THE UNIVERSITY OF HULL

Personality representation: predicting behaviour for personalised learning support

being a Thesis submitted for the Degree of Doctor of Philosophy

in Computer Science, the University of Hull

by

Dawn Helaine Wood BSc (Hons)

December 2010

Table of Contents

List of Figures

List of Tables

Acknowledgements

Thanks to my supervisor Tanko Ishaya for keeping me focused on the job at hand. I would also like to express my gratitude to the 12 Internet Computing students who took part in the studies and without whom this work would not have been completed.

A special thank you goes to Darren Stephens and Dave Webster who have provided stimulating discussions and much needed amusing activities during my time at university. Thanks also to all my university friends, both old and new, who have always had a good word to keep me going. In particular I would like to thank Janet Finlay, Tony Renshaw and Jakki Sheridan-Ross for reminding me in many ways why I started this process in the beginning.

As with all endeavours your family are always behind you and mine have most certainly given me the courage and motivation to see this through to its conclusions. My parents have never doubted my convictions, my sister set the marker for achievement by example and my extended family have provided the counter point of constant badgering, pointing of fingers and loving cajoling. Thank you all.

Last but by no means least I owe an unfathomable debt of gratitude to Lee Colley for both emotional and financial support and proof reading my spelling.

This work is dedicated to my family and friends.

Dawn Wood

Abstract

The need for personalised support systems comes from the growing number of students that are being supported within institutions with shrinking resources. Over the last decade the use of computers and the Internet within education has become more predominant. This opens up a range of possibilities in regard to spreading that resource further and more effectively. Previous attempts to create automated systems such as intelligent tutoring systems and learning companions have been criticised for being pedagogically ineffective and relying on large knowledge sources which restrict their domain of application. More recent work on adaptive hypermedia has resolved some of these issues but has been criticised for the lack of support scope, focusing on learning paths and alternative content presentation. The student model used within these systems is also of limited scope and often based on learning history or learning styles.

This research examines the potential of using a personality theory as the basis for a personalisation mechanism within an educational support system. The automated support system is designed to utilise a personality based profile to predict student behaviour. This prediction is then used to select the most appropriate feedback from a selection of reflective hints for students performing lab based programming activities. The rationale for the use of personality is simply that this is the concept psychologists use for identifying individual differences and similarities which are expressed in everyday behaviour. Therefore the research has investigated how these characteristics can be modelled in order to provide a fundamental understanding of the student user and thus be able to provide tailored support. As personality is used to describe individuals across many situations and behaviours, the use of such at the core of a personalisation mechanism may overcome the issues of scope experienced by previous methods.

This research poses the following question: can a representation of personality be used to predict behaviour within a software system, in such a way, as to be able to personalise support?

Putting forward the central claim that it is feasible to capture and represent personality within a software system for the purpose of personalising services.

The research uses a mixed methods approach including a number and combination of quantitative and qualitative methods for both investigation and determining the feasibility of this approach.

The main contribution of the thesis has been the development of a set of profiling models from psychological theories, which account for both individual differences and group similarities, as a means of personalising services. These are then applied to the development of a prototype system which utilises a personality based profile. The evidence from the evaluation of the developed prototype system has demonstrated an ability to predict student behaviour with limited success and personalise support.

The limitations of the evaluation study and implementation difficulties suggest that the approach taken in this research is not feasible. Further research and exploration is required – particularly in the application to a subject area outside that of programming.

Chapter 1
Introduction

1.1 eLearning scenario

Over the last twelve weeks Jeff had had lots of fun at university, as well as working hard. During that time he really came to appreciate that learning was not just about gaining knowledge, and his learning assistant, Buddy, had kept him organised and his workload well balanced. There had been more than one point where some of his decisions might have cost him vital time and marks.

One particular piece of work had given him a choice of topics to write about. Naturally he was going to opt for the one he felt he understood the most, but his Buddy had advised against it. Although Jeff's understanding was good in the area he had chosen, his writing skills were not suited to the style this particular topic required. *Buddy suggested two of the alternative topics*, indicating that both required the descriptions of concepts that could be represented graphically, one of Jeff's strong points, and researching these areas early would be a benefit come exam time.

Mathematical notation was not one of Jeff's strong points and several lectures he attended used this method to demonstrate new concepts. Understanding this, his Buddy initiated a search finding references to articles that explained the new concepts using non-mathematical methods. Also throughout the semester his Buddy had put together some bite-sized exercises that had allowed Jeff to practise his maths as well as covering relevant course topics. Jeff had been unaware of his new mathematical ability, but by suggesting a short quick-fire test his Buddy had showed him just how much he had improved.

Now it was revision time, none of the students, including Jeff, were really looking forward to it. Buddy had received the exam schedule a couple of weeks earlier and put together a preliminary timetable for Jeff. Requests for additional learning materials and past exam papers were submitted along with registering Jeff on specific help forums, both as mentor and mentee in anticipation of some peer group sessions.

After a weekend break Jeff was ready to get started, and requested his first revision slot. Buddy presented him with a twenty-minute presentation on a subject Jeff was fairly confident with. Asking questions at key points, and suggesting in places that Jeff should draw diagrams as aids to memorization. After this Buddy suggested he take a break, and if he was up to it, to have a look through a section in a particular text. It would assist with the second revision slot. Feeling fairly confident Jeff made a coffee, picked up his digital pad and scanned through the text a couple of times while enjoying the Java aroma.

