to develop a theory of intention. Singh [Singh, 1990] adopts a different approach to the modelling of agents in which a set of logics is presented that can be used to represent communication, knowledge, beliefs and intentions based on a branching time framework.

Development of agent theories in recent years has mainly focused on enhancing the BDI framework, and to provide solutions to some common problems faced by BDI models. One of these problems is that there exists a gap between the logic used in the BDI model and practical systems [Mora et al, 1999][Schild, 2000]. Extensions applied to the BDI model, such as BOID [Broerson et al, 2001] seek to provide further robustness to the already capable model. Work such as [Alechina and Logan, 2002][Roorda et al, 2002] continues to focus on the notion of belief for agent systems whilst Naoyuki and Takata [Naoyuki and Takata, 2002] explore the use of calculus for deduction mechanisms in BDI systems.

3.2.3 Agent Languages

Wooldridge and Jennings describe an agent language as "we mean a system that allows one to program hardware or software computer systems in terms of some of the concepts developed by agent theorists. At the very least, we expect such a language to include some structure corresponding to an agent. We might also expect to see some other attributes of agency (beliefs, goals, or other mentalistic notions) used to program agents" [Wooldridge & Jennings, 1994]

The notion of agent languages has developed from work on concurrent object languages. Hewitt's Actor Model [Agha, 1986][Hewitt, 1977] was one of earliest frameworks to support concurrent objects. It appears that the notion of a self contained concurrently executing object with some internal state not directly accessible to the outside world that responds to messages from other such objects is similar to the concept of an agent as has been defined in Section 3.2.1.

Shoham proposed a new programming paradigm based on societal view of computation [Shoham, 1990][Shoham, 1993]. Shoham proposes that his agent orientated programming (AOP) paradigm is a technique of directly programming agents in terms of the mentalistic, intentional notions that have been used by agent theorist to represent the properties of agents. Shohams AOP system is based upon three components:

1. Logical system for defining the mental state of agents.
2. Interpreted programming languages for programming agents.
3. `Agentification' process for compiling agent programs into low-level executable systems.

Shohams first implementation of this AOP paradigm was the AGENT0 framework [Shoham, 1991]. In this framework an agent's mental state is described by its beliefs, choices, capabilities and commitments, there is also a temporal component present in this framework. The language Shoham developed introduces *epistemic* and *deontic* modal operators for these notions, as in order for agents to reason about these attitudes, operators and logics for their description must be developed. Another key feature of this framework is the vocabulary of interaction between agents based on speech act theory [Cohen and Perrault, 1979][Searle, 1969].

AGENT0 was developed as a prototypical system intended to illustrate the principles of AOP. PLACA (Planning Communicating Agents) [Thomas, 1993] attempts to address some of the drawbacks of the AGENT0 system, such as the inability of agents to plan and also communicate requests for actions via high level goals. Using PLACA an agent is programmed in a similar manner to AGENT0, however, the logical component differs to AGENT0 in that operators are included for planning to do actions and achieve goals.

The main drawback with both the PLACA and AGENT0 systems is that the relationship between logic and interpreted programming language is loosely defined, so in both cases the programming language can be said not to truly execute the associated logic. Concurrent METATEM [Fisher, 1994] proposes to overcome this drawback. A concurrent METATEM system contains a number of concurrently executing agents, which are able to communicate with their peers using a synchronous broadcast message system. Temporal logic is used to program the agent, this specification is the behaviour that the agent should exhibit. It is the execution of this specification that generates its behaviour. In this manner the execution of the agent program corresponds to iteratively building a logical model for the temporal agent specification.

The field of agent orientated software engineering has developed substantially over the last 10 years as researchers strive to develop definitive methodologies and design environment tools. These tools are deigned to provide robust design environments for the development of multi agent system. A number of full blown multi agent development environments have been developed. Examples of these environments include ZEUS [Nwana et al, 1999] and MadKit [Gutknecht and Ferber, 2000]. The goal here is not only to provide systems capable of multi agent system development but also to accurately model agent entities and classes using industry standard tools such as the Unified Modelling Language (UML) [Parunak, 2001][Bauer, 2001][Arai and Stolzenburg, 2002][Heinze and Sterling, 2002].

3.2.4 Agent Overview

In order to realise the functionality of an agent as described in section 3.1.1, all agent entities can be deconstructed into 4 main functional blocks. Figure 3.2 shows these components

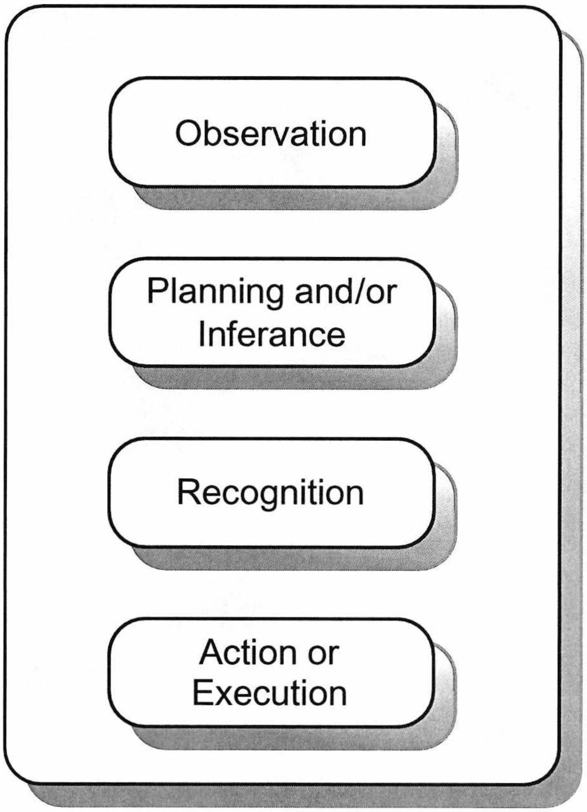


Figure 3.2 Agent functional blocks

These four blocks represent the core components of any agent-based system.

- The observation module is responsible for the acquisition of data from the agent's sensors
- The planning module is provides the generation of goal motivation action within the agent. Some agent structures do not require this module as no explicit planning occurs within the agent
- The recognition module matches the current sensor input to one of the plans that has previously been created within the preceding block
- The action module then performs the selected plan and creates the agent output conditions which are defined in the selected plan

3.2.5 Attributes of agency

As mentioned in Section 3.1 the core notion of agency is generally accepted to be the aspect of autonomy. Many agent researchers have attempted to further classify attributes which are seen to be critical properties of software agents. In this section we shall examine some of these different views on these properties and illustrate some of the difficulties in trying to provide a concrete formal definition of the attributes required for the definition of agency.

In [Franklin and Graesser, 1996] this definition problem is illustrated with a number of formal agent descriptions from various researchers, each of which is subtly different. This indicates the nature of the problem at hand when it comes to providing a specific definition of an agent. There is also much debate on what properties an agent should possess in order to be called an 'agent'.

[Etzioni and Weld, 1995] enumerate a number of attributes a software agent may possess to a greater or lesser degree depending on the requirements of the particular problem the agents is tasked with.

- **Autonomous**: an agent is able to take initiative and exercise no-trivial degree of control over its own actions

    - Goal Orientated: an agent accepts high level requests indicating what a human wants and is responsible for deciding how and where to satisfy the requests
    - Collaborative: an agent does not blindly obey commands, but has the ability to modify requests, ask clarification questions, or even refuse to satisfy certain requests
    - Flexible: the agent's actions are not scripted: it s able to dynamically choose which actions to invoke, and in what sequence, in response to the state of the external environment
    - Self starting: unlike standard programs which are directly invoked by the user, an agent can sense changes to its environment and decide when to act

- **Temporal Continuity**: an agent is a continuously running process, not a "one-shot" computation that maps a single input to a single output, then terminates

- **Character**: an agent has a well defined believable "personality" and emotional state
- **Communicative**: the agent is able to engage in complex communication with other agents, including people, in order to obtain information or enlist their help in accomplishing its goals
- **Adaptive**: the agent automatically customises itself to the preferences of the user based on previous experience. The agent automatically adapts to changes in its environment
- **Mobile**: an agent is able to transport itself from one machine to another and across different architectures and platforms

Whilst the authors agree that no single agent possess all the characteristics detailed above, it is the inclusion of these properties that differentiate agent-based systems from simple software programs.

As can be seen, there are a large number of possible characteristics that may be present in a single agent. It becomes apparent that some form of classification scheme or taxonomy is required rather than to try and classify an agent by its possible attributes. A number of these schemes have been developed by various researchers.

3.2.6 Agent Taxonomies

In [Wooldridge and Jennings, 1995] the scheme of strong and weak agents is proposed. Weak agents are characterised by the following characteristics.

- **Autonomy**: Operate without direct intervention of humans or others of control over their actions and internal state
- **Social ability**: Agents interact with other agents (possibly humans) via some kind of language
- **Reactivity**: Agents perceive their environment (may be the physical world, a user via a GUI, a collection of other agents, internet or all of these combined). They respond in a timely fashion to changes that occur in it
- **Pro activeness**: Agents do not simply act in response to their environment; they are able to exhibit goal directed behaviour by taking the initiative

Strong agents have all the characteristics of weak agents plus are either conceptualised or implemented using concepts more usually applied to humans. These concepts include mentalisitic notions such as knowledge, belief intention, and obligation.

Another method of classifying an agent is proposed from the Distributed Artificial Intelligence (DAI) community, by Moulin and Chaib-draa this technique involves classification of agent by problem solving capacity.

"A *reactive* agent reacts to changes in its environment or to messages from other agents.... An *intentional* agent is able to reason on its intentions and beliefs, to create plans of actions, and to execute those plans.... In addition to intentional agent capabilities, a *social* agent possesses explicit models of other agents." [Moulin and Chaib-draa, 1996]

In [Gilbert et al, 1995] intelligent agents are described as existing in a three dimensional space with the axis representing the properties of agency, intelligence and mobility. The authors go on to classify each of the axes thus:

"*Agency* is the degree of autonomy and authority vested in the agent, and can be measured at least qualitatively by the nature of the interaction between the agent and other entities in the system. At a minimum, an agent must run asynchronously. The degree of agency is enhanced if an agent represents a user in some way... A more advanced agent can interact with... data, applications,... services... [or] other agents. *Intelligence* is the degree of reasoning and learned behaviour: the agent's ability to accept the user's statement of goals and carry out the task delegated to it. At a minimum, there can be some statement of preferences... Higher levels of intelligence include a user model... and reasoning.... Further out on the intelligence scale are systems that *learn* and *adapt* to their environment, both in terms of the user's objectives, and in terms of the resources available to the agent... *Mobility* is the degree to which agents themselves travel through the network... *Mobile scripts* may be composed on one machine and shipped to another for execution... *Mobile objects* are transported from machine to machine in the middle of execution, and carrying accumulated state data with them." [Gilbert et al, 1995]

Figure 3.3 illustrates this taxonomy. Minimum values for each axis occur at the centre of the graph. From this taxonomy intelligent agents exist in the portion of the graph where they must exhibit some form of intelligence greater then just remembering preferences and also as mentioned by Gilbert himself must at least run asynchronously.

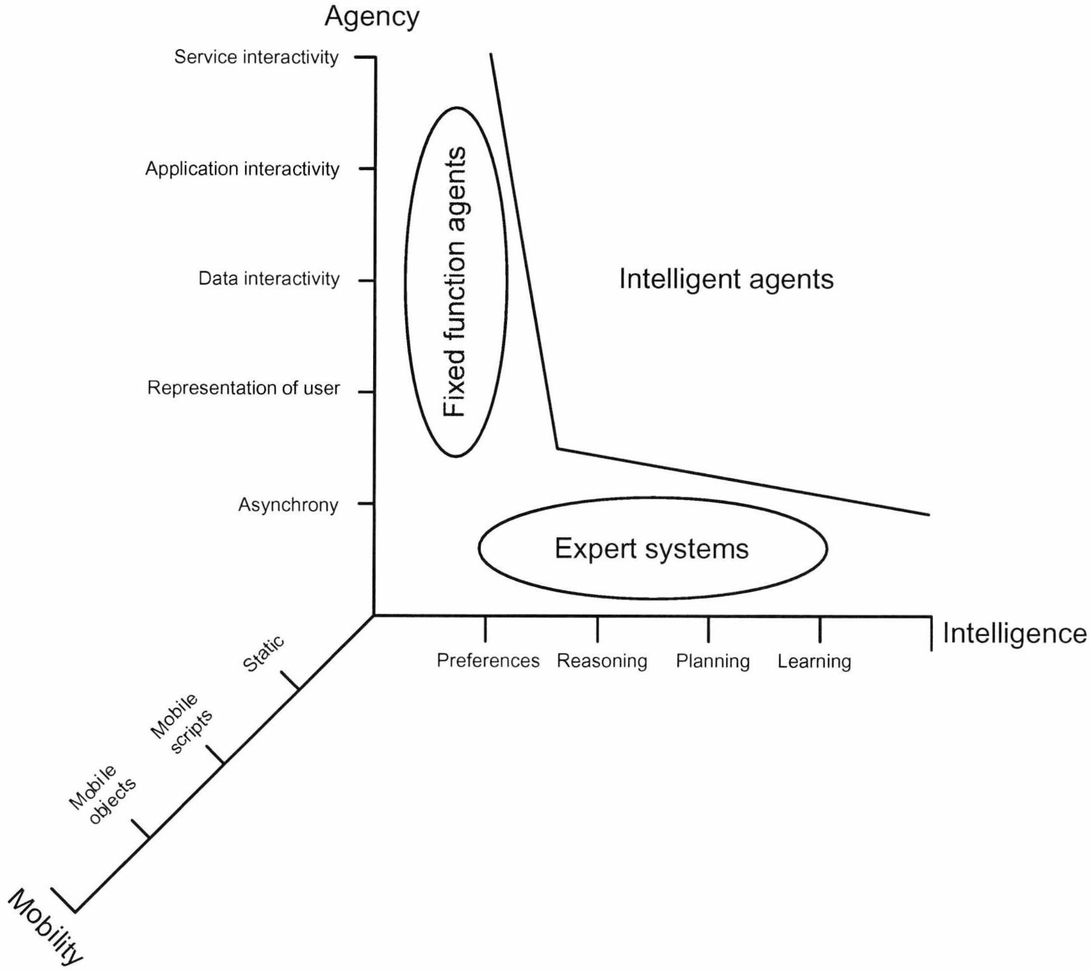


Figure 3.3 Agent taxonomy [Gilbert et al, 1995]

Another type of typology is suggested by [Nwana, 1996] in which agents are classified using other dimensions.

- Mobility, as *static* or *mobile*
- Presence of a symbolic reasoning model, as *deliberative* or *reactive*
- Exhibition of ideal and primary attributes, such as *autonomy, cooperation, learning*
- The roles which an agent may play such as *information* or *internet*

- Hybrid philosophies, which combine two or more approaches in a single agent
- Secondary attributes, such as versatility, benevolence, veracity, trustworthiness, temporal continuity, ability to fail gracefully, mentalistic and emotional qualities

From the ideal and primary attributes, Nwana derived four basic agent types, *collaborative, collaborative learning, interface,* and *smart.* These are illustrated in Figure 3.4.

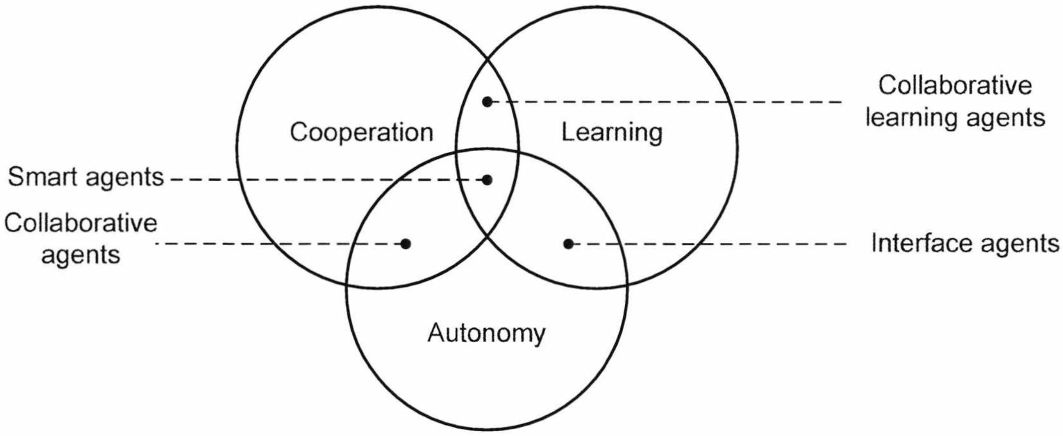


Figure 3.4 Agent Taxonomy [Nwana, 1996]

[Franklin and Grasser, 1996] propose the taxonomy shown in Figure 3.5.

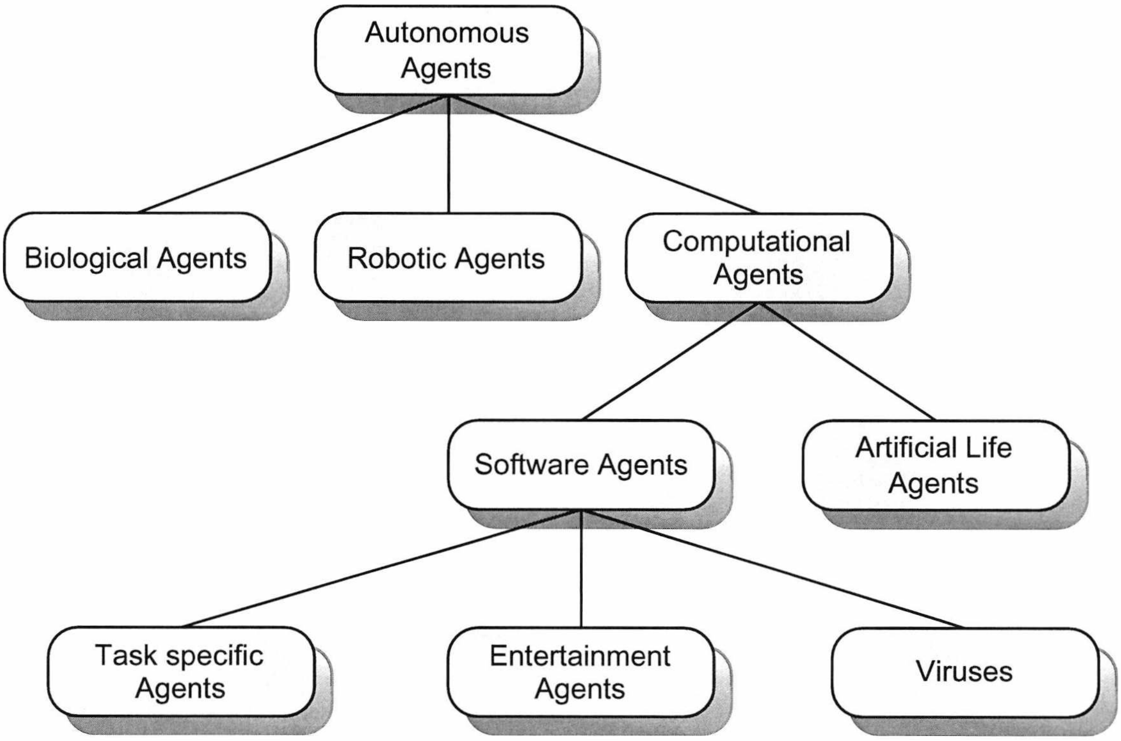


Figure 3.5 Agent typology [Franklin & Grasser, 1996]

To further classify agents, they propose that agents could be categorised by control structures, environments (e.g. Internet, network, file system, database), the language used to write the agent and also applications.

Petrie [Petrie, 1996] tackles an important issue; what distinguishes agents from other types of software - a question also raised by Odell [Odell, 2002]. Petrie notes that the majority of what are called 'agents' for web-based searching and filtering are merely one time query answering mechanisms, that could be described as 'servers'. Also the term 'mobile agent' used to describe a Java applet whose only agent like behaviour is to run processes on a foreign machine, could just as easily be described as a mobile process.

In contrast to the other attempts by researchers to define a set of characteristics for agents in general, Petrie suggests one specific agent class which he denotes as '*typed-message*' agents. This classification was performed in order to differentiate these types of agents from other types of software. A typed-message agent is capable of communication as a community using a shared message protocol (such as Knowledge Query and Manipulation Language, (KQML)).

"An individual software module is not an agent at all if it can communicate with the other candidate agents only with a client/server protocol without degradation of the collective task performance." [Petrie, 1996]

Since the field of agent research is vast it is proposed to concentrate on specific agent types rather than to try and present comprehensive coverage of the whole domain. The following section will introduce the three main types of agent architectures, reactive, deliberative or hybrid. Subsequently a number of specific agent types will be examined, mobile agents will be introduced, this type of agent presents exciting opportunities for the development of truly roaming programs. Interface agents will also be discussed, it is the field of interface agents that has assisted in the motivation for the development of the adaptive interface layer for biometric applications as introduced in Chapter 2.

3.3 Reactive Agents

3.3.1 Introduction

These types of agents can decide what action to perform without reference to any history. Their actions are based on what is happening at the present time since they respond directly to the environment in which they are situated.

3.2.2 Overview

A reactive architecture can be defined as "one that does not have any kind of central symbolic world model and does not use complex symbolic reasoning" [Wooldridge and Jennings, 1994]. Development of these reactive architectures has been driven by the various problems that exist with symbolic AI, this being the corner stone of the deliberative architecture introduced in Section 3.4.

This alternative architecture for agents was proposed by Brooks in 1985 when he developed what he termed the *subsumption* architecture [Brooks, 1986]. Brooks proposed three main theories in a number of publications [Brooks, 1991][Brooks, 1991a].

1. Intelligent behaviours can be generated without explicit representation of the kind that symbolic AI proposes.
2. Intelligent behaviour can be generated without explicit abstract reasoning of the kind that symbolic AI proposes
3. Intelligence is an emergent property of certain complex systems.

He also identified two key ideas that have educated his research

1. Situatedness and embodiment. 'Real' intelligence is situated in the world, not in disembodied systems such as expert systems or theorem provers.

2. Intelligence and emergence: 'Intelligent' behaviour arises as a result of an agent's interaction with its environment. Also, intelligence is 'in the eye of the beholder: it is not an innate, isolated property.

Brook's motivation for this work was his dissatisfaction in classical AI approaches with respect to building control mechanisms for autonomous robots. His hypothesis argued that in order to build a system that is intelligent, it is required to have the representations in the physical world [Brookes, 1991a]. This particular hypothesis is important as it is a radical concept and alleviates the need for a physical symbol system hypothesis which traditional AI systems rely on. This hypothesis circumvents the need for symbolic representation of the world or models because the world becomes its own best model. The model is kept current since the system is connected to the world via actuators and or sensors.

Brook's *subsumption* architecture consists of a number of modules based on Augmented Finite State Machines (AFSM). Based on the input value to the AFSM the unit may be triggered if the threshold is exceeded, although there are inhibition and suppression signals that are also inputs to the AFSM which can affect the triggering. These modules are grouped into layers which work in an asynchronous fashion such that modules in a higher layer can inhibit lower level modules. Higher level modules are responsible for more long term complex goals whilst the lower levels generally deal with more primitive kinds of behaviour. Each layer is responsible for a certain behaviour e.g. to avoid obstacles.

Systems developed using this reactive approach are considered to be extremely simplistic in computational terms, due to the fact that there is no pattern matching or explicit reasoning unlike the approach taken in symbolic AI systems. This being said Brooks has managed to demonstrate robots performing tasks using this *subsumption* architecture whose performance was deemed to be remarkable if they had been developed using symbolic AI.

It can be said that the most elementary reactive architecture is based upon situation action rules which is derived from some work conducted by [Suchman, 1987]. A *situation* can be described as a potentially complex combination of external and internal events and states [Connah, 1994], situation action agents react in an appropriate manner according to the current situation. Situation action agents have been used in a number of systems, one of the most notable being PENGI [Agre and Chapman, 1987]. Work on this system was carried out around the same time that Brooks was describing his initial results with the *subsumption* architecture. Chapman was looking at alternatives to the AI planning approach due to the theoretical difficulties he also envisaged with the symbolic AI model.

It was theorised that abstract reasoning was not required for everyday activities, and that for most tasks once learned, can be carried out in a routine manner and that these routines may only change to deal with new contingencies. They also stated that intelligent behaviour can result from the interaction of what can be described as 'simple machinery' in a complex environment. A claim also made by Herb Simon much earlier in 1969 [Simon, 1969].

In [Agre and Chapman, 1987][Agre, 1997] a new participatory theory of representation is presented which is called *indexical-functional*, or *deictic* representation. Using this method an agent deals with its environment through constant interaction with it rather than through the construction and management of models.

Another approach was presented in [Kaelbling and Rosenschein, 1991]. Using this technique an agent is specified in declarative terms then this is compiled into a digital system which satisfies the specification. The system does not perform any symbol handling and there is no representation of symbolic expressions.

"Specification of the semantics of the [agent's] inputs ("whenever bit 1 is on, it is raining"); a set of semantic facts ("whenever it is raining, the ground is wet"); and a specification of the state transitions of the world ("if the ground is wet, it stays wet until the sun comes out"). The programmer then specifies the desired semantics for the output ("if this bit is on, the ground is wet"), and the complier ....[synthesise] a circuit whose output will have the correct semantics... All the declarative "knowledge" has been reduced to a very simple circuit" [Kaelbling and Rosenschien, 1991]

Reactive agents can be used to simulate natural phenomena and also artificial worlds. [Ferber, 1994] illustrates how reactive agents were used to simulate ant societies where each ant was represented as an agent, three types of agents modelled a limited ecosystem, biotapes, fishermen and shoals of fish. Ferber also hypothesised that through the use of reactive agents a computer could become a 'virtual laboratory', in this 'laboratory' researchers could change experimental parameters and validate the model being investigated using quantitative and qualitative data. Another system that followed this approach was the work described by Nwana [Nwana, 1993], this system used Agent behaviour language (ABLE) to simulate children playing in a playground.

One of the problems with reactive agents is the extent of their applicability which can be said to be fairly limited. It can be seen that most of the applications based on reactive agents are mainly simulations or games.

3.4 Deliberative Architecture

3.4.1 Introduction

By definition deliberation is the explicit consideration of alternative courses of action. This type of agent has the capability to generate and choose a suitable course of action based on the state of the environment and the goals the agent is currently pursuing.

3.4.2 Overview

A deliberative agent can be defined as "one that possesses an explicitly represented, symbolic model of the world, and in which decisions (for example about what actions to perform) are made via symbolic reasoning" [Wooldridge, 1995]. This architecture is derived from symbolic AI which is based upon the physical symbol system hypothesis.

"A physical symbol system consists of a set of entities, called symbols, which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure). Thus, a symbol structure is composed of a number of instances (or tokens) of symbols related in some physical way (such as one token being next to another). At any instant of time the system will contain a collection of these symbol structures. Besides these structures, the system also contains a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction and destruction. A physical symbol system is a machine that produces through time an evolving collection of symbol structures. Such a system exists in a world of objects wider than just these symbolic expressions themselves." [Newell and Simon, 1976]

The authors use this notion of a physical symbol system in order to hypothesise:

"A physical symbol system has the necessary and sufficient means for general intelligent action" [Newell & Simon, 1976]

This hypothesis initially seems very attractive in its prospect of being able to deliver intelligent action through physical symbol representation and manipulation, although there are two fundamental issues that need to be addressed:

1. Representation/reasoning problem. This is the problem of how to symbolically represent information about sophisticated real world entities and process, and also how to get agents to reason with this information in time for the results to be useful.

2. The transduction problem. This is the problem of translating the real world into an accurate, adequate symbolic description, in time for this description to be useful.

In order to fully realise the notion of symbolic AI a large amount of research has been conducted into addressing both these issues. Work on the first problem led to research in the area of knowledge representation, automatic planning and reasoning etc. The second problem has encouraged research in the areas of learning, understanding vision and speech etc. It is generally accepted by most researchers in this field that neither of the two problems are anywhere near solved. It appears that the fundamental problem lies with the general complexity of symbol manipulation and the difficulty of theorem proving in even very simple logics. Although it was these issues that would eventually lead to the rejection of this approach by some researchers in order to investigate other approaches such as the reactive technique, it did not prevent the development of numerous systems that employed a planning component in their architecture.

Work in this field led to a number of agent systems being developed that used symbolic reasoning and planning in order to meet their design objectives. It has long been thought that in the symbolic AI community that a form of planning component would be the central component of any artificial agent. One of the first planning systems was STRIPS devised by Fikes and Nielson [Fikes and Nielson 1971]. This system used a symbolic description of the desired goal state and the world, and a set of action descriptions. Using means end analysis the system attempts to determine a sequence of actions that will accomplish the desired goal. The algorithm employed was very simple and was found to be ineffectual on problems of moderate complexity. Subsequently a large amount of work conducted in order to develop more effective techniques. Even though there were advances made in the field of planning algorithms with two innovative techniques, non-linear and hierarchical, it would appear that the idea of symbolic AI would ultimately fail [Chapman, 1987]. This shift in thinking led a number of researchers to explore other approaches which are detailed in section 3.2. Examples of other deliberative agents with planning modules are IPEM [Ambros-Ingerson and Steel, 1988] and MCS [Doran et al, 1990] both of these systems employing a non linear planning technique.

A number of other frameworks have been developed that also employ mentalistic attitudes as introduced in section 3.1.2, IRMA [Bratman et al, 1988] and also GRATE* [Jennings, 1993]. Also present in some of these frameworks was the notion of cooperation amongst agents for common problem solving. The GRATE* system is one such example, in this particular architecture there are modules responsible for implementing a model of joint responsibility [Jennings, 1992]. This model specifies the action of agents towards each other and also themselves. This aspect of *collaboration* is the core notion of systems such as DVMT [Durfee et al, 1987], ADEPT [Wiegland and O'Brien, 1996], MII [Titmus et al, 1996] and Pleiades [Mitchell et al, 1994].

Joint collaboration and learning is examined in [Weiß, 2000], techniques for conflict resolution in these environments proposed by [Arai and Sycara, 2000] and problem solving [Allen et al, 2002] both these aspects are critical to any collaborative system.

The more practical aspects of collaboration and learning are explored through the use of robots operating in environments such as the RoboCup [Kitano et al, 1997] research initiative. The goals of this initiative were to investigate the areas of multi-agent teamwork, agent modelling, and multi-agent learning [Kitano et al, 1997]. Work in this particular field has been prolific [Baral et al, 1998][Veloso et al, 1998][Marsella et al, 1999][Beetz et al, 2002] and has helped stimulate research in the team based collaborative tasks and learning.

More recently a simulation tool has been developed ÜberSim [Browning and Tryzelaar, 2003] to aid in the development of robot control systems. Collaboration between robots has also been investigated through exploration of unknown environments [Yamauchi, 1998] and world modelling [Liu and Wu, 1999], as well as schemes for the enhancement of performance for space constrained multi robot based tasks [Ostergaard et al, 2001].

More recent work in this field and also the field of reactive agents has led to the combination of both these schemes, this type of agent is described in Section 3.5.

3.5 Hybrid Agents

3.5.1 Introduction

An agent of this type combines the characteristics of two or more of the agent philosophies that have been previously described. Each type of agent hat has been examined previously has inherent strengths and weaknesses. The principle behind the hybrid approach is that for certain applications within a specific domain, the combination of these philosophies within a single agent leads to greater benefits then the gains from the same agent based entirely on a single philosophy.

3.5.2 Overview

For example an agent may use both the reactive and deliberative philosophies. For example the reactive module would take priority over the deliberative component. This would lead to a number of benefits: adaptability, faster response times and robustness. The deliberative component would be responsible for handling the longer goal orientated issues. A number of these hybrid agents have been proposed by researchers across a broad application range.

The TouringMachine architecture was developed by Ferguson [Ferguson, 1992] and is described as "an architecture for dynamic, rational and mobile agents" [Ferguson, 1992]. The mobility mentioned here refers to autonomous robots as opposed to the mobile agent paradigm. The architecture Ferguson proposed is similar to that of Brook's subsumption architecture, consisting of three control layers: - the reactive layer, planning layer and modelling layer. There are two other subsystems, perception and action which interface directly with the environment the agent is situated in.

The reactive layer uses situation action rules akin to that of Brook's system in order to devise courses of actions in reaction to events that occur too quickly for the other layers to cope with. The planning layer consists of two modules and is responsible for constructing plans and selecting action to execute in order to attain the agents goals.

The modelling layer is responsible for the identification and resolution of goal conflicts where the agent cannot achieve its goals due to unforeseen interference. This layer contains symbolic representations of the cognitive state of other objects within the environment, it is these models that are manipulated in order to identify and resolve goal conflicts.

These layers can communicate with each other using a message passing system, and are embedded in an encompassing control framework. Unlike Brook's architecture, the TouringMachine system uses vertical layer approach as opposed to the horizontal method employed by Brooks. In effect this means that all the layers have access to the perception and action modules. The control layer uses a system of control rules in order to deal with conflicts occurring from the different layers. These control rules are very similar to that of the suppression/inhibition system used by Brooks in the architecture he proposed.

Another layered hybrid system is that of InteRRap [Muller and Pischel, 1993]. Each successive layer represents a higher level of abstraction than the one below it. Each of these layers operates with different models in the knowledge base of the agent. The method of control in this system is both data and goal orientated. As perceptual input changes it is managed by the world interface which results in changes to the world model. This change in the world model results in various patterns of behaviour to be executed or dropped. The plan based module or the cooperation module may be requested to generate plans based on pattern of behaviour execution, in order to realise to the goals of the agent. Ultimately, this results in primitive actions and messages being generated by the world interface.

The InteRRap architecture has been evaluated through the development of several applications. In [Fischer et al, 1995] a test bed is described for the development of multiagent applications. This test bed known as AGENDA consists of two separate layers, the architectural layer describes a methodology for designing agents. The system development layer provides the basic knowledge representation formalism, general inference mechanisms which are used by the decision making modules of the architectural layer. The architectural layer in the test bed is provided by the InteRRap architecture whilst the system development level is provided by the MAGSY system [Fisher, 1993]. This system provides general purpose inference mechanisms as well as frame based knowledge representation formalism. The first application that was developed using this framework was an interacting robot application known as FORKS [Muller, 1994][Muller et al, 1995][Muller, 1996]

Another system to employ the InteRRap architecture with some minor modifications is AgentMove [Bohnenberger, 1996], a system that attempts to coordinate public transport in a dynamic and distributed manner using a multi agent system. Although the author could not achieve the performance specifications in this experiment, it was concluded that the InteRRap architecture is appropriate for the use in such a complex scenario.

Hayes-Roth proposed another layered hybrid architecture in [Hayes, 1995]. This system comprises of two layers, a physical layer which is responsible for perception action coordination. This layer senses, filters, interprets and reacts to the dynamic environment in which the agent is located. The cognitive layer constructs a developing model from the perceptual input from the physical layer, and also performs interpretation, reasoning and planning. The motivation behind developing this architecture was to provide a system to construct Adaptive Intelligent Systems (AIS) that operate in specialised, but challenging niches such as intensive care unit patient monitoring. The argument presented in this paper is that AI agents are 'niche-bound' because they are 'knowledge-bound' [Lenat and Feigenbaum, 1991]. In order to realise the different kinds of adaptation required by this architecture a single theoretical concept is employed 'An agent dynamically constructs explicit control plans to guide its choices among situation-triggered behaviour'. So to accomplish this notion the physical layer is implementing the reactive behaviour, whilst the cognitive layer is performing longer term deliberative planning and scheduling, drawing from the evolving model. A number of niches are exposed where the application of AIS are seen as viable. These niches are typically described as to present dynamic variability in the required tasks, available resources, contextual conditions and performance criteria.

The implementation and evaluation of this architecture has resulted in a number of agents being constructed. One of these called Guardian which has been employed in the niche of intensive care unit patient monitoring, [Hayes-Roth, et al, 1992]. Other niches upon which the architecture has been employed are: Aibots [Hayes-Roth et al, 1993][Hayes-Roth et al, 1995] and Virtual Theatre [Doyle and Hayes-Roth, 1997][Hayes-Roth and ven Gent, 1997][Rousseau and Hayes-Roth, 1998].

Another approach to the hybrid architecture was taken by Musliner [Musliner et al, 1993]. The problem seen here was that for real time control problems most research has either limited the power of AI methods or embedded 'reactivity' into the AI system. The authors realised there was a conflict between the nature of AI and the needs of the real world, real time control systems which need constant predictable performance. In order to resolve this problem a Cooperative Intelligent Real-Time Control Architecture (CIRCA) was developed. The architecture contains an AI subsystem that reasons about task level problems, whilst a separate real time subsystem deals with control level problems that require guaranteed response times. In this manner the CIRCA system is designed to reason about guaranteeing its control level goals, but not necessarily its task level goals.

Perhaps one of the most notable and best known hybrid architectures is the Procedural Reasoning System (PRS) developed by Georgeff and Lansky [Georgeff & Lansky, 1987]. This architecture is based on the BDI model, also included are a plan library as well as explicit symbolic representations of beliefs, desires and intentions. Beliefs are expressed in first order logic and represent facts about the external world or the state of the internal system. Desires are not represented as static goal states but are rather represented as system behaviours. Knowledge areas (KA) are partially elaborated plans that are contained in the plan library. Each KA has an associated invocation condition, this determines when the KA is to be activated. These KA's can be activated in a goal driven or data driven fashion, another aspect of this is that KA's may also be reactive, this allows the system to respond to rapid changes within its environment. The intentions of the system are represented by the currently active KA's. The system interpreter is responsible for the manipulation of the data structures and also for updating beliefs, executing actions and also for invoking KA's. The PRS system has been evaluated in a number of domains, one of these being mobile robot applications [Lee et al, 1994].

A version of PRS written in C known as C-PRS has been used effectively in a number of projects involving control and supervision system for mobile robots. [Aguilar et al, 1995][Lacroix et al, 1994]. Another variant of the PRS system known as PRS-CL has been successfully deployed in a number of application that require the integration of reactive and goal based behaviour, including real time tracking [Garvey and Myers, 1993], and monitoring and control systems [Georgeff and Ingrand, 1989].

Hybrid approaches have also been employed in the field of autonomous spacecraft which will be utilised for deep space exploration [Gamble et al, 1998][Pell et al, 1997][Muscetta et al, 2002]. The flexibility of the hybrid approach enables a procedural executive to be used for tasks such as scheduled execution and timing, whilst a deductive executive is used for state inference and global goal forming, both of these features providing a robust autonomous system. The basis for these systems has been built on work performed in the planning and scheduling fields of research introduced in section 3.3 such as [Das et al, 1998][Das et al, 1999] and research also conducted by Georgeff [Georgeff and Ingrand, 1989] employing the PRS system in the monitoring of spacecraft systems.

3.6 Mobile Agents

3.6.1 Introduction

A mobile agent is a running program that can move from host to host in a network when and where it chooses. Mobile agents are one form of mobile code. A number of different variants of mobility exist, these refer to the possible variations of relocating code and state information, including the values of instance variables, the program counter, execution stack etc. A Java applet for instance has code mobility through the movement of class files from a web server to a web browser, although no state information is conveyed. Aglets [Lange et al, 1997] developed at IBM allow the values of instance variables to be conveyed along with the code as the agent is transported to the new destination, however the program counter and execution stack are not. A stronger notion of mobility [Acharya et al, 1997] allows Java threads along with the agent's code to be transferred during relocation.

3.6.2 Overview

The key premise underlying these mobile agents is that certain types of agents do not need to be stationary and there are real benefits in some applications if these agents could be mobile as opposed to their stationary counterparts.

If we consider the following scenario which Wayner [Wayner 1995] proposes where a user is required to devise a program which will allow the booking of a flight reservation based on a number of preferences, such as the window of the desired departure flight, the number of connections allowed and the destination arrival time window. A single stationary program would need to access all the flights between these times via the airline reservation databases, and then start to sift through these possibilities to narrow down the search based on the other preferences that have been specified. The amount of data needed to transfer this information to the user's PC could be quite large, not to mention the amount of time that is taken to sort though the extraneous information before a list of possible candidate flights is produced.

The alternative to this approach involves the use of a mobile agent. These preferences would still be entered as parameters to this agent, but then the agent is release into the network and can travel around querying the airline databases locally and then ultimately returning to the users PC with a list of suggested flights that meet the criteria. In this fashion the cost of communicating large amounts of data to the local PC is alleviated as superfluous information regarding flights which do not match the specified criteria is disregarded. So it can be seen that mobile agents can facilitate a number of practical advantages that elude their static counterparts.

The notion of code mobility is not a revolutionary new concept, one of the earliest being remote batch submission [Boggs, 1973]. This basic concept acted as the basis for further research and a number of projects, Accent [Rashid and Robertson, 1981] and RIG [Rashid, 1986]. These were experiments in building distributed operating systems such as MACH [Accetta et al, 1986]. The notion of a mobile agent was first established in 1994, when White described a computational environment known as "Telescript" [White, 1994]. This environment enabled executing programs to transport themselves from one location in a network to another in order to interact locally with resources at those remote locations. Telescript demonstrated the notion of strong mobility in which process migration allows the program to relocate in the middle of a loop and resume execution on the destination machine. A number of other mobile agent frameworks also support the strong mobility paradigm D'Agents [Rus et al, 1997] and Ara [Peine and Stolpmann, 1997]

The Java Development Kit [Gosling et al, 1996] provides the basic native facilities to support weakly mobile code. Weak mobility refers to the ability only to migrate the code associated with the entity across the network, consequently any state information must be packaged up before the migration can occur. Java has become a popular choice amongst implementers of mobile agent frameworks for a number of reasons, inherent platform independence, object serialisation and also Java's security model. Testament to this are the large number of multi agent frameworks built upon the Java language such as Mole [Baumann et al, 1997] JAFMAS [Chauhan and Baker, 1998], JINI [Arnold et al, 1999] HIVE [Minar et al, 1999], JADE [Bellifemine et al, 2001] and JAM [Huber, 1999] which is used to develop mobile agents based on the BDI model.

More recent work had focused on aspects of mobile agent security [Li and Lam, 2002][Maggi and Sisto, 2003][Elichai, 2004] and modifications of existing mobile platform such as JADE [Ametller et al, 2004] to support secure mobile agent systems.

Interoperability issues are investigated in Hasegawa [Hasegawa et al, 2003], with the explosion of mobile agent systems, interoperability is becoming a key issue.

3.7 Interface Agents

3.7.1 Introduction

This type of agent emphasis the autonomy and learning aspects of agency in order to perform tasks for their owners. The motivation behind this approach is that the rapid expansion of the "information superhighway" has provided a whole new range of computer based tasks and services, and that the inherent complexity of this new environment will necessitate the need for a new form of human computer interaction. This new type of interaction requires the computer itself to become an intelligent, active and personalised collaborator.

3.7.2 Overview

An interface agent is "a computer program that employs artificial intelligence techniques in order to provide assistance to a user with computer related tasks" [Maes, 1994].The fundamental metaphor behind interface agents is that of a personal assistant which is collaborating with the user in the same work environment.

Pattie Maes, a key promoter of this type of agent, sees the objective of interface agent research is to go some way towards realising the vision of Kay [Kay, 1990]. In which the notion of indirectly managed human computer interfaces is introduced. Currently the majority of computer interfaces only react to direct manipulation i.e. the interface is passively waiting to execute highly specified instructions from the user. Therefore the interface is providing little or no proactive help for complex tasks.

The motivation for the development of such interface agents is driven by the prospect of future developments on the growth of computers and networks leading to a rise in untrained users. Through the use of these interface agents it is hoped that the user can be engaged in a cooperative process with the interface in which both parties can initiate communication, monitor events and perform tasks. This is seen as a direct advantage for the human user in the specific application domain.

There are a number of mechanisms identified in which these interface agents are expected to assist the user of the system these are:-

- They can perform tasks on the user's behalf.
- They can train or teach the user.
- They can help different users collaborate.
- They can monitor events and procedures.

Interface agents were introduced in Section 3.2.6 as a monolithic class. The maturing agent field has led to a growing number of interface agents reported in literature. This has in turn led to the call for a taxonomy for interface agents to be developed. Mladenić [Mladenić, 1999] proposes one such taxonomy based on a machine learning view of interface agents.

Broadly interface agents can be classified into one of four groups as shown in Figure 3.6.

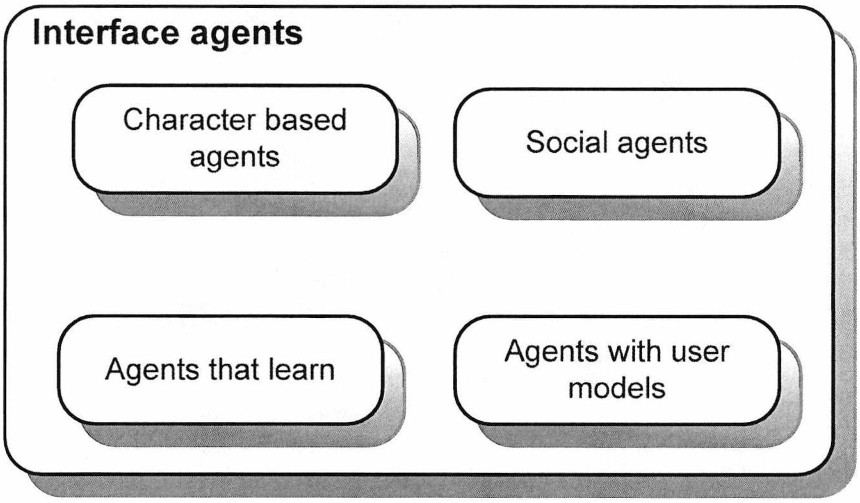


Figure 3.6 Interface agent typology

3.7.3 Character Based Agents

This type of interface agent generally employs a "character" based interface. Typically this involves the animation of a typically life like character on the screen which is intended to be a representation of the interface agent itself to the user. It is the intention of this personification to render the human computer interaction more human to human like and therefore more social. A number of benefits are expected to stem from this personification.

- The social aspects of personified agents are thought increase the believability and trustworthiness of agents. These are critical aspects if agents are to be authorised by the user [Lester and Stone, 1997].

- User engagement is increased, due to the nature of a more social interaction [Sproull et al, 1996] [Walker et al, 1994]. It has also been shown that in a learning environment, the effect of personification can positively affect the student's perception of a learning experience [Lester et al, 1996].