1.2 Background and motivation

In the scenario described above the hypothetical Buddy, is an electronic learning assistant, (often referred to as a Learning Companion (Chou et al. 2003)) that personalises support for a student throughout their learning career. Designed to be a platform independent intelligent agent, its primary purpose is to balance the student's workload, find suitable alternative resources, and give advice that supports this particular student's learning process. From these perspectives it fulfils some of the activities usually undertaken by tutors and support staff, as well as the student's peers. For electronic learning or eLearning, this is a desirable goal given today's demands on both educational institutions and their students.

Some of the activities undertaken by the Buddy are already part of some educational systems. Research into the presentation of alternative learning material (Wolf 2003) the organisation or navigation through learning material (Grigoriadou et al. 2001) and recording of student skill levels (Zarraonandia et al. 2005) are currently in progress. Systems, such as Blackboard (Blackboard 2006) which enable the tutor to present material in a specific order at predefined times effectively pacing the student's workload, are actively used in educational establishments today.

However this hypothetical system is also performing on a more subtle level, which current research is only just starting to approach. For example take the highlighted phrase in 1.1. A personality model is effecting the system's actions in such a way that, it understands that Jeff's personality will not respond well to being just told the best course of action, so it puts forward a choice leaving Jeff still in control of his decisions. It also explains it's reasoning for making such suggestions, by indicating something that Jeff is good at as well as time saving in the long run. Both these decisions are influenced by its understanding of its user's personality, by appealing to those traits that give Jeff positive feelings in being creative and efficient. This depth of personalisation has of yet not been accomplished by current profiling techniques.

Current profiling technique use attributes that are generally derived from the functionality of the system. For example to recommend products the profile usually consists of product attributes, such as brand, cost, colour, or user preferences for particular products, commonly referred to as ratings data. Quite often these attributes cannot be used in similar domains and are not transferable between domains. Profiles that focus on using ratings data don't account for individuality and those users with tastes and needs that sit outside the mainstream often receive poor recommendations.

The key point that this scenario puts forward is that: while there are some very obvious and distinct activities that systems can undertake to support students; successful integration into student learning activities is actually far more complex. Simply understanding that a student's learning style is different, and thus requires alternative material, is not the whole story. How, when and where that alternative material is presented is also important. These are the subtleties that human-to-human interactions account for and they come from understanding an individual at a much more detailed level than current profiling mechanisms offer.

The problem is that profiles used within systems for personalisation predominantly select attributes that are derived from the needs of the system and classify individuals into a small number of stereotypes. Both these aspects fail to account for individuality and limit the cross-domain use of the generated profile. In today's ever increasing range of online products and services it is important to overcome these limitations. To make better quality recommendations to all and enable users to retain control of a single profile for all domains that they can choose to share with those systems they trust.

1.3 Research aims

The core theme of this thesis is the personalisation of services. To this end this research puts forward the idea that a profiling system derived from psychological theories of personality has the potential to increase the scope of personalisation to those who lie outside the mainstream and enable a single profile to work across multiple domains. This research is in the early stages of development and this thesis will focus on the development of the profiling models and their application to a single case domain that of an educational support system. To this end this thesis seeks to answer the following research question:

> *Can a representation of personality be used to predict behaviour within a software system, in such a way, as to be able to personalise support?*

There are three key concepts in this question that require some clarification within the context of this thesis. A *representation of personality* refers to some model that includes aspects of personality derived from psychological theories. Here theories of personality will only be considered if they have been developed from a sound research base and are generally accepted within the psychological research community. There are other systems for identifying personality such as the zodiac or star sign system which are not based on research and are not considered here. It is assumed here that psychological theories will provide a more exact representation of personality and thus predict behaviour more accurately.

Here the concept of *predicting behaviour* in order to provide support is perceived as being no different to the process that underlies current recommender systems. Using one or more user profiles a recommender predicts that a particular user will have a preference for one product or service over another. The reason the product is then recommended to the user is that the user has been predicted to most likely purchase that product. The term *predict* has been used here to emphasise the link to psychological theories as well as clearly identify this particular step in the recommendation process.

The final concept is that of *personalisation*. Many systems already provide varying degrees of personalisation as discussed in 2.2. However this research is aiming to generate personalisation at an individual level rather than at a group level, which many collaborative-based systems already achieve. This research is looking to demonstrate individual patterns of support rather than groups of similar support for similar users, although both perspectives will be considered.

These three concepts form the core of this thesis and will be considered within the context of the following aims:

The aims of this thesis are:

1. to generate a generic profiling model from psychological theories of personality that has the potential to find solutions to current profiling issues as discussed in 2.1.7 (see Chapter 3)
2. to consider the practicalities and implications of this approach to personalisation within an educational support context (see Chapter4)
3. to determine the feasibility of this approach through a prototype experimentation within the domain of educational support (see Chapter 5 and Chapter 6)

4. to identify the potential scope and the limitations of this approach to a cross domain profiling system (Chapter 7)

The central claim of this thesis is that it is feasible to capture and represent personality within a software system for the purpose of personalising services. By using a profiling method that captures the essence of an individual, it is anticipated that only a single profile will be required which can then be applied across multiple domains. The perceived benefits to the user are that this single profile can remain within their control, be utilised on multiple platforms and be applied more effectively to new domains. The perceived benefits to systems developers are that data gathered by this profile from other domains may apply directly to their services and where this is not the case they need only focus on linking the profile to their services, rather than the generation of a new profile.