A study into the *persona effect* was conducted by [Mulken et al, 1998] in which the effect of personification was examined in an empirical manner to determine the benefits is any, on both objective and subjective measures for both technical and non technical domain information. Results of the study were mixed, for the objective measures the results were inconclusive, however the results from subjective measures regarding technical information appeared to support the *persona effect* whilst the non technical information did not.

3.7.4 Social agents

Social agents talk to other agents (typically other interface agents of the same type) in order to share information. This technique is often used to bootstrap new, inexperienced interface agents with the experience of older interface agents (attached to other users).

3.7.5 Learning Agents

Agents that employ a learning technology can be classified by the type of information required by the learning technique and also the way the user model is represented. One of the issues surrounding this type of agent is the aspect of how it acquires the knowledge in order to adequately assist the user with the specific task. It has been noted that the 'knowledge based' approach in which a domain and user model is constructed in order to supply the necessary domain specific knowledge, which is the method adopted by the majority of the AI community working on intelligent user interfaces, is seen to have a number of shortcomings.

The first is that the knowledge used for the models is very specific to the application under question and that this knowledge cannot be transferred to agents that will deal with other applications. The second is that once the knowledge has been programmed into the agent it is fixed and cannot and be modified to take into account individual user preferences and habits. As always the possibility of providing the agent with enough knowledge in order to fully comprehend the user's actions is unpractical.

There is another issue that can plague this type of agent and that is of trust. It has been argued that if a user was given a sophisticated interface agent from the outset, the user may be left with a feeling of loss of control and understanding [Myers et al, 1991].

The knowledge acquisition problem is handled through the use of machine learning techniques. This approach requires that the agent is given a minimum of background knowledge and through a number of mechanisms is able to acquire the knowledge is requires to assist the user. There are some conditions that need to be satisfied in order for this technique to be appropriate

- The application usage must contain an extensive amount of repetitive behaviour (within the actions of one user or amongst users)
- The repetitive behaviour is different for individual users

The metaphor of a personal assistant is paramount to the machine learning approach. It can be seen that initially the assistant is not au fait with the preferences and habits of the user. As the assistant experiences what the user is doing in certain circumstances the knowledge of the assistant increases. Not only is knowledge gleaned from the user itself but it is also possible that the user may give instructions and learning may be possible from other assistants. For a learning agent, acquisition of knowledge is possible through four different sources (Figure 3.7).

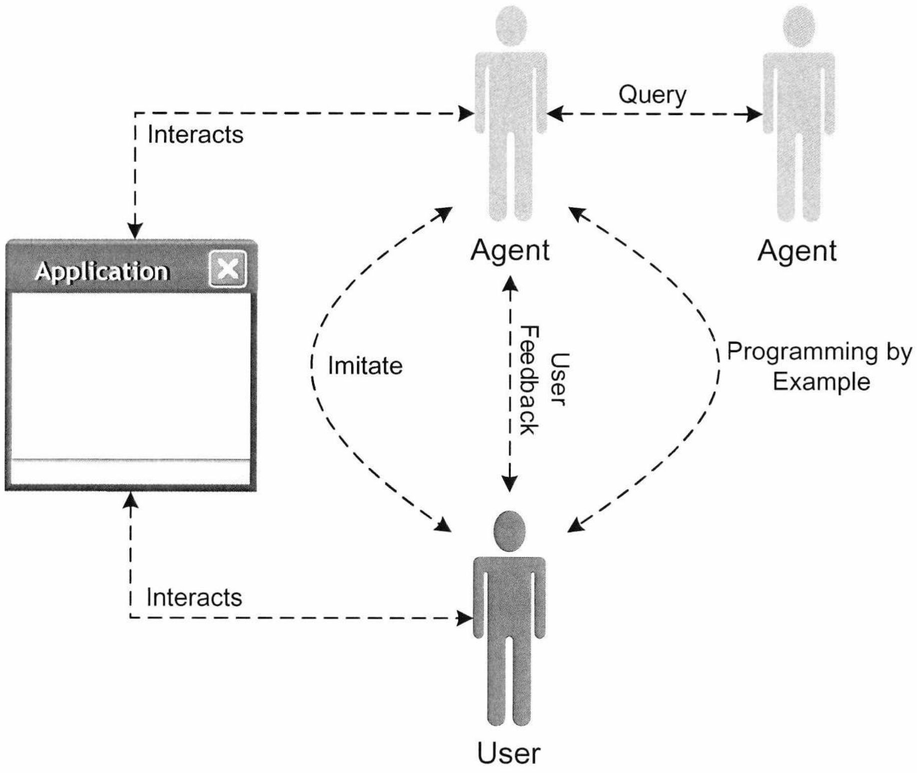


Figure 3.7 Interface agent learning mechanisms [Maes, 1994]

- Learning and imitating the user

  By "looking over the shoulder" of the user whilst the user is using the application, the agent can keep an eye on the activities of the user. Over periods of time the agent can find patterns in the operation of the interface and can offer to automate these actions

- Direct and indirect feedback

  Indirect feedback involves the user negating the suggestion of the agent and taking a different action to that of what was offered. Direct feedback is more explicit and involves the user stating "don't do that again" or "do that again" in response to the actions offered by the agent

- Explicit Instructions

  The agent can learn from examples given by the user. By giving the agent a set of hypothetical events and situations and demonstrating what the agent should do in those situations, the agent can learn from these examples

- Asking other agents

  The agent is capable of asking other agents that assist users in similar tasks for advice. These other agents may have built up more experience and may have an answer to the current situation

Apart from Maes's prolific work in this specific agent arena, a number of other researchers have contributed to this field of agent research. In [Dent et al, 1992] an interface agent is described that is used to schedule and manage meetings, known as the Calendar APprentice agent (CAP). This system shows many similarities to the work performed by [Maes & Kozierok, 1993] and their Calendar Agent although the learning techniques employed by both systems differ substantially. Whilst Maes's calendar agent uses memory based and reinforcement learning, the approach taken by Dent uses back propagation neural network and decision tree techniques. The performance of CAP is investigated in [Mitchell et al, 1994] where the results of five user years of experience with the system is examined. This period has enabled CAP to evolve and learn thousands of rules that model the scheduling preferences of each of its users.

There exists a diverse range of application domains in which interface agents have been employed. [Middleton, 2001] presents an extensive review of interface agents across a broad range of domains, email filtering, expert assistance, matchmaking, news filtering, meeting schedulers, recommender systems, web applications.

Typically these domains are ones in which the pursuit of relevant information produces large amounts irrelevant information. This 'informational overhead' is seen to be excessive as much time is wasted sifting through extraneous information, i.e. as electronic news filtering or email handling. Interface agents have been employed in some more diverse fields such as medical diagnosis [Jing et al, 2002].

Probably one of the most notable interface agents was introduced by Microsoft in Office 97 suite and also subsequently in Office 2000. The office assistant, more commonly known as "Clippy" or "Clippit", presented a caricature of a human face and was used as a presenter of the search function based on Bayesian probability. This came from work conducted by the Lumiere project [Horvitz et al, 1998] in which Microsoft research had spent a great deal of time perfecting the search function and also the presentation of these functions through the use of animated caricatures.

It must be said that Clippy was not an outstanding success with the general public. One of the main problems with this approach was that the Lumiere project ignored the human psychological and physiological reactions to peripheral movement and to the presence of faces or caricatures of faces. It has been found that humans are immediately distracted by movement of any sort on their peripheral field. Concentration is also affected by the presence of faces or caricatures on the screen. Although one of the main goals of the Lumiere project was to provide the user with useful hints at the appropriate time it was found that the computer had no way to tell when the user was concentrating and therefore should not be disturbed. This often led to the interventions from the interface to be given at the wrong time and also in an annoying manner. Needless to say "Clippy" was removed from later versions of the Microsoft Office suite.

3.8 Agent Communication Languages

An Agent Communication Language (ACL) provides an agent with the means of exchanging information and knowledge with other agents. It has even been proposed to equate agency with the ability of a system to exchange knowledge using an ACL [Genesereth and Katchpel, 1994].

Other means of exchange of information and knowledge between applications are available these include remote procedure call (RPC) and remote method invocation (RMI) and CORBA (Common Object Request Broker Architecture). ACL's are a level above CORBA for two reasons:-

1. ACL's handle propositions, rules, and actions instead of simple objects with no semantics associated with them.

2. An ACL message describes a desired state in a declarative language, rather than a procedure or method.

The ACL defines the types of messages that the agents can exchange. Typically agents participate in conversations with other agents rather than a simple course of message exchange. The conversations could be for example participation in an auction or negotiation of some description.

The origins of ACL development can be traced back to the Knowledge Sharing Effort (KSE) which was started around 1990 by the Defence Advanced Research Projects Agency (DARPA). The main concept of the KSE was that in order to share knowledge communication is required which in turn relies on a common language, it was the goal of this effort to define a common language. The KSE model defines software systems as virtual knowledge bases that exchange propositions using a language that expresses various complex attitudes. The term used for these attitudes is *propositional attitudes*. A propositional attitude comprises of a three part relationship between:

- An agent
- A content bearing proposition (for example, *it is raining*)
- A finite set of propositional attitudes an agent might have with respect to the proposition (for example, believing, asserting, fearing, wondering, hoping etc.)

The common language problem that faced the KSE is also applicable to the agent domain. An agent should be able to understand another agent's native language expressions even if the other gaunt is using a different implementation language and domain assumptions. The KSE proposed a layered approach to the common language problem. The first layer is involves the syntactic translation between languages in the same family or between families of languages. Another layer is responsible for ensuring that tokens semantic content is protected among applications. This ensures that a concept, object or entity has a uniform meaning across applications even if these applications use different names to refer to it. The last layer handles the communication between agents, it is not concerned with the low level mechanics of transporting the data but with the ability of agents to communicate attitudes about the knowledge and information they posses.

The KSE proposed to employ a particular logic language Knowledge Interchange Format (KIF) [ANSI, 1995], as a standard for describing items within computer systems such as databases, intelligent agents etc. KIF is a prefix version of first order predicate calculus with extensions to support meta-operators and definitions. The language description includes both a specification for its syntax and for its semantics. KIF was also explicitly designed to be a useful mediator in the translation of other languages.

3.8.1 Knowledge Query and Manipulation Language (KQML)

KQML is a high level message orientated communication language and protocol for information exchange independent of content syntax and applicable ontology (the vocabulary of the words in the message).

KQML Message A	KQML Message B
(ask-one	(tell
:sender joe	:sender stock-server
:content (PRICE IBM ?price)	:content (PRICE IBM 14)
:receiver stock-server	:receiver joe
:reply-with ibm-stock	:in-reply-to ibm-stock
:language LPROLOG	:language LPROLOG
:ontology NYSE-TICKS)	:ontology NYSE-TICKS)

Table 3.2 KQML message composition

The syntax of a KQML message is based upon the s-expression used in Lisp [McCarthy, 1960], a balanced parenthesis list. The initial element in the list is the *performative* (ask-one), the remaining keywords are the performative's arguments expressed as keyword/value pairs. In message A shown in Table 3.1, agent joe is querying the stock server about the price of some IBM stocks, the query is written in a language called LPROLOG which can be seen in the language field and the ontology is stated as NYSE-TICKS. A reply to this message by the stock server is shown as message B. KQML can be described as a three layer structure; these layers are briefly described below.

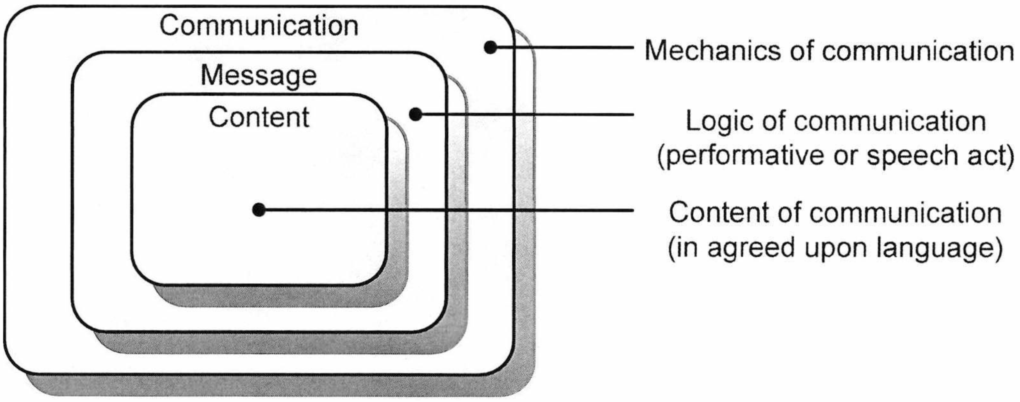

Communication
Message
Content
Mechanics of communication
Logic of communication (performative or speech act)
Content of communication (in agreed upon language)

Figure 3.8 KQML Organisational structure

Figure 3.8 illustrates a typical KQML message. The content portion of the message actually carries the information payload; this content portion can carry any representation language such as ASCII or binary notation.

The communication layer uses a set of features to describe the lower level communication parameters such as the identity of the sender and recipient. The message layer is used to attach a *performative* or a speech act to the content of the message. A speech act indicates whether the message is an assertion, a query, command or any number of other "known" performatives.

The message layer also has some additional features to describe the content language, the ontology. Although KQML has a predefined set of reserved performatives, an agent employing KQML can choose only to handle a few performatives and on the other hand a community of agents may choose to use additional performatives if they agree on their interpretation and the protocol associated with each.

The KQML performatives can be organised in seven basic categories.

1. Basic query performative (evaluate, ask-one, ask-all..)
2. Multiresponse query performatives (stream-in, stream-all..)
3. Response performatives (reply, sorry..)
4. Generic informational performatives (tell, achieve, cancel, untell, unachieved..)
5. Generator performatives (standby, ready, next, rest..)
6. Capability definition performatives (advertise, subscribe, monitor..)

7. Networking performatives (register, unregister, forward, broadcast..)

One of the design criteria for KQML was to support a variety of agent architectures. KQML has a small number of performatives that agents can use to describe metadata specifying information requirements and capabilities. KQML also introduced a special agent class called communication facilitators. These agents provide useful communication services e.g. maintaining a registry of service names and forwarding messages to named services.

3.8.2 Foundation for Intelligent Agents (FIPA)

The Foundation for Intelligent Agents [FIPA, 1996] is an association to promote the success of emerging agent based applications, services and equipment. The main goal of FIPA is to make specifications that maximise interoperability across agent based platforms. FIPA's ACL like KQML is based on speech act theory, messages are actions or communicative acts, and are intended to perform some action by virtue of being sent.

The FIPA ACL specification consists of a set of message types and the description of their pragmatics- that is the effects on the mental attitudes of the sender and receiver agents. The specification describes every communicative act with both a narrative form and formal semantics based on modal logic. It also provides the normative description of a set of high-level interaction protocols including requesting an action, contract net and several different types of auctions. The ACL is similar to KQML, the syntax is identical to KQML's except for different names for some reserved primitives. In this manner it maintains KQML's approach to separating the outer language from the inner language. The outer language defines the intended meaning of the message whilst the inner, or content, language denotes the expression to which the sender's beliefs, intentions and desires are conveyed by the meaning of the communication primitive.

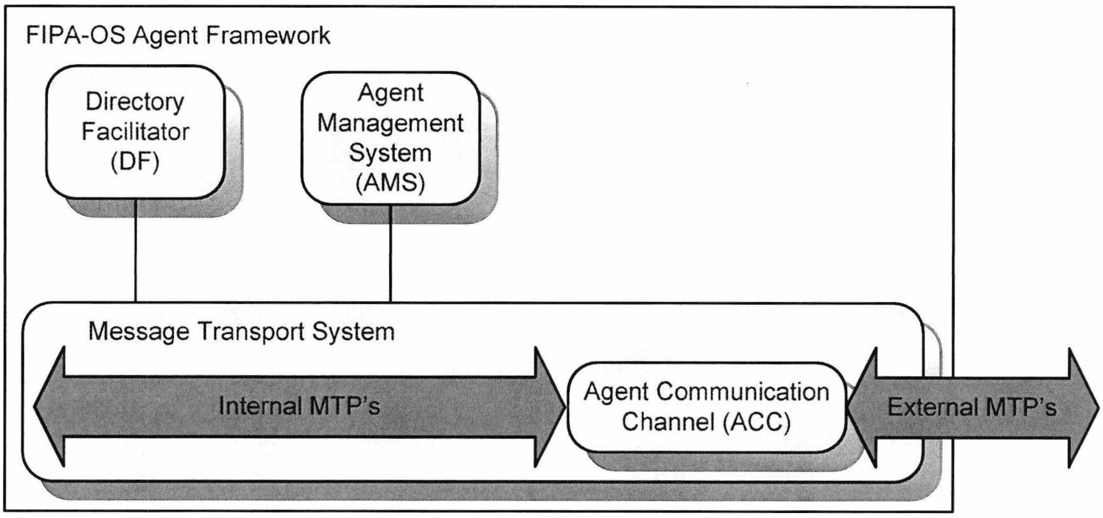

FIPA-OS Agent Framework
Directory Facilitator (DF)
Agent Management System (AMS)
Message Transport System
Internal MTP's
Agent Communication Channel (ACC)
External MTP's

Figure 3.9 FIPA-OS framework model

The FIPA reference model is illustrated in Figure 3.9. The Agent Communication Channel (ACC), Agent Management System (AMS), Internal Message Transport Protocols (MTP) and Directory Facilitator (DF) form what is known as the Agent Platform (AP). These are mandatory, normative components of the model.

The Directory Facilitator (DF) and Agent Management System (AMS) are specific types of agents, which support agent management. The DF provides "yellow pages" services to other agents. Agents may register their services with the DF or query the DF for information about the agents. Membership of a DF defines a domain, which is an agent community that reflects the logical organisation of agents. DF's can register with each other to form a network of DF's allowing the queries to span these domains.

The AMS provides agent lifecycle management for the platform. It provides an agent naming service (ANS) where it maintains a list of all the agents that are registered with the platform. This list includes the unique name of the agent and their associated transport address. The AMS is another type of agent that exerts supervisory control over access to and use of an agent platform. This includes creation of agents, deletion of agents, registration of agents on a platform and overseeing the migration of agents to and from platforms.

The ACC supports interoperability both within and across different platforms. The Internal Message Transport Protocols (MTP's) provides a message routing service for agents on a particular platform which must be reliable, orderly and adhere to the requirements specified by the FIPA specification. In order to be FIPA compliant the AP must minimally support Internet Inter-Orb Protocol (IIOP). This is the default inter platform communication method which needs to be supported for interoperability between agent platforms. This does not however preclude the use of other protocols, in addition to the IIOP.

3.9 Conclusion

In this Chapter the notion of software agents has been introduced. The Chapter begins by examining what an agent is and also the properties and attributes that an object must possess in order to be called an agent. The issue of an encompassing taxonomy for agents is exposed and illustrated with a number of proposals from various researchers. A number of specific agent types are introduced and examined, their fundamental aspects are discussed as well as a number of examples provided to illustrate the particular agent type.

Chapter 4

The IAMBIC System

In this chapter a new framework for multimodal biometric processing entitled Intelligent Agents for Multimodal Biometric Identification and Control (IAMBIC). The research reported in this chapter was conducted as part of IAMBIC. IAMBIC was a joint collaborative project between the University of Kent and two industrial partners Cardionetics and NeuSciences.

This system employed intelligent software agents in conjunction with multiple biometric modalities in order to realise the goal of a distributed health care database system. This chapter will outline the motivation for the development of such a multimodal authentication system. From the initial design concepts a number of components are identified as requisites for the successful operation of the system. These properties and operation of these entities are declared and specified using a combination of suitably expressive notation schemes. The chapter concludes with a section detailing the specific implementation path adopted for the realisation of the IAMBIC framework.

4.1 Introduction

As discussed in Chapter 2 there are a number of issues surrounding the usage of biometric technology that is hindering its wider adoption. The use of multiple biometric modalities (multimodality) is important for a number of reasons. Issues surrounding spoofing of particular biometrics are a valid concern as well as the fact that that no single biometric is generally considered sufficiently accurate and user-acceptable for a universal authentication application. By employing multiple modalities the risk of impostor penetration is reduced although there is usually an impact on usability of the system as the complexity of the user interaction is increased.

Multimodality can be beneficial from the user's point of view. Using a multimodal system if the user is unwilling or unable to provide a specific modality another may be chosen. This enables flexibility in the use of the system, whilst providing a range of modalities that also provide the broadest user base. Multimodality also enables the IAMBIC system to provide varying levels of authentication through the use of fusion algorithms which are used to combine the results from the various modalities. Non-biometric data can also be used in this fusion procedure to further enhance the robustness of the decision process.

To manage the increased complexity inherent in the use of multiple biometric modalities it was proposed to employ software agents. Software agents have been applied to a wide application range as detailed in Chapter 3. It is this approach to combine software agents and biometrics which is considered to be the novel contribution of the nature of work reported in this thesis.

4.2 Overview of the IAMBIC system

Figure 4.1 illustrates the proposed client/server system architecture for the IAMBIC framework. This generalised approach allows for access to remote resources, with a diverse range of information. The framework can also be used locally to gain access to resources held on the client side.

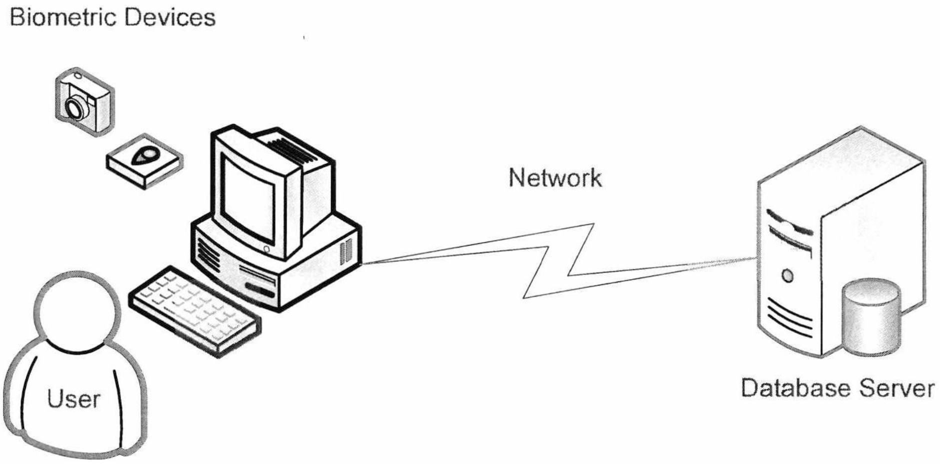


Figure 4.1 Generalised IAMBIC system overview

The goal of The IAMBIC project was to produce a system that would enable distributed multimodal biometric identification. The proposed target application was the authentication of doctors and patients at a web portal for medical informatics. These patients had recently received some form of heart surgery and were wearing a heart monitoring device as part of their recuperation. This device would record data and periodically this data would be uploaded to a remote server. A sample printout of the data recorded by this hart monitoring device manufactured by Cardionetics is presented in Appendix B. Both doctors and patients could access the data using a biometric authentication technique in order to gain access to this remote data (Figure 4.2).

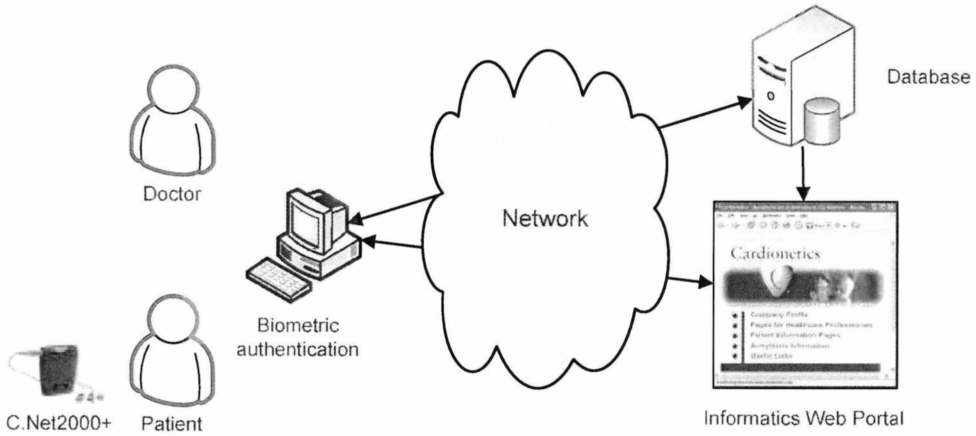


Figure 4.2 Target IAMBIC application

The requirements capture phase resulted in a number of proposed features of the system that were seen as key properties. These features included providing multiple levels of authorisation within the system within both user roles and resource access. Ease of use and an effective user interface along with providing an available range of biometric modalities in order promote acceptability within a given client base, were also seen as important requirements. A list of the requirements captured for the IAMBIC framework can be found in Appendix B. In order to realise this system a number of functional blocks were identified, these are shown in Figure 4.3. In the following section each of these components will be further investigated.

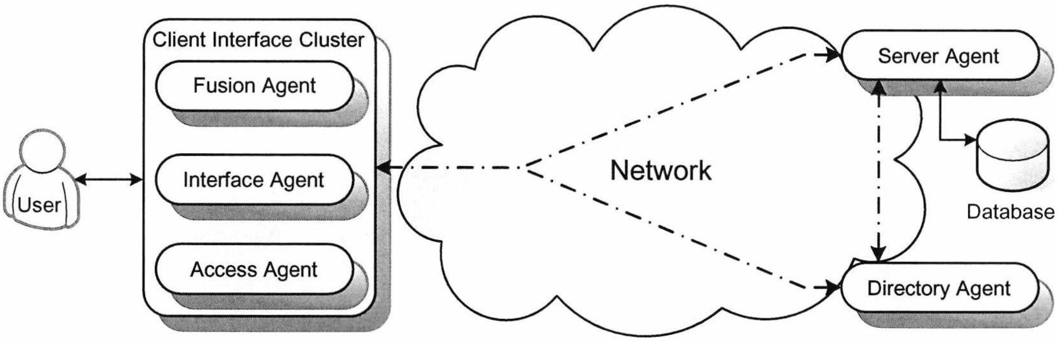


Figure 4.3 IAMBIC Functional diagram

4.2.1 Client Interface Cluster

This cluster of entities represents the client portion of the application. This cluster comprises of three components.

4.2.1.1 Interface agent

The Interface Agent is responsible for the direct interaction with the user, defining, according to the prevailing situation, the set of biometric measurements that must be taken from the user, as well as the corresponding confidence characteristics of the related biometric recognition modules. For instance, in a noisy environment, the voice recognition module is likely to be associated with a low confidence. In addition, this agent defines the level of interaction with the users according to the category of user and user characteristics such as computer literacy, familiarity with the system being used, and so on.

The interface agent acts as the main interface to the user during biometric capture operations, but is also responsible for the capture of other important non-biometric information. Additional environmental data may be captured by the available sensors (e.g., for the voice modality a sample of background noise may be captured). Analysis can be performed on these samples to determine the quality of any acquired data; this can be used to help the agent to analyse any possible enrolment and/or verification failures. The results from this type of analysis can be used to provide feedback to the user to improve future performance.

4.2.1.2 Access Agent

The Access Agent is responsible for negotiating the access to required data (e.g., medical records or other sensitive data) on behalf of the user. Essentially, this agent receives access information from the Interface Agent, locates the data, chooses the best location (in the event that the data can be found in different places), contacts the sources of the desired information and negotiates its release with the Server Agent.

Among the main goals of the Access Agent, the most important is the negotiation for the release of the requested information. As part of this negotiation, a re-measurement of the biometric samples, as well as the recalculation of the combined output, may be required under specific conditions.

4.2.1.3 Fusion Agent

The Fusion Agent is responsible for the fusion of the biometric measures taken from the user. Its main role is to define and implement the best technique for combining several different biometric measures. The design of the Fusion Agent requires knowledge of the types of biometrics measured, as well as of their corresponding characteristics and of the levels of confidence they can generate.

The main goal of the Fusion Agent is the effective combination of the evidence obtained from the different biometric samples provided by the user. The Interface Agent will provide the biometric samples and environmental information obtained from the user during the verification phase. The global confidence score produced by the data fusion will be passed to the Access Agent for transmission to the Server agent.

4.2.2 Remote Agents

Two further entities were identified in order to complete the system functionality. These are the remote agent that provides the data services and also another agent which acts as a directory for all the available data sources within the network.

4.2.2.1 Directory agent

The Directory Agent is responsible for storing and updating all relevant information about location of services within the network. In a healthcare system, for instance, this agent may store information relating to issues such as which hospitals have beds available, which databases contain information about the patients, and other similar matters. In the search for information, this agent also suggests the best way of accessing required information (for example, in the situation where several databases contain the information specified), based on network traffic, distance, and so on.

4.2.2.2 Server Agent

This agent is responsible for acting on behalf of the database in order to guarantee that the information to be released is secure. As discussed above, a negotiation process takes place between the Access and Server agents. Essentially, this negotiation deals with the level of security of the information to be released (the higher the degree of required security associated with a piece of information, the greater is the degree of confidence needed by the system that the user requesting this information is genuine and authorised), the level of encryption of the data to be transmitted and the degree of confidence that the transaction is fraud-free.

This agent is also responsible for detecting fraudulent access to the databases and keeping a log of access to the data. For example, if some unexpected pattern of data access is attempted, a security process will be activated in order to discover whether there is any suspicion of fraudulent system penetration. In addition, a process is executed at regular intervals to analyse the record log and determine if any failure in the access of the data has been detected. As part of the negotiation phase, the Server Agent has to ensure that the user wishing to access the information requested is authenticated and authorised, and acts on behalf of the "owner" of the stored information.

4.3 Typical IAMBIC interaction scenario

The use of an agent-based approach to the control of the complexity of a multi-modal biometrics authentication system is best illustrated by means of an example. In this interaction example, we shall examine how such a system may be used in a healthcare scenario. We assume that a physician requires some confidential patient information and that this information is stored remotely. It is also assumed that, prior to the users accessing the system to request for information, an enrolment process has been successfully completed to collect the required biometric templates for subsequent authorisation. The flow of information between agents is shown in Figure 4.4.

Degrees of system access can be achieved through the use of varying the value of *confidence* score required to access the particular file. The confidence score represents the value obtained as a result of the combination of the biometric verification scores performed by the fusion agent. The value of *confidence* scores in the system will be determined by the degree of confidentiality associated with the particular medical file. For example, a simple standard blood test result would be likely to have a relatively low *confidence* score compared to, say, test results for a sexually transmitted disease

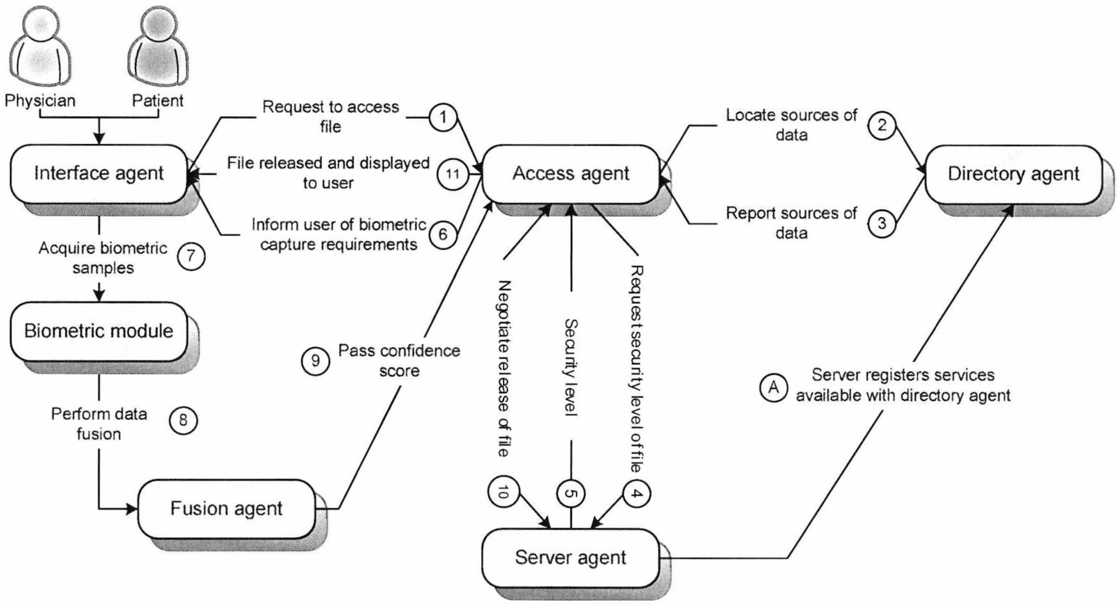


Figure 4.4 IAMBIC system interaction scenario

- When a Server Agent is initialised it will register with the Directory Agent and donate a list of the relevant data services it can provide (Figure 4.4, A).

- Upon initiating interaction, the user will be asked to provide details about the information that is required. This might consist of, for example, a patient number and the type of file required (e.g., X-Ray result, blood test result, etc).

- The Access Agent will be supplied with the information about the requested file (Figure 4.4, 1), and will attempt to register with the Directory Agent. Upon successful registration the Access Agent will encrypt the information about the requested file and transmit this to the Directory Agent (Figure 4.4, 2).

- Upon receipt of this file, the Directory Agent will search its internal dynamic list of data sources and attempt to locate the optimal server for this information. The Directory Agent will periodically refresh the list it maintains to ensure that all data sources are available and online. If multiple sources of information are found, then a list is compiled and this is transmitted back to the Access Agent with a recommended server choice (Figure 4.4, 3). Also, at this stage, the Directory Agent will check that the access class of the specific user provides the right to access the requested information. If this is not the case, the Access

Agent will be informed that the user does not have the relevant authority to retrieve the file.

- Once the Access Agent receives the server list, it will contact the recommended server, requesting authentication at the value of *confidence* score associated with the file that the user wishes to access (Figure 4.4, 4).

- The *confidence* score corresponding to this file will be ascertained and transmitted back to the Access Agent (Figure 4.4, 5).

- At this stage, the physician may be informed that a number of biometric samples will need to be collected to ensure the release of the information, depending upon the security level of the file (Figure 4.4, 6).

In the context of the present scenario, we shall assume that the physician has asked for a resource that requires a high level of *confidence* score to access.

- The Server Agent requests from the Access Agent that appropriate biometric evidence be gathered to satisfy the value of *confidence* score required to access the resource. The Interface Agent will invoke the biometric module for verification and acquire the relevant samples (Figure 4.4, 7).

- Once these samples have been acquired, the information is passed to the Fusion Agent for combination (Figure 4.4, 8).

In this example we shall assume that the *confidence* score associated with the voice sample is low, due to environmental noise, to demonstrate how the flexibility of the system copes with this situation.

- The combined *confidence* score is then passed to the Access Agent, (Figure 4.4, 9), which in turn transmits this result to the Server Agent. It is now the responsibility of the Server Agent to decide whether the score is sufficiently high to release the information. Assuming it is not, the Server Agent has several viable options:

1. Ask the Interface Agent to resample the biometrics.
2. Ask the Interface Agent to sample specific additional biometric modalities.
3. Ask the Interface Agent to invoke the Fusion Agent to re-fuse the existing data using a different fusion technique.
4. Ask the Interface Agent to request from the user appropriate non-biometric personal data that may assist in authentication.

Let us assume that, in this case, the Server agent has decided to suggest to the user that another biometric modality needs to be acquired to complete the requested action. For example, a facial image might be considered appropriate.

- The physician is informed that there has been a problem verifying personal identity, and that another biometric sample is required. The biometric module is invoked and the specified modality is sampled. Once again this information is passed to the Fusion Agent for combination. The overall *confidence* score is then passed to the Access Agent again for transmission to the Server Agent. Assuming that this new score is equal to or greater then the required *confidence* score, the server will release the file. The Access Agent will receive this file and pass it to the Interface Agent for the physician to view.

- If the *confidence* score is still not sufficient to allow release of the information, the Server Agent can enter a period of extended negotiation (Figure 4.4, 10) with the Access Agent as it attempts to ensure that identity is validated. This is achieved through the use of the options available to the Server Agent as detailed above. Once the Server agent is assured of the identity of the user the system will release the requested file to the Access Agent, and the Access Agent will decrypt the file and pass it to the Interface Agent so the user can view the file (Figure 4.3, 11).

Another case illustrating the flexibility of the system is that of a patient travelling outside the geographic catchment area of the usual physician. In this case, the patient would be able to see a physician in any location where the authentication facility is available. Any confidential information required can then be authorised through the patient also providing a biometric sample in order to confirm that the physician has the authority to access the records on behalf of the patient.

An enrolment process is invoked when the user accesses the system for the first time. This procedure uses a software wizard to guide the user through the process of enrolling with each of the modalities the system employs, and generates the user templates that will be used in subsequent verification attempts. The system will automatically attempt to obtain the best quality samples from the user during the enrolment procedure. The user will be allowed a number of attempts to enrol on each modality, but in the event that a template for a given modality cannot be generated, the system will note this and can attempt to re-enrol the user at a later stage.

4.4 Design Methodology

The initial challenge was in choosing a suitable design methodology that was appropriate for developing multi-agent systems. It is generally accepted that a comprehensive and rigorous methodology for developing multi-agent systems is lacking [Elammari, 1999], [Odell et al, 2000]. [Arazy et al, 1999] provides a comprehensive review of agent orientated analysis and design methodologies. This provided an ideal starting point for the selection of a suitable methodology in order to express the functionality of the IAMBIC system.

The methodology chosen for the IAMBIC system was GAIA [Wooldridge et al, 1999], [Wooldridge et al, 2000]. GAIA provides a system where all the aspects which are important in describing agent societies are present, individual agent aspects as well as social aspects; static aspects as well as dynamic, but only at a high level. The authors of this methodology acknowledge this weakness, however, the reasoning behind this strategy is to allow the low-level design and implementation details to be open-ended, enabling the designer to select the appropriate architecture and programming language.

It was this aspect that lower level implementation details were not covered by this methodology that was attractive for the IAMBIC architecture. It allowed the suitable representation of the agent hierarchy within the system at an early design stage, whilst further consideration of the actual implementation strategy was still under consideration between the project partners.

4.4.1 GAIA methodology description

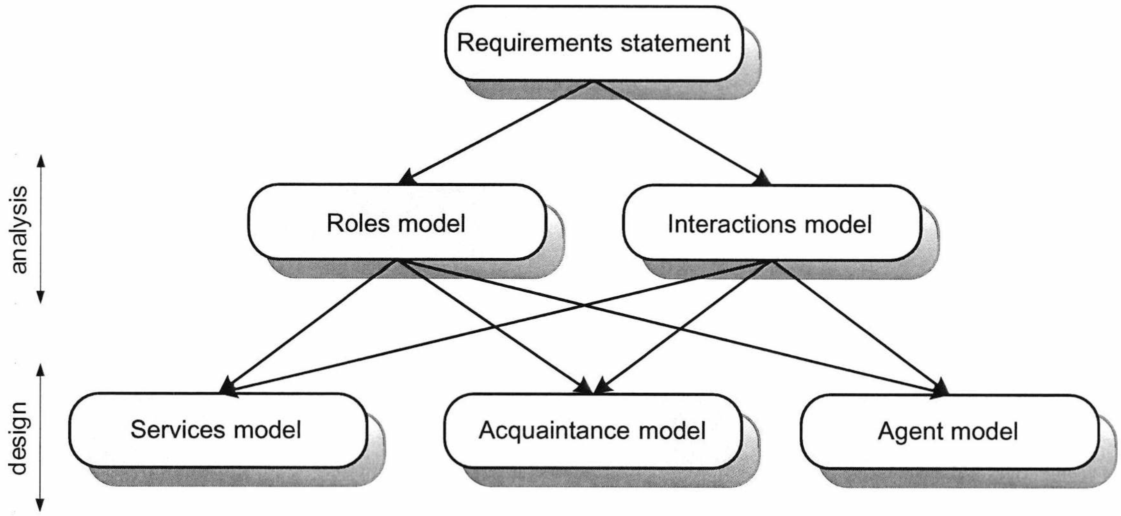


Figure 4.5 GAIA models

Figure 4.5 illustrates the main models employed in the GAIA methodology. There are two main types of entities introduced in the GAIA methodology, *abstract* and *concrete. Abstract* entities are used during the analysis phase in order to conceptualise the system, however, they may not have a direct realisation in the system. This is in contrast to *concrete* entities which are used in the design process and have direct counterparts in the final run time system.

In the application of the GAIA methodology the designer moves through a process of moving from abstract concepts to more increasingly concrete concepts. The authors describe "Each successive move introduces greater implementation bias, and shrinks the space of possible systems that could be implemented to satisfy the original requirements statement."

The GAIA methodology contains two phases, an analysis phase and a design phase. As can be seen from Figure 4.5 the analysis phase contains two models, the roles model and the interactions model.

The analysis phase is used to develop an understanding of the system and its structure. The authors make the point that an agent-based system can be viewed as an artificial society or organisation. This notion is beneficial because of the need to identify the roles which each agent may play in the overall system.

The roles model is used as an abstract descriptor of the expected function of the entity. Each role has four attributes associated with it: responsibilities, permissions, activities and protocols. This agent based system is a collection of roles that have certain relationships with one another, and also participate in patterns of interaction with other roles.

Responsibilities of a role can be expressed as determining functionality and as such are the key attribute of a particular role. Responsibilities are divided into two variants, liveness and safety properties. Liveness properties are conditions that the agent is attempting to satisfy given certain environmental conditions. These could be described as the goals of the agent. These liveness properties are generally specified using a "liveness expression" which classifies the "life cycle" of the role. These expressions are similar to the format used in FUSION [Coleman et al, 1994], although an additional operator is introduced 'ω' which is used to denote infinite repetition. These liveness expressions specify the potential execution path through the activities and interactions associated with the particular role.

Safety properties ensure that there are some invariant conditions which are required by the system across all states of execution. E.g. "Ensure that maximum vehicle speed does not exceed 30mph".

Permissions are used to identify the resources that are required in order to realise responsibilities. These permissions typically refer to information resources, i.e. a role may require the agent to read some particular information, alter another piece of information or even generate information. Permissions in GAIA make use of a formal notation that is based upon FUSION notation for operation schemata [Coleman et al, 1994].

Activities of roles are actions which do not require the interaction with other agents. These activities can be viewed as *private* actions in the sense of the object orientated programming paradigm. Protocols are used to define the interaction that the role can use to interact with other roles. E.g. a "seller" role may have associated with it the protocols "Dutch auction" and "English auction" [Anthony et al, 2001].

Role Schema: *Server Manager*

Description: The server manager ensures that no information is released from the database without the relevant security levels being satisfied. This manager negotiates with the access manager for the release of the requested information. This may involve the request for additional biometrics if the confidence is low or the user is not willing to donate the specified samples.

Protocols and Activities

NegotiateRelease, ProvideSecurityLevel, ReadSecurityLevel, RefreshDBSources, AwaitFileRequest, EncryptData, RegisterService

Permissions

Supplied	filerequest	// the file that it being requested by the user.
Generates	securitylevel	// the level of security for the requested file.
Generates	ServerService	// a list of services that the server can offer.
Generates	file	// the file the user has requested.

Responsibilities

Liveness:

([RegisterService].AwaitFileRequest.RefreshDBSources.ReadSecurityLevel.EncryptData. ProvideSecurityLevel.NegotiateRelease)$^{\omega}$

Safety:

- Must register to process filerequests.

Table 4.1 Server manager role schema

The roles model was employed in order to attempt to capture the behaviour of the IAMBIC system at this initial design stage. For each of the proposed system components shown in Figure 4.3 a role schema was constructed. One of these role schemas is shown in Table 4.1. The remainder of the schema for the project are shown in Appendix B.

4.4.2 Detailed Design Methodology

The GAIA methodology was useful to initially attempt to classify some of the core functionality required to produce the agents themselves, but for the more low level details such as the individual classes and the communications between the agents which the authors of GAIA acknowledge are not specified in any great detail. It was proposed to use the Unified Modelling Language (UML) [UML, 1997], with some specific alterations in the language to accommodate agent-specific communications, Agent UML (AUML) [Odell, 2000].

UML provides twelve diagrams divided into three categories: Four diagram types represent static application structure; five represent different aspects of dynamic behaviour; and three represent ways to organise and manage application modules. UML enables designers to model any type of application running on any type and combination of hardware, operating system, programming language and network. By using the Meta-Object Facility (MOF) metamodel, which defines *class* and *operation* as fundamental concepts, UML becomes the intuitive choice for object orientated languages and environments. Although the usage of UML is not restricted to the purely object orientated domain. Non object orientated applications and also can be modelled and UML can even be employed for business modelling and modelling of other non-software systems.

AUML seeks to build upon the existing UML constructs to provide enhancements in a number of these diagrams in order to facilitate the development of a sufficiently detailed agent based programming system. Investigations into these extensions for UML were conducted by the OMG Agent Work Group [OMG, 1997] and also FIPA. The FIPA Modelling Technical Committee was established to develop vendor-neutral common semantics, meta-model, and abstract syntax for agent-based methodologies [FIPA, 2003]

The proposed extensions cover a number of modifications in the Class, Sequence and Interaction diagrams. Work in this area has been extensive and a document has been produced detailing these proposed enhancements [Odell et al, 2004].

A number of diagrams were identified as present in the AUML that would be beneficial in order to specify further the behaviour of each of the system entities. AUML diagrams are presented in Appendix B for the IAMBIC system, although some diagrams are presented for the server agent in order to demonstrate the nature of this particular methodology.