1.4 Research methodology

This research uses a mixed methods approach incorporating both quantitative and qualitative methods for investigation and evaluation. The following texts were consulted during the design stage of the research process: Burns 2000; Gillham 2000b; Gillham 2000a; Oppenheim 1992; Creswell 2003; Rubin 1994; Corbin and Strauss 2008.

To enable the concept of a personality representation (PR) profile to be evaluated the first stage in this research was to develop a computational model of personality from psychological theories of personality. A literature review of the predominate theories of personality psychology was undertaken which informed the process of the models' development presented in 3.2. This resulted in two models a behavioural framework (Figure 3.6) and a model of personality (Figure 3.7) for use as a profiling system.

The second stage was to implement some part of the developed models (see 4.4 for proposed implementation) within a software prototype (Chapter 5) in such a way that it could be tested and evaluated with student users. To enable this development an observation study of students performing programming tasks was undertaken, presented in 5.1. This provided data for identifying areas of programming support and linking personality to specific behaviour patterns. Personality data was gathered using a psychology instrument (Johnson 2000; Johnson 2005), based on the Five Factor Model of personality (McCrae and Costa 2003) and a self-evaluated programming preference survey (Appendix A.2.). The recordings of the programming activities were analysed qualitatively in an investigative manner to identify behaviour patterns and the programming tasks they related to as discussed in 5.1.1. To confirm the identified areas for support from the observation study a focus group was invited to discuss and explore the main barriers that students faced when learning software engineering. The resulting discussion was analysed qualitatively and results are presented in 5.1.2.

The final stage of this research involved the evaluation of the Personality Representation Support System (PReSS) prototype (Chapter 6). This was done by using a second simulated lab experience, following the same procedures as the observation study, but with the PReSS prototype replacing the coding editor used in the first study. Participants were asked to evaluate their experience of the support they received from the perspective of how they perceived it was relevant to their current context (as described in 4.4 and 6.1.2). They evaluated their experiences during the study using the quick-evaluation sheet in Appendix B.1 and a post-study evaluation questionnaire in Appendix B.4. The measures selected for the evaluation process are discussed in 4.4 and the means by which they were generated are analysed in 6.1.

1.5 Thesis contribution

The primary contributions of this thesis are:

- the presentation of a model of behaviour based on psychological theories
- the presentation of a profiling model based on psychological theories of personality that combines individual differences and group similarities and uses both implicit and explicit data collection methods

- a proposal for a theoretical educational system implementing the models that highlights a number of key issues with the personalisation of services and presents some potential solutions
- an application of the personality model within a prototype environment that demonstrates the personalisation of support
- the results of the prototype evaluation highlighting the difficulties this approach presents and some potential benefits.

These contributions are expected to be of interest to those researchers considering issues around the personalisation of services. In particular practitioners in the fields of profiling, user modelling and educational technology will find this thesis valuable for future research.

When this research was first undertaken there were few indicators that psychological theories of personality were being considered as a means of personalisation. There was a small amount of research within educational systems using a few individual aspects of personality (2.2), but within recommender systems there was no interest identified. In recent years a number of researchers have developed and tested prototype systems utilising a personality-based profile, such as (Hu 2010). These have been applied in the same means as the concept of collaborative profiling, focusing on group similarities, which supports some of the ideas presented in 4.3. However this research is seeking to identify a more holistic model of personality. The models (Figure 3.6 and Figure 3.7) presented and evaluated in this thesis included the use of the Five Factor Model (also referred to as the Big Five or OCEAN), at the facet layer as well as the factor layer. They also included a number of other personality elements that are focused on individual differences. As far as it is possible to know current research is not examining the use of a holistic model of personality.

1.6 Scope and limitations

There is a wealth of literature that could be covered given the topic of personalisation. The concepts of profiles and user models are used in many branches of computer science. For this reason it is important to identify here the boundaries of this thesis. The two core areas investigated during this research were, profiling techniques within recommender systems, which provided the motivation for this research, the issues to be addressed, and personalisation techniques within educational systems, which informed the case study used to evaluate the feasibility of the personality profile.

Human Computer Interaction (HCI) is an area that applies a range of user models and may have been of interest here. However it was decided not to go down this route as the focus of this research was on generating personalisation and not on mapping interactions. Another area that may also have been of interest was that of Artificial Intelligence (AI) which models a wide range of human processes. Some of these models may have added value to this work or provided a starting point in some way. It was decided that utilising an existing computational model may result in missing personality aspects or insights into personality that may be more appropriate to the problem of personalisation.

Given the nature of the case study another area that could have been reviewed was the practice of teaching and learning software programming and adult education in general. At the time it was decided to draw on the local experience of programming tutors rather than pursue this area directly.

There are limitations to the generalisability of the results of this research. The initial aim of this research was to produce a generic profiling model that could be used across domains. As only a single case study has been evaluated here, that of programming support, it is unclear at this stage how effective this model would be within other contexts.

During the evaluation phase of this research a number of changes had to be made due to actions outside the control of the researcher. The original intention was to undertake a longitudinal study across two semesters with two groups of first year undergraduate students. One group would interact with the prototype using the full profile (both dynamic and static aspects of personality), with the individual dynamic aspects being populated from each interaction. The second group would interact only in a single instance using only the collaborative profile data. The aim here was to test the consistency from one interaction event to the next as well as evaluate how effective the collaborative aspects of the profile would predict to new users. However due to a change in programming language for first year students, from Java to JavaScript, it was not feasible to redevelop the prototype for this group. This changed the pool of available students to second years and upwards, including postgraduates. This also resulted in various time delays limiting the research to two observation studies.