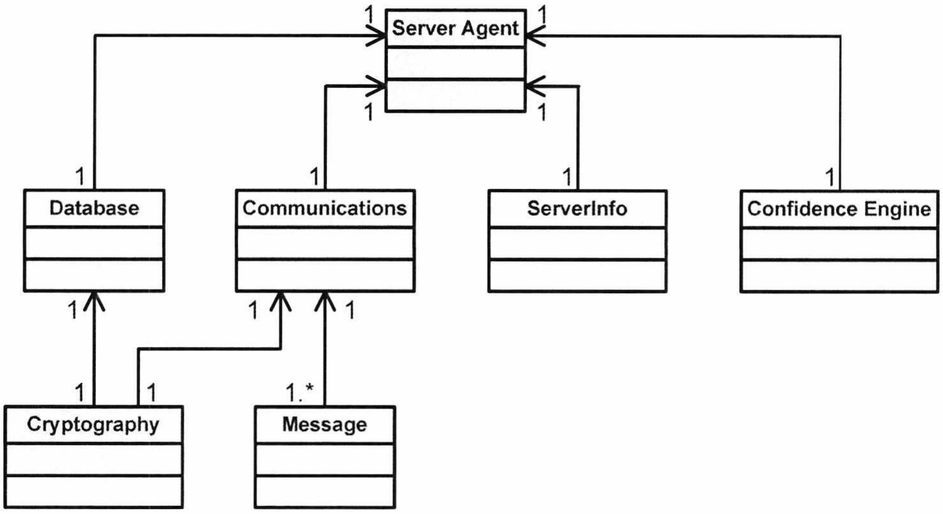


Figure 4.6 Server agent class diagram

The classes encapsulate the required functionality of the system, a brief description of each class is given below.

- ServerInfo

This class represents the information services that the server can offer. The information contained in this class will be transmitted to the Directory Agent upon registration. If the details of the services offered change during the lifetime of the server, this information can be conveyed to the Directory Agent so that it can update its dynamic information list.

- Confidence Engine

This class will be used during the negotiation phase of the conversation with the Access Agent. The knowledge contained within this class will enable an informed decision to be made whether the level of confidence received from the user with respect to the requested file is sufficient to warrant the release of the file. This class can also make suggestions based on the information received in order to attempt to validate the user through other means such as re-sampling of biometrics or modification of the fusion process or even the supply of non-biometric personal data.

- Database

This class is responsible for the maintenance of the information contained within the server's database. It will also retrieve the security level associated with the requested file and extract the file itself. Provision will be made in this class for facilities to modify the data contained within the database as required.

- Communications

This class is responsible for implementing the particular ACL that will be employed in the system. It will provide methods for sending and receiving messages and reporting any message failure conditions that may occur.

- Message

This class encapsulates the ACL message that will be passed to the communications class for transmission.

- Cryptography

This class will be responsible for the cryptography used in order to protect the content portion of any message passed between the agents. It will also be used to protect sensitive data contained within the Server database.

Another useful diagram that is present in AUML is the Agent Interaction Protocol (AIP). These diagrams are used in order to depict the flow of communications between separate agents. Figure 4.7 illustrates the AIP for the interaction between the Server agent and the Access agent.

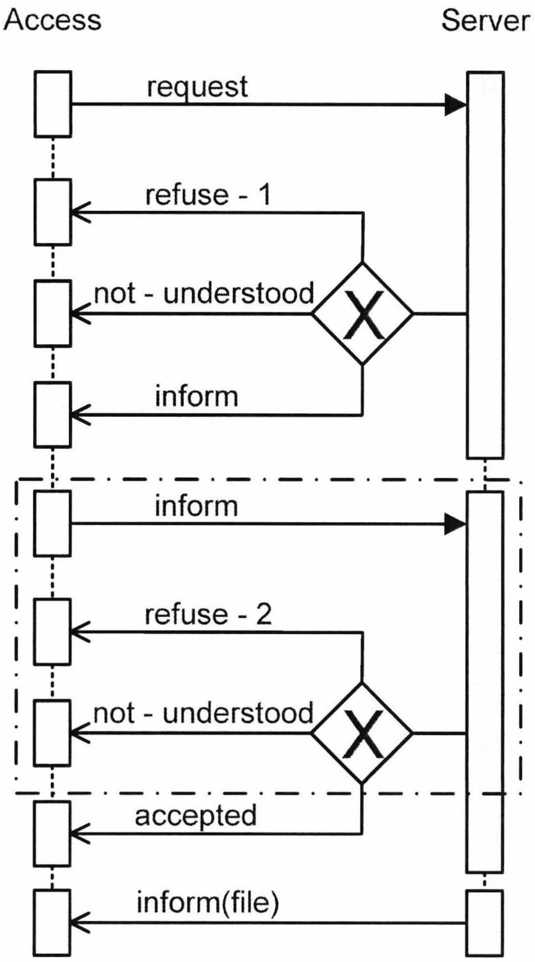


Figure 4.7 Server agent AIP

The Access Agent requests the file from the server. The server can respond in one of three ways:

1. The Server Agent can refuse. This failure condition could represent the condition where the requested file was not found within the server's database.
2. The message could be corrupted in which case the server responds with the not-understood condition. The receipt of this type of message would require a retransmission of the specified message by the Access Agent.
3. If the file is found then the Server Agent replies to the Access Agent with the relevant security level associated with the file.

The Access Agent then replies with the global confidence score obtained from the user biometric after the data fusion process. The Server Agent is then is a position where the following messages could be sent back to the Access Agent.

1. The server has examined the confidence score and has determined that the received level is not high enough to warrant the release of the file. The message returned will indicate this and a suggestion will be conveyed back to the Access Agent from the confidence analyzer class.
2. The message could be corrupted in which case the server responds with the not-understood condition. The receipt of this type of message would require a retransmission of the specified message by the Access Agent.
3. The confidence score received has been deemed satisfactory by the confidence analyser class and the server can release the file. The Access Agent is informed of this.

Upon successful authorization the file is released as denoted by the last message condition on the Agent Interaction Protocol (AIP).

The dashed box shown in Figure 4.7 indicates possible conversation iterations as the Access Agent may need to send several modified confidence scores to the Server Agent in an attempt to authenticate the user for the requested resource.

The message types in this AIP are general; the actual *performatives* used in the implementation would depend on the ACL that is used. It is assumed that the particular ACL employed will cater for the condition where messages cannot be delivered as well as the acknowledgment of messages and hence these details are not explicitly included in the interaction diagrams presented in Figure 4.7. The ACL used in the IAMBIC system is described in the Section 4.5.

4.5 Implementation

In Section 4.4 the functionality of the IAMBIC system was described using two methodologies that could capture the requirements at a suitable level for subsequent implementation. This section describes the possible choices in implementation strategy for the system and provides reasons for the adoption of a particular type of technical solution.

4.5.1 Agent Architecture

From the GAIA and AUML constructs for IAMBIC it was decided to utilise a reactive based approach to the implementation of the agents within the system. Using a reactive approach (Section 3.1) was attractive for a number of reasons. From the design methodology it became clear that the agents within the system would not need to perform abstract reasoning. In order to construct an agent that can perform some form of deliberative reasoning, a significant amount of work is required to describe and build the knowledge base and associated goal directed behaviour rules. Both these factors were seen as an unnecessary complexity as the environment in which the agents would be located could be explicitly catered for a design time.

The GAIA role schema for each entity clearly defined a rigid life cycle for each entity which could be easily translated into code. The interaction diagrams within AUML also provided a clear and unambiguous pattern of interaction amongst agents exchanging messages across the network. This further enabled the passage from design to implementation to be straightforward.

4.5.2 Modality choice

The IAMBIC system required a number of modalities in order to demonstrate the benefits of multimodality. As seen in Chapter 2 there is a large and diverse range of possible biometrics to choose from. One of the key factors in making the decision to employ a particular biometric was one of cost, other factors considered included the physical hardware required to obtain a biometric sample and also the level of user interaction required to obtain a sample of the chosen biometric. After performing an initial review of the biometric technology available at that time, it was decided to employ three modalities, fingerprint, voice and face. These modalities can be collected in a cost effective manner and also present a low transactional complexity towards the user.

4.5.2.1 Fingerprint

The hardware chosen for this biometric came from Secugen. This company produces a number of fingerprint based devices for biometric use. The Secugen IntelliMouse III© FDU02™ incorporated into fingerprint sensor in the side of a mouse. This allows for simple sample donation from the thumb of the right hand as the optical area is in that particular region. This is illustrated in Figure 4.8. By incorporating the mouse and the biometric sensor in one package it allows the user to continue using the computer in a normal fashion whilst enabling the gathering of biometric samples.

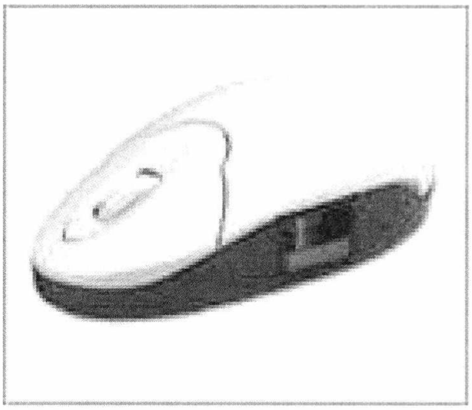


Figure 4.8 Secugen IntelliMouse III ©

To enrol the user must donate two samples from the same finger. At the time of capture there is a *quality* parameter which can be specified in software, this relates to the minimum level of image quality that is necessary for a valid image to be acquired. The *quality* parameter can be employed to indicate that the sample the subject has given contains the necessary features and distinctiveness to promote efficient feature extraction

To enrol a *security* level is also specified, this figure can be used in order to manage the FAR of the system. The *security* level relates to the variability permitted in the distances returned from the pattern matching algorithm, when comparing a number of samples in order to classified as a successful match. The more variability permitted results in a system which provides more tolerance towards poor quality samples the user provides, however, increased instances of false accept may occur due to the degree of variability allowed in the matching process. Higher levels of *security* level the subject can be enrolled and subsequently verified at indicate with more confidence that the subject is whom they claim rather then an impostor. A table illustrating this is provided by the manufacturers and is shown in Table 5.2. Both the *security* level and *quality* setting for this biometric will be covered in more detail in the Chapter 5 in which the adaptive interface layer is presented.

If enrolment is successful a 400 byte template is created from the information contained in the two samples. Verification is performed by loading the template and then passing a new sample to the verification method also specifying a security level, the method returns whether the operation was successful or not, indicating verification success or failure at the value of security level passed to the method. [Secugen, 2001]

4.5.2.2 Face

This facial recognition package chosen for use within IAMBIC was supplied by Visionics [Identix, 1982]. The FaceIt® verification software development kit [Identix, 2001] was the particular software package employed to provide a facial recognition component in the system. The majority of facial recognition systems available on the market are designed to work with a diverse range of video input devices, including standard webcams connected to the user's personal computer, this enables this modality to be employed with the minimum of additional cost to the end user. Visionics was chosen primarily as it was the market leader in the facial biometric sector.

To enrol the user is required to sit in front of the webcam for a period of fifteen seconds (software programmable). During this period the software is capturing images of the face, usually a minimum of four facial images are required for enrolment. A template is constructed from the images in the enrolment battery. The template file is in a proprietary format although the enrolment images can be extracted from the constructed template file by using the SDK. Depending on the number of enrolment images acquired the template size is approximately twenty kilobytes.

Verification is a similar procedure to enrolment for the user. The template is loaded and then the user is prompted to sit in front of the camera whilst verification occurs. The period of time allowed for verification is software programmable. The manufacturers recommended value of twelve seconds was employed. During this time the software is attempting to perform matching on the supplied facial image, this generates a score from 0 to 10. There are three recommended threshold values for verification 8.2, 8.6, 9.2 representing the lowest acceptable verification score to the highest. A user scoring equal or above the specified threshold is considered to have passed verification. These recommended threshold values for verification were extracted from the manufacturer's SDK literature

4.5.2.3 Voice

The final modality that was employed was voice-based. This solution was provided by VeriVoice [VeriVoice, 2000]. This particular package utilised text dependant speaker recognition using Hidden Markov Models (HMM). Enrolment required the subject to repeat twelve differing number strings each of five numbers (e.g. 12043, 32051, 42510). From these twelve number strings a user template was constructed at the end of enrolment. The template is approximately 16 kilobytes in size.

Verification involved using the previously recorded template and one of the number sequences spoken during enrolment selected randomly. The verification method returns a value which indicates whether the subject has passed verification or failed.

Not only was cost a motivating factor in employing these modalities but also the user acceptance of the chosen modalities. The modalities chosen also represent biometrics where the interaction made with the user during enrolment and subsequent verification should be minimal, i.e. the interaction with these biometric devices should be straightforward and intuitive.

After the selection of the necessary biometrics the next step was to determine the programming language that would be utilised in order to interface with the devices and perform vendor specific enrolment and verification routines. The Application Programming Interface (API) that was supplied by each vendor supported a number of development environments (Microsoft Visual Basic, Microsoft Visual C++ and Borland C++ Builder). It was decided to employ the Borland C++ Builder environment due to experience already held with that particular environment. This enabled application development to start after the initial design phase had been concluded, rather than incorporating a delay whilst familiarisation with a new programming environment was made.

4.5.3 Data Storage Considerations

Throughout the distributed components within the IAMBIC systems there exists a clear need for a form of standardised data container for exchanging data. There are a number of data sources within the system, both on the client and on the server. Client side data sources include the biometric application, and also the access agent which is responsible for negotiating the file from the server and ultimately receiving the file. The server side database is the primary data source, however, the server agent is also generating messages for the access agent.

The first issue to address was the format in which the inter agent communication would occur. The content of the messages that are passed in the system can be represented in XML (eXtensible Markup Language) [OASIS, 1998], a markup language similar in syntax to HTML (Hyper Text Markup Language). XML was designed to describe and encapsulate data. XML can also be used to exchange data between systems. This means that the agents in the system can all interact using the XML payload in the message, and this can be used to retrieve data from the server database directly. The reason for choosing XML is its universal syntax that allows ease of translation, transformation, parsing, presentation and validation with a variety of standard mechanisms. XSL (eXtensible Stylesheet Language) can be used to transform and format XML messages so that the interface agent can directly display the retrieved data in a web browser window.

A Document Type Definition (DTD) was drawn up to act as a template for the generation of the XML messages. This DTD can be used for XML data binding within the Borland JBuilder programming environment, this effectively generates classes for the XML document and enables it to be accessed as an object in its own right.

XML was also used to record the transactions the user made with the biometric application. This file provided the relevant information so the fusion agent can calculate a confidence score from the values recorded from verification samples in the biometric application. On the server side incoming XML messages are transformed into SQL queries which are executed on the server database which contains the relevant information.

4.5.4 Agent Communication

An important consideration for the IAMBIC system was the type of communications approach that would be employed for passing messages between the various agents within the system. As mentioned in Chapter 3 there are a number of ‘off the shelf’ solutions available for inter-agent communications. It was decided to employ a Remote Method Invocation (RMI) based technique for communications known as SACI (Simple Agent Communications Infrastructure) [Hübner and Sichman, 2000].

The SACI Java API allowed rapid prototyping of an agent with the ability to perform KQML *performative* based messaging. Creating an agent was achieved by deriving a class from the provided base agent class. This creates a basic template for an agent with communicative functionality. The programmer is left to write the code to produce the desired behaviour of the agent.

This SACI API also provided a number of other useful facilities to the programmer. A ‘Yellow Pages’ service is available which can be used in order for agents within the society to register their skills and also query what skills are offered by other agents. A ‘White pages’ service is present which enables the messages to be sent using just the receiving agent’s name. The Facilitator or White Pages service enables the agent’s location in the network to be transparent to the agent sending the message. Both these services are convenient for the implementation of communications amongst distributed agents.

4.5.5 Component Implementation

This section details the specific implementation of each of the system components identified in Section 4.2.

4.5.5.1 Client Interface Cluster

The decision to employ Java to provide the network communications led to an issue with the client application, which required another programming language to be used in order to interface with the biometric devices. To alleviate this problem the decision was made to separate the previously described user interface cluster into two separate applications as shown in Figure 4.9.

The Java Native Interface (JNI) [JNI, 1999] provides a mechanism where Java code can operate with libraries and application written in other languages such as C/C++. To use this technique with third-party vendor tools requires a large amount of code transformation in order to implement this native interface. The alternative approach is to build an executable in another language (such as C++), and use the shell commands available within the Java language to launch the executable. This second technique was adopted as it was deemed the least technically complex and also allowed development of both applications to continue without incorporating delays as the intricacies of managing the native interface were exposed.

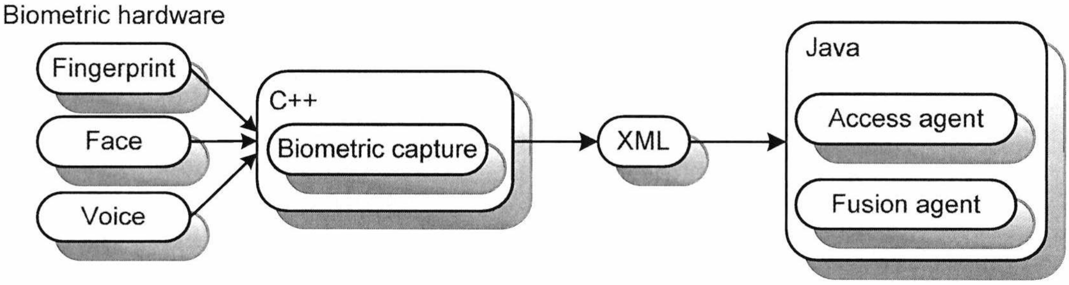


Figure 4.9 Modified IAMBIC client structure

The rationale behind this approach was to enable the application that enrolled and verified the users to act as a standalone application. This application would be solely responsible for capturing samples and reporting these results in the XML history file of the current user.

The access agent and the fusion agent were combined together as an application. This application was programmed using Java and could launch the biometric application in order to gather biometric samples in order to verify the identity of the current user.

4.5.5.2 Access and Fusion Component

As already discussed these entities were written in Java. Using this application the user can initially log in to the system by providing a set of biometric samples. This application also acts as the database interface for the user, using this interface the database can be browsed and files sent and received. Details of the fusion algorithm employed can be found in Chapter 6. Screenshots of this application can be found in Appendix B.

4.5.5.3 Biometric Capture Application

This application now becomes the primary interface between the user and the biometric systems. The interface is presented in a 'wizard' fashion, in which the user is guided through the donation of the necessary samples. There are two main modes of operation of this interface, enrolment and verification. Enrolment is required before system use, and the verification mode is employed at all other times the user interacts with the system. Screenshots of these interfaces are shown Appendix B.

4.5.5.4 Directory agent

In the initial system design specification this entity was identified as providing a role very similar to that of the Facilitator within the SACI API. The role specification of this agent describes an entity that is responsible for storing and updating relevant information about the location of services within the network. The decision was made to remove this entity from the system, although it was agreed that in order to still realise the proposed functionality of the system, individual server agents could register with the facilitator and advertise the services they posses in a particular domain. Client agents could then interrogate the facilitator to look up the server agents which contain the data that is required. This replicates the functionality of the directory agent, although there would be no need to explicitly create an agent to achieve this, thus reducing the programming overhead associated with creating this agent from scratch. Figure 4.10 illustrates the final system implementation.

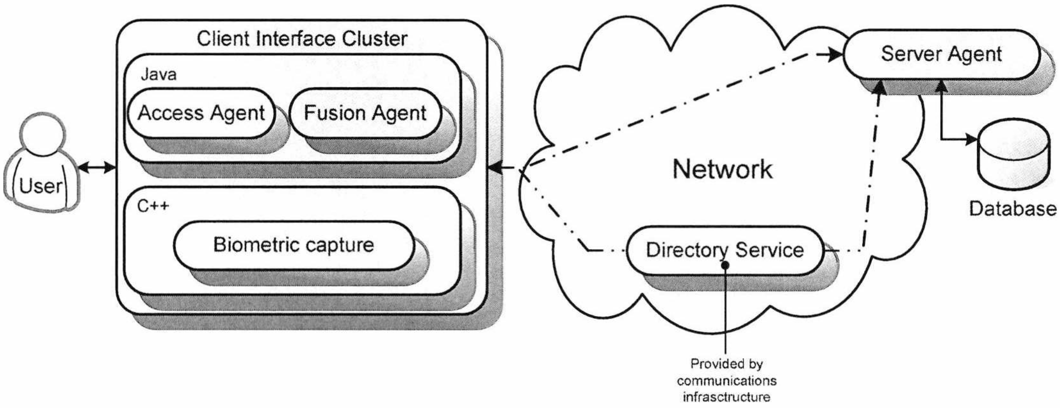


Figure 4.10 Final system implementation

4.5.5.5 Server agent

This implementation of this entity was the responsibility of NeuSciences. From the initial GAIA and AUML constructs it was decided that the agent would be developed using Java [Gosling et al, 1996]. This implementation approach was chosen for a number of reasons, NeuSciences had a long history of using Java for their commercial applications and also Java provides feature rich libraries to connect to databases. As seen in Chapter 3 the Java programming language lends itself to the construction of multi agent system through its intrinsic network capabilities and also the security model that it employs.

4.6 Conclusions

In this chapter the IAMBIC system has been introduced. The motivation behind the project has been established and discussed. The design section outlines the methodology that was employed in order to effectively capture the properties of the desired system. The next section addresses the relevant identified components that are required to realise the system requirements. These are presented and their behaviour specified.

The implementation section discusses the particular biometrics that were employed in the system and the rationale behind their choice. The granularity of the system is increased as specific implementation strategies are considered for the biometric interface, communication language and the client and server application language. Some changes to the system architecture are also proposed and concluding this chapter the final system implementation for IAMBIC is illustrated

Chapter 5

Adaptive Interface Layer

In this chapter the adaptive interface layer will be introduced. The motivation for the development of such an interface for use in a biometric system will be established. Design strategies for the adaptive interface will be discussed and core components required to produce the functionality will be explored and specified. Towards the end of this chapter implementation specific details are examined and documented.

In order to determine some of the system parameters required for the operation of the adaptive interface layer a number of experiments were conducted, these are detailed in the relevant section of this chapter.

5.1 Adaptive User Interface Motivation

It was during the biometric sample gathering trial phase conducted as part of the IAMBIC project at the University of Kent [Fairhurst et al, 2002] that it was observed that user interface issues with biometric systems could have a major impact on usability and user acceptance. These trials were conducted using individual software applications written for each modality. A supervisor was present during the trials to assist the subject in the donation of samples. The application interface itself offered no form of guidance in the donation of samples neither did it provide any failure analysis.

The trial was conducted with cooperative subjects who had not been previously exposed to any form of biometric technology. Table 5.1 illustrates the FTE rate for each modality.

Modality	Failure To Enrol (FTE %)
Fingerprint	14.4
Face	27.1
Voice	12.5

Table 5.1 FTE rates for IAMBIC biometric trial phase

What is surprising about these figures is that it appears that even modalities which require very little interaction (i.e. the face modality) still exhibited a FTE rate. The voice modality appears to be the most robust when enrolling subjects. Subsequent verification performance is not based upon the ease of enrolment, from the results obtained it was clear that at least these modalities posed a problem for a portion of the test subjects to interact with to produce samples of sufficient quality for enrolment. It appears that since users are experiencing a degree of difficulty utilising the system then some form of assistance in donating samples might prove useful to this group of users, both during the initial enrolment stage and also the subsequent verification phase.

Another obstacle facing the operation of biometric systems is the need to have a supervisor present during enrolment to ensure that the necessary steps are being taken in order to generate templates of sufficient quality. This training phase which is required to promote user familiarity towards the device is seen as an added expense and complicates the deployment of biometric systems. It has been noted that user performance can improve as the user becomes familiar with the device and the donation of samples, a process known as habituation [Mansfield and Wayman, 2000][INCITS, 2005].

There exists a need for a suitable mechanism that can be seen to accelerate this habituation process. If habituation can be accelerated then user performance would improve and also user acceptance of the device would grow. In order to affect this process some form of user assistance and training would have to be provided but this would also have to occur unsupervised to fully realise the notion of an unsupervised biometric system.

Primarily the greatest contributing factor to the poor user acceptance of a biometric system is seen to be the number of false reject conditions occurring. Obviously this is the most palpable question to a person from a non-technological background is "Why did the system not recognise me?" The answer to that question could have a number of contributing factors extending from user error to some form of hardware fault. What is required is some manner of automated assistance that can be invoked so that the user can have a more suitable and expressive explanation for failure other than "Verification Failed", in essence a dynamic interface that could respond in an appropriate fashion towards the user during the sample acquisition phase.

Other issues which can also affect performance in any biometric system are related to template issues. Poor enrolment templates can hinder subsequent verification attempts and during the operational lifetime of such a system. The template aging process cannot be ignored. (Section 2.10.2)

It is the user interface aspect as well as the underlying template management issues that prompt the need for the development of such a system which would attempt to address these problem areas. It is in the realm of software agents that a possible solution to this problem may exist. In Section 3.6 Interface agents were introduced as entities that could be employed to assist the user in some task or provide training in a given task. If the user interface could be designed in such a manner as to provide informative advice about the donation of samples and also provide a degree of fault analysis on verification failures, these factors could effectively aid in assisting and training the user.

The adaptive interface layer is expected to exhibit a number of agent properties. *Adaptivity* "the ability to modify an internal representation of the environment through sensing of the environment in order to change future sensing, acting and reacting for the purpose of improving assistance". *Autonomy* "the ability to sense, act and react over time within an environment without direct intervention". *Collaboration* "the ability to communicate with other agents, including the user to pursue the goal of offering assistance to the user".

5.2 Design

As well as providing a robust user interface, a number of other components which are coupled to the interface can be identified as contributing to overall system behaviour. These include template management utilities, sample analysis and calibration routines. From these entities the basic system architecture can be constructed, as shown in Figure 5.1.

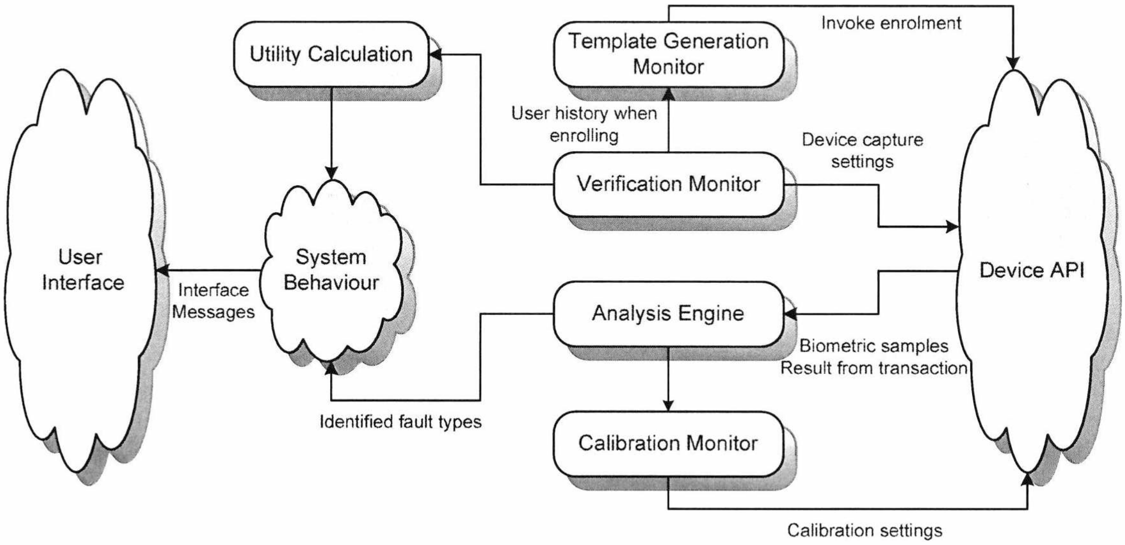


Figure 5.1 Adaptive interface architecture

Figure 5.1 illustrates the proposed system architecture for the adaptive interface layer. It can be seen that these components are designed to act as an intermediate layer between the user interface and the device API which is responsible for sample acquisition, enrolment and verification procedures.

To demonstrate this concept it was proposed that this adaptive interface layer could be applied to a specific modality in order to asses its effectiveness. The biometric chosen for this experiment was the fingerprint modality. This particular biometric provided the necessary pre-requisites for the translation to an adaptive interface approach. Two parameters existed in the particular commercial system which were used for enrolment and verification procedure which could be used in order to generate differing level of system behaviour.

- '*Security* Level'. This parameter (threshold) is related to the False Accept/False Reject rates of the system and can be set by the end user. The higher the *security* level, the lower the expected probability of false accepts, although the probability of false rejects generally increases. There are nine levels of security settings, Table 5.2 illustrates the manufacturer's data regarding security levels and false accept/reject rates. These *security* levels represent distinct threshold values at which the matching distance between given sample and template must satisfy in order to be classified as a match.

Security Level	FAR %	FRR%
1	0.0496	0
2	0.0110	0
3	0.0030	0.11
4	0.00025	0.22
5	0.00013	0.22
6	0	0.45
7	0	0.67
8	0	1.45
9	0	1.79

Table 5.2 Secugen IntelliMouse III© FDU02™ FAR/FRR rates

The value of *security* level is used in both the enrolment and verification phase. During the enrolment phase two samples are acquired from the user. These two samples are passed to a registration method which also takes the *security* level as an argument. The Boolean result returned as the result of this operation indicates whether the template could be created from the supplied samples at the specified *security* level.

Acquiring samples from the subject that enable enrolment at higher levels of *security* can result in a system that theoretically should exhibit lower rates of false acceptance. This is because the security level is also employed at the verification phase, however, a trade off is made against the rate of false rejection. Enrolling a subject at higher security levels province enhanced robustness towards instances of false accept for the system. Templates can be enrolled at the highest possible *security* level by employing a template generation procedure shown in Figure 5.2. This calls the enrolment method in a loop, incrementing the *security* level every iteration, the loop exits when the enrolment method fails. At this point the enrolment security level is equal the value of the loop counter.

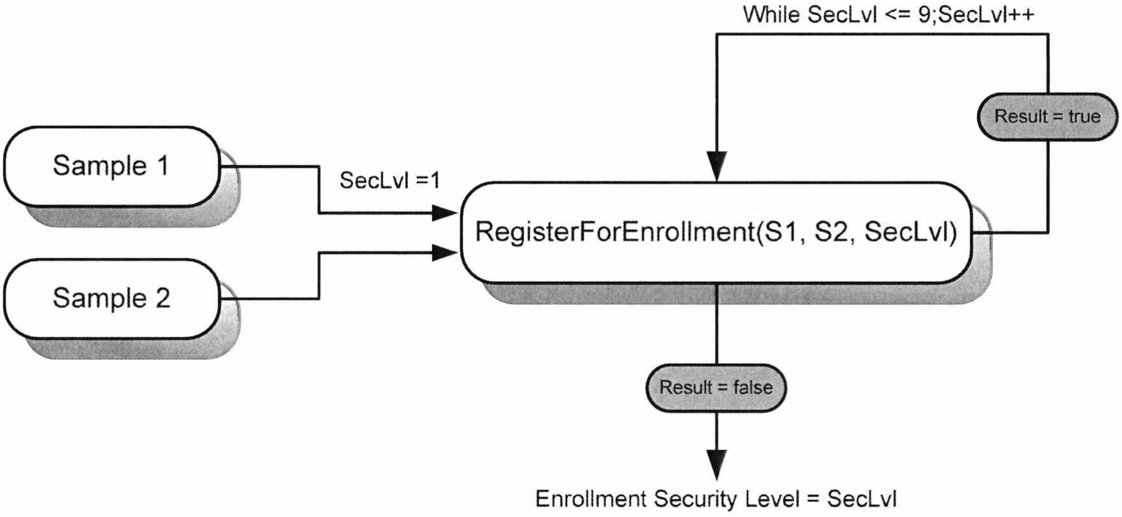

While SecLvl <= 9;SecLvl++
Result = true
Sample 1
SecLvl =1
RegisterForEnrollment(S1, S2, SecLvl)
Sample 2
Result = false
Enrollment Security Level = SecLvl

Figure 5.2 Template acquisition method

During verification the template created during enrolment is passed to a verification method along with the *security* level. The Boolean result returned from this operation indicates whether the subject sample successfully matched at the specified *security* level. Matching can only take place up the security level at which the enrolment template was created, attempting to match at higher levels causes unpredictable results. This further reinforces the notion that during enrolment the highest possible *security* level should be used when creating templates. The security level employed during verification can also provide degrees of flexibility towards user authentication on a dynamic basis based on the content of accessible resource. A subject providing samples which can be matched at higher levels of security indicates greater confidence in the claimed identity of the user against the stored template. Subjects who can also provide samples both for enrolment and subsequent verification which can be matched at higher levels of security tend to indicate users that can confidently use the system.

- 'Quality Setting'. This parameter is related to the image quality that is acquired by the device sensor. The value of this quality setting ranges from 0 - 100. This parameter is employed during the sample acquisition phase. If the figure specified for image quality is not met then no sample is acquired and the result of this operation is marked as failure to acquire.

Figure 5.3 illustrates how these parameters are employed in the sample acquisition and matching process within the system.

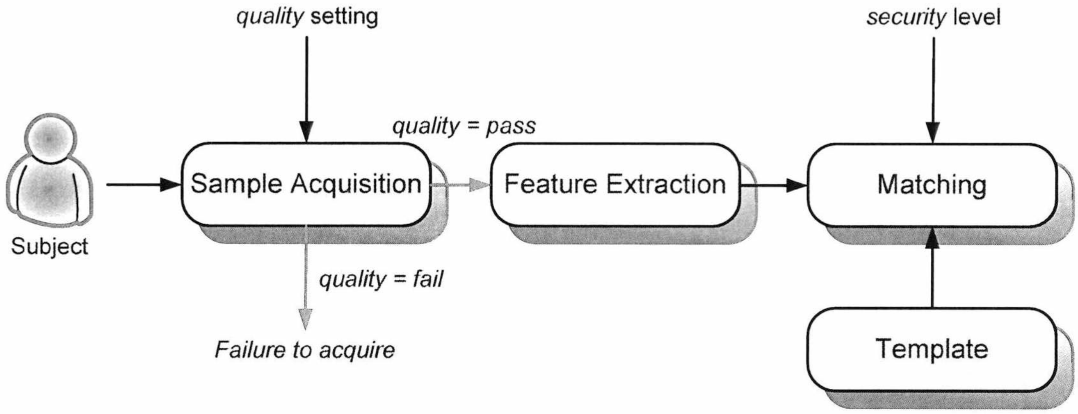


Figure 5.3 System parameter diagram

Also the nature of the images produced by the device itself lends itself to basic image processing techniques which could be used to determine possible causes of failure, further enhancing the robustness of the system.

5.2.1 Utility

To achieve the desired behaviour of the interface agent it was proposed to use a model of utility to determine what feedback was provided to the user during the operation of the system.

The use of the utility concept for user modelling in adaptive systems is not new [Brown et al, 1998], however, it is proposed to employ this notion in a different manner. In this proposed system this refers to the agent's utility with respect to the levels of security that the user is achieving. In other words the agent's degree of "satisfaction" is directly the capability of the subject to provide samples which result in enrolment and matching during verification occurring at higher *security* levels and also samples which can meet specified *quality* settings.

The goal of the system is to try and maximise its utility over the period of time for which the system is being used. The value of utility is directly used to change the behaviour of the agent with respect to the user. For example, an agent with a low utility score will aggressively attempt to aid the user with the donation of higher quality samples through the use of extended user assistance, whilst an agent with a relatively high utility score is less likely to offer the degree of assistance the low utility agent is exhibiting. It will still be able to offer help if the agent determines the user is experiencing difficulty in donating samples.

The process by which this value of utility is calculated is detailed below. Since the modality we are dealing with gives two attributes when dealing with samples donated from the user, Multi Attribute Utility Theory (MAUT) [Winterfield and Edwards, 1986][Schäfer, 2001]. MAUT can be used in this instance to generate an overall utility score from these two factors. According to MAUT, the overall evaluation *v(x)* of an object *x* is defined as a weighted summation of its evaluation with respect to its relevant *value dimensions*. In the system we are evaluating the user interaction based on the value dimensions of Security level and Quality setting. The overall evaluation is defined by the following *overall value* function which gives us the utility value:

$$v(x) = \sum_{i=1}^{n} w_i v_i(x) \qquad (5.1)$$

Here, $v_i(x)$ is the evaluation of the object on the i-th value dimension and w_i the weight determining the impact of the i-th value dimension on the overall evaluation (also called the *relative importance* of a dimension), n is the number of different value dimensions, and

$$\sum_{i=1}^{n} w_i = 1 \qquad (5.2)$$

To utilise each of the attributes using the MAUT approach, a scale must be constructed to characterise the *value function* of the attribute. For the purposes of this experiment simple a simple linear scaling function was applied to both the security and quality attributes. This simple scaling factor was chosen as an initial starting point to investigate system behaviour. The actual values employed and the rationale for these values is presented in Section 6.6.5. Figure 5.4 illustrates this value function for the security level attribute.

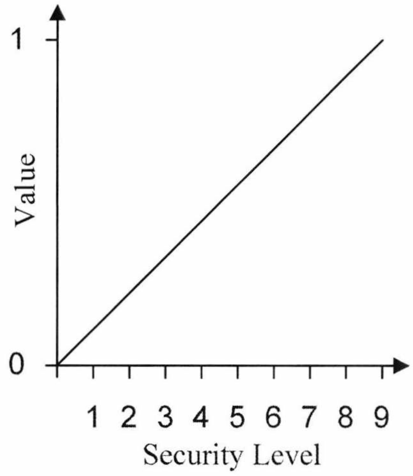


Figure 5.4 Value Function

There are two terms which need to be defined in order to adequately encompass the nature of the interaction which a subject will experience with the system.

- **Transaction**

  A transaction represents a single attempt by the subject to verify against the enrolment template which has already been created.

- **Session**

  A session encompasses a number of transactions which may occur within a fixed time frame. E.g. a subject attempts to use the system to verify their identity, at the first attempt the subject fails to verify themselves but at the second verification attempt is successful. In this case the subject has performed two transactions within one session.

5.2.2 System behaviour

The behaviour of the overall agent entity is based upon the current utility value for the user. After each session the user logs will be analysed to determine the overall utility score. The method with which this normalised session utility score is calculated is described below.

A score is produced for each session, this is based upon the parameters involved and equation 5.2. This initial session score is scaled based on the maximum figure of utility that the user can produce based on the security level and quality setting at which the user templates were acquired. This indicates the relative performance of the user when compared to the baseline enrolment visit. To calculate the normalised session utility the sum of the normalised session scores are divided by the total number of transactions that have occurred. This normalised session utility score represents the performance of the user over the last session. Periodic analysis of these normalised scores occurs during the operation of the system at these points the normalised session scores are averaged and employed to determine the behaviour level of the system (Table 5.3) with respect to the subject. This determines the degree of user feedback the system exhibits in order to attempt to increase the performance of the user.

Normalised Session Utility %	Behaviour Level
$\geq 0 \leq 25$	1
$> 25 \leq 50$	2
$> 50 \leq 75$	3
$> 75 \leq 100$	4

Table 5.3 System behaviour levels

A numerical example demonstrating the mechanics of the normalised session utility score is illustrated in Appendix B.

5.2.2.1 Behaviour Level 1

This level represents the lowest performance band. In this case the user is achieving poor verification scores. In this state the system is actively trying to increase the performance of the user. The system will examine any identified fault types and will enter into a dialog with the user in an attempt to determine the cause of this poor recognition rate. Some possible solutions may be offered to the user after this dialog. Another mechanism is to examine the identified fault types occurring and determine whether these faults are directly related to the poor scores. There is a threshold for the number of concurrent sessions that produce a score in this particular level, this threshold dictates the point at which the system gracefully declines from offering such active assistance. This action is taken due to the fact that in biometric systems users who cannot produce samples of sufficient quality exist, and no matter how much assistance is provided no improvement in system usage can be obtained, these users are known as 'goats'. For this type of user the use of a biometric system is a poor choice and some other form of identity authentication mechanism will have to be employed.

Naturally the user can indicate at the relevant time if this extended assistance is required, and the agent will take the necessary action.

5.2.2.2 Behaviour Level 2

This level represents a level of performance from the user which is still considered below average. The system may decide to take proactive action based on the threshold for the number of sequential level 2 sessions. A similar procedure may be taken towards fault analysis as exhibited in level 1, in which fault analysis and interactive user dialog is initiated which is hoped may lead to increased user performance.

5.2.2.3 Behaviour Level 3

This behaviour level represents a level of performance by the user that is considered adequate. The agent will monitor the performance of the user but will not actively provide assistance to the user unless explicitly requested. It is expected that this band will be the normal level of operation for the majority of the users of the system. Although the performance of the user is deemed adequate in this band no active attempt is made at reenrolment to acquire improved template samples.

5.2.2.4 Behaviour Level 4

In this level the user is achieving good verification scores. The system does not need to actively aid the user in the donation of samples, however, is still monitoring for specific fault types and will offer assistance on these if they occur. Sustained scores in this range indicate to the template generation agent that there may be a possibility of increasing the user's security level of enrolment templates (if possible) at the appropriate time. Although this particular commercial API only supports the combination of two images into the subject's template, it is conceivable that in this particular level of operation images which were deemed to be of good quality could be added to the template file, further enhancing the capability of the system to successfully verify a subject.

5.2.3 Verification Monitor

This agent is responsible for the monitoring and setting of the two parameters that contribute to the utility rating. It is recommended that the level of *security* that is used for verification purposes is kept at the level at which enrolment occurred or below as attempting a match at higher levels may lead to incorrect results. The matching procedure automatically attempts to match up to the enrolment *security* level. The result from this operation is used as one parameter to calculate the *utility* of the transaction. The *quality* setting of any acquired sample is handled in a different manner. Since this parameter plays a role in the calculation of utility the agent here has the opportunity to adjust this factor in order to increase the utility rating. This is achieved in the following manner: -

The manufacturers recommended value of the *quality* setting for verification transactions is 40. If the user was enrolled at a much higher level of *quality* (say 60), the system assumes that user should realistically be able to produce a sample of at least this value during the operational lifetime of the software. In order to proactively adjust this quality setting this agent examines the results obtained from the session logs.

If a user is achieving successful verification, at a rate above 75% (see Section 5.2.2) over the last session then the agent will attempt to increase the *quality* setting for the next user session. There is a roll-back mechanism in place here as mentioned below so if the analysis agent determines that the image capture settings are unrealistic for the user then these settings can be reduced, to a level which the user can be expected to achieve. The speed at which this agent increases the *quality* setting is determined the capability of the user to donate sustained high quality samples, however, there is one more factor which govern the speed at which this process can develop, this is:

The number of concurrent sessions in which the user has achieved the specified behaviour level required for an increase in the *quality* setting. The value of concurrency is chosen such that system gradually increases these *quality* values over a period of time as the interaction of the user with the system stabilises.

The values chosen for each of these parameters during the experimental phase and the rationale behind the choice of these values is detailed in Section 6.6.5.

5.2.4 Analysis Engine

This component is responsible for the examination of any user-donated sample. By examining the biometric sample and knowing the result of the transaction (i.e. match or non-match) this component can attempt to classify the nature of any failure using the error types described below.

5.2.4.1 TYPE A User Fault

These types of fault involve the poor usage of the device. The user is not donating a sample of sufficient quality (i.e. the image is too dark or too light). The system will have some initial parameter limits for establishing this fault type (Section 5.3.1). This type of fault also encompasses the condition where finger placement is possibly the cause of the verification failure.

5.2.4.2 TYPE B Device Error

To classify image acquisition problems we can take image samples from the sensor and calculate simple parameters to determine optical irregularities. A key test here is to attempt to determine whether the optical device is clean: this being an important factor that can influence the quality of the acquired image.

5.2.4.3 TYPE C Template Aging

The aspect of template aging here of course rests on what particular modality is being used. The majority of biometric modalities have to ensure that templates are kept current as the user characteristic upon which the modality is based gradually changes over time. These types of error will be difficult to classify. The only constraint we have used is based on a time limit for the validity of the template. If the period of time a template is considered valid (set in software) has been exceeded then the possibility exists that the error may be caused by template aging.

5.2.4.4 TYPE D Acquisition Parameter Error

This type of error may be flagged if the system believes that the current image acquisition parameters are unrealistic for the current user. This type of fault can be identified by keeping track of the current and previous acquisition parameters, stored in file for that user. After an unsuccessful transaction event the user is made aware of the possible nature of the fault as determined by the analysis engine.

The results from the analysis agent are logged in an XML file. This file is examined periodically and is used to determine which behaviour band the user is currently in. The parameters employed in the periodic examination of this file are illustrated in Section 6.6.5.

5.2.5 Template Generation Monitor

This component is responsible for the generation of user templates and also the reenrolment of the user at the appropriate time. During enrolment the goal of the agent is to attempt to acquire the best quality templates a user can realistically donate. At this time the quality setting is set to the manufacturers recommended setting for enrolment, this setting is used as the default to ensure that the largest number of users can actually enrol in the system. The templates are enrolled at the highest security level that is possible, this is achieved through employing the mechanism shown in Figure 5.2. This is in order to minimise the false reject rate by generating high quality templates.

Another mechanism for invoking reenrolment is triggered by the Analysis Engine described above. These methods of template management are intended to alleviate any possible template aging issues the user may develop with the system. The user can opt to reenrol at any time without having to be prompted by the application itself. In some applications the use of such a mechanism would require the user to supply some form of non-biometric data, either a token or a password in order to provide sufficient authentication to perform reenrolment.

At the time of reenrolment, however, a different process is invoked. Since it is known what levels the user has been achieving over the period of time since enrolment, the analysis of these results can determine whether there is a possibility that either the *quality* or *security* settings for this user can be modified. This process is augmented by another feature of this entity that it can accept samples from the analysis agent obtained from the user during the verification phase. In this manner the agent can increase its confidence in the ability of the user to produce a sample that is capable of reenrolment at higher *security/quality* levels, before actually asking the user to reenrol at these levels.

5.2.6 Calibration Monitor

Many biometric devices have the capability to be calibrated. In the fingerprint modality examined here, this is an important feature. Every subject will use the device in a slightly different way and environmental conditions may impact the acquisition of samples. Since the device requires the finger to be placed on the sensing area, each subject may apply a different amount of pressure. If the same calibration data is used for all the users this may lead to unacceptable image quality for a portion of the subject set.

In an attempt to rectify this problem, this system records calibration details for each user individually. The calibration process which is vendor specific, involves an automatic exposure control process. The subject presents their finger to the sensor, during the calibration procedure the software adjusts levels of brightness, contrast and gain in order to provide the optimum image acquisition parameters for subsequent feature extraction.

This calibration data is initially saved before the user can initiate enrolment, and is valid only for a specified amount of time. This value can be determined on a case by case basis for the target environment. If the analysis agent determines that the samples acquired are failing, possibly due to old calibration data then the user will be prompted to recalibrate the device.