Unforeseen difficulties with capturing certain profile aspects resulted in only a part of the profile being tested within the prototype as discussed in 5.3. Part of this profile was also generated by an untested questionnaire (as discussed in 5.1) which may have been the reason for the dynamic aspects performing less successfully than the static aspects of the profile. All of which demonstrates the difficulties found with implementing a personality based profile.

1.7 Structure of this thesis

Chapter one introduces the concept of personalisation and presents the main research aims and outcomes of this thesis. Chapter two reviews two relevant areas of literature that of profiling and recommender systems and the use of personalisation within educational systems. From this a number of issues are identified with current profiling techniques. This chapter concludes by suggesting that a personality based profile may provide the solution to a number of these issues.

Chapter three presents the concept of personality as defined by this research. It then presents the process undertaken to develop a set of models via the review of a number of psychological theories of personality. The developed models and their contribution are presented at the conclusion to this chapter. Chapter four theoretically discusses how the models developed in Chapter three can be applied in the context of an educational support system. With the models providing the means for generating personality based profiles for the purpose of predicting behaviour and personalising learner support. The conclusion to Chapter four presents the initial evaluation plan for a case study that implements the models within a prototype system with the domain of educational support.

Chapter five presents the design and implementation of the prototype system. This includes the first study used to observe the behaviour patterns of participants undertaking programming tasks. Chapter six evaluates the prototype system using a second observation study, where participants experienced and evaluated the personalisation of learning support. The final chapter, Chapter seven concludes this research by reflecting on the evaluation results and the process of implementation and presenting the final conclusion and areas for future work.

Chapter 2
Personalisation

In the context of this thesis personalisation is the idea of making something personable and personally relevant to an individual as opposed to impersonal. This is more than just identifying an individual’s preferences, which currently is the focus of many online systems. It is about being able to applying those preferences in such a way as to make the interaction more amenable and relevant to the user. This is particularly important in the context of systems designed to support learning. In the face-to-face world the act of giving instruction or direction, does not inspire or motivate a response in itself. Something more is needed from the tutor, something that generates a relationship of trust and commitment. While it is accepted at this stage that the idea of generating that level of personal connection is not yet a realistic endeavour, this thesis aims to move the idea of personalisation of services in that direction.

This chapter reviews two key areas of research that are relevant to the concept of personalisation within software systems. The first is the domain of profiling within recommender systems, examined in 2.1, exploring types of attributes and methods used in their exploitation within various domains. This shows that current profiling techniques are powerful and useful tools within their rightful domains, but as yet have a number of unresolved issues (section 2.1.7) that would impact on their performance in broader domains as educational support.

The second is how personalisation is being implemented within educational technology (section 2.2) and to what extent recommender techniques are being applied (section 2.2.4). There are some marked differences in the approach between recommenders and educational technology (discussed in 2.2.5).While the scope of these systems is still fairly limited they are moving more towards a more personable concept of personalisation than current recommendation systems.

The final section (2.3) of this chapter brings the findings from these two areas together and clearly identifies the problem within the context of this research.

2.1 Profiling techniques

The idea of a profile has been around for hundreds of years and has slight variations depending on the domain of reference. In the arts it refers to the outline of an object. In literature it refers to a biographical sketch or character study of an individual (Ilson et al. 1987; Harper 2001). According to the Encyclopaedia Britannica (2003) the term profile is *a set of data often in graphic form portraying the significant features of something*. Computer science also has several uses for the term, depending on its context which can be either hardware or software orientated.

From the perspective of computer science, profiling has always been applied with the intention of making the user's interactions with systems more intuitive and less arduous. However, while a psychological profile is a complex cross-domain concept, profiling within computer science is a limited set of values that are domain restricted. This restriction is potentially limiting in today's cross-domain and multi-cultural applications which users want to tailor more closely to their individual specifications.

The type of profile that is examined here has its roots in the domain of Information Retrieval (IR). IR was formed at the intersection of librarianship and computer systems (Baeza-Yates;

Ribeiro-Neto 1999). It focuses on the processes surrounding the storage, organisation and retrieval of information.

User profiling was developed to better represent a user's informational needs by adding context to the user's queries. Profiles consist of user topic preference, or user feedback on previously retrieved information, or a combination of both. The profile can be used to extend the user's query (Nanas et al. 2003) or act as a filter on the query result set as shown in Figure 2.1. As an extension to the query the aim is to make it more specific. For example, if the user enters the search term *Tigers* and their profile indicates that the user has an interest in natural history, the system may remove or demote results that come from other topic areas, such as literary or sports references. The profile can also be used after the results have been formulated as a means of ranking or filtering. Both these interaction points will produce fairly similar results, but have an effect on the systems architecture and the types of algorithms used.