This concludes the design section of the adaptive interface layer. Principal components of this layer have been introduced and their desired operation explained. It is envisaged that these components will work in a cohesive manner in order to affect the expected functionality of the adaptive interface layer. A UML class diagram of the adaptive interface layer is shown in Figure 5.5.

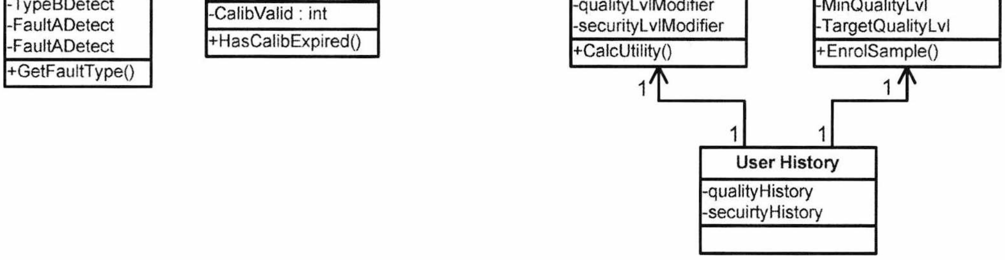


Figure 5.5 Adaptive layer UML class diagram

5.2.7 Data Storage Format

XML had already been chosen as the data format of choice for the IAMBIC project. Since the overall application was recording various parameters from the interaction the user had with the system, all that was required were some minor modification to the DTD in order to achieve the functionality required for the adaptive interface layer. These modifications primarily involved adding some data structures to hold data pertaining to details such as, device calibration and also parameters involved with the fingerprint device interaction (*quality* and *security* level).

Another XML file is created for each user that will be used to contain the results from the session mining, this file contains a history of the user's usage of the system with respect to overall behaviour level, and also records any instances where the system has increased the *quality* level setting as per the specified behaviour of the adaptive layer. The DTD's employed by this component are shown in Appendix B.

5.3 Implementation

The framework required for the implementation of the interface application has been developed by the author. The adaptive interface layer is presented as a natural extension to the already capable biometric capture application. A number of classes were implemented to satisfy the components identified during design time. These are detailed in the following section. These classes were written in C++ and were implemented under the Borland Builder C++ integrated design environment. This was performed in order to provide complete compatibility with the overall IAMBIC application.

5.3.1 Analysis Engine

5.3.1.1 Type A

As detailed in the previous section this component is responsible for the analysis of user donated samples. There are four differing conditions which this component is required to detect. The first fault type (Type A) is determined by examining the donated image. There are a number of basic image parameters which can be employed in order to detect this fault condition. An image analysis component was employed to extract the following parameters; the mean pixel value, the number of non-zero pixels as well as the standard deviation of the non-zero pixels contained in the image. By using these three parameters it is possible to classify a number of conditions relating to poor usage conditions. Primarily the conditions which should be identifiable by this process are illustrated in Table 5.4.

Condition
No Finger on sensor
Faint Image
Image too dark

Table 5.4 Proposed identifiable fault conditions

Finger placement issues are critical to the correct operation of the system, this component can tentatively identify incorrect placement of the finger on the sensing device. If a sample fails verification but cannot be classified into one of the conditions shown in Table 5.4 and also doesn't fall into one of the other fault categories then the fault is classified as a possible placement error.

An experiment was devised that involved utilising fingerprint images captured during the biometric trial conducted at the University of Kent. It was theorised that if these images could be inspected in order to obtain various image parameters and then cross referenced to whether the user passed or failed verification then it would be possible to determine whether a correlation existed between these extracted image parameters and successful verification.

Experimental Setup

In the biometric trial conducted at the University of Kent there were a total of 220 subjects. During the two phases of this trial a large number of fingerprint images were acquired, subjects were allowed to attempt verification no more than three times during the enrolment phase and a further three during the verification phase, which was conducted some time later. The images acquired were in 8bit greyscale and were 121 * 153 pixels in size. All these images were analysed during this stage of the experiment.

In order to extract these image parameter values image processing software was utilised under the Borland C++ Builder 6 environment that enabled an image to be loaded and then analysed in such a manner to extract the values depicted in Table 5.5.

Image Parameter
Mean pixel value
Number of non-zero pixels
Standard deviation of non zero pixels

Table 5.5 Extracted Image Parameters

These three features allow the likelihood for a number of possible of fault conditions to be identified. These three values were chosen as they required simple image analysis techniques which could be applied in a computationally efficient manner whilst still being potentially robust descriptors for the proposed identifiable fault conditions. A description of how each parameter is expected to aid in classification is detailed below.

Mean pixel value

The images acquired by the device are 8bit greyscale. The mean pixel value can therefore be useful in determining the overall intensity of the image which may prove constructive in classifying failures.

Number of non-zero pixels

This parameter can also be used as a gauge to the size of the area covered by the finger on the sensor. A zero value indicates the colour of the pixel is pure black. Therefore the number of non zero pixels in the image gives a broad indication of the darkness of the acquired image.

Standard deviation of non-zero pixels

This parameter is a measure of the distance the remainder of the pixels which are not totally black are likely to lie from the average values for that particular image. In essence this figure can be used to characterise the overall range of pixel values in the sample image. A sample with a low standard deviation represents an image where the pixel values are spread less widely around the mean pixel value, this could be used in order to estimate the distribution of pixel values within the image. This parameter gives a guide to the texture of the acquired image.

None of these individual parameters are expected to be sufficiently descriptive to be employed singularly to determine failure categories. It is through the combination of some or all of these chosen parameters and the selection of suitable threshold values, it is envisaged that these failure conditions could be detected.

The verification results from the IAMBIC biometric gathering phase are illustrated in Table 5.6. Verification session1 was conducted after initial enrolment with the devices. Verification session 2 occurred 4 – 6 weeks later.

	Verification Session 1	Verification Session 2
Total Verification attempts	649	632
False Reject	90	212

Table 5.6 IAMBIC verification results

A graphical breakdown of these results is shown in Figure 5.6.

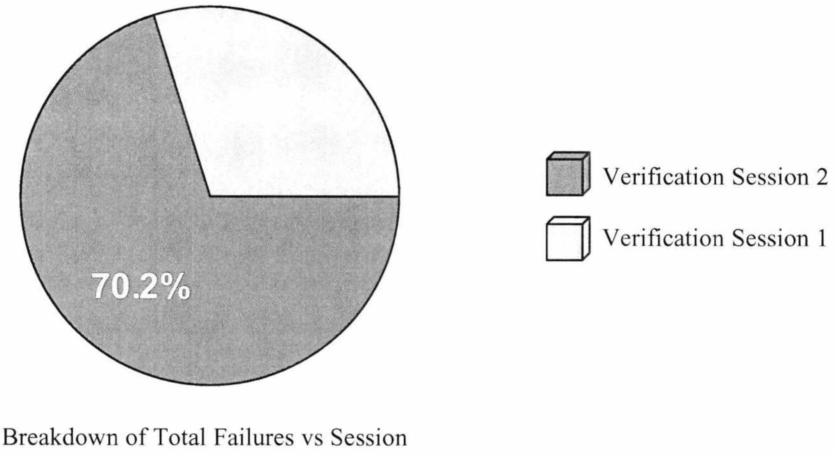


Figure 5.6 Graphical verification results

One of the key observations of the breakdown of results was that:

- During the verification phase the number of false reject conditions observed increased dramatically. One of the reasons for this occurrence could be due to the period of time that elapsed since the user last donated a sample. This period was at least four weeks for the majority of users, it is therefore feasible that familiarity towards the device could have dissipated.

Figure 5.7 depicts the histogram of the verification passes observed during the experiment, the results tend to indicate clustering of values around the mean amongst the test set. This highlights the general tolerance of the matching algorithm present in the Secugen API, the extremes of the graph demonstrate this aspect. These limits represent images which are particularly dark or light but have still been sufficiently feature rich to enable verification.

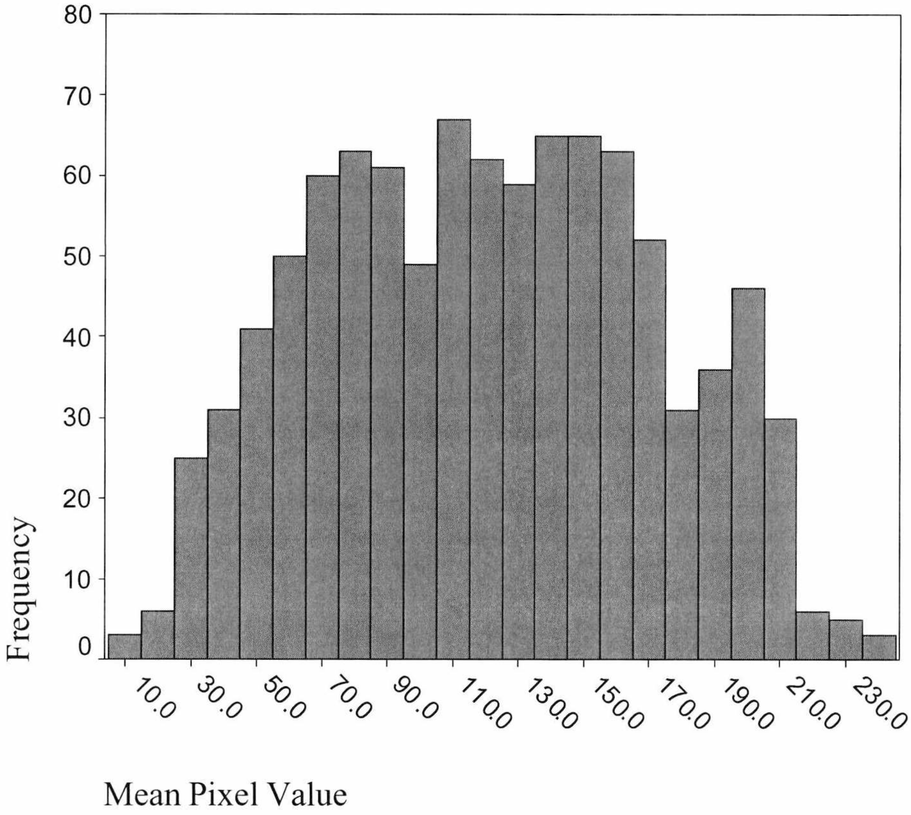


Figure 5.7 Histogram of mean pixel value for successful verification test

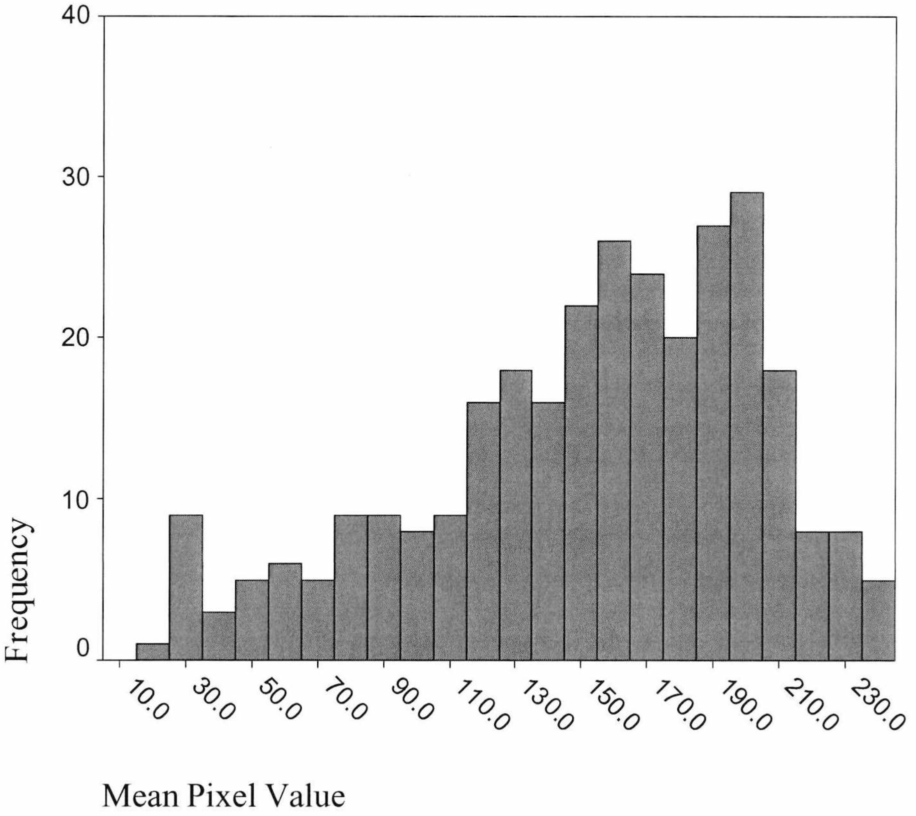


Figure 5.8 Histogram of mean pixel value for unsuccessful verification test group

An observation made from Figure 5.8, also provides some possible indications that could aid with the diagnosis of incorrect operation of the device. The histogram illustrates that in this failed test set the distribution of values has been shifted towards the lighter end of the available spectrum. Images which were at the extreme range of the histogram represented a larger proportion in this test population (i.e. the images were deemed very dark or light). This particular condition represents the class of faint images, which by their very nature do not present a suitably contrasted image in order for the algorithm to extract the necessary features, if present at all. To investigate this condition further, as it is clear that the value of the mean pixel value was not on its own a suitable classifier, the parameter values were plotted against each other as shown in Figures 5.9, 5.10 and 5.11.

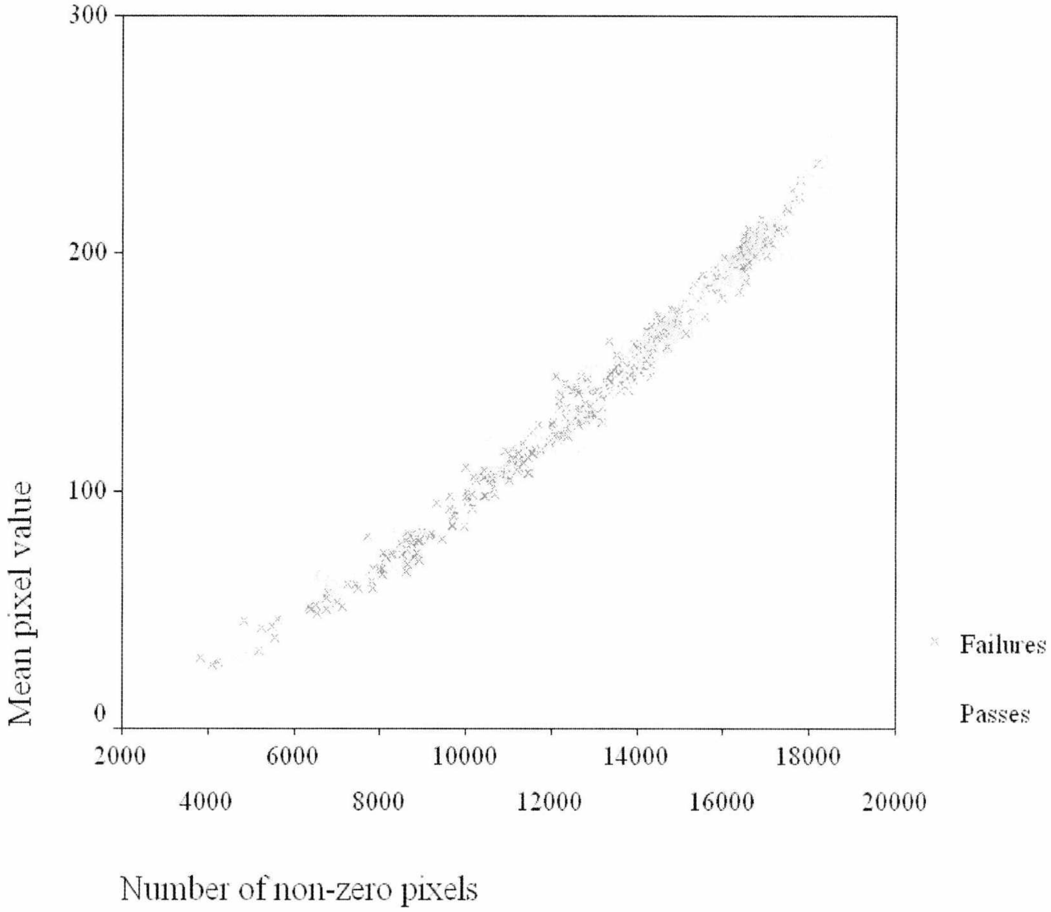


Figure 5.9 Graph of mean pixel value Vs number of non-zero pixels

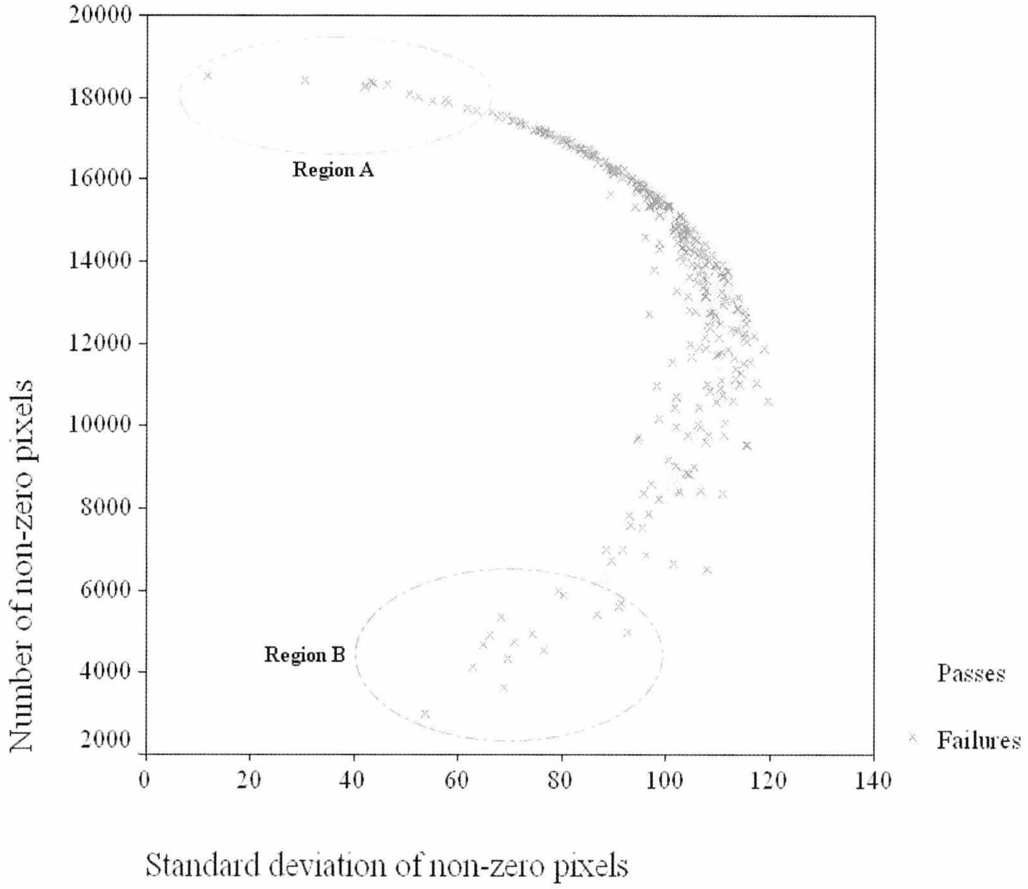


Figure 5.10 Graph of Number of non zero pixels Vs Standard deviation of non-zero pixels

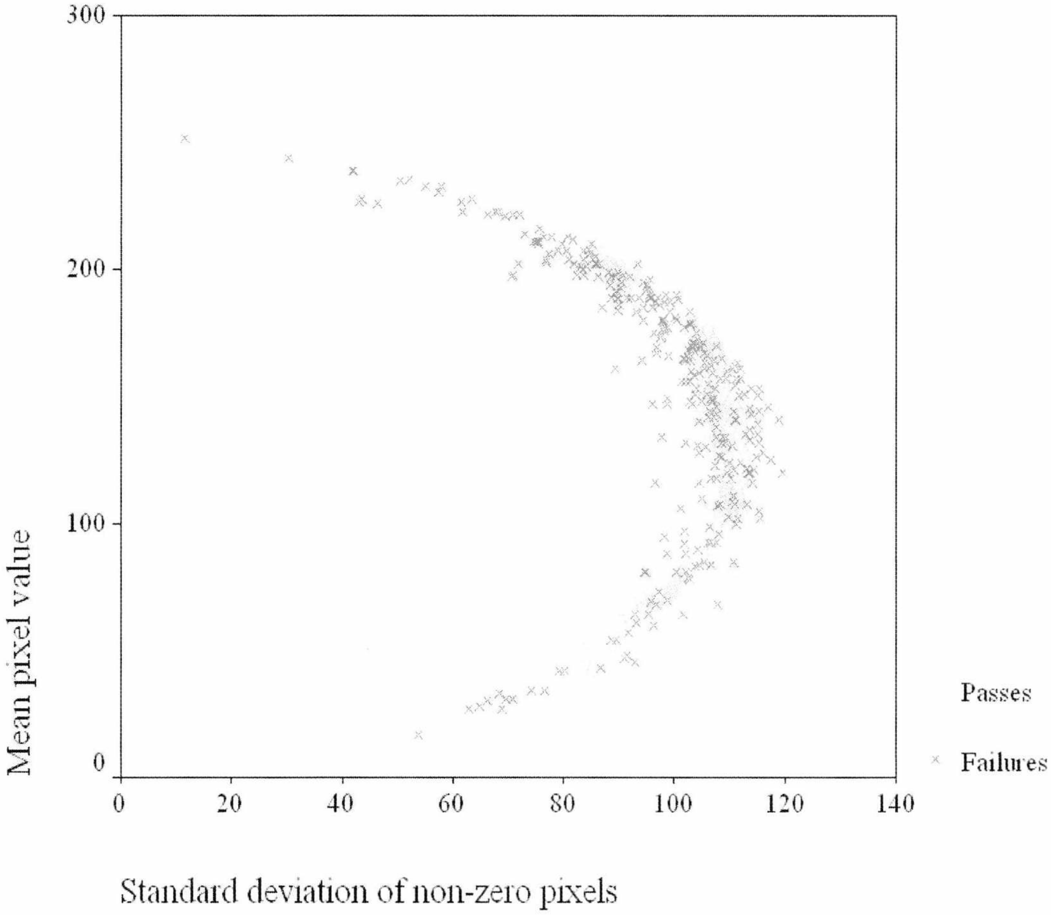


Figure 5.11 Graph of Mean pixel value Vs Standard deviation on non-zero pixels

Figure 5.10 illustrates a condition where a portion of the observed failures occurred in a specific range. This range is once again towards the lighter end of the available intensity range for the image. This can be seen from the top tail of the plotted values indicated by the area marked Region A, on Figure 5.10. From this a number of values can be extracted in order to detect this particular fault.

Parameter	Value
Number of non-zero pixels	> 17000
Standard deviation of non-zero pixels	< 65

Table 5.7 Parameter values for images that are too light

The rationale behind the selection of these particular values is that the range of values accurately encompasses images which can be identified as possibly being too light in order for successful verification to occur.

A feature present in Figure 5.10 can be utilised in order to detect that an image is possibly too dark for verification. The clustering of failures towards the bottom of the curve marked as Region B, illustrates the possibility of this type of fault. Parameter thresholds derived from this observation are presented in Table 5.8.

Parameter	Value
Number of non-zero pixels	< 7000
Standard deviation of non-zero pixels	< 100

Table 5.8 Parameter values for images that are too dark

In order to detect that there is no finger at all presented to the sensor a number of mechanisms can be employed. The first is to determine whether the sample acquisition process occurs without failure. An interaction where an image cannot be acquired is rare and can be attributed primarily due to no finger being present on the sensing area. The second mechanism occurs once an image has been successfully acquired but the image parameters that have been extracted fall into the following bands shown in Table 5.9.

Parameter	Value
Number of non zero pixels	>17000
Standard deviation of non zero pixels	< 20

Table 5.9 Parameter values for no finger present

This band is subtly different to the data figures presented in Table 5.8, encompassing parameters which tend to indicate that no finger is present.

Another procedure can be employed during this analysis phase to further enhance the robustness of the fault analysis routines. Enrolment images can be analysed in order to establish the mean pixel value of these samples, this can be used to determine another condition which was observed in the test population which was that subjects who provide particularly light or dark enrolment images subsequently experienced difficulty successfully verifying with images which were the inverse (i.e. dark enrolment images followed by light verification images). This state can be easily detected, and users can be informed in a suitable manner. It is envisaged that by employing the stringent calibration schedule as detailed in Section 5.2.6 it will minimise this particular occurrence.

The results from this part of the experiment are interesting for a number of reasons. The graphs from the experiment (Figures 5.9, 5.10 and 5.11) illustrate that the parameters for images that passed verification and also the images that failed verification were similar across the variables examined. This would seem to indicate that attempting to classify images that would pass or fail based directly on these parameters would be difficult at best. The analysis of the image does allow a number of basic factors to be ruled out before finger placement becomes the prevailing prominent fault condition. Therefore the user could be alerted to the probable cause of failure being finger placement.

The examination of these results provides evidence to support that the correct positioning of the finger on the sensing area and ensuring that a sufficiently feature rich sample is acquired is of paramount importance. This further emphasises the need either for user training or a method of providing real-time feedback to the user to ensure that verification is successfully achieved with the minimum of effort to the user and associated frustration towards the device.

The identification of incorrect finger placement would be greatly aided by exposure to lower-level functions within the device API, such as the number of minutiae identified within the sample provided and the underlying number of minutia required for successful verification. These functions are not available to the end user and are essentially hidden due to the vendor not permitting an end user access to these methods.

5.3.1.1 Type B

The next condition this component is responsible for identifying is the Device Error (Type B). As previously mentioned this component should attempt to determine if the sensing area of the device is deemed sufficiently clean for optimum image acquisition. The grease deposited from a fingerprint on the sensing area can hinder the general operation of the device and may contribute to verification failures. Once again this type of error could be potentially identified using basic image analysis techniques. An experiment was conducted in order to determine what suitable values were required in order to determine if the sensing area on the fingerprint device was considered dirty.

The experimental approach here was to attempt to obtain a sample from the sensor without employing any form of quality assessment, so that an image would be automatically acquired. It was proposed that any form of deposit on the sensing area might be observable in the acquired image. It was soon discovered that there was an issue with the type of approach employed to acquire the image itself. It was possible to attempt to capture a blank image from the sensor but even with the sensing area heavily contaminated with deposit there was no discernable change in the parameters of the image acquired. Without a suitable reflective surface near the sensor, the image acquired was always pure white.

This experimental result forced a design change in the manner of which this particular condition could be classified. Although other image acquisition faults such as lower level hardware errors could be identified from the device itself, the cleanliness of the sensing area could not be determined in a suitably efficient automated manner by employing the proposed techniques.

It was proposed to shift this function from the software itself and to attempt to make the act of ensuring that the optimum acquisition surface is present to be the responsibility of the user during the interaction protocol. The user would therefore be reminded before the beginning of a session that the surface should be examined and cleaned if necessary. Naturally this dialog could be disabled by the user once it is felt this reminder is no longer required. Also the process by which the user is reminded periodically to ensure the sensing area is kept clean reinforces this aspect of good usage which may assist the further minimisation of verification failures.

5.3.1.2 Type C

The template aging condition (Type C) that this component needs to detect can be handled in a simple manner. Values are available for how each particular biometric modality is affected by the process of human aging (Section 2.10.2). This value for the fingerprint modality can be used as a measure for the time that a template file is considered valid. Although this error is difficult to accurately identify and diagnose, it is still important to alert the user to a developing potential template aging situation. Updating user templates is another technique that the user has in order to possibly reduce verification errors and therefore must be made aware of this procedure at the appropriate times. In order to achieve this functionality all that is required is to record a timestamp when the user initially enrols, and then periodically check that the period of time elapsed since enrolment has not exceeded the pre-determined value. Suitable values for these time intervals are proposed in Section 6.6.5.

5.3.1.3 Type D

The final condition this component is tasked with involves the detection of inappropriate image acquisition parameters (Type D). This can occur if the system has been incrementing these values due to high quality verification samples being acquired from the user over a period of time. It is entirely feasible that the user may have a verification session where overall performance is poor when compared to previous attempts but still sufficient for verification, however, the *quality* setting may need to be reduced in order to facilitate this. This parameter is rolled back gracefully using the same increments that were used to increase the value if required. The value of *quality* is not allowed to decrease below the level employed for enrolment.

The calibration monitor is a simple component to implement, once again this particular module employs a time based value to ensure that the calibration data for the specific user is valid. Calibration data is time stamped and stored within the user's XML file, this value is checked every time the user initiates a verification session. If the threshold value is exceeded, e.g. four weeks, then the user is prompted to recalibrate the device.

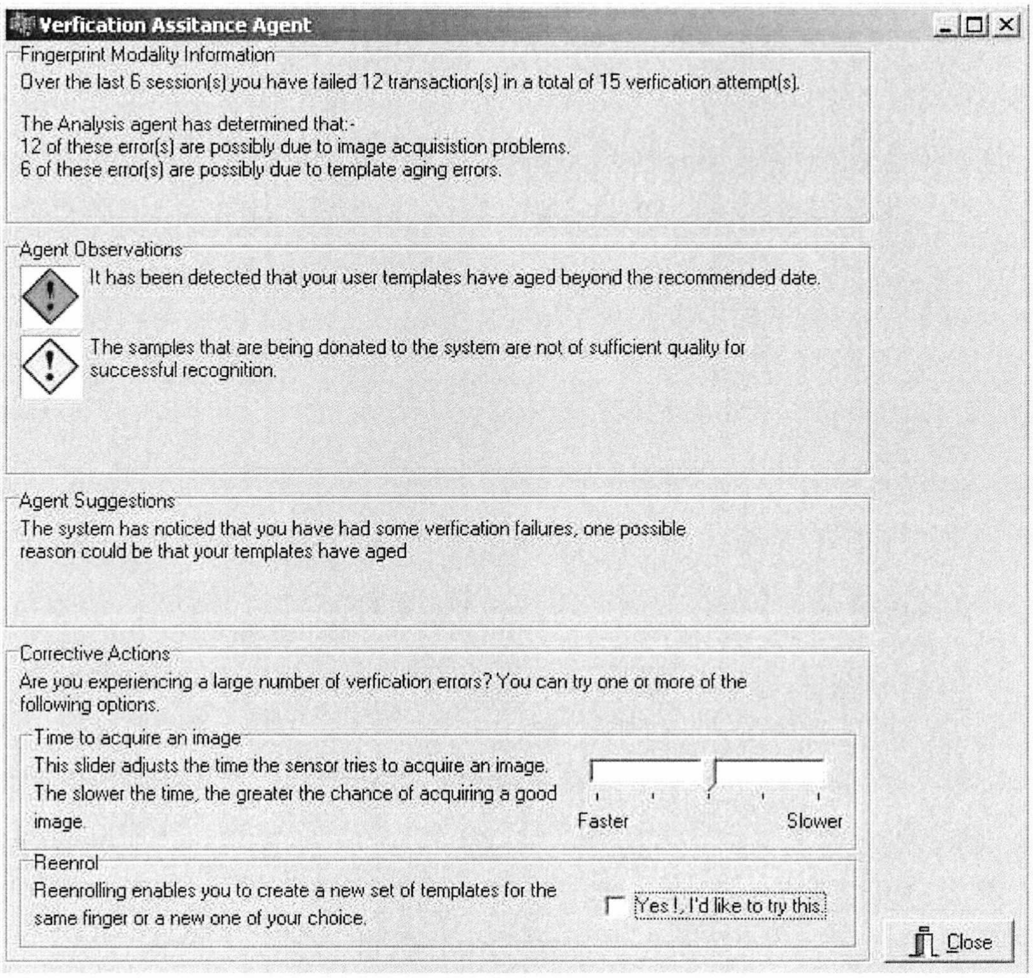
5.3.2 Template Generation Monitor

In order to implement the desired behaviour for this component two mechanisms must be developed. The first is a process by which the samples acquired from the subject during enrolment can be enrolled at the highest security level possible. This can be achieved by calling the enrolment function in a loop, incrementing the security level during each iteration. The loop exits when the maximum security level for the donated sample is reached (Figure 5.2).

The second is the operation of this module during re-enrolment, which is slightly different. During the period of time elapsed since the initial or last enrolment the analysis engine has been examining the performance of the user. At the time which the user history data file is analysed, the system will indicate that a possibility exists to increase one of the image acquisition parameters, *security* or *quality*. This component will be invoked and passed the values from the XML file of the user. If the necessary conditions exist for increasing the *security* level or the *quality* setting then this module attempts to acquire images at these new settings.

5.3.3 System Behaviour

The behaviour of the interface towards the user is controlled by this component. By determining the user's current system behaviour band and also the number of specific fault types, the system will offer appropriate feedback. The manner in which this feedback is presented is illustrated in Figure 5.12, which is shown after a sufficient number of biometric transactions have occurred. This form demonstrates the interface displayed to the user, this dialog indicates any identified fault types and also what the system deems to be the 'severity' of the identified fault. The severity rating indicates what courses of action the user could pursue in order to minimise the reoccurrence of this particular fault. This is illustrated in the 'Agent Observations' panel in Figure 5.12. If multiple fault types have been identified then these are presented in a priority order to combat the probable fault with the minimum of user interaction.



Verfication Assitance Agent

Fingerprint Modality Information

Over the last 6 session(s) you have failed 12 transaction(s) in a total of 15 verfication attempt(s).

The Analysis agent has determined that:-
12 of these error(s) are possibly due to image acquisistion problems.
6 of these error(s) are possibly due to template aging errors.

Agent Observations

It has been detected that your user templates have aged beyond the recommended date.

The samples that are being donated to the system are not of sufficient quality for successful recognition.

Agent Suggestions

The system has noticed that you have had some verfication failures, one possible reason could be that your templates have aged

Corrective Actions

Are you experiencing a large number of verfication errors? You can try one or more of the following options.

Time to acquire an image

This slider adjusts the time the sensor tries to acquire an image. The slower the time, the greater the chance of acquiring a good image.

Reenrol

Reenrolling enables you to create a new set of templates for the same finger or a new one of your choice.

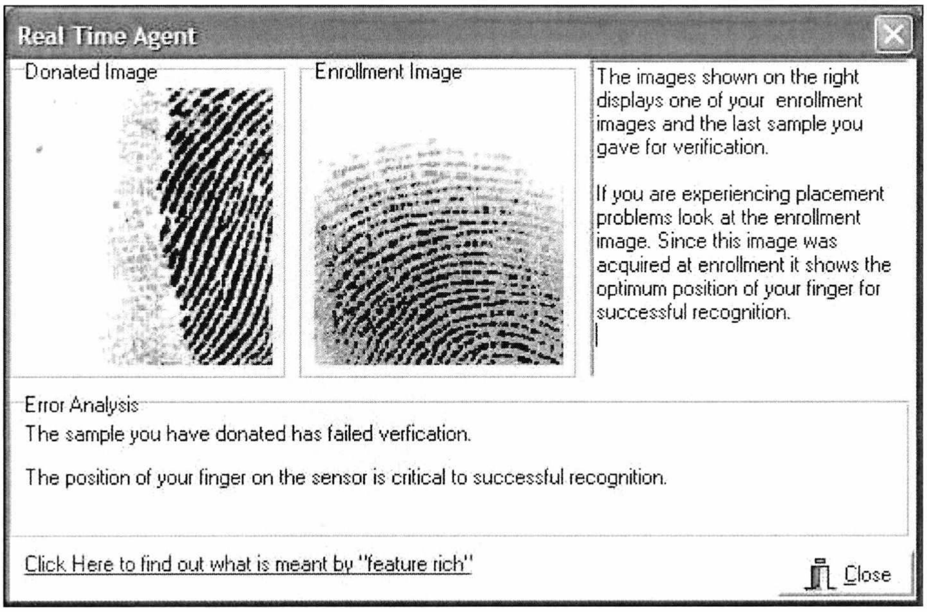
□ Yes !, I'd like to try this.

Figure 5.12 Periodic user feedback dialog

Help strings are provided for each fault type and the order in which these are presented is randomised in order to provide differing methods of delivering what is essentially the same manner of assistance for the fault type to the user. Examples of these help strings are given in Appendix B.

Whilst these help prompts are informative the user does have some more proactive actions available to combat poor verification results. These actions include reenrolment and also the ability to adjust the period of time that the device attempts to acquire an image from the sensor. Reenrolment is initiated at the next session the user attempts, whilst image acquisition time is altered immediately.

While this feature gives an overall guide to user performance during the operational lifetime of the system, another mechanism is present which gives advice specifically at the point of failure. This 'real-time' dialog uses the identified fault types and alerts the user to the probable cause of failure. This feature is shown in Figure 5.13.



The images shown on the right displays one of your enrollment images and the last sample you gave for verification.

If you are experiencing placement problems look at the enrollment image. Since this image was acquired at enrollment it shows the optimum position of your finger for successful recognition.

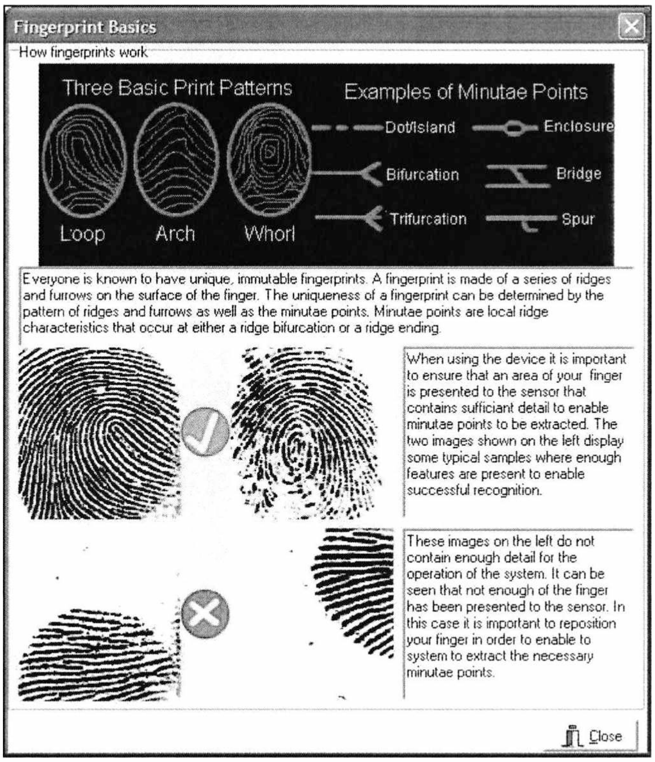
The sample you have donated has failed verfication.

The position of your finger on the sensor is critical to successful recognition.

Click Here to find out what is meant by "feature rich"

Figure 5.13 Real time dialog

Another issue that can help in the training of the user is a basic understanding of the underlying technological process that is occurring when using such a biometric device. In the case of this modality, the user can be advised about the importance of how presenting a feature rich area of the finger is critical to the operation of the system. This can be achieved through the use of in application help such as illustrated in Figure 5.14 or in a more active fashion by demonstrating these facts to the user through a dialog containing sample images and short video files when possible finger placement issues have been detected.



Everyone is known to have unique, immutable fingerprints. A fingerprint is made of a series of ridges and furrows on the surface of the finger. The uniqueness of a fingerprint can be determined by the pattern of ridges and furrows as well as the minutae points. Minutae points are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending.

When using the device it is important to ensure that an area of your finger is presented to the sensor that contains sufficiant detail to enable minutae points to be extracted. The two images shown on the left display some typical samples where enough features are present to enable successful recognition.

These images on the left do not contain enough detail for the operation of the system. It can be seen that not enough of the finger has been presented to the sensor. In this case it is important to reposition your finger in order to enable to system to extract the necessary minutae points.

Figure 5.14 Modality information dialog

5.4 Conclusions

In this chapter the notion of the adaptive interface has been explored. The interface itself provides a number of probable fault detection mechanisms which are hoped to aid the user in successful donation of samples, coupled with the behavioural ability of the system to target aid specifically to those users who are experiencing difficulty thus is seen as a key benefit over non-adaptive systems. In Chapter 6 the adaptive interface will be assessed and the proposed approach presented here will be tested.

Chapter 6

Experimental Setup and Testing

This chapter details the experimental procedures used to assess the both the performance of the IAMBIC system and also the adaptive interface layer. The experimental results obtained from these procedures are also presented.

The first half of the chapter involves detailing the test protocol utilised in order to determine the performance of the IAMBIC system. The results are presented and analysis is offered to extract a number of system performance metrics. During this analysis phase some suggestions are also offered for enhanced system robustness based on the experimental results obtained.

The second half of this chapter introduces the test protocol employed in order to assess the functionality of the adaptive interface. The performance metrics of both the non-adaptive and the adaptive system are compared, along with a number of other results which illustrate the benefits of applying interface agent methodology in biometric applications.

6.1 IAMBIC testing overview

The underlying motivation behind the final system testing was to determine how the system reacted to simulated 'real-life' operation. This type of testing is known as scenario testing [Mansfield and Wayman, 2002], where a complete system is evaluated rather than individual biometric devices or algorithms. In order to achieve this, a robust test protocol must be devised to adequately test the various parameters under which the system was being assessed.

To mimic the operational use of such a system, a number of users will be assigned a role in the system. The total number of test subjects utilised in this trial was ten. The scope of this scenario testing was not as broad as the initial biometric gathering phase in which over two hundred subjects were utilised. Therefore, it was proposed that these test subjects could be recruited from within the Electronics Department at the University of Kent, enabling this testing phase to be completed without any initial delay as test subjects were recruited.

These ten test subjects were distributed amongst the three user roles in the scenario.

- Doctors
- Patients
- Analysts

The first two roles, doctors and patients, contain four users each. The analyst role contained two users. This role breakdown was performed in order to maximise the groups that contained roles which had a high level of system interaction. For each role in the system there are a number of permissions associated with that role, these are illustrated in Table 6.1. The analyst provides an administrative role in the proposed system and as such the permission for this role only allows access to the filenames contained on the information server.

Permission	Patient	Doctor	Analyst
Can upload cardio files	√	√	X
Can view patient files	√	√	X
Can view multiple patient files	X	√	X
Can view all patient files	X	X	X
Can retrieve all filenames on server	X	X	√

Table 6.1 IAMBIC role permissions

The two permission settings (uploading of a file, viewing a file) each have an associated confidence score required for the execution of said operation. These permissions were part of the original requirements capture for the IAMBIC project, stating that the system had to provide multiple levels of authorisation within the system, both in relation to different categories of user and different levels of confidence necessary to access information.

It was also part of this testing regime to determine whether the confidence values initially associated with system access were realistic for the given group of users. Table 6.2 illustrates the confidence values that are required by the system for each given operation.

User Type	Action	Confidence Required (%)
Doctor	Upload a file	60
	Retrieve a file	80
Patient	Upload a file	50
	Retrieve a file	60
Analyst	Retrieve filenames	70

Table 6.2 IAMBIC role confidence thresholds

The values chosen for these various levels were arbitrary and represented values which were deemed as achievable based on previous biometric trial data. The confidence score is based upon a weighted summation of the individual scores produced for each biometric modality. The technique employed to produce this score is detailed in Section 6.1.1.

6.1.1 Fusion algorithm

In order to assess the system effectiveness an simple fusion scheme was employed that combined equally the scores of each modality, so in the case of a user enrolling in all three modalities each would contribute 33.3% towards the overall confidence score, or for a user only enrolling in two modalities this figure would now be 50% for each of the enrolled modalities. Score normalisation was performed on each modality using the scaling factors shown in Table 6.3. The scaling is required to ensure that each particular modality contributes an equal weight towards the values required for the role confidence thresholds (Table 6.2).

	Scaling Factor		
Modality	**3 Modalities**	**2 Modalities**	**1 Modality**
Finger	3.70	5.55	11.1
Face	3.33	5.00	10
Voice	0.074	0.11	0.22

Table 6.3 Modality scaling factors

These scaling factors were obtained by dividing the output score for each modality by the maximum score through the verification process. These maximum values are illustrated in Table 6.4. The values for the finger and voice modality were extracted from the documentation from the within the relevant API. Whilst the value for the voice modality was chosen from values recorded during the IAMBIC biometric gathering phase. If a subject fails the verification process for a particular modality then the corresponding contribution for that modality is zero.

Modality	Maximum possible verification score	Minimum possible verification score	Verification threshold
Finger	9	0	5
Face	10	0	8.8
Voice	450	- 450	0

Table 6.4 Maximum modality verification scores

6.2 Test Protocol

In this section the protocol used in the scenario testing of the IAMBIC system will be specified. The protocol covers specific enrolment and verification arrangements along with the proposed scheme that will be employed in order to test the resilience of the system to 'live' impostor attacks.

6.2.1 Enrolment

Each volunteer will be asked to enrol in the system using the enrolment wizard component of the IAMBIC project. This enrolment procedure will be supervised to ensure the acquisition of good quality templates. The user will be given three attempts to enrol on each specific modality. If no templates can be acquired after these three attempts then a failure to enrol condition is flagged for this user on the specific modality. Template quality is monitored by the application itself, strict limits are placed on these samples to ensure that samples of sufficient quality are acquired in the enrolment procedure. The order in which modalities are enrolled is as follows: firstly the fingerprint, secondly the voice and finally the face modality.

6.1.2 Verification

Volunteers will make a number of verification attempts. During verification the user is allowed to attempt verification on each modality a maximum of three times. If the user fails three attempts no further verification occurs on that specific modality, verification then moves on to the next modality. Failures of this type can be examined at a later time by examining the contents of the transaction file.

Each user group will be asked to attempt specific actions pertaining to that user group once a day. E.g. for a patient the following actions would be requested.

1. The user logs in and donates a set of samples. This allows the system to generate a confidence score for this current session.
2. The user is then asked to attempt to upload a cardio file to the server database. This action encapsulates one of the two authorisation levels.

3. If the user's generated confidence level is not sufficiently high to perform this operation, the user is prompted to attempt verification again.
4. The user is asked to attempt to view one of their own uploaded files on the server, this action requires a higher confidence score then produced in step 2 above.
5. If the user's confidence is not sufficiently high to perform this operation, the user is prompted to attempt verification again.
6. Once this phase is over the user is asked to log out of the system and is prompted to attempt to act as an impostor to each of the other user types. The user logs in as another genuine user and donates samples in order to actively attack the system.

For the doctor role the set of actions are similar to that of a patient and the interaction protocol would be the same as the case described above. The analyst role only has one action available, as shown in Table 6.1. In this case the protocol described below is employed.

1. The analyst logs in and donates a set of samples. This allows the system to generate a confidence score for this current session.
2. The user is then asked to attempt to view all the filenames available on the server. This is the only action available for the analyst role.
3. If the user's confidence is not sufficiently high to perform this operation, the user is prompted to attempt verification again.
4. Once this phase is over the user is asked to log out of the system and is prompted to attempt to act as an impostor to each of the other user types. The user logs in as another genuine user and donates samples in order to actively attack the system.

6.2.3 Impostor Testing

Impostor testing will be conducted in a live fashion during the verification phase. Test subjects will be asked to act as an impostor to previously enrolled subjects. In this manner of impostor attacks, every user attempts to attack the identity of each user at least once during the operational execution of the trial.

In summary, each user will attempt to genuinely use the system three times, and will attempt impostor attacks against three different users once per session. During the impostor testing phase the attacker will be subject to the same conditions as the 'normal' subjects. That is the maximum number of verification attempts given per modality is three for each attack.

Each session is conducted on a separate day. Impostor testing is an important aspect of this trial and it is expected to demonstrate the benefit of employing multiple biometrics as a possible safeguard against possible impostor penetration.

6.2.4 Equipment and environmental setup

The proposed testing environment was the biometric suite located in the digital research and vision group within the Electronics Department at the University of Kent. The biometric suite contains a number of PC's equipped with biometric equipment which can be employed for this kind of biometric trial. The IAMBIC client software was deployed onto one of these machines. The hardware utilised for biometric capture is illustrated in Table 6.5.

Biometric Modality	Hardware
Finger	Secugen FDU02
Voice	Creative microphone SD-50
Face	Logitech Webcam Quickcam Express V-UM14

Table 6.5 Hardware setup for IAMBIC trial

The server software was installed on a laptop computer running Window Server 2000 and SQL 8.0. Also the SACI environment was installed on this laptop in order for this machine to create an agent society for the rest of the agents to participate in. This machine was also responsible for the management of the 'yellow pages' and 'white pages' services performed under the SACI API.

Both the client and the server machine were connected to the Local Area Network (LAN) within the department in order to provide a simple and convenient communication channel. Although this test was conducted locally over the hard wired network there are no foreseeable deployment issues with this system working over a dial up internet connection or a similar remote link

6.3 Results

In order to assess the performance of the IAMBIC system a number of factors will be considered in the examination of the trial results. These include:

- Failure to Enrol (FTE) Rates
- Failure to Acquire (FTA) Rates
- False Accept Rate (FAR)
- False Reject Rate (FRR)

Also a questionnaire was produced to be completed by trial subjects. This questionnaire focused on issues surrounding usability of the IAMBIC system and covered the aspect of employing biometrics as a method for identity authentication. The results from these questions also provided a useful insight into how biometric technology is perceived by general users of such a system. This questionnaire can be found in Appendix B.

6.3.1 IAMBIC Test Results

The results from this experiment will be divided into two sections. The first will examine the data obtained from 1:1 client matching portion of the experiment. The second will investigate the results obtained from the impostor testing segment of the experiment.

6.3.1.1 Enrolment Details

Ten users were employed for the purposes of this trial. Of these users only one user was not able to enrol in all three modalities, this user could not enrol in the finger modality.

6.3.1.2 Client to Client Matches

Users were allowed a maximum of three attempts per modality, if the user was successfully verified then no further attempts were allowed and the user moved onto the next modality. If a user failed to meet the required confidence level for a specific operation then the biometric verification wizard was launched and the user was prompted to donate another set of samples in order to increase the user confidence. The user was allowed three attempts in order to increase this confidence level. The following section details the breakdown of results for each modality.

6.3.1.3 Finger

Total Number of matches made	59
Total number of passes	54
Total number of fails	5
False Reject Rate %	**8.48**

Table 6.6 Finger modality FRR

The false reject associated with this modality was confined to one user who expressly experienced these errors. This particular subject managed to successfully enrol in the modality, however, during subsequent verification the samples provided were extremely poor quality. In particular this was identified as finger placement and lack of pressure on the sensing area contributing to these failures. The remaining portion of the user set (eight) did not experience any false reject with this modality.

6.3.1.4 Voice

Total Number of matches made	64
Total Number of Passes	57
Total Number of Fails(including acquisition errors)	7
Total Number of Acquisition Errors	7
False Reject Rate (including errors) %	**10.93**
False Reject Rate (excluding errors) %	**0**

Table 6.7 Voice modality FRR

This modality suffered from a number of acquisition errors, this occurred due to software problems and also possibly environmental noise pollution causing spurious results. A number of issues had arisen during the integration of the voice software into the overall IAMBIC framework. One of these issues was that the overall stability of the software was reduced leading to instances where a sample would not be captured by the software and result in an acquisition failure. Although no user failed the voice modality over the allowed three attempts, a significant number of these acquisition failures did occur.

6.3.1.5 Face

Total Number of matches made	**67**
Total Number of Passes	**67**
Total Number of Fails	**0**
False Reject Rate %	**0**

Table 6.8 Face modality FRR

As can be seen from the results shown in Table 6.8, there was no false reject of any users using the face modality over the allowed three attempts.

6.3.1.6 System False Reject Rate Calculations

Through the use of multiple biometric modalities individual instances of false reject should not be as frustrating as those in a uni-modal system. This is due to the fact that a single instance of false reject may not deny the user access to the system as long as the other samples the user donates are of sufficiently high quality to enable the confidence score to meet or exceed the value required for access.

Moreover, it is the choice of the fusion scheme and the associated confidence score required for specific actions that now determines whether the user has been rejected by the system for the particular required action. So in order to assess the false reject rate exhibited by the system it is proposed to calculate the false reject rate based on whether a subject can gain access at a given confidence level within the allowed attempts. A false reject rate can thus be calculated for each of the particular confidence scores required for system access. These calculations are based on the test population as a whole ignoring the role a subject may play. This was performed in order to more accurately assess the performance of the test subjects as a whole. Table 6.9 illustrates these figures. These figures are examined and their impact on the usability of system is detailed in Section 6.3.1.7.

Confidence Value	50	60	70	80
FRR (%)	13.4	41.7	68.6	92.5

Table 6.9 FRR Vs confidence value

6.3.1.7 Confidence Value Generation

The confidence value generated by the software for transmission to the server is based upon the mechanics detailed in Section 6.1.1. The confidence values for all roles recorded during the experiment are illustrated in Figure 6.1.

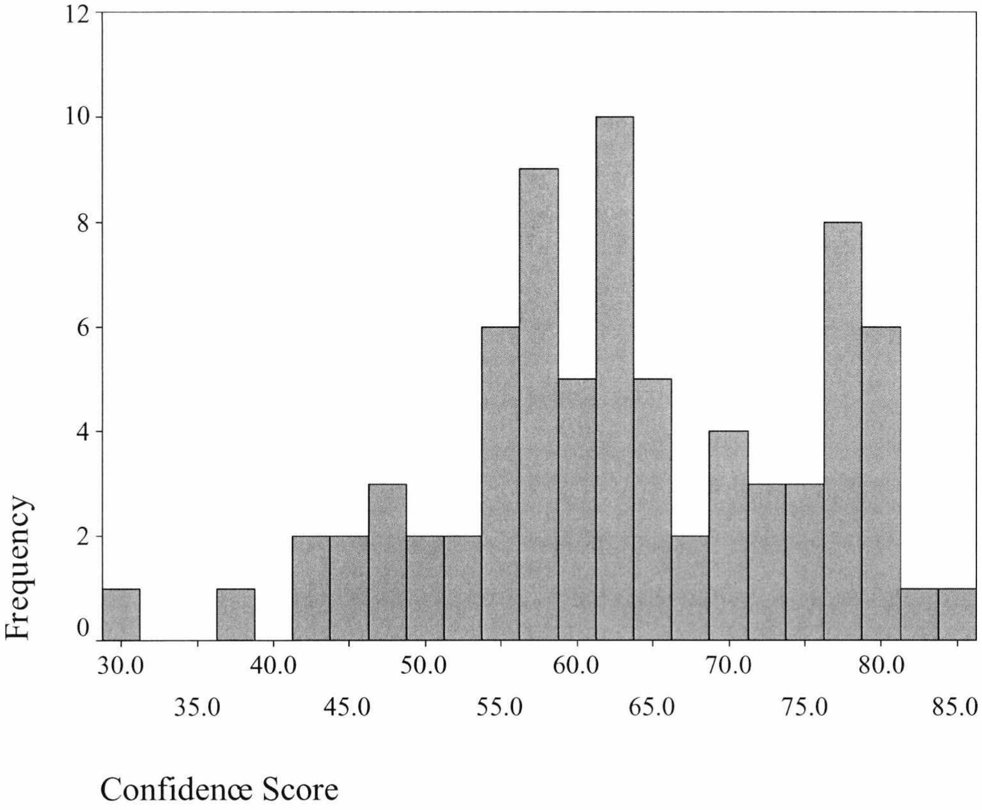


Figure 6.1 Confidence score distribution

These results are interesting as they highlight an area in which overall system performance can be improved. The distribution indicates that the mean value observed in this particular test group is around the value of 63. Also the frequency of users who can achieve a confidence score of 80 required for the highest level is system access is small when compared with the remainder of the distribution.

This is an important aspect, due to the fact that both the doctor and the patient roles require a confidence score of at least 80 in order to retrieve a file from the server. Obviously these results indicate a potential usage problem where a number of subjects are experiencing difficulty reaching the proposed figure for system interaction. This led to the reduction of the values that were required for system access and are detailed in more detail in Section 6.5.

6.3.1.8 Impostor testing results

Each user acted as an impostor to every other user during the trial. Each user was given three attempts at each modality the selected user enrolled with. Verification thresholds were kept the same as in the client to client portion of the trial. The results from these matches are illustrated in Table 6.10.

Modality	Total Matches	Number of False Accepts	False Accept (%)
Finger	243	0	0
Voice	243	3	0.82
Face	243	3	0.82

Table 6.10 FAR rates for IAMBIC modalities

In the total number of impostor attempts there were five distinct cases where impostors successfully verified using the thresholds which had been employed during the client to client matching portion of the experiment. Since the verification scores for each modality are used to generate an associated confidence score which ultimately determines the level of access the impostor now gains, it is important to determine whether these impostors have penetrated the system. The mechanics of the confidence score calculation has already been exposed in Section 6.1.1. Table 6.11 depicts the verification scores obtained by these impostors and also the modalities that were susceptible to this manner of impostor attack.

Impostor Case	Face Score	Voice Score
1	N/A	31
2	8.92	4
3	N/A	103
4	8.84	N/A
5	8.86	N/A

Table 6.11 Impostor scores

Figure 6.2 graphically illustrates the confidence values recorded from these impostor attacks.

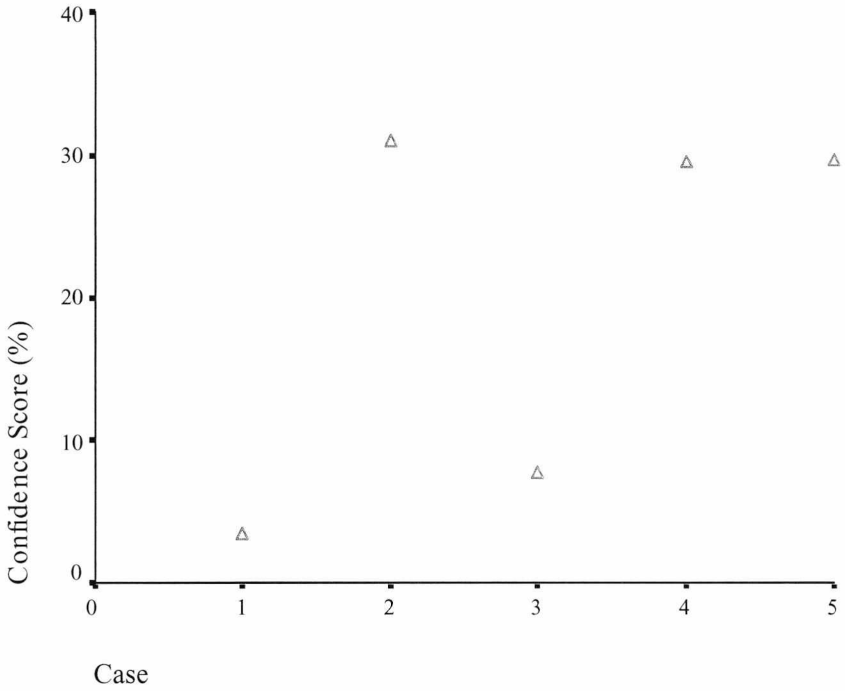


Figure 6.2 Impostor confidence scores

The results indicate that although there have been instances of false accept at the verification thresholds used in the trial, the overall confidence score for the verification attempt is still well below that required for system access at any level (see Table 6.2). Another interesting feature from these results is the low numbers of false accept conditions exhibited by the finger modality. Finger prints are well known for their distinctiveness and due to the small number of subjects who participated in this trial it would be extremely disturbing if any false accept occurred on this modality during the duration of the trial. From these results it can be observed that if a uni-modal system had been employed, utilising either the face or voice modality on its own would have resulted in system penetration. By employing multiple biometrics and also a rudimentary fusion scheme, the IAMBIC system has demonstrated resistance to active impostor attacks.

Confidence Value	50	60	70	80
FRR (%)	13.4	41.7	68.6	92.5
FAR (%)	0.0	0.0	0.0	0.0

Table 6.12 IAMBIC system performance

6.4 IAMBIC biometric gathering phase results

During the project lifetime a large scale trial was conducted into the performance of the three biometric modalities ultimately utilised in the IAMBIC project. Details of the protocol employed and also the demographic of the trial population can be found in [Fairhurst et al, 2002]. The overall results from this phase of the project are presented in Table 6.13.

Modality	Failure to enrol (%)	False accept rate (%)	False reject rate (%)
Fingerprint	14.4	0.0	15.6
Voice	27.1	0.9	3.4
Face	12.5	2.0	36.0

Table 6.13 IAMBIC biometric gathering phase results

6.5 IAMBIC Conclusions

The overall results from this trial phase were encouraging. Results from the IAMBIC system scenario testing illustrated that through the use of multiple biometrics the system was capable of providing a means of authentication that provided multiple levels of system access. The benefit of employing multiple modalities was to provide added flexibility at the verification stage.

By employing a combination of biometric modalities and a rudimentary fusion scheme subjects are provided with a suitably robust biometric authentication system. Although not illustrated in Table 6.12 if the confidence value was set to 30 for general system access, then the observed values in the test population indicate that the FRR of the system would be as low as 1.3 % and FAR 0 %. Employing a fusion scheme allows added flexibility during the verification phase, as an instance of false reject on one particular modality does not preclude the subject from system access, as long as the subject can provide suitably feature rich samples to the other modalities. This even extends to situations where a user may not want to donate a sample for a particular modality for whatever reason, the system can still provide a level of system access based on the remaining modalities.

One further conclusion that can be drawn from these results is the performance of the fingerprint modality towards false acceptance was the best of all three modalities under test. This indicates that for this test set it could be used on its own to prevent instances of false acceptance, without the need for the assistance of other modalities to provide additionally robust impostor resistance.

These test results did highlight an area where the modification of the confidence levels required for system access was required. Section 6.3.1.7 illustrates the confidence values obtained for the subjects in this trial, this demonstrated that the value chosen for the highest level of access (uploading a file) was too high for majority of subjects in this trial to achieve. In order to rectify this issue it was proposed to decrease the overall levels required for system access to values which would be more readily achievable. The final levels of confidence required for system interaction are shown in Appendix B.

6.6 Adaptive Interface Testing Overview

The purpose of this evaluation is to attempt to demonstrate that employing an adaptive interface on a single modality can lead to an increase in user performance. This should equate to a quantifiable increase in recognition rates over a non-adaptive system. The testing procedure must be well defined in order to determine the effects of the adaptive system over the possible benefits of user habituation towards the modality, which may also be exhibited by the test corpus. The tests will be designed not only to determine the potential benefits of an adaptive system but also to test each of the functional components present in the adaptive system.

This testing procedure is mainly concerned with 1:1 matching of the proposed trial population. Currently there is no provision to test the system against impostor attacks. Work in the area of impostor penetration with this specific modality has been completed during the biometric trial phase for offline impostor testing, and also online impostor testing conducted as part of the final IAMBIC scenario testing.

It is also important to state that this experiment was largely unsupervised. The subjects were presented with a document detailing what is required from them for the duration of the experiment. This document is available in Appendix B. This aspect of demonstrating that a biometric system can be used in an unsupervised manner is an important feature, as well as showing that through the use of an adaptive system, user habituation can be accelerated thus leading to a consistent level of user performance and in turn an increase in user appreciation of such a biometric system.

6.6.1 Protocol Overview

It was proposed that twenty subjects will participate in this trial. All efforts were made to make sure that the gender split in this group is equal. The subject population also contained a range of ethnic diversity. The test subjects were students and staff from within the Department of Electronics at the University of Kent.

In this trial only one modality will be employed, this is the fingerprint modality for which the adaptive interface layer has been primarily designed. There are two main phases required in order to use the system. The enrolment phase consists of the user donating two matching samples of the same finger, these samples are then passed to a registration method and the samples are then registered at the highest security level that is possible.

The second and main phase consists of the users attempting verification against the previously stored templates. During this phase a score will be produced for each biometric transaction, this score represents the *utility* of the transaction, as defined in Section 6.6.5.

It is also proposed to test the systems facility to reenrol users using a method of template aging. This method does not affect any template parameters but sets a low value for the time period associated with the validity of a template. This value is absolute and is measured from the date and time the user initially provided a set of samples which were used to construct a template. This is performed in order to determine if the system is able to make a decision based on previous user history to attempt as to increase the *security* level at which the user is enrolled with. The value chosen for template validity for the experiment is detailed in Section 6.6.5.

6.6.2 Protocol Definition

This section describes the actual experimental procedure that was employed to test the adaptive and non-adaptive systems.

The subjects will be split into two groups of ten subjects each. It is proposed that the first group of subjects will be started on the non-adaptive system and the second group will be exposed to the adaptive system. As mentioned above, the first phase common to both user groups, is the enrolment phase.

6.6.3 Enrolment Phase

During the enrolment phase for the adaptive system the system will be fully operational. That is the verification monitor will be examining the supplied enrolment images for possible faults and feedback will be presented to the user at this stage due to the critical nature of this operation to the future success of the verification phase. If the user's enrolled security level is deemed to be low then the application will ask the user if the enrolment procedure can be reattempted in order acquire templates of a higher quality.

The non-adaptive group will be presented with a 'dumb' application. This application will not provide any extended feedback on the donation of enrolment samples, or provide a mechanism to ensure that the user templates are of a specified quality before the user can proceed.

6.6.4 Verification Phase

This phase will comprise the majority of this experiment. Users will be asked to attempt verification during each day for the duration of five days. Subjects are allowed three attempts to succeed. If the subject successfully verifies then no further attempts are allowed. Subjects of both systems are bound by the same test protocol.

During the trial period the adaptive system will be monitoring user performance and providing levels of feedback based on the performance of the user. Whilst this process is occurring the adaptive system will also be monitoring the *quality* and *security* level of the samples donated by the user. If the user is performing particularly well the system may attempt to start to adapt to the user by increasing the *quality* setting initially and then if applicable the security level at the time of reenrolment. This process is driven by the *utility* score for the particular user. This process of template management is transparent to the user.

Another mechanism is also at work for the adaptive system, this is the calibration monitor (see Section 5.2.6). The period of time calibration values are valid for are based on an absolute period of time. If this period of time has expired since the user has calibrated the device they are prompted to do so. Users of the adaptive system will also be prompted to recalibrate the device if the system determines this may be the cause of any failure to verify conditions.

The non-adaptive user group will not be provided with any form of feedback if the donated sample fails verification. All that the user will be informed is that they have failed to verify. Neither will they be subject to any of the advanced template management or device calibration techniques the adaptive system offers.

At the end of the trial, the log files of all the subjects will be examined. This will facilitate the evaluation of the reenrolment procedures. It is expected that by this time the adaptive system may have a number of users that are performing well enough to be able to adjust either the security level or quality setting. To determine whether these conditions have occurred, the subject data logs can be examined to investigate any proposed changes that the adaptive interface has suggested to either parameter.

The results obtained at this stage are important, it is expected that by this time users will have conducted many transactions with the biometric device and may be exhibiting signs of habituation. This habituation is expected to manifest itself in sustained security levels at or near the subjects enrolled security level. Habituation in the adaptive system is expected to be noticeable by the system changing the acquisition parameters for the relevant subjects.

6.6.5 Adaptive Interface Layer testing parameters

Some parameters are explained in Section 5.2.1 for the calculation of utility. These particular parameters need suitable values for this experiment. These parameters are illustrated in Table 6.14 Theses values represent the weight each of the parameters contributes to calculation of the utility score. The security level parameter is the most important factor in the calculation of utility, as it directly relates to the probability that the user is whom they claim, therefore the weighting factor given must represent this. The value of the weighting for quality is dictated by Equation 5.2. These factors are weighted based on the judgement that the *security* level modifier should contribute more to the overall *utility* value.

Parameter	Value
Security Level Modifier	0.8
Quality Level Modifier	0.2

Table 6.14 Utility parameter values

Due to the short duration of the experiment any behaviour in the form of user messages or changes the adaptive layer makes to the capture parameters need to be apparent. This necessitates the values chosen for the number of transactions that are made before the users XML file is examined, as well as the subsequent periodic inspection interval, to be low. The values chosen are shown in Table 6.15.

Parameter	Value
Number of sessions before initial examination	2
Number of sessions before periodic inspection interval	1
Number of sessions at level 4 before increasing utility	2

Table 6.15 Temporal values for.data mining

Before testing could begin there were a number of final parameters which were related to temporal-based conditions which the adaptive system had to manage. These are shown in Table 6.16.

Parameter	Value
Calibration Valid	5 days
Template Valid	5 days

Table 6.16 Temporal values for calibration and template validity

Once again the values of these parameters were chosen so that the invocation of these modules could be observed during the post trial analysis phase.

6.7 Results

The results from this experiment will be presented in a number of ways. Firstly, there will be some statistical data relating to the FRR and FTE rates for both systems. Secondly, there will be data based on the specific *security* and *quality* levels achieved by each user during the verification transactions these figures are used to calculate the *utility* score for a transaction. The *utility* value is used to drive the system behaviour and it is hoped to observe the results from the manner in which the interface is reacting to the subject. Either in the form of messages or more underlying mechanisms such as the adjustment of the capture parameter values which are observable through analysis of the subject's data files.

6.7.1 Enrolment phase results

This initial phase was common to both interfaces that were under test during this experimental phase. Figure 6.3 and 6.4 illustrate the distribution of enrolment security levels that were achieved by each test group.

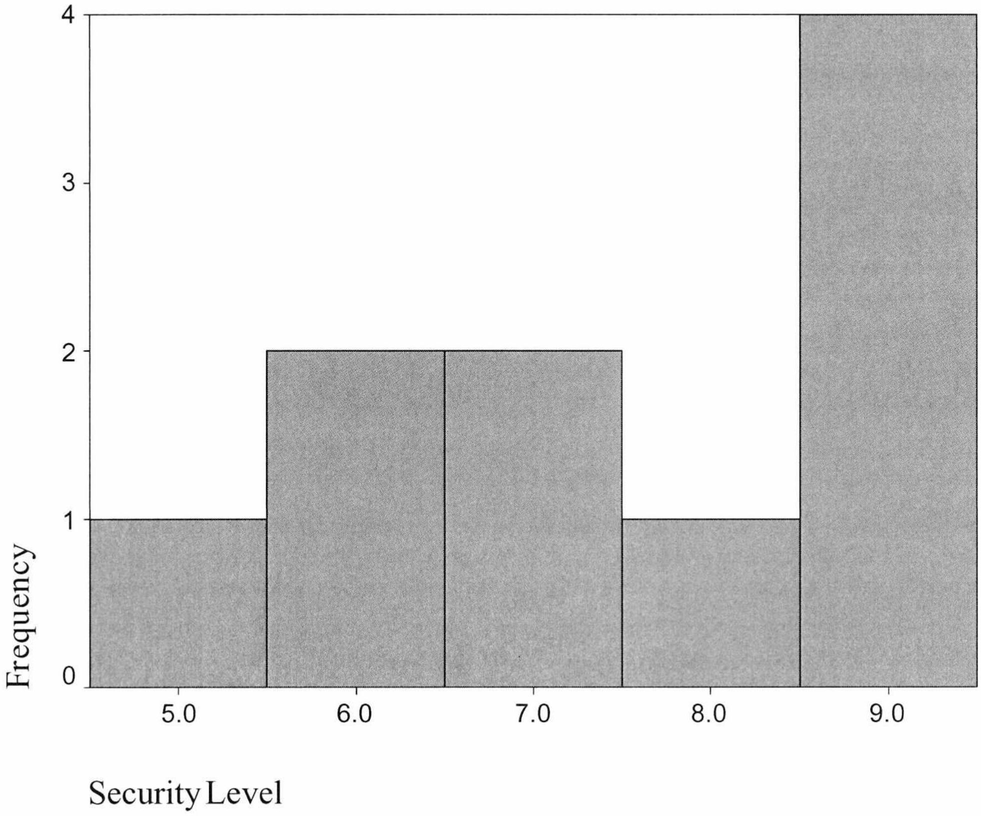


Figure 6.3 Distribution of adaptive interface enrolment security levels

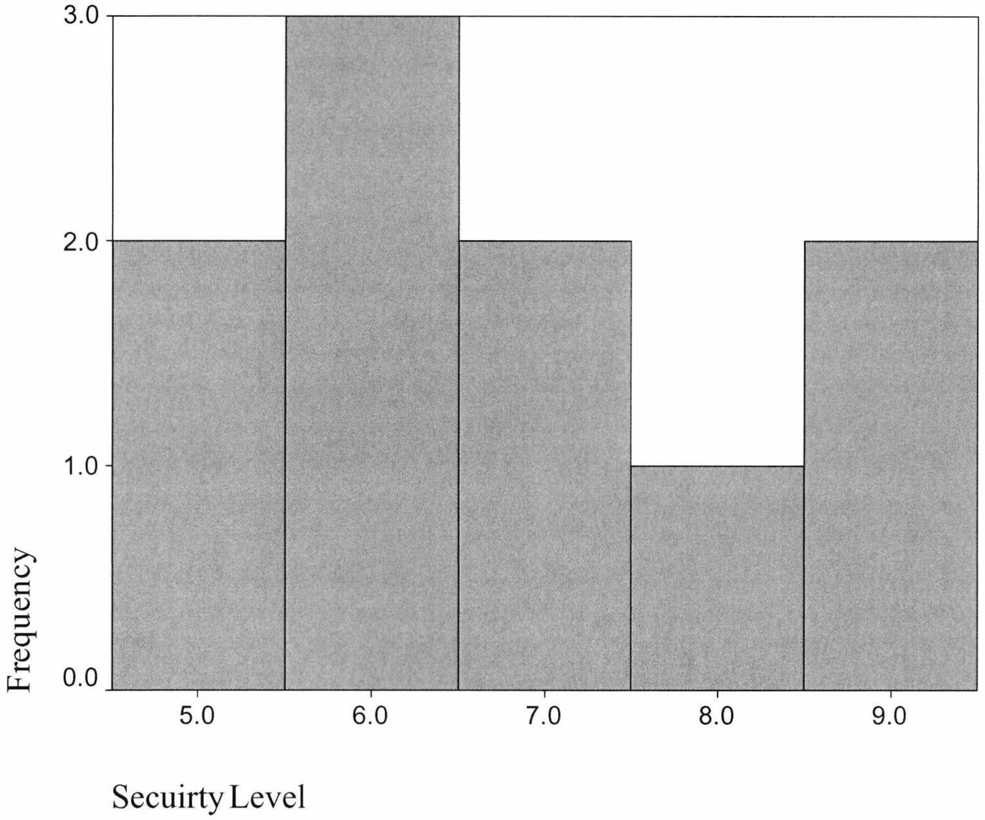


Figure 6.4 Distribution of non-adaptive interface enrolment security levels

Table 6.17 shows the FTE rates observed for each interface.

System	FTE (%)
Adaptive interface	0.0
Non adaptive interface	10.0

Table 6.17 Failure to enrol rates for systems under test

Although only one user explicitly failed to enrol in the given three enrolment attempts as specified by the test protocol, a number of enrolment attempts were required by a subset of users across both systems. The number of users and the attempts required for successful enrolment is illustrated by Figure 6.5.

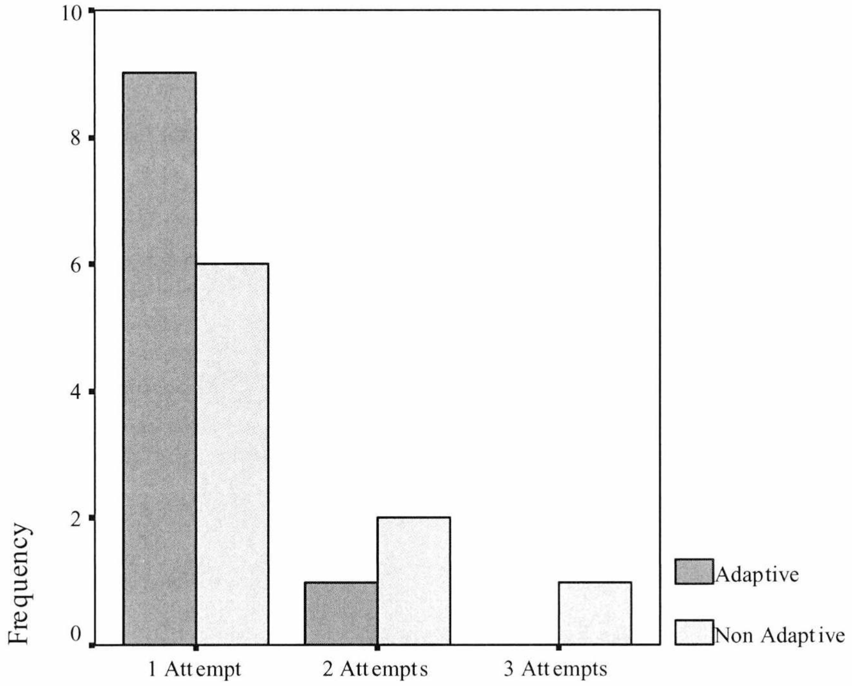


Figure 6.5 Enrolment attempt distribution

These initial enrolment results are interesting as they illustrate the potential benefit of employing a mechanism which provides assistance during the enrolment phase. For the non-adaptive interface one user required up to three attempts to successfully enrol. The adaptive system exhibited a reduction in the number of repeat enrolment attempts. This could be possibly due to the feedback presented to the user during a failed enrolment attempt. In both cases a supervisor was present. The only assistance given during this phase was that given by the interface alone, the supervisor did not give any guidance to subjects who failed enrolment attempts.

The enrolment security levels distributions are dependant on the ability of the user to provide an adequately feature rich sample to the sensor. It can be seen that in Figure 6.3 the percentage of users that managed to enrol at the highest security level is 40% of the total test group, whilst in the non-adaptive test group this figure is only 22% (Figure 6.4). The adaptive interface required users to calibrate the device before attempting enrolment. This goes some way to ensure that the samples donated are of the highest possible quality so that an optimum image is passed to the subsequent feature extraction phase. The distribution of the adaptive test group results shows a shift towards the higher end of the scale, illustrating that on average users in this test group achieved a higher enrolment security level than those in the non-adaptive group.

Whilst these results are not conclusive evidence that the adaptive interface is enabling users to perform better than their counterparts in the non-adaptive test group, it does indicate at this stage that the adaptive interface may be providing a more robust environment for enrolment.

6.7.2 Verification phase

The results from this phase will be presented in a number of ways. Standard performance measures such as failure to verify rates will be reported. The data presented will also focus on some of the operational features present in the adaptive system. The functionality of these features will be tested and their overall impact on the operational effectiveness will be assessed.

6.7.2.1 System results

Table 6.18 illustrates the failure to verify rates that were observed for both systems under test.

Interface	Total Matches	Verification Failures	Failure To Verify (%)
Adaptive	54	4	7.4
Non-adaptive	55	10	18.1

Table 6.18 Failure to verify rates for both systems under test

6.7.2.2 Adaptive system components

In this section the focus will be on reporting the results that verify the correct operation of the functional components in the adaptive system.

6.7.2.3 Calibration monitor

Users of the adaptive system had to calibrate the device before enrolment could be initiated. This stored a copy of the calibration data from the subject's first contact with the system. This calibration data is only valid for the period of time as detailed in Table 6.16. Once this period has been exceeded the subject is alerted to the fact that the calibration data is out of date, by a scrolling message on the interface window. This is illustrated by the screenshots in Appendix B.

The operation of this component was verified in the post trial analysis phase. The operation of the component was verified by observing that the relevant message was displayed for the specific user after the absolute value for calibration validity had expired.

6.7.2.4 Feedback behaviour

The behaviour of the adaptive interface towards the subject with respect to providing informative help and assistance is governed by the performance of the subject. It was expected that during the operation of the system the majority of users would be performing in a band deemed as satisfactory or above (see Section 5.2.2). It is at the extremes of the behaviour levels, i.e. 1 and 4 that the system exhibits observable behaviour towards the user.

Figure 6.6 illustrates the values of *utility* achieved per user over the duration of the experiment for the adaptive interface. Instances where a user has failed to verify can be observed by the low utility score for that particular session. In order to explore the potential system behaviour user data can be examined and used to verify the operation of the behaviour levels.

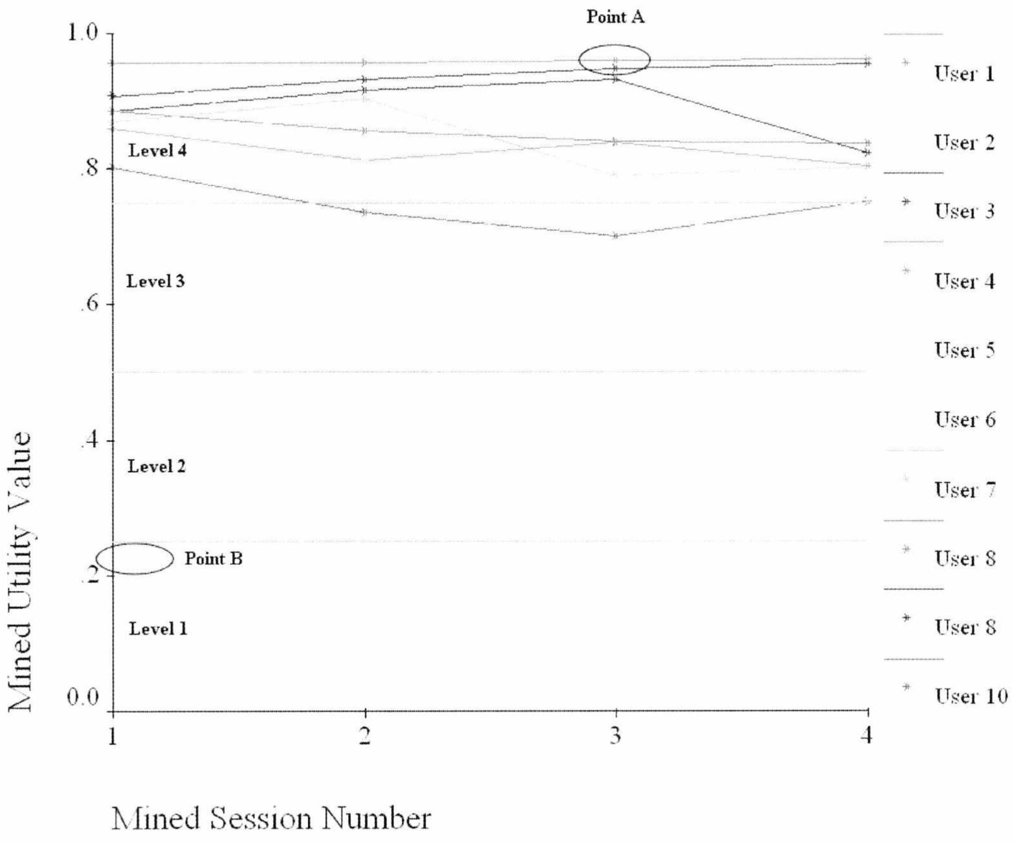


Figure 6.6 Behaviour levels for adaptive system

Figure 6.6 depicts how the behaviour levels of the adaptive interface changed over the extent of the trial. The reference lines on Figure 6.6 indicate the various behaviour levels ranges as specified in Section 5.2.2. This data illustrates that a number of subjects are performing particularly well, achieving the maximum security level possible over a number of system interactions. As detailed in Section 5.2.2.4 if a subject is in Behaviour Band 4 for more than 2 sessions (as stated in Section 6.6.5) then the system will attempt to increase the *quality* setting of the acquisition parameters in order to achieve a higher *utility* score.

This mechanism can be seen at work at point A in Figure 6.6, at this stage two users have met the condition stipulated for an increase in the *quality* level and this is exhibited by an observable increase in the utility score for this subject. This particular result illustrates the adaptivity demonstrated toward the subjects that are deemed to be performing well. A number of other users had performed well enough for the manifestation of a *utility* increase. This is not apparent on the graph as subsequent verification performance by these subjects was not sufficient to produce high values of *utility*.

The inverse of this situation is when a subject is having repeated difficulty supplying samples which can be successfully verified. In this case if the system determines that the user is residing in Behaviour Band 1 then a dialog is launched to attempt to advise the user as to why these verification failures are occurring. During this short experiment there was only one user whose performance was poor enough to launch this particular mechanism. This is illustrated by the user whose first mined session is below the threshold for level 1 (0.25), depicted as point B on Figure 6.6.

6.7.2.5 Non-adaptive system

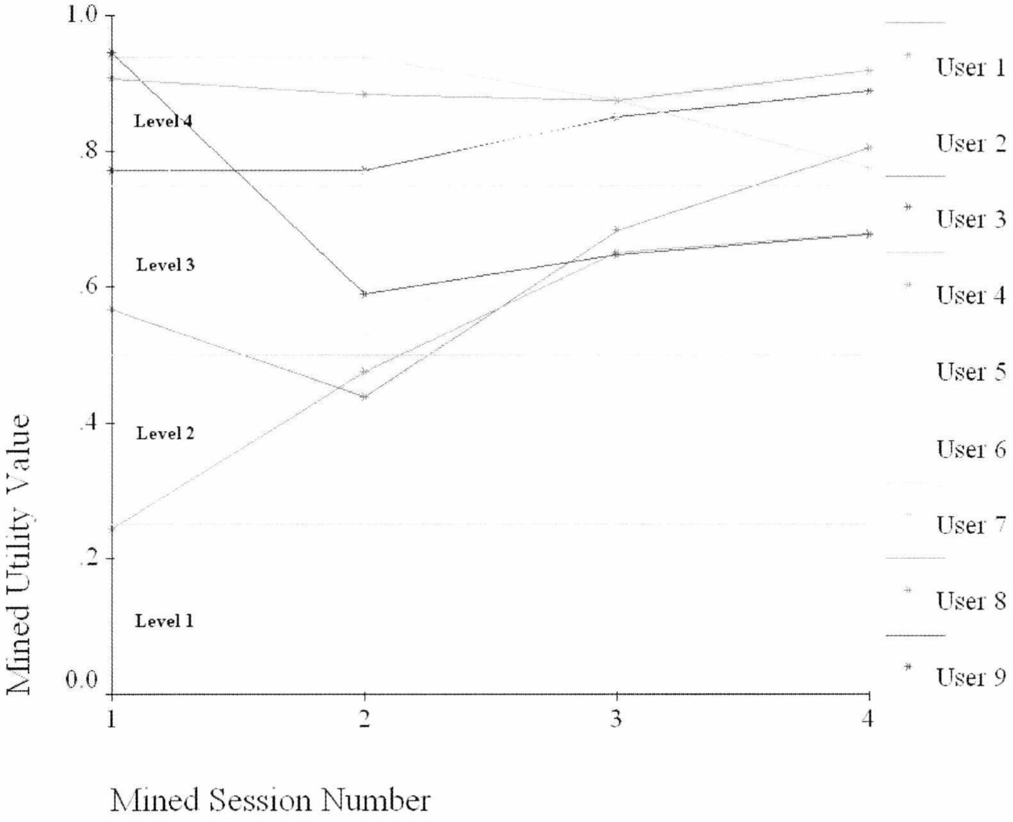
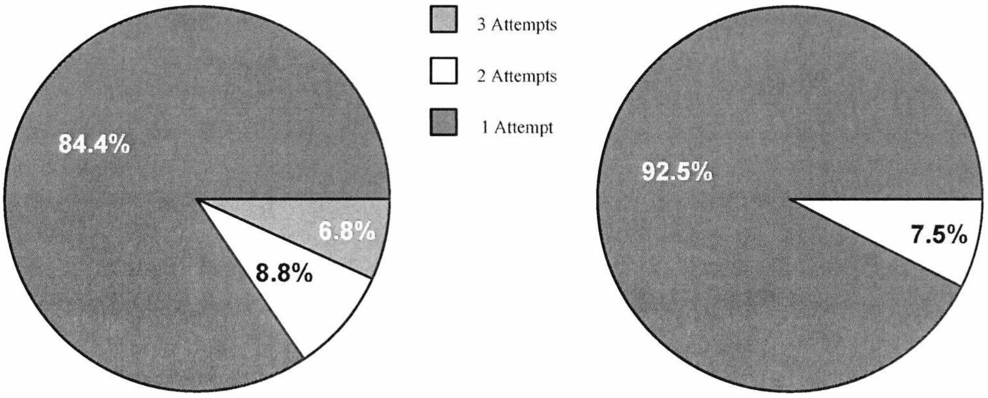


Figure 6.7 Behaviour levels for non-adaptive system

Figure 6.7 depicts how the behaviour levels for the non-adaptive test group varied over the period of the trial. Users of this system also exhibited the similar condition under which the samples which were being donated were in behaviour level 1 over a number of attempts. The non-adaptive system does not 'reward' users in the same manner that the adaptive system does. The lack of adaptation in this particular system limits the capacity of the system to determine if the user is able to provide better samples at some time in the future based on previous history. This leads the process of reenrolment or template updating to be an uneducated procedure in which the system has no knowledge if the user could possibly be enrolled at a higher *security* level than previous attempts. Effective management of the complex interaction over the lifetime of a biometric application dictates the use of some form of adaptivity.



Percentage of users successfully verified in *n* attempts for non adaptive system

Percentage of users successfully verified in *n* attempts for adaptive system

Figure 6.8 Repeat verification charts

Figure 6.8 depicts the percentage of users that successfully verified in the specified number of attempts. An interesting feature present in this data is where a subject initially fails a verification attempt. The non-adaptive interface did not offer any feedback on individual failed verification attempts and it can be observed that the non-adaptive test group exhibited a greater proportion of repeated verification attempts when compared to the results from the adaptive interface test group.

It is also important to try and assess the nature of the samples which failed verification across both test groups. Although the non-adaptive interface was not providing feedback on any failed transactions it was still recording the identified possible fault types to a log for subsequent analysis. Figure 6.9, illustrates the breakdown of fault types that occurred across both of the systems.

It can be observed that there are some interesting trends in the distribution of fault types across both the interfaces under examination. The non-adaptive system shown in Figure 6.9 exhibits a broad range of identified possible fault types, within the Type A fault category. The main contributions in this particular group come from the condition where a faint image has been presented to the sensor and also from the condition where finger placement was identified as the probable cause of failure.

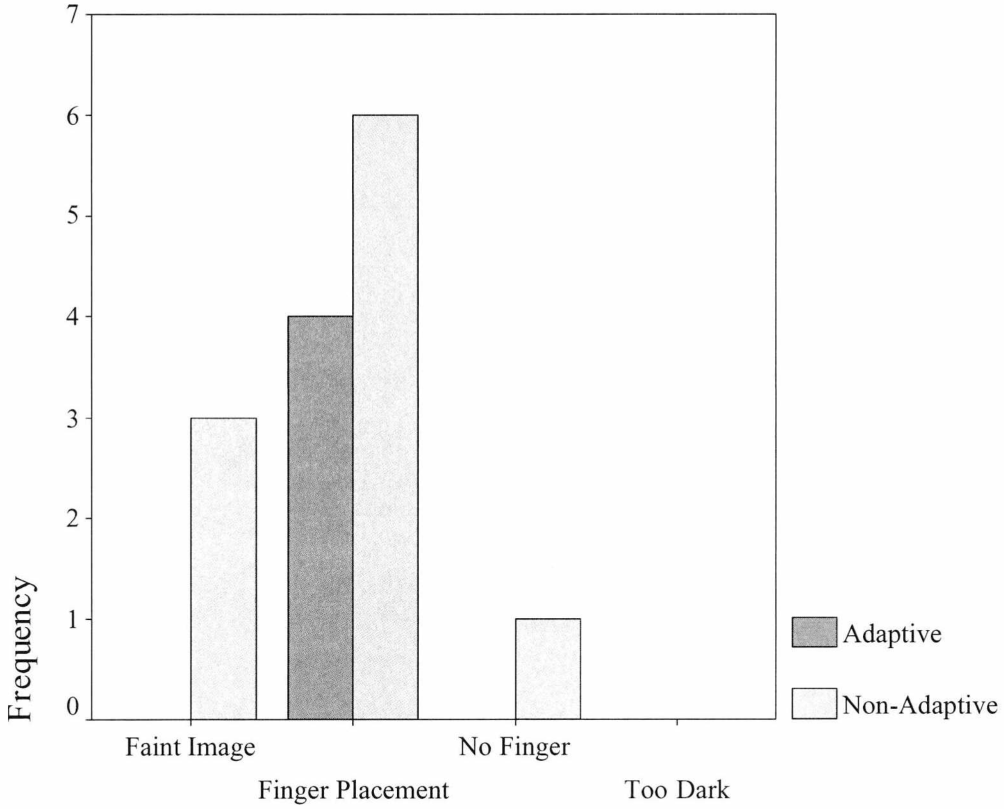


Figure 6.9 Verification failure breakdown by type

In the adaptive test group the distribution of failures is concentrated in the area where finger placement has been identified as the most likely reason for failure to verify. Also of interest in these results is the absence of conditions where not enough pressure has been applied to the sensor, which results in faint images being acquired. This could be attributed to a number of probable mechanisms, one is the presence of the calibration monitor in the adaptive system which through its operation attempts to ensure optimum acquisition settings. The other is perhaps the provision of the in-application help which demonstrates how to donate an adequately feature rich sample in order to facilitate successful verification. Another feature which could be aiding adaptive users is the presence of the real-time agent dialog which provides feedback pertaining to the particular fault type identified at the point of failure. This allows the user to take proactive measures immediately and aids in the correction of the problem that initially caused verification failure. The process of habituation is also expected to contribute further to the minimisation of the frequency of this particular fault across both systems.