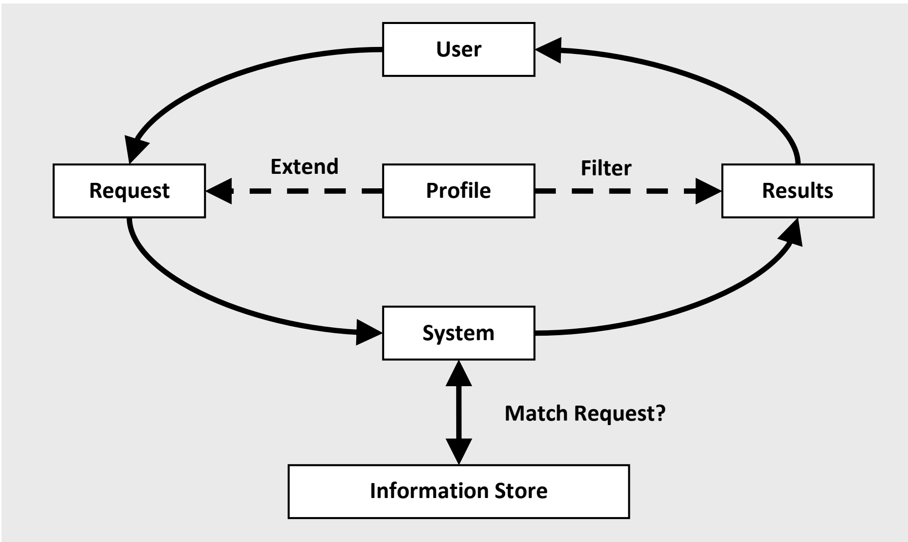


Figure 2.1 The use of a profile within a search request

With the development of the Internet and World Wide Web (WWW) the need for more effective information retrieval systems is required. Search engines are the primary gateway to the content of the Web, providing a pull based service to users. Some employ categorisation techniques like the Yahoo (2009) directory. Others use full-text indexing methods like Lycos

(2009). The hyperlink, unique to the WWW provides a structure that can also be used in the retrieval and ranking of information, utilised by Google (2009).

Search engines are not the only beneficiaries of IR techniques. For businesses, education and other information providers, the number of potential customers has also increased. Local newspapers traditionally had only a local readership, and local stores were stocked for local needs. To develop a presence on the Web, businesses and organisations have had to account for a multitude of differences in their customer's requirements and technical abilities. Their solution to this was to search on behalf of the customer. In other words recommend to the customer a selection of products and services in anticipation of their requirements. Thus a new technique of information search was developed, referred to as a recommender system.

2.1.1 Recommender systems

Recommendation systems are primarily concerned with news, products (both digital and none-digital) and marketing. They utilise many of the techniques developed in IR to provide a push based service to users (Fan et al. 2005). In particular they have adopted and adapted the use of profiling as one of their solutions to their retrieval problem. Profiles now contain a variety of user related attributes, as well as interests and feedback. They are applied to individuals or groups and use information gathered on or submitted by both. Recommendation systems have become one of the primary research areas for profile development.

Another area that has a major influence on recommendation systems, particularly in enhancing performance of profile application, is AI and more specifically the methods used within machine learning (Godoy and Amandi 2003). The use of machine learning technologies within profiling systems has been driven by various factors. Some have been used to combat problems inherent in basic profiling techniques, others to improve the intuitiveness of the overall system.

Unlike searching, recommendation processes are designed around the nature of the profile. Figure 2.1 shows how a profile is added to a search system for the purpose of improving returned results, at either the point of query expression or ranking algorithm. By comparison

Figure 2.2 shows how the profile is more integral to a recommender system. The profile acts as a record of user interaction, described by a set of attributes and values. User feedback, either explicit or implicit, is used to update the profile over time, thus keeping pace with changes in the user's perspective on the data being recommended. The profile is then used directly to identify the data to be recommended to the user, not as a secondary process on an already defined sub-set of potential items. This process can include simple matching of profile attributes to item attributes or a more complex, decision-based process. Without the profile the recommendations would be based only on the current user interaction or the most recent record of user feedback.

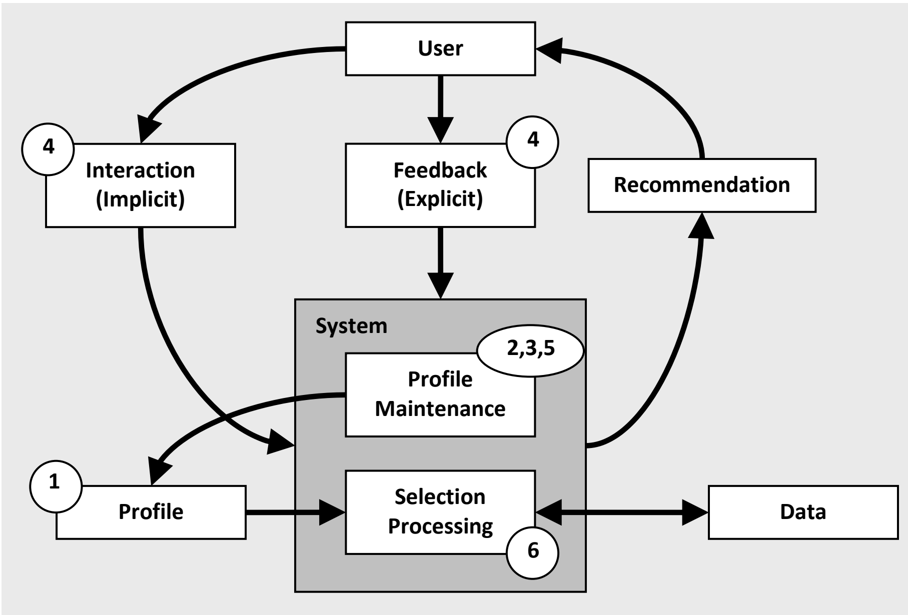


Figure 2.2 The point of profile use within a recommendation system

The profile is the critical design point to the recommender development. Godoy and Amandi (2003) suggest that the following four questions should be asked during the design process: What are the attributes?; How to obtain attribute data?; How to apply this data to the recommendation set?; How the attribute data is updated? In Montaner et al's (2003) classification of agent recommenders they identify six similar design decisions that need to be taken into account when developing a recommender. These critical decisions are linked, via the circled numbers in

Figure 2.2 to the main elements of a recommendation system and are as follows:

1. Profile representation - the aspects of the user or their activities that are used to define them within the system. These aspects are referred to as the profile's attributes.