6.8 Conclusions

In this chapter the functional testing of both the IAMBIC system and also the adaptive interface has been presented. The first half of this chapter is concerned with the presentation of the test protocol and associated results from the scenario testing of the complete IAMBIC system. These results highlight areas where some system parameters could be modified in order to provide a more robust system. These modifications are in part required for the simple fusion scheme applied during this testing phase. The overall performance of the IAMBIC system was found to be satisfactory, although some system improvements could be implemented in light of some of these experimental results. These are covered in more detail in Section 6.5.

The second half of this chapter focuses on the test protocol and related results for the adaptive interface. The purpose of the testing was not only to ensure that the functional components of the system were operating in a cohesive manner but also to test the hypothesis that an adaptive system would provide an enhanced biometric sample acquisition environment when compared to a non-adaptive system.

The results from this experiment tend to indicate that in the trial population, the subjects in the adaptive group witnessed a reduction of FTE and also FRR conditions. The adaptive system exhibited varying behaviour towards a subset of the test population. In particular where users were donating samples of high quality, the system began adapting to these users by adjusting image acquisition parameters. At the other end of the spectrum, the users in the adaptive test group whose performance was deemed poor were subject to the 'periodic-failure' dialog. This failure dialog attempts to identify the probable causes of failure and suggests some steps the user can take to minimise the failure reoccurring.

The performance of the users in the non-adaptive test group was found to be acceptable, however, this test group experienced a greater proportion of repeat verification attempts. In some cases requiring the subject required three attempts to successfully verify. These instances may have been reduced if the system had presented some form of real-time analysis and feedback for samples failing verification, as provided by the adaptive system.

The users of the non-adaptive system had no automated facility available for template management. The results from this particular test group appear to indicate that even during the short duration of this experiment subjects were able to deliver sustained verification successes at their enrolled security level. This indicates that the potential for template adjustment over time is entirely feasible. Whilst users who are enrolled at the highest security level do not benefit as much from the template management mechanism than those enrolled at lower security levels, it still allows for added user flexibility as the system can make informed decisions at the time of reenrolment regarding security levels and quality settings. The benefit of employing a scheme of template management during the lifetime of a biometric application is clear as the nature of the interaction with the system is fundamentally dynamic and therefore the system should incorporate some form of dynamism in its operation. Whether this is performed at an API level or as middleware in the application it is an important feature and should not be overlooked.

In Chapter 7 these results will be reviewed in more detail and some recommendations will be proposed in order to enhance the functionality of both the IAMBIC system and also the adaptive interface layer.

Chapter 7

Conclusions and further research

This final chapter presents a summary of the overall research presented within this thesis. The most relevant conclusions arising from the research are outlined in this chapter and the contribution of this work is highlighted. Suggestions are provided for the future research and exploitation of the developed concepts.

7.1 Summary

The main focus of the research presented in this thesis addressed the aspects of intelligent interfaces and their usefulness with respect to biometric systems. The other topic of research detailed in this thesis is based upon the design and implementation of a distributed multi-modal biometric authentication system (IAMBIC) which was successfully developed and trialled.

In Chapter 1 a brief introduction to the fields of biometrics and software agents was presented. Physiological and behavioural characteristics that are suitable for use as a biometric are explored briefly in this introduction along with an overview of a biometric system and the generalised operations that occur within such a system. The domain of software agents was introduced and the main research areas in this field are outlined. A simplified taxonomy is also presented along with a brief synopsis of each identified agent type.

In Chapter 2, the broad topic of biometrics is examined. This chapter explores in some depth the features which are deemed suitable for use as a biometric, also introducing the measures that are employed in order to characterise the performance of such biometric systems. A number of specific modalities are examined and their operation analysed. There are some broader issues which the biometric industry is facing as a whole and which can be said to some extent are hindering the widespread deployment of these systems. These issues are documented and examined and proposals to overcome some of these issues are discussed.

Chapter 3 introduces the domain of software agents. This particular field of research has been gaining momentum since it was first proposed in the late 1950's. The domain can be split into three main research areas which encompass the necessary operations required to initially describe the functionality and behaviour of the agent entity, architectures which can implement the desired behaviour and languages that enable the realisation of agent concepts.

Chapter 3 also introduces some relevant agent types and in particular focuses on the class of interface agents which could be seen as beneficial if applied to biometric applications.

Chapter 4 introduces the principles behind the IAMBIC system and also provides initial design work on overall system structure and the identified system entities required for the complete operation of the system. The IAMBIC project was an investigation in to the design and implementation of a distributed multimodal identity authentication system. The motivation for the development of such a system was to provide an application which could provide a reliable distributed multimodal authentication system which also offered multiple levels of authorisation to provide access to different categories of information.

The IAMBIC system was designed and developed using two agent-based methodologies, GAIA and AUML. Both these were seen to adequately encapsulate the desired behaviour of the system throughout the development process. Key to the development of IAMBIC was the integration of software agents. These were employed to manage a number of aspects such as the complexity of the multi-modal interaction and communications infrastructure. Implementation of IAMBIC was based on a client C++/Java application which managed the hardware interfaces which the biometric devices, communications and database interaction. The Java server agent handled database queries. In order providing a standard data container throughout the system, XML was employed due to its flexibility in data representation and ease of integration with the languages used in the IAMBIC implementation.

One of the main phases of the IAMBIC project, was the biometric data gathering phase. During which a large group of volunteers (in the region of 220) were sampled in order to collect a number of different modalities over a time period. The impetus behind this phase was to observe in real terms the individual performance of the biometric modalities when compared to the manufacturers quoted performance and also to observe how users reacted to the individual biometric technologies. A number of observations were recorded from the results of this phase. The first was that the manufactures stated performance claims were often exaggerated to what was observed during 'real-life' trials. The second was that the nature of the software application itself can have a real impact on the performance of the user. It was this observation that led to the hypothesis that elements of agent based engineering could be applied at the interface level in order to combat some of these issues which users had been experiencing with the biometric sample donation process. A possible solution to these problems came from within the interface agent research community. Amongst the possible uses for this type of agent are entities which can provide training or assistance in some given task.

It was suggested that through the application of the interface agents the user experience towards the particular biometric device becomes much more robust and engaging. In the biometric research community a great deal of research has been given to improving performance though improving acquisition and recognition algorithms but the interface aspect of the application has been neglected. It was this aspect of combining the use of interface agents into a biometric application that was considered novel. Chapter 5 introduces this notion of the adaptive interface layer. This layer was primarily designed to work with the fingerprint modality which was employed in the IAMBIC project. This particular modality was chosen because it possessed the necessary requisites for straightforward integration into the proposed adaptive layer.

The planned operation of the layer is covered in some detail in Chapter 5, however, the main goals of the layer were to provide assistance in situations where the user has failed some operation (either enrolment or verification) and to provide some rudimentary user training in the device in order to promote good working practice, once again this is designed to minimise failures and in turn reduce the potential of user frustration towards the device. The layer also was responsible for the underlying template management and calibration routines that need to be addressed in any biometric system, these features are totally transparent to the user. These features were expected to provide significant enhancements for the user over non-adaptive systems.

The IAMBIC project itself was considered a great success by the DTI, the project objectives had been met or exceeded and overall user feedback from the system was positive with the exception of a number of users complaining about the overall stability of the voice recognition package. This particular aspect has already been discussed in Section 6.3.1.4. The robustness of employing multiple biometrics in a single authentication session was shown to be invaluable with respect to unauthorised fraudulent access. The results from the IAMBIC testing in Chapter 6 illustrated that if a uni-modal approach had been taken here a real possibility of system penetration by an impostor could have occurred. This also highlights the fact that although biometric system are promoted almost as a panacea for the human identity authentication problem, some of the current biometric solutions available in the market still have some way to go before these systems can offer fool proof identity authentication. The final IAMBIC system trial demonstrated that users of the system could use a number of biometric modalities as the authentication medium to provide differing levels of authentication for information retrieval over a network.

This feature of utilising multiple biometrics also brings to the fore another aspect which can have an impact on the benefits of such a system. The aspect of data fusion combining the verification scores from modalities and also possibly non-biometric sources of data is a pivotal subject for research, with the mechanics of combining this data being an ongoing focus for research. Even from the simplistic linear scheme for score fusion employed in the final IAMBIC system trial it demonstrated the situation where the overall levels required for system access had to be adjusted in light of the experimental results. This indicates that the confidence levels for system access must be carefully chosen in order to prevent unnecessary instances of false rejection. The benefits of using such a combination scheme may become more apparent in a system where multiple levels of authentication are required (even more than that required in the IAMBIC system).

The results obtained from the testing phase of the adaptive interface were encouraging. The performance of the subjects that utilised the adaptive interface experienced a number of benefits over those subjects who used the non-adaptive system. These benefits manifested themselves in a number of ways:

- Reduction of repeat verification attempts.
- Reduction of repeat enrolment attempts.

Whilst the reduction of the various rates presented above were perhaps not pronounced as initially expected this does illustrate that for subjects who are experiencing difficulty donating samples, the adaptive system can provide constructive assistance at the point of failure. This action can assist the user in the probable cause of failure, ensuring that the user is suitably advised as to certify that the next verification attempt can be met with success.

More importantly the adaptive system identified that a system that can provide a mechanism where the performance of the subject is analysed over the lifetime of the application. This analysis allows for the adjustment of the acquisition parameters as the subject experiences the effects of habituation. This enhances the robustness of the system as the interaction between the subject and the device itself it essentially dynamic in nature. The system should provide facilities to manage this dynamic interaction. By utilising a static approach to the design of a biometric system precludes the ability of the system to make decisions based on previous user history. Although the system cannot make pre-emptive decisions about future performance, due to the dynamism of the interaction, previous subject usage history could be useful in determining overall performance trends. This data could be employed at specific times i.e. reenrolment, in order to establish whether the user is capable of producing samples that meet or exceed the quality of previous samples included in the original template file.

In summation the main contribution of this work can be characterised as follows:

- The design, development and testing of an agent-based multimodal biometric framework

The size of the test population for the adaptive interface was not large enough to draw absolute conclusions from the observed results. Although the results indicate the potential of the following point:

- Interface agent technology can be effectively deployed to enhance the robustness of biometric applications which can lead to improvements in observed user performance

7.2 Future Research Suggestions

The orientation of the research in this thesis has been two fold. Firstly, the research was directed towards the development of the IAMBIC system which provided a distributed multi-modal authentication system. Secondly, research was conducted on the concept of an adaptive interface layer for biometric applications employing the notion of interface agents. This layer was attached to a commercial fingerprint system and primarily provided user assistance and training. Other functional components of the adaptive layer were designed in order to enhance the robustness of the biometric system towards user interaction.

Both these investigations have provided the groundwork for further possible investigation in the future. In the following section some points will be outlined for possible future lines of research.

7.2.1 Multi-modal authentication systems

The IAMBIC project demonstrated that multi-modal authentication system could be built at low cost with Commercial Off The Shelf (COTS) products available in the market today. Although there were some implementation issues such as API integration, due to no all encompassing standard API for biometric devices (Section 2.10.4), it was possible to develop a multi-modal application without excessive complication. The aspect of multi-modal systems is considered a fruitful area of research especially in the area of data fusion. Investigations in the area of data fusion could be conducted in order to determine the effects of various fusions schemes.

- Studies into the performance of various other fusion schemes such as voting
- Effects of combining non-biometric data

The notion of a distributed authentication scheme is also another area where further research could be conducted. There are a large number of issues surrounding distributed systems as a whole e.g. security. This issue is especially prevalent in the biometrics domain. Although encryption can go some way to provide secure communication channels, trust is another aspect which needs to be focused on in a distributed domain.

Research could be channelled into the development of standards and systems for trust and security in distributed biometric systems.

- Techniques to guarantee that biometric devices attached to client computers are trustworthy
- Standards to ensure adequate security of biometric data throughout the distributed system

7.2.2 Adaptive Interface Layer

The adaptive layer is not only limited to the fingerprint modality explored in this research. The application of this adaptive layer could feasibly be implemented on a number of modalities such as face or voice for instance. The system components can be used with little modification all that is required is to determine the most suitable manner to implement the components.

System components such as the analysis engine would have to be converted to operate with the proposed modality. For facial-based biometric systems the analysis engine would need to be modified to analyse the images based by this particular biometric rather than the fingerprint images the layer was originally designed to operate on. The principles of operation are similar, the facial image can be analysed in order to determine various image based parameters to determine a set of suitable fault characteristics that would enable images that failed to be classified into a specific fault type. The fault types for this modality could share some commonality with the ones identified for the fingerprint modality, such as the image was too light or dark for instance.

The calibration monitor component is useful for any biometric that employs an imaging unit attached to the client biometric system that can be calibrated. A facial-based biometric system that utilises a camera, such as a web cam could benefit from some form of calibration routine. This is especially prevalent if the API itself does not offer some form of auto-tuning facility in order to facilitate optimum image acquisition settings for sample gathering. The temporal value for calibration validity would have to be determined on a case by case basis for the proposed target application. Capture surroundings which offer a high degree of variability in environment parameters, would probably benefit from a lower value in this case.

The template generation monitor is another critical component that could conceivably be employed for another biometric modality. Although the particular biometric modality that the adaptive layer was developed on did not support the addition of extra samples to the user template, it is known that other biometric systems do allow this. The facial recognition package employed for the IAMBIC project provided such a feature. This feature is important as it facilitates the inclusion of additional samples that are deemed to be sufficiently rich to be included in the subject template. By increasing the number of images present in the template file it can maximise the robustness of the system towards successful recognition of the user.

Adaptive data fusion could also be another potential area of research that could be explored. The fusion process could be managed by agent-based technology. The fusion process itself may take many parameters, not only the raw result scores from the modality itself. Information such as geographic location or IP addresses are two such sources of information that could be used in some manner in the fusion routine. Adaptive fusion could provide a system where this calculation can be performed dynamically based on parameters which the end application dictates, the weightings of which may vary over the lifetime of the application. Providing such a facility serves only to enhance the capability of the biometric system it is attached to.

Another potential area of further research could be conducted into the resilience of the adaptive layer against spoofing attacks. Although the layer was designed to aid a user towards successful verification, this assistance could conceivably be misused by an impostor in an attempt to gain system access. This warrants an investigation in order to establish whether the adaptive layer could assist in spoofing based attacks.

Overall it can be seen that the principles of the adaptive biometric interface layer are described at a high level of abstraction. This allows for the future potential development of the adaptive biometric interface layer for various other modalities. In conclusion both the main areas of research presented in this thesis provide the foundation for significant possible future development to enhance the performance and usability of biometric systems

References

[Accetta et al, 1986] Accetta, M., Baron, R., Bolosky, W., Golub, D., Rashid, R., Tevanian, A., and Young, M. (1986). Mach: A new kernel foundation for UNIX development. *Summer USENIX Conference*, pp. 93-112.

[Acharya et al, 1997] Acharya, A., Ranganathan, M., Salz, J. (1997). Sumatra: A Language for Resource aware Mobile Programs. *Lecture Notes in Computer Science* No. 1222, pp. 111-130.

[Agha, 1986] Agha, G. (1986). ACTORS: A Model of Concurrent Computation in Distributed Systems. *The MIT Press*, Cambridge, MA.

[Agre, 1997] Agre, P. (1997). Computation and Human Experience. *Cambridge University Press*, UK.

[Agre and Chapman, 1987] Agre, P., and Chapman, D. (1987). Pengi: An Implementation of a Theory of Activity. *Proceedings of AAAI-87*, pp. 268-272.

[Aguilar et al, 1995] Aguilar, L., Alami, R., Fleury, S., Herbb, M., Ingrand, F.F. and Robert, F. (1998). Ten autonomous mobile robots (and even more) in a route network like environment. *Proceedings of IROS 95, LAAS Report N°95250.*

[Alechina and Logan, 2002] Alechina, N., and Logan, B. (2002).Ascribing beliefs to resource bounded agents. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 881-888.

[Allen et al, 2002] Allen, J, Blaylock, N., and Ferguson, G. (2002). *A problem solving model for collaborative agents*, pp. 774-781.

[Ambros-Ingerson and Steel, 1988] Ambros-Ingerson, J. A., and Steel, S. (1998). Integrating planning, execution and monitoring. *Proceedings of the 15th National Conference on Artificial Intelligence*, pp. 83-88.

[Ametller et al, 2004] Ametller, J., Robles., and Ortega-Ruiz, J. (2004). Self-Protected Mobile Agents. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 362-367.

[Anthony et al, 2001] P. Anthony, P., Hall, W., Dung Dang, V., and Jennings, N. (2001). Autonomous agents for participating in multiple online auctions. *IJCAI01 Workshop on E-Business 4 the Intelligent Web*, pp. 54-64.

[Arai, 2002] Arai, T., and Stolzenburg, F. (2002).Multiagent systems specification by UML statecharts aiming at intelligent manufacturing. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 11-18.

[Arai and Sycara, 2000] Arai, S., Sycara, K. (2000). Multi-agent reinforcement learning for planning and conflict resolution in a dynamic domain. *Proceedings of the fourth international conference on Autonomous agents*, pp. 104-105.

[Arazy and Woo, 2002] Arazy, O., and Woo, C. (1999). Analysis and Design of Agent-Oriented Information Systems. *The Knowledge Engineering Review*, 17(3):215-260

[Arnold et al, 1999] Arnold, K., Wollrath, A., O'Sullivan, B,. Scheifler, R., and Wald, J. (1999). *The Jini Specification*. Addison-Wesley.

[ANSI, 1995] American National Standards Institute. (1995). Knowledge Interchange Format Specification. *http://logic.stanford.edu/kif/specification.html*

[ASNI, 2003] American National Standards Institute. (2003).Biometric Information Management and Security for the Financial Services Industry (X9.84). *http://www.x9.org/*

[Baral et al, 1998]Baral, C., Floriano, L, Hardesty, A., Morales, D., Nogueira, M., and Son, T. (1998). From theory to practice: the UTEP robot in the AAAI 96 and AAAI97 robot contests. *Proceedings of the second international conference on Autonomous agents*, pp. 32-38.

[Bauer, 2001] Bauer, B. (2001). UML classes diagrams and agent-based systems. *Proceedings of the fifth international conference on Autonomous agents*, pp. 104-105.

[Baumann et al, 1997] Baumann, J., Hohl, F., Rothermel K., and Strasser M. (1997). Mole - concepts of a mobile agent system. *World Wide Web Journal*, 1(3):12-137.

[Beetz et al, 2002] Beetz, M., Buck, S., Hanek, R., Schmitt, T., and Radig, B. (2002). The AGILO autonomous robot soccer team: computational principles, experiences, and perspectives. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 805-812.

[Bellifemine et al, 2001] Bellifemine, F., Poggi, A., and Rimassa, G. (2001). JADE: a FIPA2000 compliant agent development environment. *Proceedings of the fifth international conference on Autonomous agents*, pp. 216-217.

[Benthem, 1983] van Benthem, J. (1983) Modal logic and classical logic. *Bibliopolis,* Naples.

[Billsus & Pazzani, 1999] Billsus, D., and Pazzani, M.. (1999). A Personal News Agent that Talks, Learns and Explains. *Proceedings of the. Thirs international conferance. on Autonomous Agents*. pp. 268-275.

[BioAPI, 2001] The BioAPI Consortium. (1995).Consortium Announces Release of Final Specification and Reference Implementation. *www.bioapi.org.*

[Biometrics Consortium, 1995] Biometrics Consortium. (1995). An introduction to biometrics. *http://www.biometrics.org/html/introduction.html*

[Boggs, 1973] Boggs, J. (1973). IBM Remote Job Entry Facility: Generalized Subsystem Remote Job Entry Facility. *IBM Technical Disclosure Bulletin, 75.*

[Bohnenberger, 1996] Bohnenberger, T. (1996). AgentMove Distributed Dynamic Public Transport Scheduling. *http://citeseer.ist.psu.edu/bohnenberger96agentmove.html*

[Boon, 1998] Boon, G. (1998). Concept features in Re:Agent, an intelligent e-mail agent. *The Second International Conference on Autonomous Agents*, pp 10-13.

[Bratman et al, 1988] Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349-355.

[Broerson et al, 2001] Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., and van Torre, L. (2001). The BOID architecture: conflicts between beliefs, obligations, intentions and desires. P*roceedings of the fifth international conference on Autonomous agents*, pp. 9-16.

[Brooks, 1986] Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14-23.

[Brooks, 1991] Brooks, R. (1991). Intelligence without reason. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 569-595.

[Brooks, 1991a] Brooks, R (1991). Intelligence without representation. *Artificial Intelligence*, 47:139-159.

[Brown et al, 1998] Brown, S., Santo, E., and Banks, B. (1998). Utility theory-based user models for intelligent interface agents. *Proceedings of the Twelfth Canadian Conference on Artificial Intelligence*, pp. 379-393.

[Browning and Tryzelaar, 2003] Brwoning, B., and Tryzelaar, E. (2002). ÜberSim: a multi-robot simulator for robot soccer. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 948-949.

[Casterfranchi, 1995] Casterfranchi, C. (1995).Guarantees for autonomy in cognitive agent architecture. *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pp. 56-70.

[Chapman, 1987] Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence*, 32:333-378.

[Chauhan and Baker, 1998] Chauhan, D. and Baker, A. (1998). JAFMAS: A multiagent application development system. *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 100-107.

[Chen & Sycara, 1998] Chen, L., and Sycara, K. (1998). WebMate: A personal agent for browsing and searching. *Proceedings of the Second International Conference on Autonomous Agents*, pp. 132-139.

[Chong et al, 1997] Chong, M., Ngee, T., Jun, L., and Gay, R. (1997). Geometric framework for Fingerprint Classification. *Pattern Recognition*, 30(9):1475-1488.

[Cohen and Levesque, 1990] Cohen, P., and Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42:213–261.

[Cohen & Perrault, 1979] Cohen, P., and Perrault, C. (1979). Elements of a plan based theory of speech acts. *Cognitive Science*, 3:177-212.

[Coleman et al, 1994] Coleman, D., Arnold, P., Bodoff, S., Dollin, C., Gilchrist, H., Hayes, F., and Jeremaes, P. (1994). Object-Oriented Development: The fusion Method. *Prentice Hall International*, pp. 26-31.

[Connah, 1994] Connah, D. (1994). The Design of Interacting Agents for Use in Interfaces. *Human-Machine Communication for Educational Systems Design*, NATO ASI Series, Series F, Computer and Systems Sciences 129, Springer Verlag.

[Das et al, 1998] Das, S., Caglayan, A., and Gonsalves, P. (1998). Increasing agent autonomy in dynamic environments. *Proceedings of the second international conference on Autonomous agents,* pp. 309 – 316.

[Das et al, 1999] Das, S., Krikorian, R., and Truszkowski, W. (1999). Distributed planning and scheduling for enhancing spacecraft autonomy. *Proceedings of the third annual conference on Autonomous Agents,* pp. 422 – 423.

[Daugman, 2001] Daugman, J. (2001). Combining multiple biometrics. *http://www.cl.cam.ac.uk/users/jgd1000/combine/*

[Dennet, 1987] Dennett, D. (1987). The Intentional Stance. *The MIT Press*.

[Dent et al, 1992] Dent, L., Boticario, J., McDermott, J., Mitchell Y., and Zabowski, D. (1992). A personal learning apprentice. *In Proceedings of Tenth National Conference on Artificial Intelligence*, pp. 96-103.

[Doran et al, 1990] Doran, J., Carvajal, H., Choo, Y. J., and Li, Y. (1990). The MCS multi-agent testbed: Developments and experiments. *Proceedings of the International Working Conference on Cooperating Knowledge Based Systems*, pp. 240–254.

[Doyle and Hayes-Roth, 1997] Doyle, P., and Hayes-Roth, B. (1997). An intelligent guide for virtual environments. *Proceedings of the first international conference on Autonomous agents,* pp. 508-509.

[Driscoll et al, 1991] Driscoll, E., Martin, C., Ruby, K., Russel, J., and Watson, J. (1991). Method and Apparatus for Verifying Identity Using Image Correlation. *US Patent No 5067162.*

[Durfee et al, 1987] Durfee, E-H., Lesser, V., and Corkill, D. (1987). Coherent Cooperation among Communicating Problem Solvers. *IEEE Transactions of Computers* C-36(ii), pp. 1275-1291.

[Eshera & Fu, 1984] Eshera, M., and Fu, K. (1984). A Similarity Measure Between Attributed Relational Graphs for Image Analysis. *Proceedings of the seventh international conference on Pattern Recognition*, pp75-77.

[Elammari and Lalonde, 1999] Elammari, M., and Lalonde, W. (1999). An agent-oriented methodology: High-level and intermediate models. *Proceedings of the first international workshop on Agent-Oriented Information Systems*, 12.

[Elichai, 2004] Elichai, A. (2004). RDS: Remote Distributed Scheme for Protecting Mobile Agents. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 354-361.

[Etzioni and Weld, 1995] O. Etzioni, O., and Weld, D. S. (1995). Intelligent agents on the internet: Fact, fiction, and forecast. *IEEE Expert*, 10(4):44-49.

[Fairhurst et al, 2002] Fairhurst, George, J., and Deravi, F. (2002). Scenario based data collection trials for the evaluation of multi-modal biometric processing: a preliminary report. *Proceedings of the Sixth International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, pp.1217-1221.

[Ferber, 1994] Ferber, J. (1994).Simulating with Reactive Agents. *Many Agent Simulation and Artificial Life*, Amsterdam: IOS Press, pp.8-28.

[Ferguson, 1992] Ferguson, I. (1992). TouringMachines: Towards an architecture for Adaptive, Rational, Mobile Agents. *Proceedings of the 3rd European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pp. 249-262.

[Fikes and Nielson, 1971] Fikes, R., and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2):189-208.

[FIPA, 1996] Foundation for Intelligent Physical Agents. (1996). *http://www.fipa.org*

[FIPA, 2003] Foundation for Intelligent Physical Agents Modeling Technical Committee. (2003). *http://www.fipa.org/activities/modeling.html*

[Fischer, 1993] Fischer, K. (1993). The Rule based Multi Agent system MAGSY. *Proceedings of the CKBS'92 Workshop*.

[Fisher, 1994] Fisher, M. (1994). A survey of Concurrent METATEM - the language and its applications. *Proceedings of the First International Conference (LNAI Volume 827)*, pp. 480-505.

[Fischer et al, 1995] K. Fischer, K., Muller, J. P., and Pischel, M. (1995). Cooperative transportation scheduling: an application domain for DAI. *Journal of Applied Artificial Intelligence,* 10(1):1-34.

[Fitz & Green, 1996] Fitz, A., and Green, R. (1996). Fingerprint Classification Using Hexagonal Fast Fourier Transform. *Pattern Recognition*, 29(10):1587-1597.

[Franklin and Graesser, 1996] Franklin, S., and Graesser, A. (1996). Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Institute for Intelligent Systems*, pp. 21-36.

[Galton, 1878] Galton, F. (1878). Composite Portraits, *Nature*, (18):97-100.

[Galton, 1888] Galton, F. (1888). Personal Identification and Description I, *Nature*, 38(973):173-177.

[Gamble et al, 1998] Gamble, E., Pell, B., Gat, E., Keesing, R., Millar, W., Nayak, P., Plaunt, C., and Williams, B. (1998).A hybrid procedural/deductive executive for autonomous spacecraft. *Proceedings of the second international conference on Autonomous agents*, pp. 369 - 376.

[Garvey and Myers, 1993] Garvey, T., and Myers, K. (1993). The intelligent information manager. *Final Report SRI Project 8005*, Artificial Intelligence Center, SRI International.

[Gilbert et al, 1995] Gilbert, D., Aparicio, M., Atkinson, B., Brady, S., Ciccarino, J., Grosof, B., O'Connor, P., Osisek, D., Pritko, S., Spagna, R., and Les Wilson, L. (1995). IBM intelligent agent strategy. *White paper*.

[Genesereth and Katchpel, 1994] Genesereth, M., and Katchpel, S. (1994). Siftware Agents. Communications of the ACM, 37(7):48-53.

[Georgeff and Lansky, 1987] Georgeff, M., and Lansky, A. (1987). Reactive Reasoning and Planning. *Proceedings of the conference of the American Assoc. of Artificial Intelligence*, pp. 677-682.

[Georgeff and Ingrand, 1989] Georgeff, M., and Ingrand, F. (1989). Monitoring and Control of Spacecraft Systems Using Procedural Reasoning. *Proceedings of the Space Operations-Automation and Robotics Workshop.*

[Gold & Rangarajan, 1996] S. Gold, S., and Rangarajan, A. (1996). A Graduated Assignment Algorithm for Graph Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4): 377-388.

[Gosling et al, 1996] Gosling, J., Joy, B., and Steele, G. (1996). The Java Language Specification. *Addison-Wesley.*

[Griffen et al, 1994] Griffin, C., Matsui, T., and Furui, S. (1994). Distance measures for text-independent speaker recognition based on MAR model. *ICASSP*, pp. 309--312.

[Gutknecht and Ferber, 2000] Gutknecht, O., and Ferber, J. (2000). The MadKit agent platform architecture. *Agents Workshop on Infrastructure for Multi-Agent Systems*, pp. 48-55.

[Hasegawa et al, 2003] Hasegawa, T., Cho, K., Kumeno, F., Nakajima, A.., and Honiden, S. Interoperability for mobile agents by incarnation agents. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems,* pp. 1006-1007.

[Hayes, 1995] Hayes-Roth. B. (1995). An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence: Special Issue on Agents and Interactivity*, 72:329-365.

[Hayes-Roth, et al, 1992] Hayes-Roth, B., Washington, R., Ash, D., Collinot, A., Vina, A., and Seiver, A. (1992). Guardian: A prototype intensive-care monitoring agent. *Artificial Intelligence in Medicine,* 4:165-185.

[Hayes-Roth et al, 1993] Hayes-Roth, B., Lalanda, P., Morignot, P., Pfleger, K., and Balabanovic, M. (1993). Plans and Behavior in Intelligent Agents. *Technical Report KSL-93-43,* Knowledge Systems Laboratory, Computer Science Department, Stanford.

[Hayes-Roth et al, 1995] Hayes-Roth, B., Pfleger, K., Lalanda, P., Morignot, P., and Balabanovic, M. (1995). A domain-specific software architecture for adaptive intelligent systems. *IEEE Transactions on Software Engineering*, 21(4):288-301.

[Hayes-Roth and ven Gent, 1997] Hayes-Roth, B., and van Gent, R. (1997). Story-marking with improvisational puppets. *Proceedings of the first international conference on Autonomous agents*, pp, 1-7.

[Heinze and Sterling, 2002] Heinze, C., and Sterling, L. (2002). Using the UML to model knowledge in agent systems. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 41-42.

[Henry, 1900] Henry, E. (1899). Classification and uses of Finger Prints. *Routledge*, London, pp. 54-58.

[Hewitt, 1977] Hewitt, C. (1977). Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3):323-364.

[Hintikka, 1962] Hintikka, J. (1962). Knowledge and Belief. *Cornell University Press*.

[Horvitz et al, 1998] E. Horvitz, E., Breese, J., Heckerman, D.,Hovel, D., and Rommelse, K. (1998). The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence,* pp. 256-265.

[Huber, 1999] Huber, M. (1999). JAM: A BDI-theoretic mobile agent architecture. *Proceedings of the Third International Conference on Autonomous Agents*, pp. 236-243.

[Hübner and Sichman, 2000] Hübner, J, and Sichman, J. (2000). Simple Agent Communication Infrastructure. *http://www.lti.pcs.usp.br/saci/*

[IBIA, 1988] International Biometric Industry Association (1988). *www.ibia.org*

[INCITS, 2005] International Committee for Information Technology Standard. (2005). Effects of User Habituation and Acclimatization Version 2.0. *http://www.incits.org/tc_home/m1htm/docs/m1050139.pdf*

[Hong & Jain, 1999] Hong, L, and A. K. Jain, K. (1999). Classification of Fingerprint Images. *Proceedings of the 11th Scandinavian Conference on Image Analysis*, pp 56-68.

[Identix, 1982] Identix. (1982). *http://www.identix.com*

[Jain et al, 1997] Jain, A., Hong, L., Pankanti, S., and Bolle, R.(1997). An Identity Authentication System Using Fingerprints. *Proceedings of the IEEE*, 85(9):1365- 1388

[Jain et al, 1999] [18] Jain, A., Prabhakar, S., and Hong, L. (1999). A Multichannel Approach to Fingerprint Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):348-359.

[Jain et al, 2004] Jain, A., Pankanti, S., Prabhakar, S., Hong, L., Ross, A., and Wayman., J. (2004). Biometrics a grand challenge. *Proceedings of the International Conference on Pattern Recognition, (ICPR)*, 2:935-942.

[Jennings, 1992] Jennings, N. (1992). On Being Responsible. *Decentralized Artificial Intelligence,* (3), pp. 93-102.

[Jennings, 1993b] Jennings, N. (1993). Specification and implementation of a belief desire joint-intention architecture for collaborative problem solving. *Journal of Intelligent and Cooperative Information Systems*, 2(3):289-318.

[Jing et al, 2002] Jing, Y., Brown, K., and Taylor, N. (2002). Intelligent interface agents for a system to diagnose eye disorders. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 794-795.

[JNI, 1999] Sun Microsystems Inc. (1999). Java native Interface. *http://java.sun.com/j2se/1.3/docs/guide/jni*

[Kaelbling and Rosenschein, 1991] Kaelbling, L., and Rosenschein, S. (1991). Action and Planning in Embedded Agents. *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, The MIT press, pp. 35-48.

[Kay, 1990] Kay, A. (1990). User interface: A personal view. *The art of human computer interface design*, Addison-Wesley, pp. 191-207.

[Kay, 1994] Kay, A. (1990). Computer Software. *Scientific American*, 251, pp. 53-59.

[Kitano et al, 1997] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). RoboCup: The Robot World Cup Initiative. *Proceedings of the first international conference on Autonomous agents*, pp. 340-347.

[Kitchel and Elliot, 2004] Kitchel, J., and Elliot, S. (2004). Effects of Template Aging on Facial Recognition. *Annual CERIAS Research Symposium*. (Poster).

[Konolige, 1983] Konolige, K. (1983). A deductive model of belief. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pp. 377-381.

[Lacroix et al, 1994] Lacroix, S., Chatila, R., Fleury, S., Herrb, M., and T. Simeon, T. (1994). Autonomous navigation in outdoor environment : Adaptative approach and experiment. *IEEE International Conference on Robotics and Automation (ICRA94)*, pp. 426-432.

[Laird, 2001] Laird, J. (2001). It knows what you're going to do: adding anticipation to a quakebot. *Proceedings of the fifth international conference on Autonomous agent*, pp. 385–392.

[Lange and Oshima, 1999] Lange, D., and Oshima, M. Seven good reasons for mobile agents. *Communications of the ACM*, 42(3):88-89.

[Lange et al, 1997] Lange, D., Oshima, M., and Kosaka, K. (1977). Aglets: Programming Mobile Agents in Java. *Proceedings of the World-Wide Computing and its Applications (WWCA'97)*, Lecture Note Computer Science, Vol. 1274, pp. 253-266.

[Lee et al, 1994] Lee, J., Huber, M., Durfee, E., and Kenny, P. (1994). UM-PRS: An implementation of the procedural reasoning system for multi robot applications. *Conference on Intelligent Robotics in Field, Factory, Service and Space, CIRFSS94,* pp. 842-849.

[Lenat and Feigenbaum, 1991] Lenat, D., and Feigenbaum, E.(1991). On the thresholds of knowledge. *Artificial Intelligence*, 47, pp. 185-250.

[Lester et al, 1996] James C. Lester, C., Converse, S., Kahler, S., Barlow, S., Stone, B., and Bhogal, R.. (1996). The persona effect: Affective impact of animated pedagogical agents. *Proceedings of the Conference of Human Factors in Computer Systems, (CHI-97),* pp. 359-366

[Lester and Stone, 1997]Lester J., and Stone, B. (1997). Increasing believability in animated pedagogical agents. *Proceedings of the First International Conference on Autonomous Agents*, pp. 16-21.

[Li and Lam, 2002] Li, T., and Lam, K. (2002). Detecting anomalous agents in mobile agent system: a preliminary approach. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 655–656.

[Liu and Wu, 1999] Liu, J., and Wu, J. (1999). Evolutionary robots for collective world modeling. *Proceedings of the Third International Conference on Autonomous Agents*, pp. 48-55.

[Maes, 1994] Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7):31-40.

[Maes, 1997] Maes, P (1997). Software agents tutorial. Conference on Human-Computer Interface (CHI-97). *http://pattie.www.media.mit.edu/people/pattie/CHI97/index.htm*

[Maes & Kozierok, 1993] Maes, P., and Kozierok, R (1993). A learning interface agent for scheduling meetings. *Proceedings of the ACM SIGCHI International Workshop on Intelligent User Interface*s, pp. 81-88.

[Maggi and Sisto, 2003] Maggi, P., and Sisto, R. (2003). A configurable mobile agent data protection protocol. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 851-858.

[Regelous, 2000] Regelous, S. (2000). Massive. *www.massivesoftware.com*

[Mansfield and Wayman, 2000] Mansfield, A., and Wayman, J. (2000). Best Practices in Testing and Reporting Performance of Biometric Devices. NPL Report CMSC 14/02 National Physics Laboratory. *www.npl.co.uk/scientific_software/ publications/biometrics/bestprac_v1_1.pdf*

[Mansfield et al, 2001] Mansfield, T., Kelly, G., Chandler, D., and Kane, J. (2001). Biometric Product Testing Final Report. National Technical Authority for Information Assurance. *www.cesg.gov.uk/site/ast/**biometrics**/media/BiometricTestReportpt1.pdf*

[Marsella et al, 1999] Marsella, S., Adibi, J., Al-Onaizan, Y., Kaminka, G., Muslea, I., and Tambe, M. (1999). *Proceedings of the third annual conference on Autonomous Agents*, pp. 221–227.

[Matsumoto et al, 2002] Matsumoto, T., Matsumoto, H., Yamada, K., and Hoshino, S. (2002). Impact of Artificial 'Gummy' Fingers on Fingerprint Systems. *Proceedings of SPIE*, vol. 4677.

[McCarthy, 1960] McCarthy, J. (1960). Recursive functions of symbolic expressions and their computation by machine. *Communications of the ACM*, 3(4):184-195.

[McCarthy, 1978] McCarthy, J. (1978). Ascribing mental qualities to machines. *Technical report*, Stanford AI Lab.

[Menczer,2003] Menczer, F. (2003). Complementing search engines with online Web mining agents. *Decision Support Systems*, 35(2):195-212.

[Middleton, 2001] Stuart E. Middleton, S. (2001). Interface agents: A review of the field *http://www.informatik.uni-trier.de/~ley/db/journals/corr/corr0203.html#cs-MA-0203012*

[Minar et al, 1999]N. Minar, N., Gray, M., Roup, o., Krikorian, R., and Maes, P. (1999). Hive: Distributed agents for networking things. *Proceedings of the First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents ASA/MA'99*, pp. 118-129.

[Mitchell et al, 1994] Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994). Experience with a Learning Personal Assistant. *Communications of the ACM*, 37(7):81-91.

[Mladenić, 1999] Dunja Mladenić, D. (1999). Text-learning and related intelligent agents: A survey. *IEEE Intelligent Systems*, 14(4):44-54.

[Moore, 1985] Moore, R. (1985). A formal theory of knowledge and action. *Formal Theories of the Commonsense World*, pp.319-258.

[Mora et al, 1999] Mora, M., Lopes, J., Viccari, R., and Coelho, H. (1999). BDI models and systems: Reducing the gap. *Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL-98),* LNCS 1555.

[Moulin and Chaib-draa, 1996] Moulin, B. and Chaib-draa, B. (1996). An Overview of Distributed Artificial Intelligence. *In Foundations of Distributed Artificial Intelligence, Sixth-Generation Computer Technology Series*, New York: John Wiley & Sons, pp. 3-55.

[Muller, 1994] Muller, J. (1994). A conceptual model of agent interaction. *Draft Proceedings of the Second International Working Conference on Cooperating Knowledge Based Systems (CKBS-94)*, pp. 389-404.

[Muller, 1996] Muller, J. (1996). The Design of Intelligent Agents. A layered Approach. *Lectures notes in Artificial Intelligence*, volume 1177. Springer.

[Muller and Pischel, 1993] Muller, J., Pischel, M. (1993). The Agent Architecture InteRRap: Concept and Application. *Technical Report RR-93-26.*

[Muller et al, 1995] Muller, J., Pishel, M, and Thiel, M. (1995). Modelling Reactive Behaviour in Vertically Layered Agent Architectures. *Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages*, pp. 709-713.

[Mulken et al, 1998] Mulken, S., Andr, E., and J. Muller J. (1998).The persona effect: How substantial is it? *Proceedings of HCI on People and Computers XIII*, pp. 53-66.

[Muscetta et al, 2002] Muscetta, N., Dorais, G., Fry, C., Levinson, R., and Plaunt, C. (2002). Idea: Planning at the core of autonomous reactive agents. *Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space.*

[Musliner et al, 1993] Musliner D., Durfee E., and Shin K. (1993). CIRCA: A Co-operative Intelligent Real-time Control Architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6):1561-1574.

[Myers et al, 1991] Brad A. Myers, B., Cypher, A., Maulsby, D., Smith, D., Shneiderman, B. (1991). Demonstrational interfaces: Coming soon? *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology,* pp. 393–396.

[Naik et al, 1989] Naik, J., Netsch, L., and Doddington, G. (1989). Speaker verification over long distance telephone lines. *Proceedings of the 1989 International Conference on Acoustics, Speech, and Signal Processing*, pp. 524-527.

[Naoyuki and Takata, 2002] Naoyuki, N., Takata, S. (2002). Deduction systems for BDI logics using sequent calculus. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 928-935.

[Newell and Simon, 1976] Newell, A., and Simon, H. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the Association for Computing Machinery*, 19(3):113-126.

[Nwana, 1993] Nwana, S. (1993). Simulating a Children's Playground in ABLE. *Working Report*, Department of Computer Science, Keele University, UK

[Nwana, 1996] S. Nwana, S. (1996). Software agents: An overview. *Knowledge Engineering Review*, 11(3):1-40.

[Nwana et al, 1999] Nwana, S., Ndumu, D., Lee, L., and Collins, J. (1999). ZEUS: A tool-kit for building distributed multi-agent systems. *Applied Artificial Intelligence Journal*, 13(1):129-186.

[Parunak, 2001] Parunak, H. (2001). Representing Social Structures in UML. *Proceedings of the fifth international conference on Autonomous agents*, pp. 100-101.

[Peine and Stolpmann, 1997] H. Peine, H., and Stolpmann, T. (1997). The Architecture of the Ara Platform for Mobile Agents. *First International Workshop on Mobile Agents, MA'97*, pp. 50-61.

[Pell et al, 1997] Pell, B., Bernard, E., Chien, S., Gat., E., Muscettola, N., Nayak, P. Wagner, M., and Williams, B. (1998). An Autonomous Spacecraft Agent Prototype. *Autonomous Robots*, 5(1):29-52.

[van der Putte, 2001] van der Putte, T. (2001). Spoofing biometrics as easy as 1, 2, 3 ?. *Biometrics 2001*, London, UK.

[Podio et al, 2001] Podio, F., Dunn, J., Reinert, L., Tilton, C., O'Gorman, L., Collier, M., Jerde, M., and Wirtz, B. (2001). The Common Biometric Exchange File Format (CBEFF). National Institute of Standards and Technology, *NISTIR 629.*

[OASIS, 1998] Organisation for the Advancement of Structured Information Standards. (1998). Extensible Markup Language (XML) 1.0. *http://www.xml.org*

[Odell et al, 2000] Odell, J., Parunak, V., and Bauer, B. (2000). Extending UML for Agents. *Proceedings of the Agent-Oriented Information Systems Workshop*, pp. 3-17.

[Odell, 2002] Odell, J. (2002). Objects and Agents Compared. *Journal of Object Technology*, 1(1):41-53.

[OMG, 1997] Object Management Group.(1997). *http://www.objs.com/agent/*

[Ostergaard et al, 2001] Ostergaard, E., Sukhatme, G., Matari, M. (2001). Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks. *Proceedings of the fifth international conference on Autonomous agents*, pp. 29-30.

[Parthnasardhi et al, 2002] Parthnasardhi, S., Derakhshani, R., Schuckers, S., and Hornak, L. (2002). Initial results of Spoofing and Liveness Detection in Fingerprint Scanners. *Biometrics 2002*, London, UK.

[Penav and Atick, 1996]. Penev, P., Atick, J. (1996) Local Feature Analysis: a General Statistical Theory for Object Representation. *Network: Computation in Neural Systems*, 7(3):477-500.

[Petrie, 1996] Petrie, C. (1996). Agent based engineering, the Web and Intelligence. *IEEE Expert*, 11(6):24-29.

[Ranade &.Rosenfeld, 1993] Ranade, A., and Rosenfeld, A. (1993). Point Pattern Matching by Relaxation," *Pattern Recognition*, 12(2):269-275.

[Rao and Georgeff, 1991] Rao, A., and Georgeff, M.(1991). Modeling rational agents within a BDI-architecture. *International Workshop on Knowledge Representation (KR'91)*, pp. 473-484.

[Rao and Georgeff, 1993] Rao, A., and Georgeff, M.(1993). A model-theoretic approach to the verification of situated reasoning. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pp. 318–324.

[Rao and Georgeff, 1995] Rao, A., and Georgeff, M. (1995). BDI agents: From theory to practice. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95),* pp. 312-319.

[Rashid, 1986] Rashid, R. (1986). From RIG to Accent to Mach: The Evolution of a Network Operating System. *Proceedings of AFIPS 1986 Fall Joint Computer Conference*, pp. 1128-1137.

[Rashid and Robertson, 1981] Rashid, R., and Robertson, G. (1981). Accent: A communication oriented network operating system kernel. *Proceedings of the Eighth ACM Symposium on Operating Systems Principles*, pp. 64-75.

[Roorda et al, 2002] Roorda, J., Hoek, W., and Meyer, J. (2002). Iterated belief change in multi-agent systems. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 889-896.

[Rosenschein and Kaelbling, 1986] S. Rosenschein, S., and Kaelbling, L. (1986). The synthesis of digital machines with provable epistemic properties. *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 83-98.

[Rousseau and Hayes-Roth, 1998] Rousseau, D., and Hayes-Roth, B. (1996). A social-psychological model for synthetic actors. *Proceedings of the second international conference on autonomous agents*, pp. 165-172.

[RSA, 2004] Are Passwords Really Free? A Closer Look at the Hidden Costs of Password Security. White Paper, RSA security, *www.rsasecurity.com.*

[Rus et al, 1997] Rus, D., Gray, R., Kotz, D. (1997). Transportable information agents. *International Conference on Autonomous Agents*, pp. 228-236.

[Russell and Norvig, 1995] Russell, S., and Norvig, P. (1995). Artificial Intelligence a modern approach. *Prentice Hall*, pp.31.

[Schäfer, 2001] Schäfer, R. (2001). Rules for Using Multi-Attribute Utility Theory for Estimating a User's Interests. *Proceedings of the Ninth GI-Workshop:ABIS*, pp. 21-28.

[Schild, 2000] Schild, K. (2000). On the relationship between BDI logics and standard logics of concurrency. *Autonomous Agents and Multi agent Systems*, 3, pp. 259-283.
[Searle, 1969] Searle, J. (1969). Speech Acts: An Essay in the Philosophy of Language. *Cambridge University Press*.

[Secugen, 2001] SecuGen PC Peripherals SDK Manual 01.23.01.pdf

[Seel, 1989] Seel, N. (1989). *Agent Theories and Architectures*. PhD thesis, Surrey University.

[Sen et al, 1997] Sen, S., Haynes, T., Arora,N. (1997). Satisfying user preferences while negotiating meetings. *International Journal of Human-Computer Studies*, 47:407-427

[Simon, 1969]. Simon, H. (1969). Sciences of the artificial .*MIT Press*. pp. 63-66.

[Singh, 1990] M. P. Singh. Towards a theory of situated know-how. *Proceedings of the Ninth European Conference on Artificial Intelligence (ECAI-90)*, pp. 604-609.

[Shoham, 1990] Shoham, Y. (1990). Agent-oriented programming. *Technical Report STAN-CS-1335-90*, Computer Science Department, Stanford University, Stanford, CA.

[Shoham, 1993] Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60(1):51-92.

[Shoham, 1991] Shoham, Y. (1991). AGENT0: An Agent-Oriented Language and its Interpreter. *Proceedings of the Ninth National Conference on Artificial Intelligence*, pp. 704-709.

[Sproull et al, 1996] Sproull L., Subramani, M., Kiesler, S., Walker, J., and Waters, K. (1996). When the interface is a face. *Human-Computer Interaction*, 11, pp. 97–124.

[SSM-HESY, 2003] (2003). SSM-HESY – the mobile phone generation of the future. *http://www.hesy.de*

[Suchman, 1987] Suchman, L. (1987). Plans and situated actions: the problem of human-machine communication. *Cambridge University Press.*

[Thomas, 1993] Thomas, S. (1993). PLACA, an Agent Oriented Programming Language. PhD thesis, Computer Science Department, Stanford University, Stanford, CA.

[Titmus et al, 1996] Titmuss, R., Winter, C., and Crabtree, B. (1996). Agents, Mobility & Multimedia Information. *Proceedings the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 96)*, pp. 693-708.

[Thalheim et al, 2002] Thalheim, L., Krissler, J., Ziegler, P. (2002). Body Check: Biometrics Defeated. Extreme Tech, *http://www.extremetech.com/article2/0,3973,13919,00.asp*

[Turk and Pentland, 1991] Turk, M., and Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71-86.

[UML, 1997] Object Management Group. (1997). Unified Modeling Language. *http://www.uml.org/*

[Uludag et al, 2003] Uludag, U., Ross, A., and Jain, A. (2003). Biometric template selection and update: a case study in fingerprints. *Pattern Recognition*, 37: 1533 – 1542.

[Verlinde and Acheroy, 2000] Verlinde, P., and Acheroy, M. (2000). A contribution to multi-modal identity verification using decision fusion. *Proceedings of PROMOPTICA*, pp. 1-16.

[Verlinde et al, 2000] Verlinde, P., Chollet, G., and Acheroy, M. (2000). Multimodal identity verification using expert fusion. *Information Fusion*, 1:17 – 33.

[VeriVoice, 2000] VeriVoice. (2000). *http://www.verivoice.com/*

[Veloso et al, 1998] Veloso, M., Stone, P., and Han, K. (1998). The CMUnited-97 robotic soccer team: perception and multiagent control. *Proceedings of the second international conference on Autonomous agents*, pp. 78-85.

[Walker et al, 1994] Walker, J., Sproull, L., and Subramani, R. (1994). Using a Human Face in an Interface. *Proceedings of ACM Conference on Human Factors in Computing Systems, (CHI-94)*, pp. 85-91.

[Wayner 1995] Wayner, P. (1995a).Free Agents, *Byte*, March, pp. 105-114.

[Weiß, 2000] Weiß, G. (2000). Planning and Learning Together. *Proceedings of the Fourth International Conference on Autonomous Agents*, pp. 102-103.

[White, 1994] J.E. White, J.(1994). Telescript Technology: the Foundation for the Electronic Marketplace, *General Magic White Paper*, http://www.genmagic.com

[Wiegland and O'Brien, 1996] Wiegland, M., and O'Brien, M. (1996). Adept : An application viewpoint. *Proceedings of Intelligent Systems Integration Programme Symposium*.

[Winterfield and Edwards, 1998] Winterfield, D., and Edwards, W. (1998). Decision Analysis and Behavioral Research. *Cambridge University Press.*

[Woodward, 1997 Woodward, J. (1997). Biometrics: Privacy's Foe or Privacy's Friend? *Proceedings of the IEEE*, 85(9):1480-1492.

[Wooldridge, 1995] Wooldridge, M. (1995). Conceptualising and Developing Agents. *Proceedings of the UNICOM Seminar on Agent Software,* pp. 40-54.

[Wooldridge, 1999] Wooldridge, M. (1999). Intelligent Agents. Multiagent systems: A modern approach to distributed artificial intelligence. *Cambridge University Press,* pp.27-77.

[Wooldridge and Jennings, 1994] Wooldridge, M., and Jennings, N. (1994*)*. Agent Theories, Architectures, and Languages: A Survey. *Proceedings of the Workshop on Agent Theories, Architectures, and Languages, ECAI-94,* pp.1-39.

[Wooldridge and Jennings, 1995] Wooldridge, M., and Jennings, N. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115-152.

[Wooldridge et al, 2000] Wooldridge, M. Jennings, N., Kinny, D. (2000). The Gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3): 285-312.

[Yamauchi, 1998] Yamauchi, B. (1998). Frontier-based exploration using multiple robots. *Proceedings of the second international conference on Autonomous agents*, pp. 47-53.

List of publications

Intelligent Agents for the Management of Complexity in Multimodal Biometrics, F Deravi, M C Fairhurst, R M Guest, N Mavity and A D M Canuto, Int. J. Universal Access in the Information Society. vol 2, issue 4, November 2003, pp. 293-304.

Intelligent Management of Multimodal Biometric Transactions, M C Fairhurst, F Deravi, N J Mavity, J George, KES'2003, Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems, University of Oxford, United Kingdom, 3, 4 & 5 September 2003.

Adaptive User Agents for Multimodal Biometric Interfaces, N J Mavity, M C Fairhurst and F Deravi, RASC 2002, 4th International Conference on Recent Advances in Soft Computing, Nottingham, United Kingdom, pp 72-77, 12 & 13 December 2002

Adaptive User Agents for Biometric Applications, N J Mavity, M C Fairhurst, F Deravi, COST 275 Workshop - The Advent of Biometrics on the Internet, Fondazione Ugo Bordoni, Rome, Italy, pp 109-113, November 7-8, 2002

Design of Multimodal Biometric Systems for Universal Authentication and Access Control, F Deravi, M C Fairhurst, R M Guest, N Mavity, A D M Canuto, Proceedings of the Second International Workshop on Information Security Applications, WISA 2001, Seoul, Korea, pp 289-295, September 13-14, 2001

Face Verification Competition on the XM2VTS Database. K Messer, J Kittler, M Sadeghi, S Marcel, C Marcel, S Bengio, F Cardinaux, C Sanderson, J Czyz, L. Vandendorpe, S Srisuk, M Petrou, W Kurutach, A Kadyrov, R Paredes, B Kepenekci, F Tek, G Akar, F Deravi, N Mavity.Lecture Notes in Computer Science, Vol 2688, August 2003, pp. 964 – 974

Appendix B

This appendix contains primarily supplementary design documentation for the IAMBIC project. Also presented are screenshots for the various applications described in this thesis.

The GAIA role schema for the functional components of the IAMBIC system are illustrated in the following tables.

Role Schema: *Interface Manager*
Description: The interface manager handles the interaction with the user of the system. The manager will initiate the enrolment and verification process handled by the separate biometric module.
Protocols and Activities InquireSecurity.AcquireSamples.VerifyUser.PassBioData **Permissions** Reads enrolstatus //true or false, enroll the user or not. Supplied secuirtylevel// the level of security corresponding to the file requested Supplied biodata //the biometric verification scores Supplied environdata // associated environmental information Generates filereuqest // some information about the file that needs to be accessed. Generates enrolrequest // a request to be sent to the biometric interface Generates verifyrequest// the level of verification required for the requested file Generates userinfo // class of user and ID Generates biodatavalid// a message to indicate the biometric data is ready
Responsibilities **Liveness:** InterfaceManager = (AcquireSamples | InquireSecurity.VerifyUser.PassBioData) $^{\omega}$ **Safety:** • Enrolstatus = true → enrolrequest = true. • logonOK = true.

Table B.1 Interface manager GAIA role schema

Role Schema: *Access Manager*

Description: The access manager controls the requests to the server for the release of the information requested by the user. It is responsible for the issuing of the information to the directory agent to request the location of the requested file, then the agent will negotiate with the server agent to determine the level of security needed to release the file to the user.

Protocols and Activities

SearchDB. RequestSecurityLevel. NegotiateSecurityLevel. AwaitAccessRequest. EncryptData. DecryptData

Permissions

Supplied confdata // the overall confidence score from the fusion engine

Supplied dblocation//the location(s) of the requested data

Supplied filerequest// information about the file passed from the interface manager

Supplied secuirtylevel// level of security for file generated from server agent

Supplied userinfo// class of user and ID

Responsibilities

Liveness:

(AwaitAccessRequest.SearchDB.EncryptData.RequestSecurityLevel.NegotiateSecurityLevel.DecryptData) $^{\acute{\omega}}$

Safety

- confdata = valid
- userinfo = true.

Table B.2 Access manager GAIA role schema

Role Schema: *Directory Manager*

Description: The directory manager a dynamic list of the locations of the data sources available to the system. The manager also monitors the state of the communications network to provide the optimum transport channel for the requested information at time the user requires it.

Protocols and Activities

RequestDBsources.CalcBestLocation.CalcNetworkStatus.AwaitEnquiry.CheckService

Permissions

Reads DBSources //the sources of the information currently available

Supplied filerequest //information about the file being requested.

Generates DBLocation // the location(s) of the requested file

Responsibilities

Liveness:

(AwaitInquiry. RequestDBsources.CheckService. CalcNetworkStatus. CalcBestLocation)ώ

Safety

- filerequest = true.

Table B.3 Directory manager GAIA role schema

Role Schema: *Fusion Manager*

Description: This agent is responsible for the fusion of the biometric samples taken from the user. The technique of fusion will depend on the confidence of each sample as well as environmental factors that may influence the acquisition of a particular sample.

Protocols and Activities

AwaitData, AnalyseBioData, ReturnBioData

Permissions

Supplied biodatavalid // a message to indicate that the biometric data is ready

Supplied biodata// the biometric verification scores

Supplied environdata // associated environmental information.

Generates confscore // the confidence score produced by data fusion

Responsibilites

Liveness

FusionAgent (AwaitData.AnalyseBioData.ReturnBioData) $^{\omega}$

Safety

- biodatavalid = true.

Table B. 4 Fusion manager GAIA role schema

Utility Calculation

In the following section of this appendix a numerical example of the mechanics of calculating the utility score for a single subject is given.

Enrolment Phase

Security Level	7
Quality Level	40

Table B.5 Enrolment phase example parameters

In order to proceed with this example some figure need to be employed for the values of parameters that are acquired at a subjects enrolment. These are illustrated in Table B.1.

Verification phase

Session Number	Transaction Number	Security Level	Quality Setting
1	1	6	40
2	1	5	40
3	1	6	40
4	1	0	40
	2	4	40
5	1	0	40
	2	5	40

Table B.6 Verification phase example parameters

The values in Table B.6 represent fictitious data recorded during the verification phase in order to illustrate the generation of the figures used to determine system behaviour.

To calculate the normalised session utility the following procedure is followed. The numerical example below will use the first session scores to calculate the normalised session score.

$$SessionUtility = ((0.8 \times (SecLvl \times 0.111) + (0.2 \times Quality)))$$

$$SessionUtility = ((0.8 \times (6 \times 0.111) + (0.2 \times 0.40))) = (0.532) + (0.08) = 0.612$$

The normalisation process takes into account the security levels and the quality setting the user was subject to at the enrolment phase. Therefore, the normalised session score represents how well the subject is performing in relation to the samples they donated during the enrolment phase.

The enrolment normalisation factor is calculated in the same manner as the session utility score, however the values observed during enrolment (Table B.5) are used in the example given above.

$$EnrolmentNormalisationFactor = ((0.8 \times (7 \times 0.111) + (0.2 \times 0.4))) = (0.621) + (0.08) = 0.701$$

A figure for normalised transaction utility can be generated for verification session 1 using the equation below.

$$NormalisedTransactionUtility = \frac{SessionUtility}{EnrolmentNormalisationFactor}$$

$$NormalisedTransactionUtility = \frac{0.612}{0.701} = 0.87$$

If a user has failed any verification attempts during a session then the score produced for the failed transaction is zero. When it comes to calculating a score for that session then the following equation is applied.

$$NormalisedSessionUtility = \frac{NormalisedTransactionUtility1 + NormalisedTransactionUtility \cdot \cdot n}{NumberofTransactions}$$

Therefore the session score now reflects the difficulty the subject experienced in donating samples during that particular session. Normalised session scores for the example shown above are depicted in Table B.7.

Periodically these normalised session scores are used to calculate the behaviour band the subject is residing in. The calculation is based upon two parameters as introduced in Section 6.6.5. The first, determines how many sessions have to be completed before this initial examination occurs. The second, determines how many sessions occur before further examination of the subjects data occurs. This process is illustrated below using the data shown in Table B.7.

Session Number	Normalised Session score	Mined Session Number	Mined Session Score
1	0.612		
2	0.524	1	0.568
3	0.612	2	0.546
4	0.217	3	0.381
5	0.262	4	0.321

Table B.7 Example session scores

After the initial number of sessions has occurred (in this case two), an average normalised score for these two sessions is calculated. This figure represents the subjects first mined session score. This figure is used to place the user in one of the four available behaviour bands.

In this example the periodic interval before re-inspection is one session. Therefore, the next mined session score is calculated by averaging the first mined session score and the relevant normalised session score. This process continues over the lifetime of the application.

AUML Diagrams for the IAMBIC system

The next section of this appendix illustrates the IAMBIC system components specified in terms of various descriptive AUML constructs.

Interface Agent

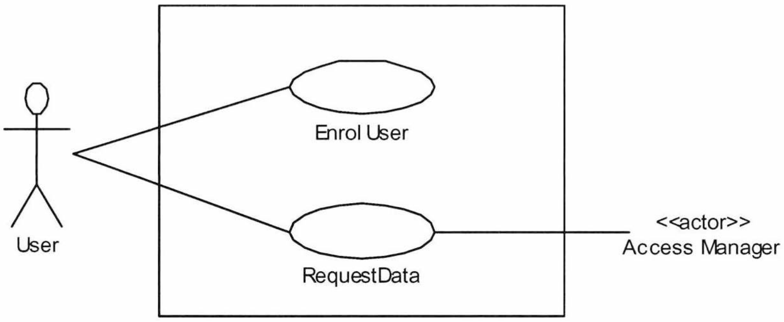


Figure B.1 Interface agent use case

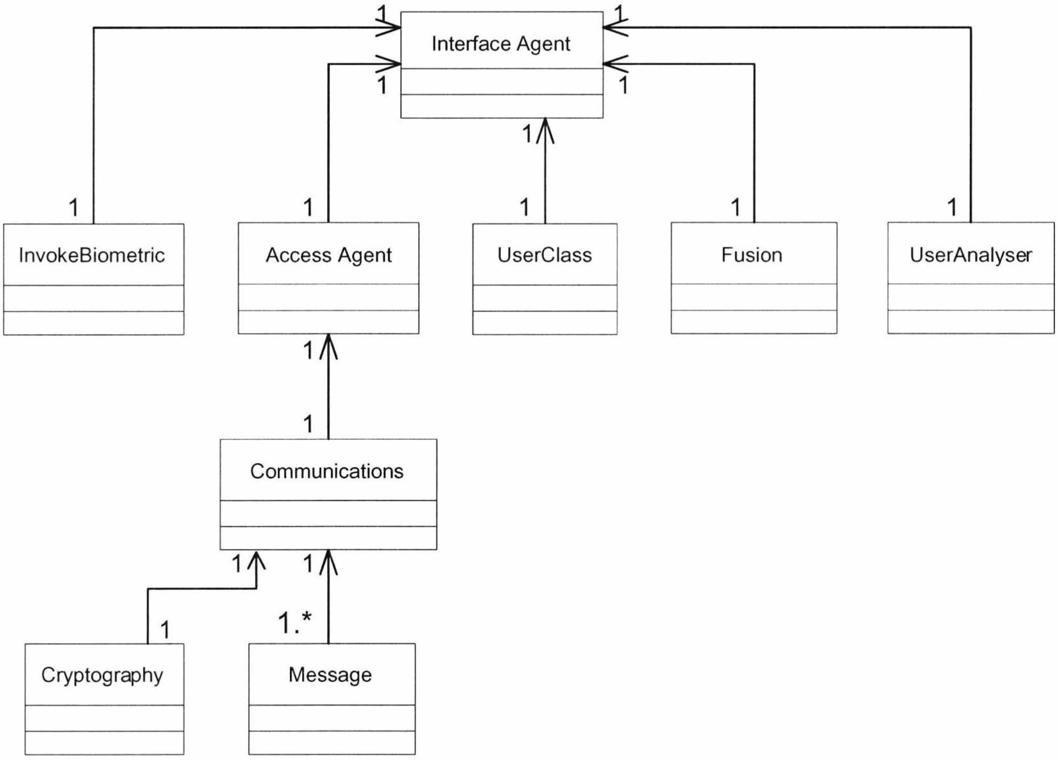


Figure B.2 Interface agent class diagram

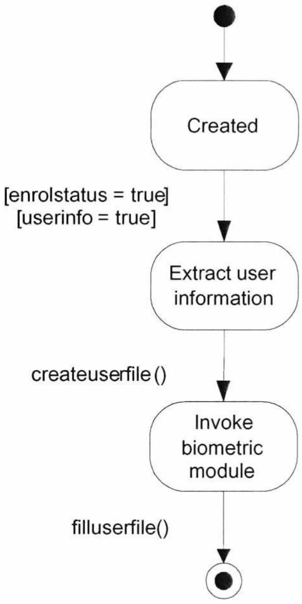


Figure B.3 Enrol user state chart

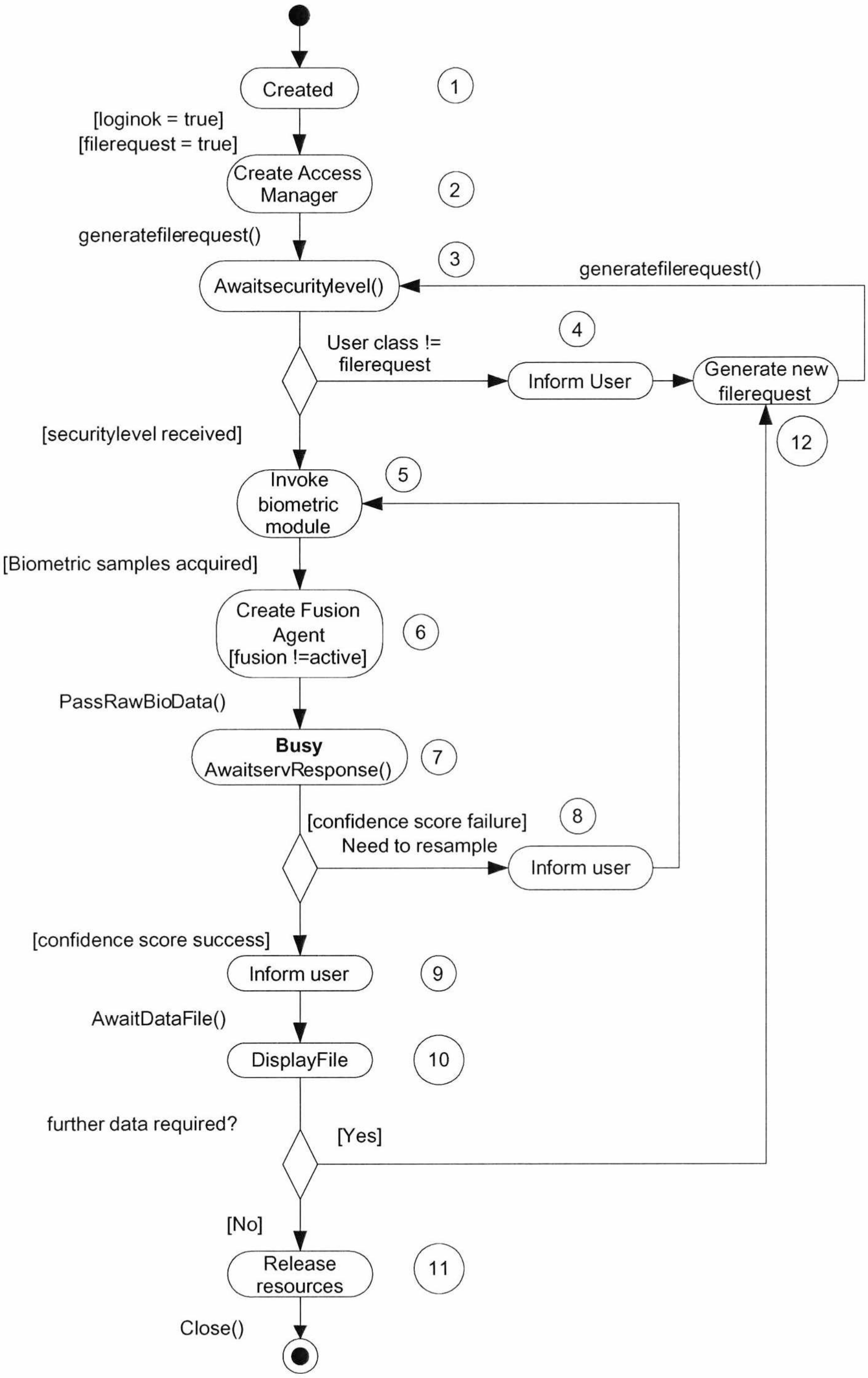


Figure B.4 Request data state chart

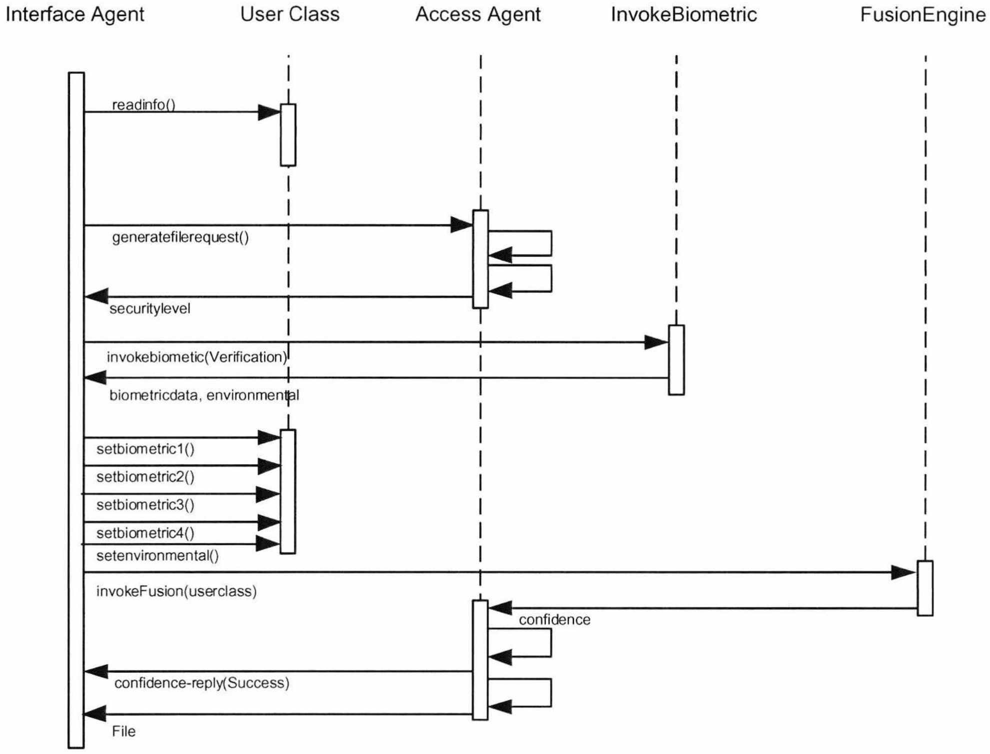


Figure B.5 Interaction diagram for file retrieval

Fusion agent

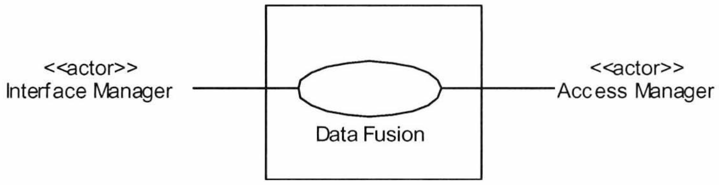


Figure B.6 Fusion Agent use case

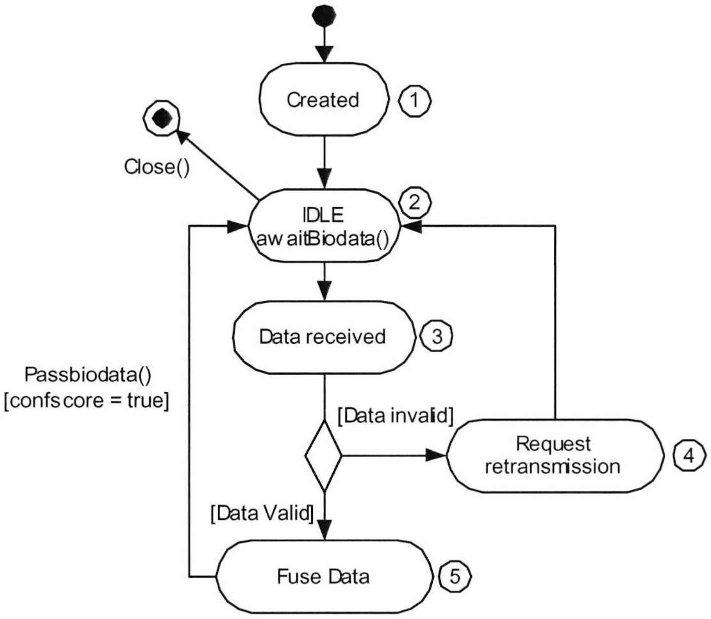


Figure B. 7 Fusion Agent state chart

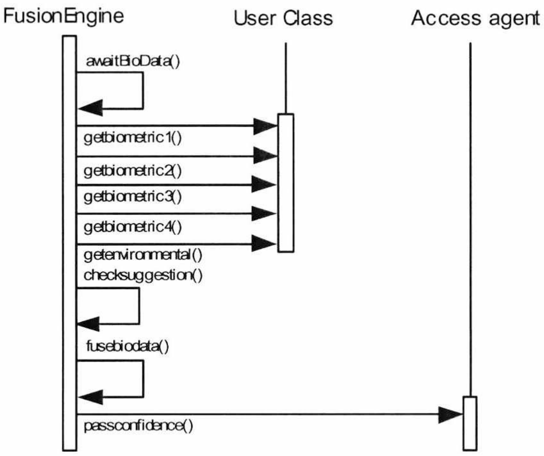


Figure B.8 Fusion agent interaction diagram

Access Agent

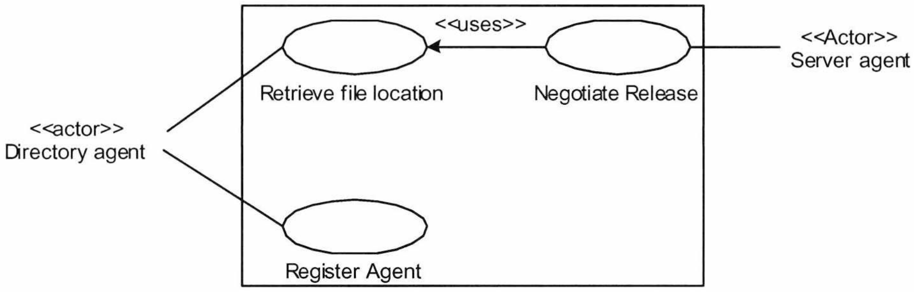


Figure B.9 Access agent use case

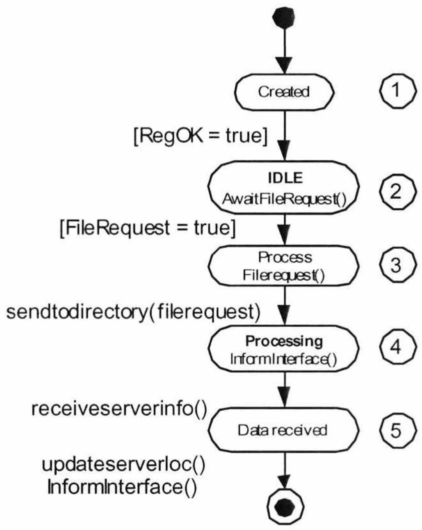


Figure B.10 Retrieve file location state chart

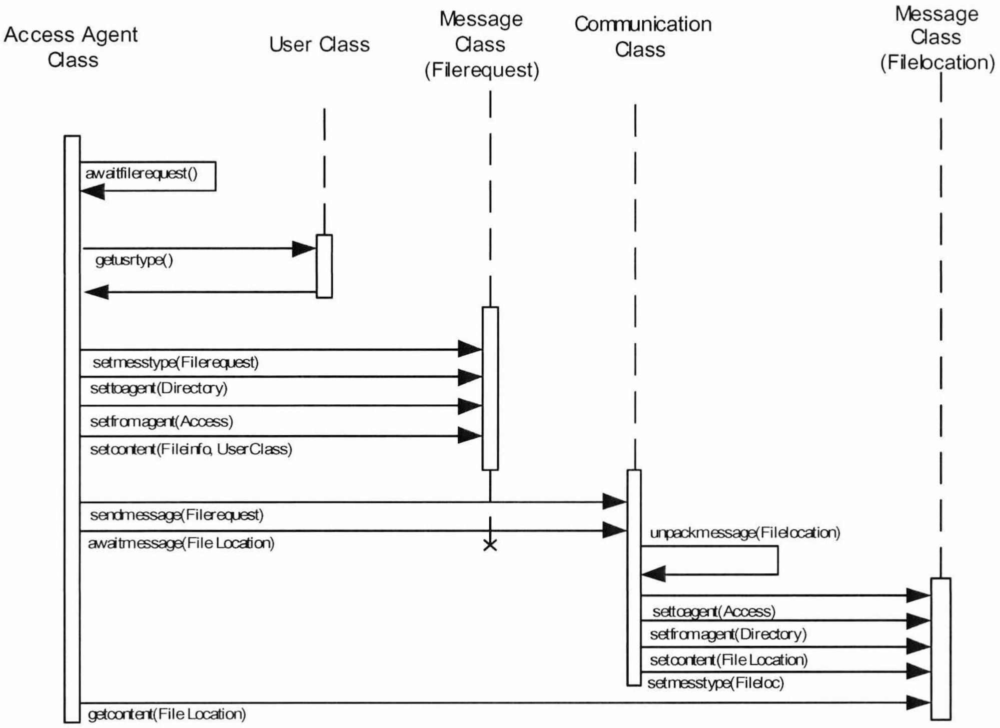


Figure B.11 Retrieve file location interaction diagram

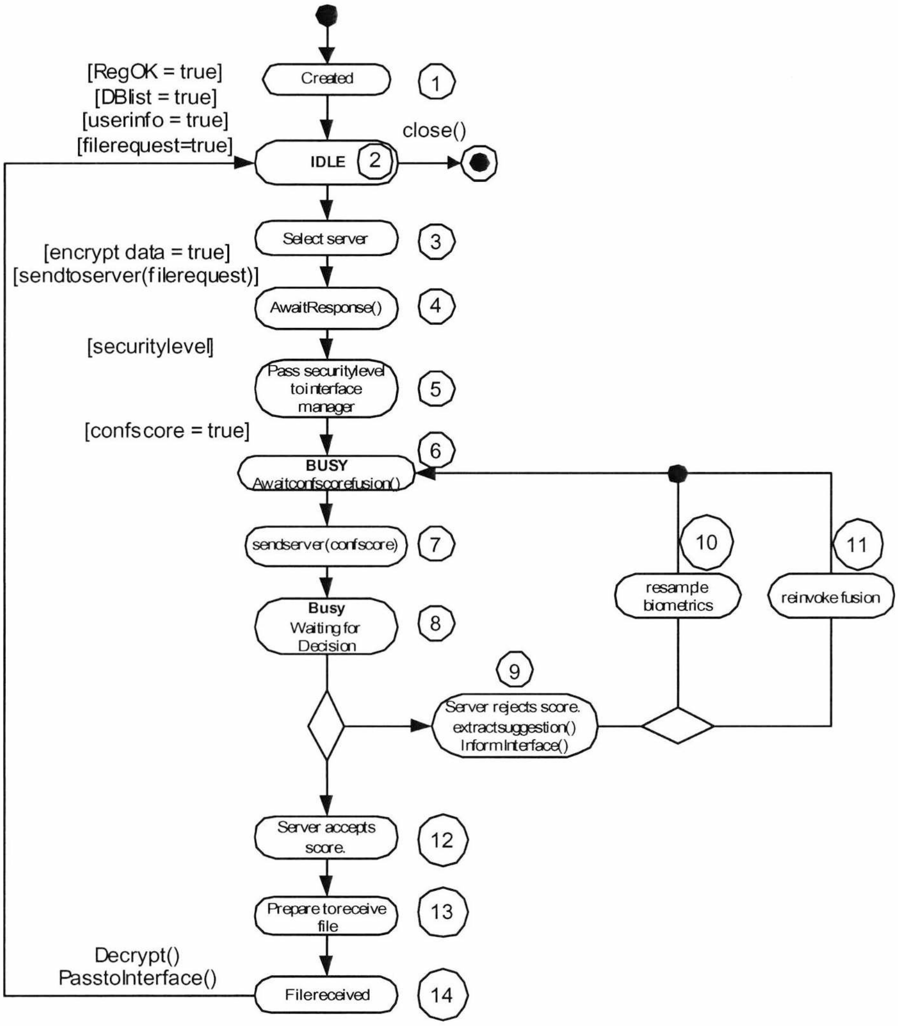


Figure B.12 Negotiate release state chart

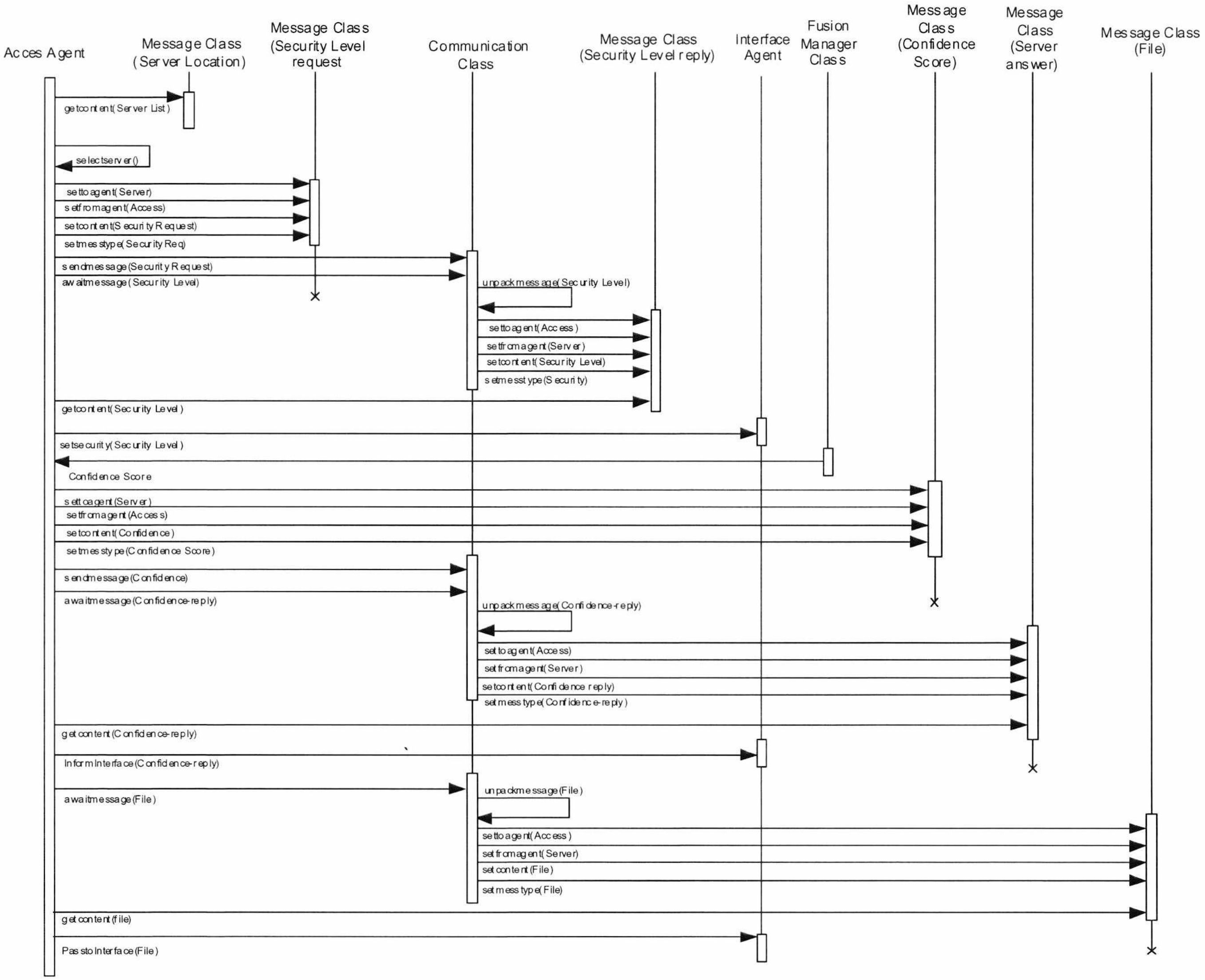

Acces Agent
Message Class (Server Location)
Message Class (Security Level request
Communication Class
Message Class (Security Level reply)
Interface Agent
Fusion Manager Class
Message Class (Confidence Score)
Message Class (Server answer)
Message Class (File)
getcontent(Server List)
selectserver()
settoagent(Server)
setfromagent(Access)
setcontent(Security Request)
setmesstype(Security Req)
sendmessage(Security Request)
awaitmessage(Security Level)
unpackmessage(Security Level)
settoagent(Access)
setfromagent(Server)
setcontent(Security Level)
setmesstype(Security)
getcontent(Security Level)
setsecurity(Security Level)
Confidence Score
settoagent(Server)
setfromagent(Access)
setcontent(Confidence)
setmesstype(Confidence Score)
sendmessage(Confidence)
awaitmessage(Confidence-reply)
unpackmessage(Confidence-reply)
settoagent(Access)
setfromagent(Server)
setcontent(Confidence reply)
setmesstype(Confidence-reply)
getcontent(Confidence-reply)
InformInterface(Confidence-reply)
awaitmessage(File)
unpackmessage(File)
settoagent(Access)
setfromagent(Server)
setcontent(File)
setmesstype(File)
getcontent(file)
PasstoInterface(File)