2. Profile generation – how the initial values of the profile attributes are generated. The method used here determines the need for profile learning techniques.

3. Profile learning technique – in automated profile generation systems, different techniques are used that reflect the nature of the profile and its performance.

4. Relevance feedback – the information that is gathered about current user interactions or desires. This may be positively or negatively orientated. It may also be implicit, inferred from interaction types, or explicit, directly inputted by the user.

5. Profile adaptation technique – how the profile is updated or adapted to reflect changes in the user. This may include the removal of profile attributes and values or an adjustment to the values of specific attributes.

6. Profile exploitation method – how the profile is applied to the content or objects that are to be recommended. This process consists of two parts: how the potential recommendation set is filtered; and whether that filtering process has a profile to item relationship or a profile-to-profile relationship.

The exploitation technique effectively provides the bridge between the profile and the recommendation set. How that bridge is used and the method of its process or calculation is reflected on the attributes used within the profile and thus on the rest of the design decisions. It is the nature of this technique that is used predominately to categorise the different types of recommender system.

Three exploitation techniques are identified by Montaner et al. (2003), content based, demographic based and collaboratively based; and the use of a hybrid combination. Later work, such as Bogers and Bosch (2007), distinguishes only between content and collaborative. Others, such as Fan et al. (2005) and Godoy and Amandi (2003) refer to the same processes, but label them from the historical perspective of their algorithms, as either being derived from IR or from

Machine Learning (ML). IR methods tend to be associated with content filtering and ML methods with collaborative, but this is not always the case. For the purpose of this research Montaner et al. (2003) provides the most comprehensive method of classification and this review focuses on content, demographic, collaborative and hybrid filter.

2.1.2 Content filtering

The term *content* tends to imply that this method is only dealing with textual objects. This method is concerned with the similarities between object descriptions as is shown in Figure 2.3. In the case of documents this may be extracted keywords or topics. In the case of consumer products this might be the categorisation of the product, a description of its appearance or its practical applications. This descriptive profile is then compared to the descriptions of other items within the potential recommendation set and those that match, or have a specific number of matches, are recommended to the user.

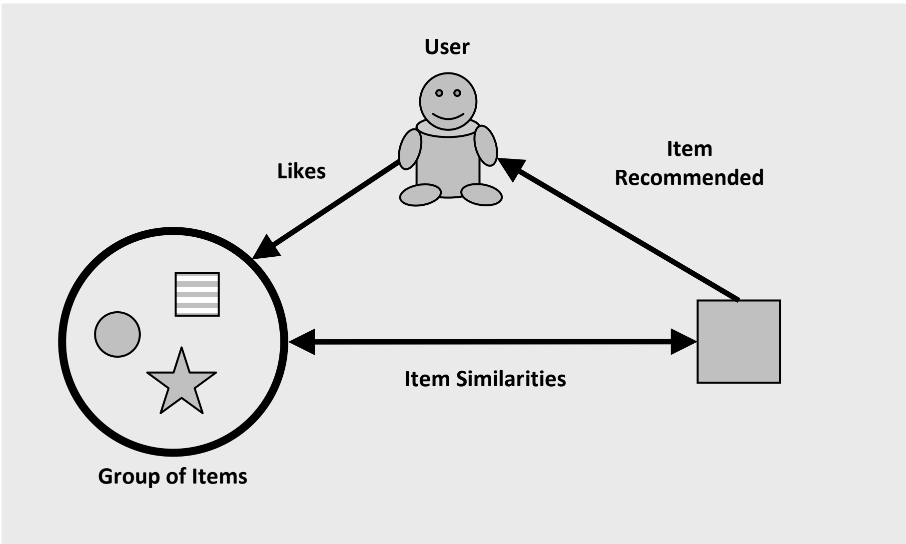


Figure 2.3 Content profiling

Amalthaea (Moukas 1996) an early recommender system was developed in response to the assumption that excessive results from search engines, was as much to do with poor keyword choice on the user's part, as poor indexing capabilities on the search engines part. The profile for this system is a list of interests extracted from the user's favourites (browser-stored links) and their browsing history. It is focused on three tasks: data discovery via meta-search using weighted keywords for document selection; continuous news feed filtering; and monitoring existing sources of interest for changes. The results of these processes are presented to the user in the form of a digest. Users are then invited to give feedback as to the relevance of the selected resources. The feedback is in the form of ratings scaled from one to five. This system is an agent based architecture, the feedback assigns *credit* to those agents that produce positive recommendations, giving them greater weight in future predictions.

What is particularly interesting about Amalthaea is the freedom it gives the user to add to and adjust the profile. The user is able to give the system relevant documents and URLs (Uniform Resource Locator) as well as selecting which keywords are more relevant than others. Recommendations produced by agents that are assigned user selected content are also given more weight, providing a second layer to the profile application.

Another early, yet highly cited work is Letizia, developed by Lieberman (1995). Like the Amalthaea system, its profile consists of keywords extracted from URL's, but Letizia identifies relevant URL's by studying the user's behaviour during browsing. It does this by using a set of simple heuristics. For example:

- The user saves a link to the web page (adds to favourites) - shows a strong interest
- The user follows a link within a page – shows a tentative interest in the link and a degree of interest in the page the link is embedded in
- The user returns to the referencing page within a short space of time – shows the link was of little interest, but there is an increasing interest in the initial page.

Letizia also takes user activity within a single page into account. In the West users tend to read left to right, top to bottom. Using this fact the system can construe that unselected links lying above a selected link are of no interest. This examination of behaviour and its relation to the relevance of a piece of text is unusual and intriguing, shifting the focus from the recommendation set and towards the user.

Seyskill and Webert (Ackerman et al. 1997) is an agent system that recommends web pages to users that focuses exclusively on explicit user feedback. This feedback consists of a hot (like) or cold (dislike) ratings for each visited page. Using cross document word frequencies to generate keywords, the system submits new queries to search engines as well as analysing the documents hyperlinked to those in the current known set. The results are then presented to the user as annotated links on a web page. With a probability rating, generated using a Bayesian classifier (Duda and Hart, 1973 as cited by Ackerman et al. 1997) indicating the possible level of interest in each document.

These three systems highlight one area of contention within all profiling systems, whether to use implicit or explicit data. Content filtering, particularly for the purpose of document or web resource recommendation, tends towards implicit feedback derived from favourites list (Moukas 1996), previous documents (Nanas et al. 2003), browsing history and other user interactions (Godoy and Amandi 2003). The benefit of inferring interest from behaviour, current and previous, is that it is a less obtrusive process for the user (Godoy et al. 2004), but inference can be prone to inaccuracies. Information gained directly from the user may be more accurate, as in the initial document selection for Bogers and Bosch (2007) and the user ratings of Ackerman et al. (1997) but is time consuming and requires the user to be familiar with the relationship ratings attributes and the type of result generated. Like Fan et al. (2005) suggests, *it is no different from the formulisation of a query within a search engine and imposes the same difficulties on those lacking this skill.*

Another issue that is raised by the work done by Ackerman et al. (1997) is a problem of using a profile generated within one domain across others. Profiles are generated for each individual in each topic. These were then tested across topics using a Bayesian classifier and the predictions were no better than a guess. Ackerman and Billsus et al. (Ackerman et al. 1997) suggests two reasons for this. The first is simply that the words have different contexts in different domains, which varies their usefulness in describing that domain. This is a problem inherent in many text-based systems, which lack an understanding of semantics and context. The second is that a simple decision boundary exists within a single domain and that the more domains added the more complex that boundary becomes. This effectively becomes a clustering problem and will require an increasing number of examples to gain useful clusters. To combat this, the authors also experimented with decision trees (Russell and Norvig 2003) and multi-layer neural nets (Russell and Norvig 2003; Mccorduck 2004).

The move towards semantic data and the application of ontology has become more prevalent in the last five years providing content context to the recommendation processes. By including semantics within content filtering weight can be added to those terms that have particular types of relationship. Xuan et al. (2011) has used semantic trees to generate semantic weightings between terms for both the user profile and the recommendation set, with some positive results.

The use of multiple profiles, such as used by Ip et al. (2000), is required when the profile attributes are based on descriptions of the potential recommendation set rather than that of the user. This is also evident in Nanas et al. (2003) using a concept map for each topic of interest and Godoy et al. (2004) models the user's interests in separate categories. This is often referred to as over specialisation, more of the same (Montaner et al. 2003), on the part of the system.

Linked to this is also the problem of the lack of serendipitous recommendations, *partly because seemingly unrelated topics are often related through non-obvious connections* (Lieberman 1995). While humans can understand and express tentative links between topics for machines, things either belong or do not belong to a set. The broader the rule set, the more irrelevant items included, and the narrower the less likely that an oddity will spark interest in a new area. Defining the division point between two profiles can have serious effects on the performance of the system (Ackerman et al. 1997).

For the user multiple profiles can be difficult to manage, update and keep track of. There are many concerns about this distribution of such personal information (Kay 2006) and a more ideal solution would be a single cross domain profile that resides locally with the user as suggested by Moukas (1996) and Kay (2006).

Others have looked towards AI for solutions for recommendation of text based resources. Godoy and Amandi (2003) uses agent-based architecture in utilising user profiles for recommending documents and web resources. Russell and Norvig (2003) use case-based reasoning to describe a user's experiences of the resources they have reviewed. These stored experiences are used to make comparisons to current activities so that solutions to these activities can be found and recommendations made. The cases represent the document as a *bag-of-words* (Russell and Norvig 2003) weighted according to the structure of the document, after stopping, (removing small words such as the, and, off) and stemming (replacing words with their root such as computing to compute) have been applied. The context of the document visit is taken into account by: the time spent by the user in relation to the documents length; the URL; and the level of user interest. A confidence measure keeps track of the number of times a case has been successful or unsuccessful.

Lieberman (1995) also notes that profiling methods need to consider persistence of interest and the development of a realistic method of old interest fading and possibly being forgotten entirely is needed. So far the focus has been around the textual content of documents and articles. Textual descriptions are also used to describe products and multi-media web content to facilitate recommendation. This process is itself subjective and personal to the describer, for these reasons there are also many concerns about the use of content filtering with non-textual elements and items.

2.1.3 Demographic filtering

This method of filtering is often misplaced within the collaborative group and there are very few examples of its use. Montaner et al.'s (2003) review of thirty-seven recommender systems identifies only a single case of demographic filtering, LifeStyle Finder (Krulwich 1997).

Mobasher (2007) classifies systems that use demographic data as Knowledge based recommenders, but does not provide any examples of such.