Figure B.13 Interaction diagram for negotiate release use case

Directory Agent

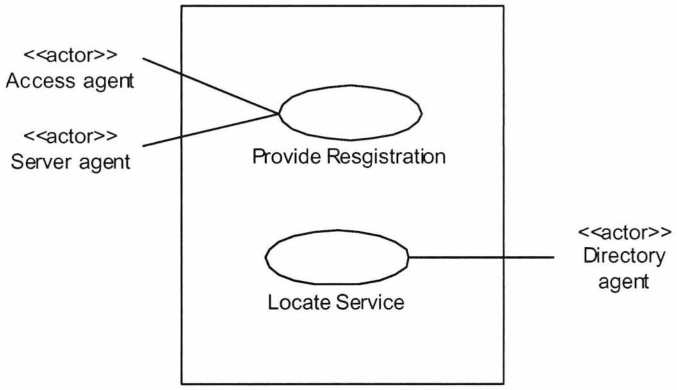


Figure B.14 Use case diagram for directory agent

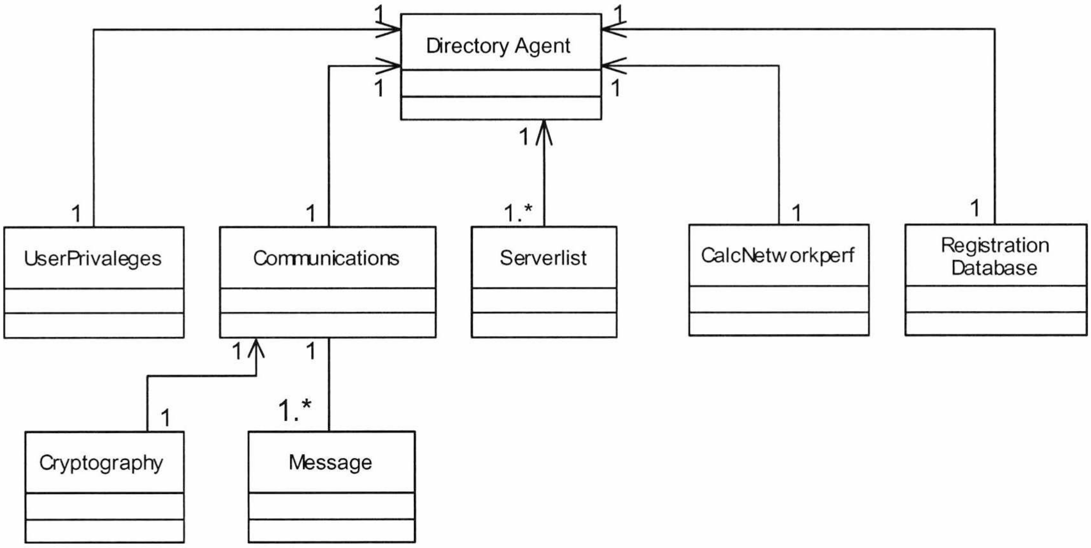


Figure B.15 Directory manager Class diagram

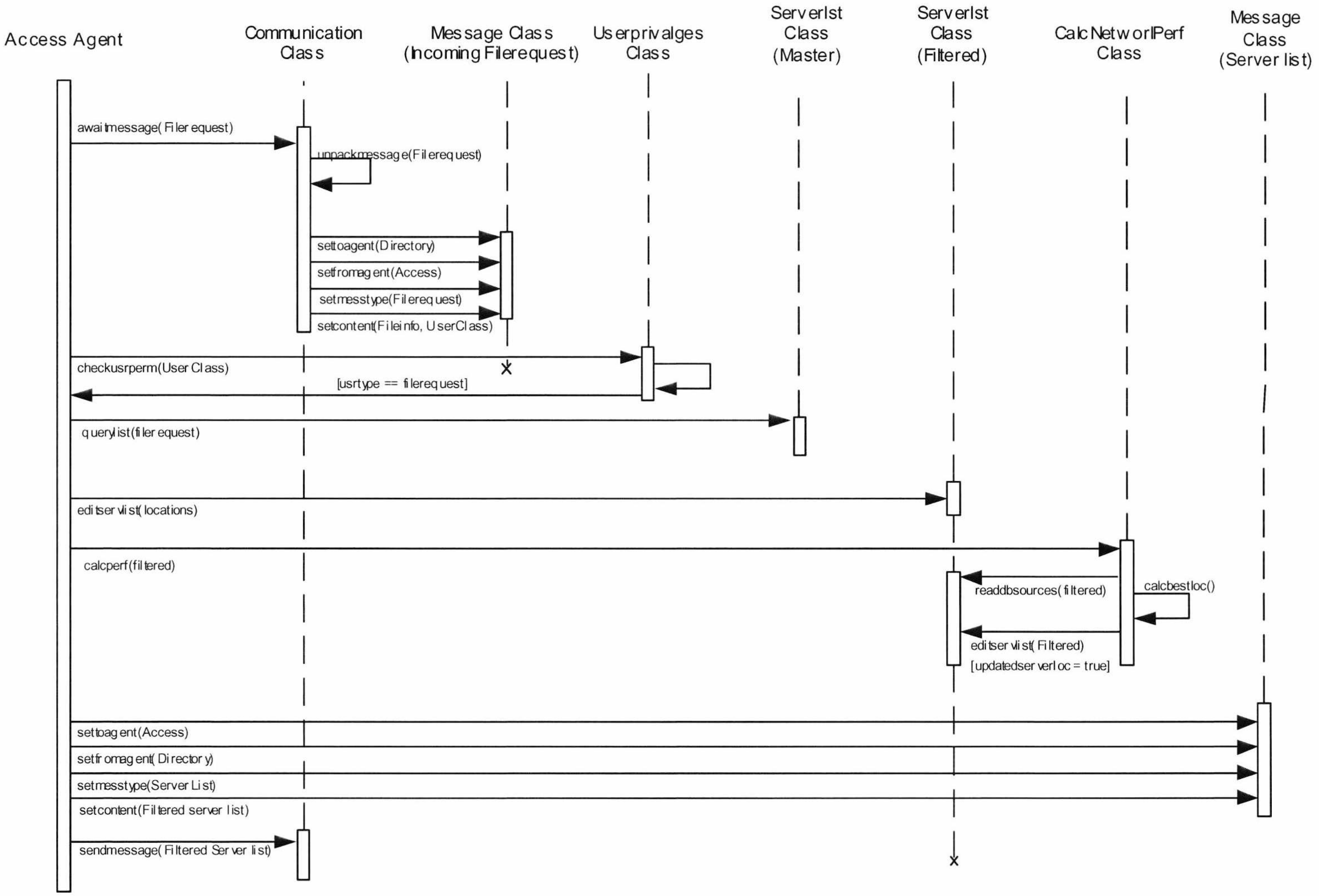


Figure B.16 Locate Service Interaction diagram

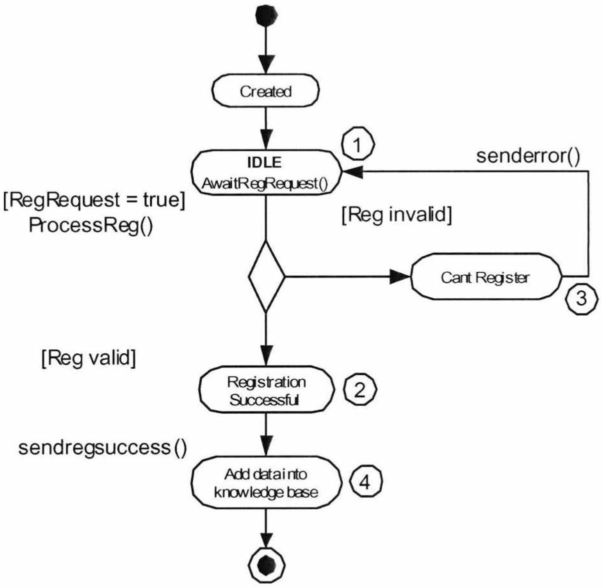


Figure B.17 Provide registration state chart

Server agent

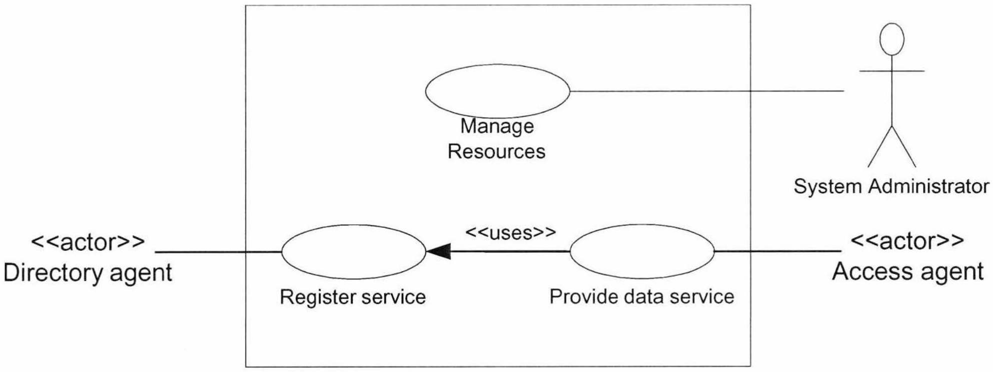


Figure B.18 Use case diagram for server agent

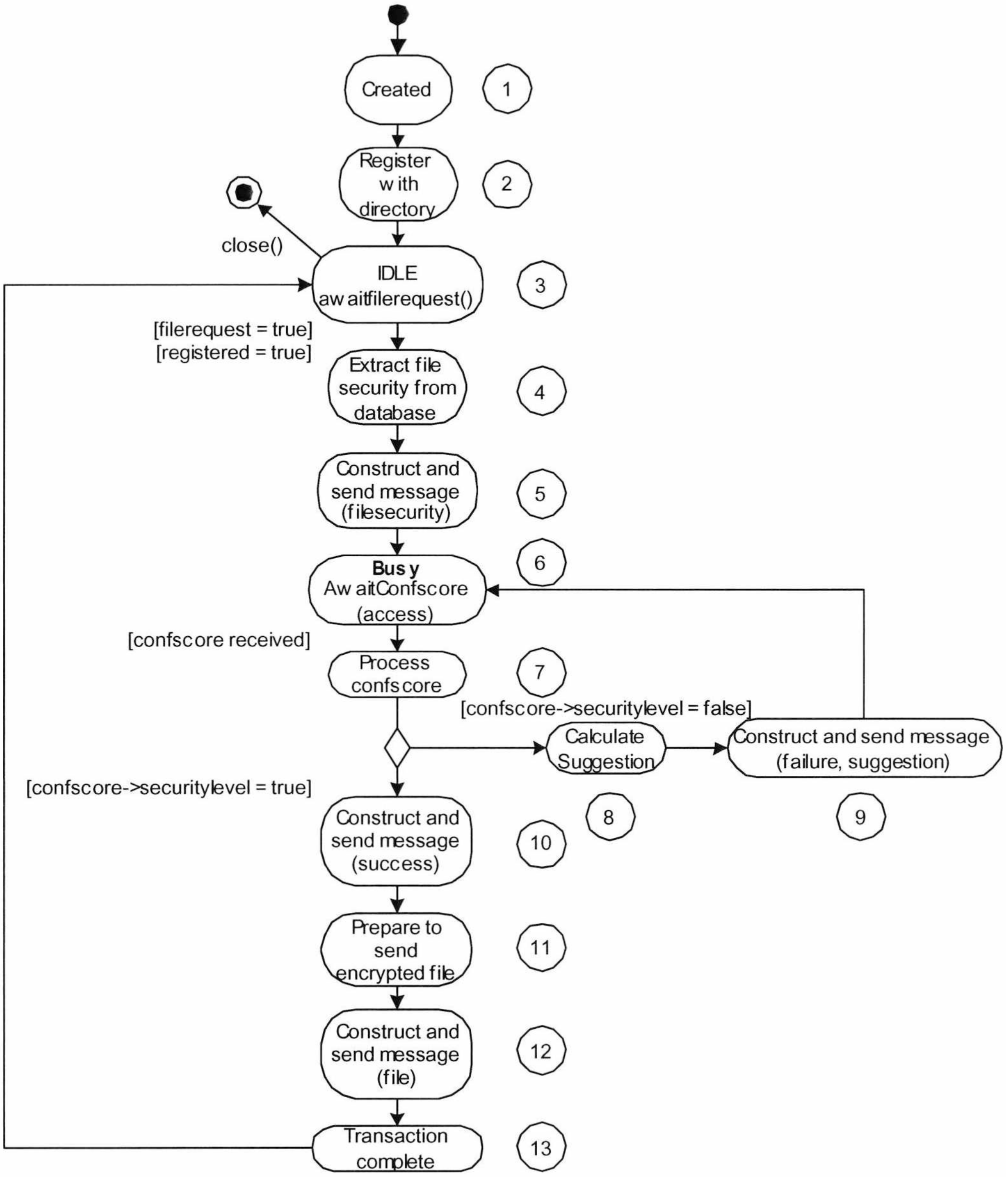

Created
1
Register with directory
2
close()
IDLE
aw aitfilerequest()
3
[filerequest = true]
[registered = true]
Extract file security from database
4
Construct and send message (filesecurity)
5
Busy
Aw aitConfscore (access)
6
[confscore received]
Process confscore
7
[confscore->securitylevel = false]
Calculate Suggestion
Construct and send message (failure, suggestion)
8
9
[confscore->securitylevel = true]
Construct and send message (success)
10
Prepare to send encrypted file
11
Construct and send message (file)
12
Transaction complete
13

Figure B.19 Provide data state chart

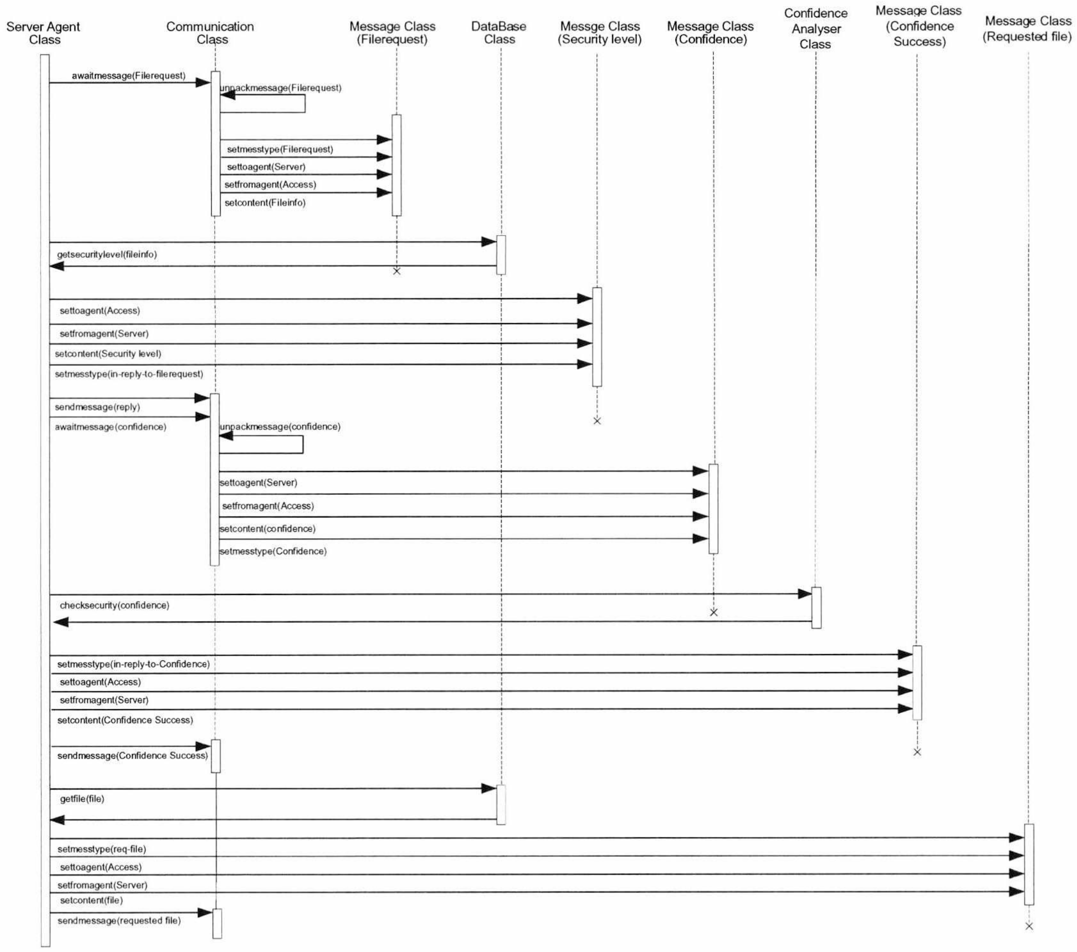

Server Agent Class
Communication Class
Message Class (Filerequest)
DataBase Class
Messge Class (Security level)
Message Class (Confidence)
Confidence Analyser Class
Message Class (Confidence Success)
Message Class (Requested file)
awaitmessage(Filerequest)
unpackmessage(Filerequest)
setmesstype(Filerequest)
settoagent(Server)
setfromagent(Access)
setcontent(Fileinfo)
getsecuritylevel(fileinfo)
settoagent(Access)
setfromagent(Server)
setcontent(Security level)
setmesstype(in-reply-to-filerequest)
sendmessage(reply)
awaitmessage(confidence)
unpackmessage(confidence)
settoagent(Server)
setfromagent(Access)
setcontent(confidence)
setmesstype(Confidence)
checksecurity(confidence)
setmesstype(in-reply-to-Confidence)
settoagent(Access)
setfromagent(Server)
setcontent(Confidence Success)
sendmessage(Confidence Success)
getfile(file)
setmesstype(req-file)
settoagent(Access)
setfromagent(Server)
setcontent(file)
sendmessage(requested file)

Figure B.20 Provide data sequence diagram

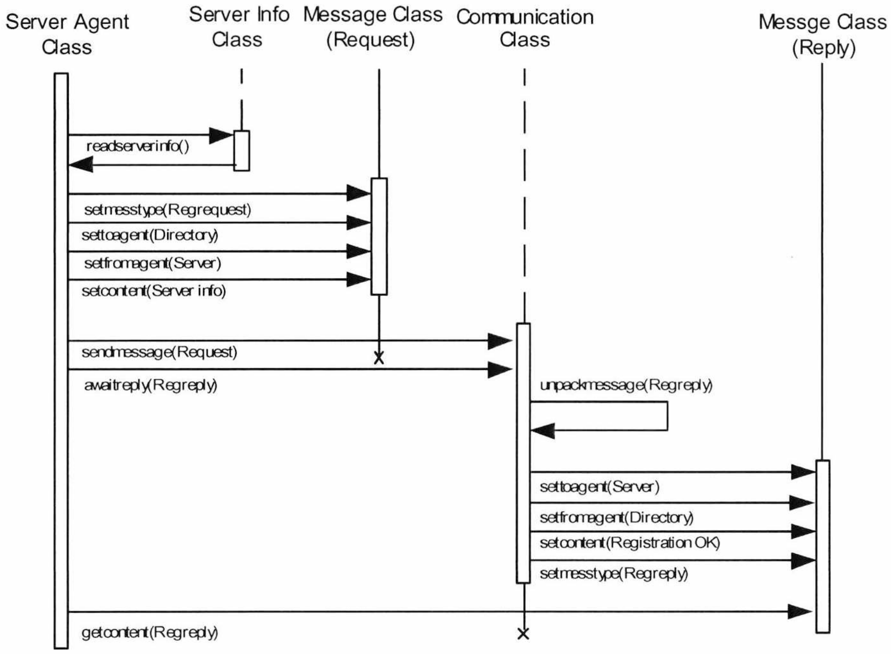


Figure B.21 Register interaction diagram

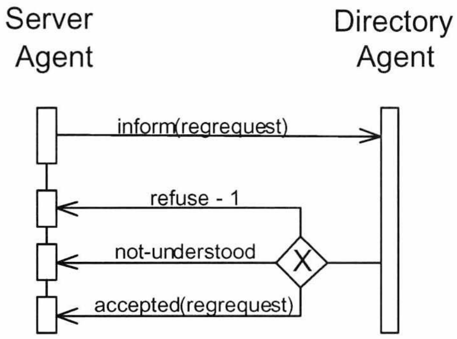


Figure B.22 Registration AIP

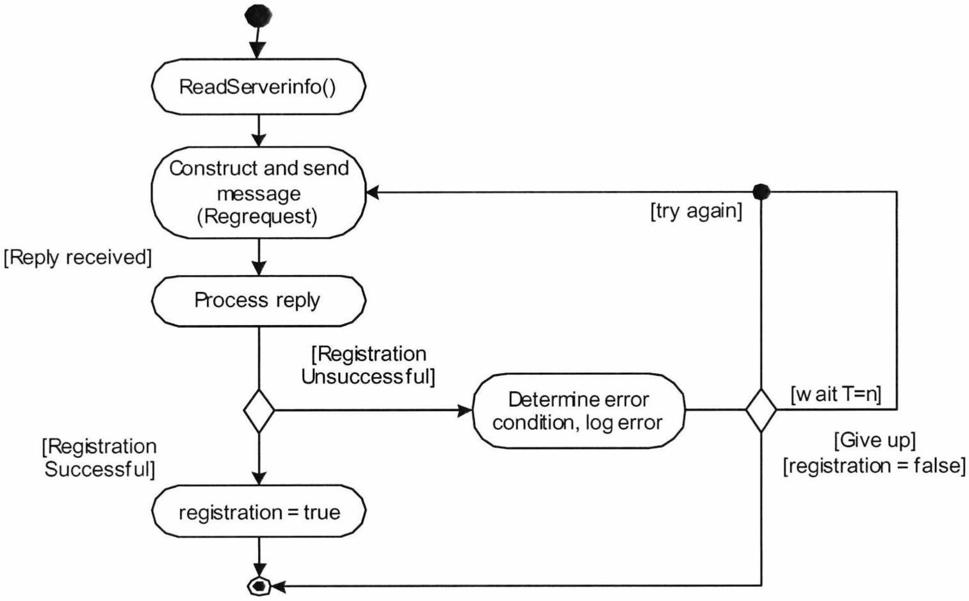

ReadServerinfo()
Construct and send message (Regrequest)
[try again]
[Reply received]
Process reply
[Registration Unsuccessful]
Determine error condition, log error
[w ait T=n]
[Registration Successful]
[Give up]
[registration = false]
registration = true

Figure B.23 Registration state chart

IAMBIC Framework Requirement Capture Document

1. It is important that IAMBIC provides multiple levels of authorisation within the system, both in relation to different categories of user and different levels of confidence necessary to access information;

2. Ease of use is of fundamental importance since users are accustomed to their current systems and may be resistant to the introduction of a new system. This includes a user-friendly interface, fast access and reliable enrolment facilities;

3. Adopting multiple biometrics should provide security if one of the available modalities fails. As added protection, simple PIN numbers can be used as an ultimate fall back measure.

4. Avoid unauthorised use of the system: users can leave computers unattended and an unauthorised person may then access the system. This problem can be avoided either by using continuous monitoring of identity or by including an automatic log out mechanism.

5. Training the users to use the IAMBIC system is very important since users may not be familiar with its techniques and might find it difficult to use biometric measures in order to access information, at least initially;

6. Audit trail: a log file must be kept in order to store information about the users who have accessed the system in a certain period of time (for example, users who have accessed the system in the last month). The name as well as a set of captured biometrics (face, signature or voice) of the users will be recorded to provide traceable evidence of access and non-repudiation;

7. The cost of implementing a new system must be such that the benefits are perceived as off-setting this. An implication of this is likely to be that standard (i.e. "off-the-shelf") components should be used wherever possible;

8. Any system ultimately produced should be thoroughly tested to ensure reliability. This is related to the points made above;

9. A range of biometrics should always be readily available. Not only will this support improved performance and reliability (the fundamental reason for adopting a multimodal architecture) but it is also important so that choices can be made which can guarantee acceptability for any given client base.

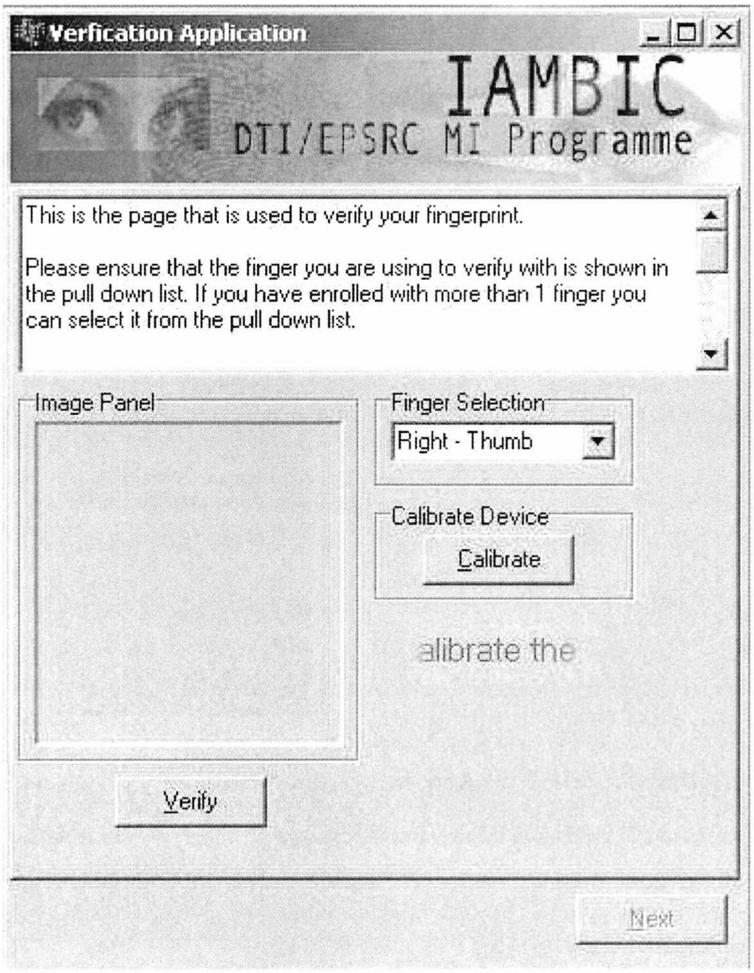
Project Document Type Definition (DTD's)

Shown below is the DTD that was employed in the main biometric application for the storage of the subject's interaction with the biometric devices.

```
<!ELEMENT Transaction (TransDetails, UserDetails,
ClientPCDetails, FingCalDet, BiometricRecord,
UserBioPref)>
<!ELEMENT TransDetails EMPTY>
<!ATTLIST TransDetails
       Transstart   CDATA #REQUIRED
       Transstop    CDATA #REQUIRED
>
<!ELEMENT UserDetails EMPTY>
<!ATTLIST UserDetails
            UserNum CDATA #REQUIRED
            UserLevel    CDATA #REQUIRED
>
<!ELEMENT FingCalDet EMPTY>
<!ATTLIST FingCalDet
          Contrast CDATA #REQUIRED
          Brightness CDATA #REQUIRED
          Gain CDATA #REQUIRED
          LastCal CDATA #REQUIRED
>
<!ELEMENT ClientPCDetails EMPTY>
<!ATTLIST ClientPCDetails
       ClientIP          CDATA #REQUIRED
       ClientPCName  CDATA #REQUIRED
>
<!ELEMENT BiometricRecord (FingerBio+, VoiceBio+,
FaceBio+)>
<!ELEMENT FingerBio EMPTY>
<!ATTLIST FingerBio
            SecurityLevel CDATA #REQUIRED
            ImQual      CDATA #REQUIRED
            Result       CDATA #REQUIRED
            XtraInfo     CDATA #IMPLIED
            VTime        CDATA #REQUIRED
>
<!ELEMENT FaceBio EMPTY>
<!ATTLIST FaceBio
            FaceFindMode CDATA #REQUIRED
            FaceThresh   CDATA #REQUIRED
            Result  CDATA #REQUIRED
            XtraInfo     CDATA #IMPLIED
            VTime       CDATA #REQUIRED
>
<!ELEMENT VoiceBio EMPTY>
<!ATTLIST VoiceBio
            Result  CDATA #REQUIRED
            XtraInfo     CDATA #IMPLIED
            VTime        CDATA #REQUIRED
>
<!ELEMENT Transactions (Transaction+)>
```

The DTD shown below was employed in order to store the various behaviour levels the user had achieved over the duration of the application

```
<!ELEMENT History (NSUS, QInc, AcqTime)>
<!ELEMENT NSUS EMPTY>
<!ATTLIST NSUS
       NSUSTStamp CDATA #REQUIRED
       NSUSVal CDATA #REQUIRED
       BEHAVLVL CDATA #REQUIRED
       SessMined    CDATA #REQUIRED
>
<!ELEMENT QInc EMPTY>
<!ATTLIST QInc
       QIncVal CDATA #REQUIRED
       QIncTStamp CDATA #REQUIRED
>
<!ELEMENT AcqTime EMPTY>
<!ATTLIST AcqTime
       ACqTimeVal CDATA #REQUIRED
>
<!ELEMENT Histories (History+)>
```



This is the page that is used to verify your fingerprint.

Please ensure that the finger you are using to verify with is shown in the pull down list. If you have enrolled with more than 1 finger you can select it from the pull down list.

Figure B.24 Interface showing calibration message

IAMBIC System Survey Questionnaire

Dear Volunteer

As one of the objectives of the IAMBIC trials we would like to collect your views about your experience with the system. Please could you answer the following questions:-

Question 1

Did you find the IAMBIC system easy to use? Please answer "YES" or "NO".
Comments:

....

....

Question 2

Do you think the multi modal aspect of the IAMBIC System provides a secure method of identity authentication Please answer "YES" or "NO". Comments:

....

....

Question 3

Did you find the user interface helpful in the assistance given to donate the biometric samples ? Please answer "YES" or "NO". Comments:

....

....

Question 4

Would you like to see a multi modal verification system employed as a means to authenticate your identity ?

Please answer "YES" or "NO". Comments:

....

....

Patient Data file

The following section illustrates to data file that is generated from the data recorded by the heart monitoring device. This file can be retrieved by the relevant patient, or by the physician.

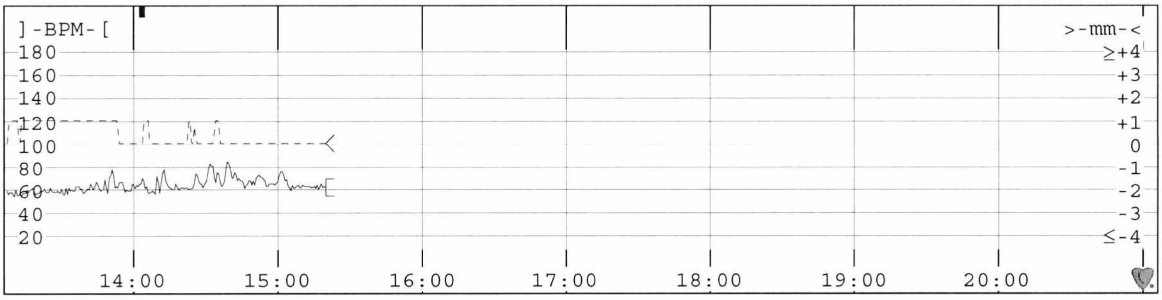
```
Cardionetics C.Net2000+  7.0  Report        13:06 Fri
04/10/2002    Page 1 of 4
----------------------------------------------------------------
----------------
|Patient    ATest                        |Indications/Notes
|
|                                        |
|
|Address    UKC                          |
|
|                                        |
|
|                                        |
|
|                                        |
|
|                                        |
|
----------------------------------------------------------------
--
Test Details
 Duration 02:14:18  Artefact 00:01:58  Analysed 02:12:20  (98%)
 Reference 1205-C701H-4-M16WNT5002195 2002_1004_02_ ATest.car
----------------------------------------------------------------
--

Sinus Rhythm Summary
    Mean Heart Rate                   64
    Maximum Heart Rate                90
    Minimum Heart Rate                50
    Tachycardic Episodes               0     over 120 BPM
    Bradycardic Episodes               0     under 50 BPM

ST Segment Depression Summary
    ST Depression Events               0

AF Summary
    AF                                       Not Detected

Arrhythmia Summary
    Pauses of 1.7+ seconds             0
    Arrests of 3.0+ seconds            0
    Missed beats                       0
    Aberrant beats                     0

Ectopic Beat Summary
    Atrial Ectopic Beats               3     rate  1  per hour
    Ventricular Ectopic Beats          0
    MF Ventricular Ectopic Beats       0
---------------------------------------------------------------
Analysis
    24 Hour Heart Rate and ST Segment Summary. See page 2 of 4
    Atrial Ectopic beats detected. See page 3 of 4
    1 Symptom button event recorded. See page 4 of 4

Cardionetics C.Net2000+  7.0  Report        13:06 Fri
04/10/2002    Page 2 of 4
----------------------------------------------------------------
----------------
|Patient    ATest                          |Indications/Notes
|
|                                          |
|
|Address    UKC                            |
|
|                                          |
|
|                                          |
|
|                                          |
|
|                                          |
|
----------------------------------------------------------------
--
```

Test Details

```
 Duration 02:14:18  Artefact 00:01:58  Analysed 02:12:20  (98%)
 Reference 1205-C701H-4-M16WNT5002195 2002_1004_02_ATest.car
----------------------------------------------------------------
--
```

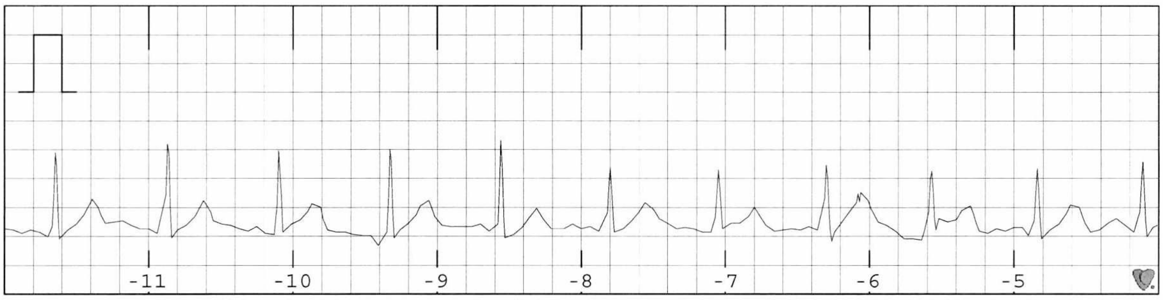
24 Hour Heart Rate and ST Segment Summary

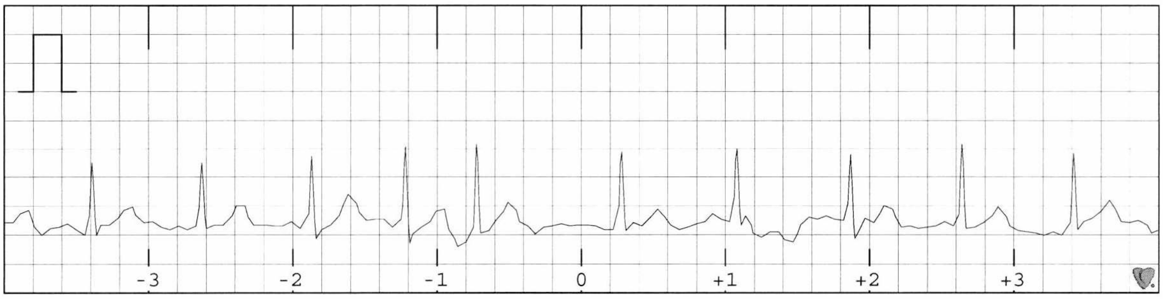
```
   Heart Rate     Duration      %       ST Segment  Duration
%
   Tachycardic    00:00:00      0       over +1 mm  00:50:08
38
   50 to 120 BPM  02:10:08    100      -1 to +1 mm  01:20:00
61
   Bradycardic    00:00:00      0       under -1 mm 00:00:00
0
```



```
Cardionetics C.Net2000+  7.0  Report        13:06 Fri
04/10/2002    Page 3 of 4
```

```
------------------------------------------------------------------
----------------
|Patient    ATest                         |Indications/Notes
|
|                                         |
|
|Address    UKC                           |
|
|                                         |
|
|                                         |
|
|                                         |
|
|                                         |
|
------------------------------------------------------------------
Test Details
    Duration 02:14:18  Artefact 00:01:58  Analysed 02:12:20
(98%)
    Reference 1205-C701H-4-M16WNT5002195 2002_1004_02_ATest.car
------------------------------------------------------------------
Atrial Ectopic Analysis
    Single          3 rate   1 per hour at 13:50     Shown
    Trigeminy       0
    Bigeminy        0
    Double          0
    Triple          0
    Salvo (4-7)     0
    Episode (8+)    0
```





```
Cardionetics C.Net2000+  7.0  Report         13:06 Fri
04/10/2002    Page 4 of 4
```

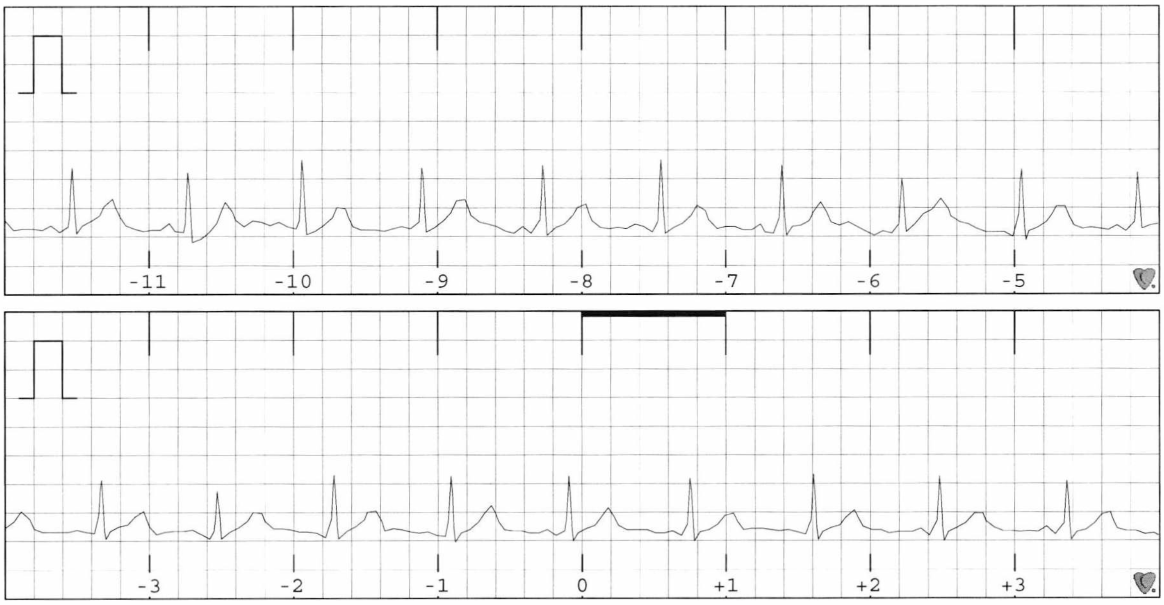
```
----------------------------------------------------------------
--
|Patient    ATest                        |Indications/Notes
|
|                                        |
|
|Address    UKC                          |
|
|                                        |
|
|                                        |
|
|                                        |
|
|                                        |
|
----------------------------------------------------------------
```

Test Details

```
    Duration 02:14:18  Artefact 00:01:58  Analysed 02:12:20
(98%)
    Reference 1205-C701H-4-M16WNT5002195 2002_1004_02_ATest.car
----------------------------------------------------------------
```

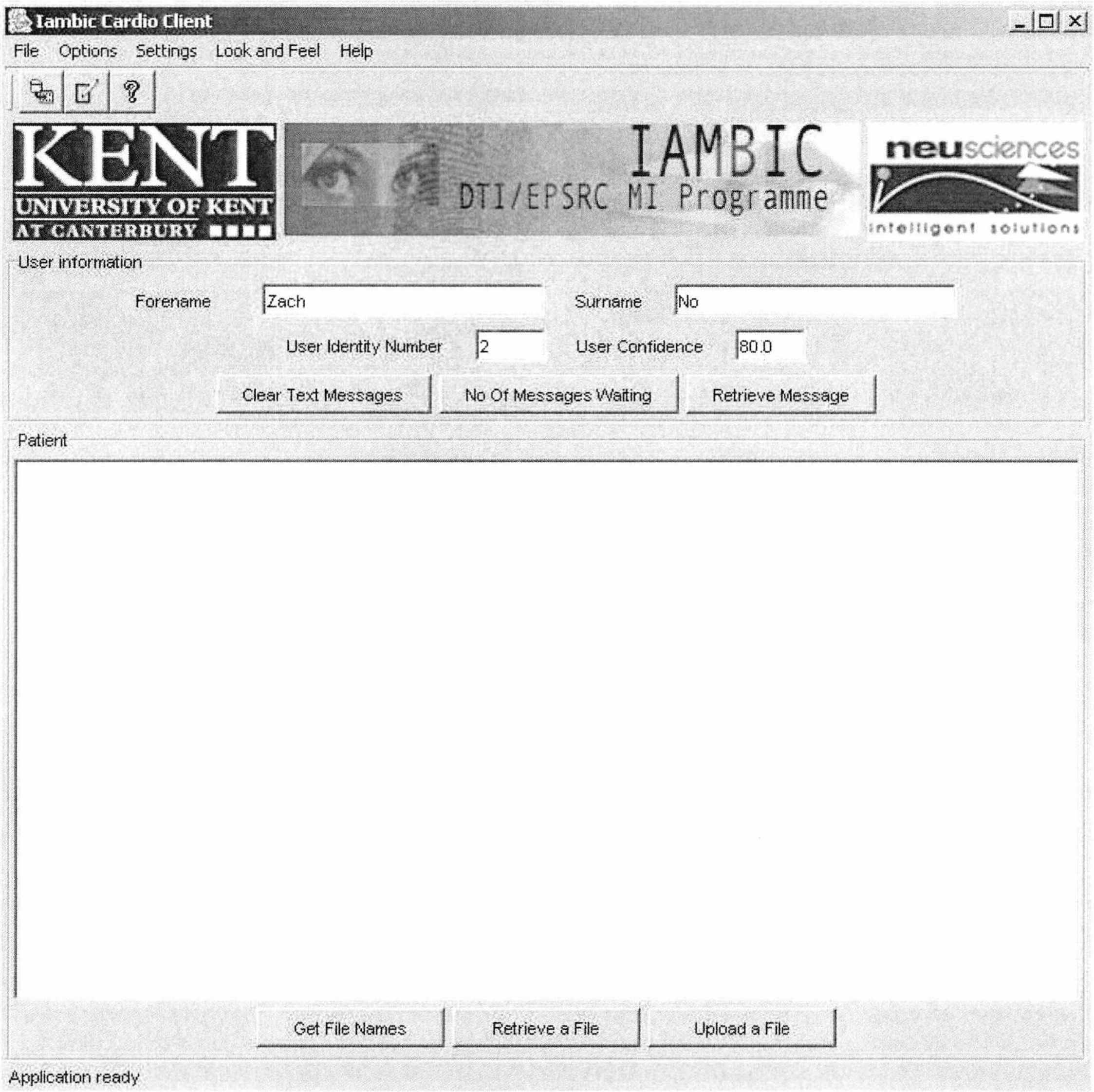
Symptom Button Pressed - First Event

```
    Time of trace        14:03

    Heart Rate           72
```



Database Interface Screenshots

The figures below illustrate the layout of the Java™ application that acted as the database interface for the client application.



User information

Forename Zach

Surname No

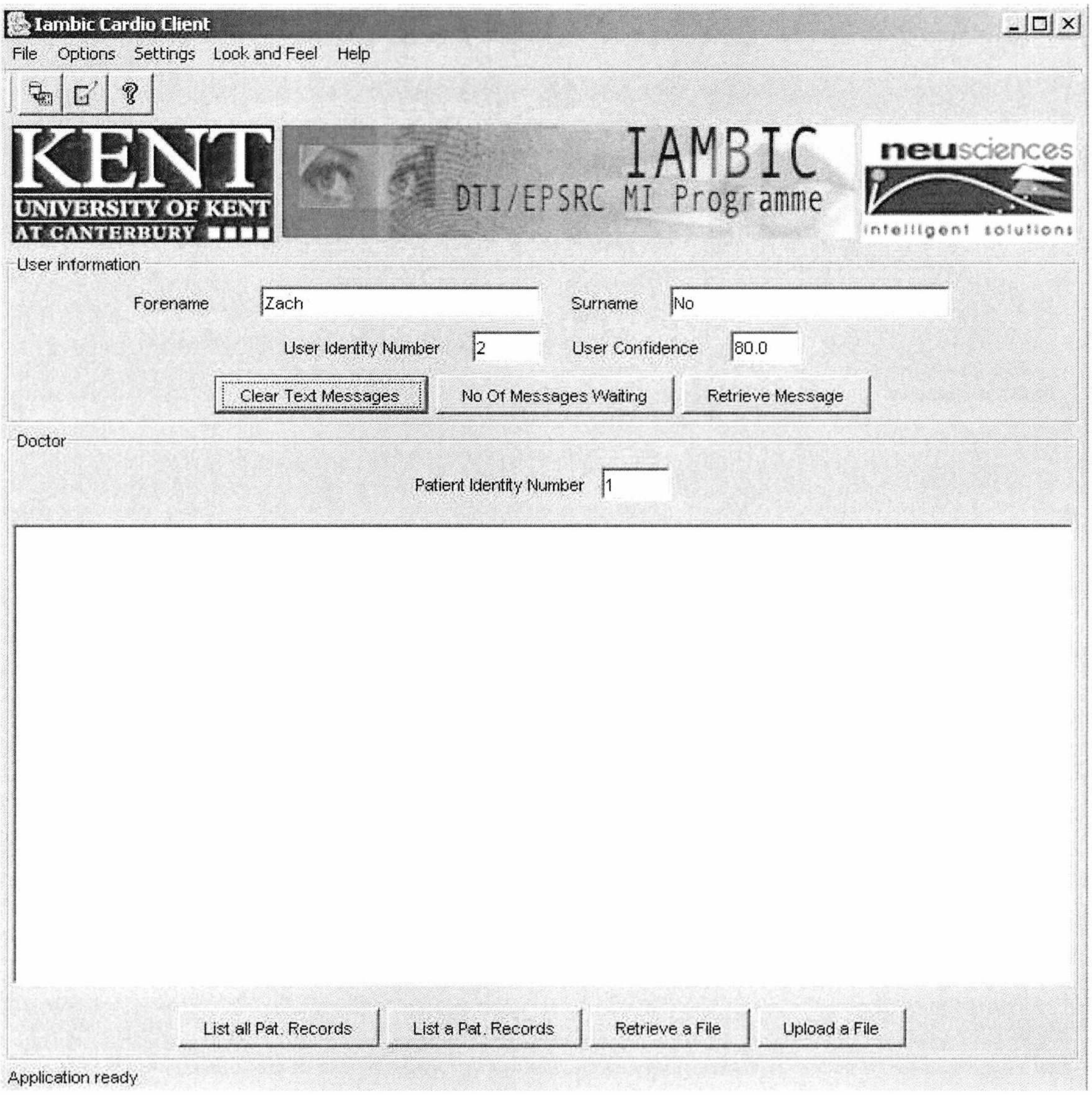
User Identity Number 2

User Confidence 80.0

Patient

Application ready

Figure B.25 Patient interface



Forename: Zach

Surname: No

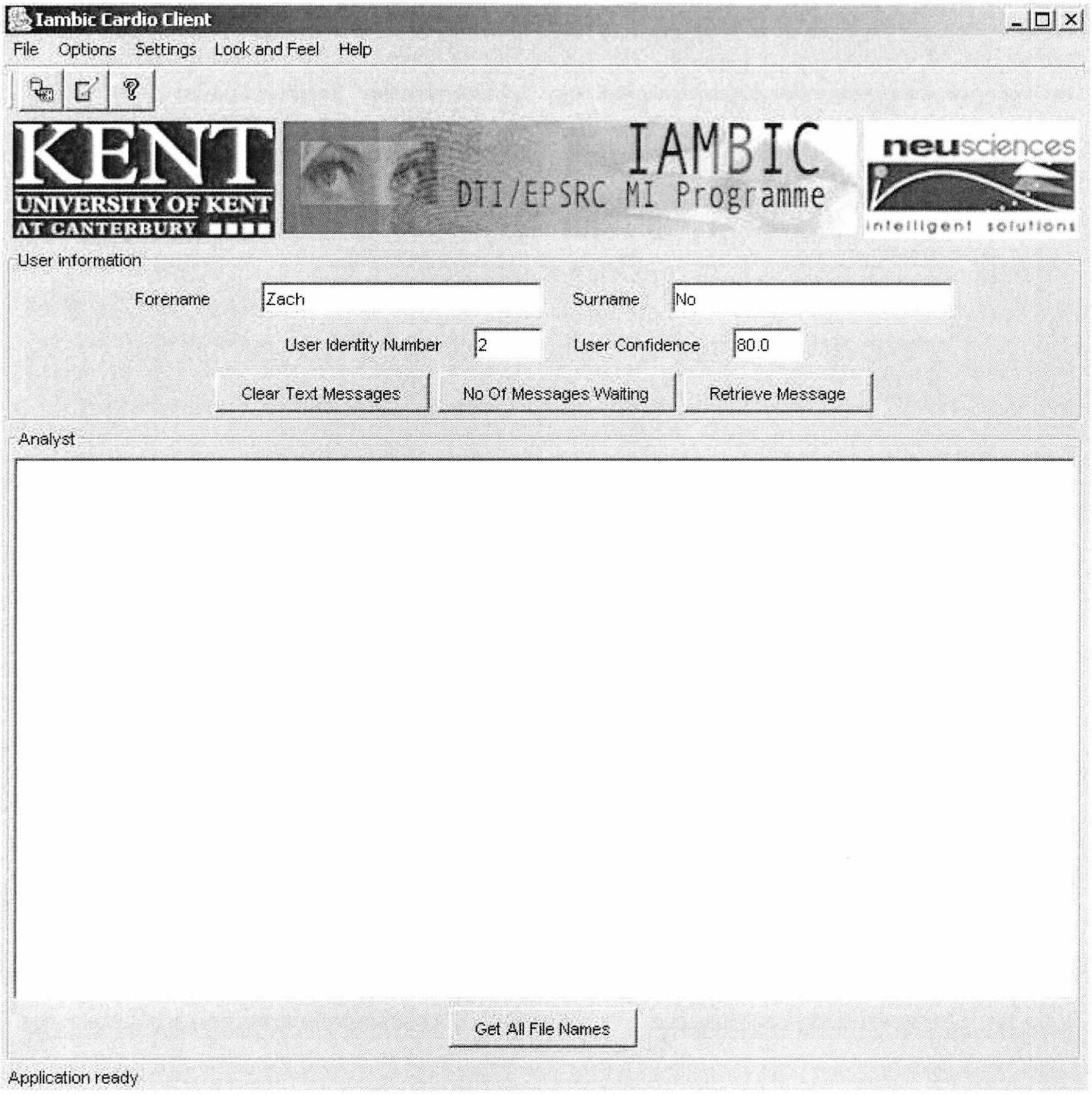
User Identity Number: 2

User Confidence: 80.0

Patient Identity Number: 1

Application ready

Figure B. 26 Doctor interface



User information

Forename Zach

Surname No

User Identity Number 2

User Confidence 80.0

Analyst

Application ready

Figure B.27 Analyst interface