The profile for this type of filtering consists of descriptions of the user. These descriptions can be physical, as in their clothing sizes; psychological, such as their preference for a particular colour; or behavioural, the user performs a particular pattern of actions to achieve a particular task. Figure 2.4 shows the relationship of a single user to a recommended item. The attributes within the user profile are compared to the attributes of other users. Where similar profile values are found, the items these users have had a positive response to are recommended to the user. This user-profile-matching (Montaner et al. 2003) usually uses one of two methods (Shapira et al. 1997); clustering techniques to identify the commonalities across users (Krulwich 1997); or a series of predefined categories to which attributes are assigned via experts in such areas. Thresholds can be set on each variable so that partial matches can be used to assign a user to a particular cluster. Users can also belong to more than one cluster depending on the domain types the clusters are linked to.

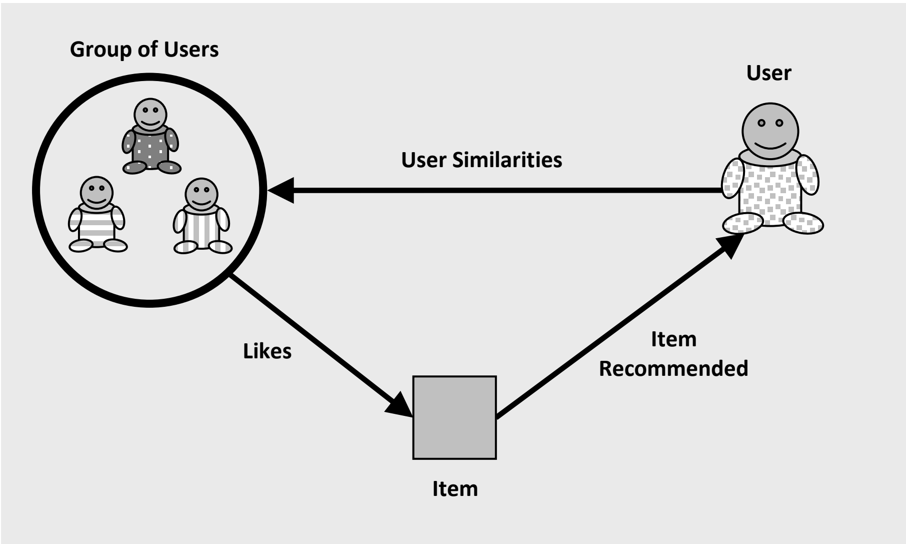


Figure 2.4 Demographic profiling

Lifestyle Finder (Krulwich 1997) uses demographic data from a commercially available database called PRIZM now available from The Nielsen Company (2010). The data set used was based on the surveys of more than 40,000 people across 600 variables that included characteristics such as purchasing whisky, playing golf or owning a dog. For each user the probability of which of the sixty-two demographic clusters they belong to is calculated. If there is more than one most likely cluster then a partial profile is generated from those values that are similar in the matching clusters. The authors suggest that as little as half a dozen demographic variables are needed for this process, which, in their opinion is considerably less than the other approaches they reviewed.

Krulwich (1997) evaluates their profiling technique by embedding it in a web site recommendation service called *Waldo the Web Wizard*. By inferring more abstract clusters from the data, such as genera of TV watching, and linking this to a question and answer set for the user to complete. The *Waldo* system is able to infer the user's most probable lifestyle and thus recommend a set of URLs they will be interested in. Their results show that at least 60% of users preferred the profile based recommendations to those randomly generated.

The process of grouping demographic data is also referred to as stereotyping, introduced by Rich in 1979 (as cited by Shapira et al. 1997). This method has its roots in methods of user modelling used within AI (Krulwich 1997), particularly with Expert and Tutor systems (Shapira et al. 1997) which are discussed in 2.2.1. Again this draws on aspects of the user to create a profile, but then maps the individual to a set of attributes under a given stereotype heading rather than a generated cluster. An example of this is the KNOME support system developed by Chin, (1989 as cited by Shapira et al. 1997).

One of the issues associated with this type of system is the degree of homogeneity within stereotypes. The use of stereotypes assumes that there is a more than arbitrary relationship between users assigned to a particular stereotype. Zhang and Han (2005) tests this by making the assumption that individuals in a given stereotype would share similar levels of knowledge about specific domains, and that by testing the level of knowledge across individuals within a stereotype, the performance of that stereotype can be evaluated.

Their results show that knowledge level predicts the stereotype of librarian far more accurately than that of undergraduate. This disparity highlights the complex relationship between stereotype, the variables used to attribute membership to that stereotype and its relationship to the applied area. Reinforcing that there needs to be a strong conceptual link between the intended application of the profile and the attributes used within the profile. They report similar disparities between the other seven stereotypes they tested.

Like content filtering, the use of demographic filtering produces no serendipitous results, fixing the potential recommendation set. The data gathering process here is explicit and as noted earlier (2.1.2) while high quality (Keenoy et al. 2004), it is also subject to issues of time and trust on the part of the user. Cost also comes into account when commercial data sources are used or experts are brought in to define the attributes of the stereotypes.

2.1.4 Collaborative filtering

The differences between demographic and collaborative filtering (CF) are subtle, as aspects of one have grown into the other. Demographic filtering is closely associated with the process of creating stereotypes, as described above. Stereotypes are groups of users with similar attributes, but within collaborative filtering these attributes are not restricted to being aspects of the individual and tend more often than not to be the opinions of an individual gathered by using ratings data of one sort or another. For example Amazon.co.uk (2009), asks users to rate children's toys on their educational value, durability and fun, on a scale of one to five.

Out of all the methods discussed here CF has attracted the most attention from the research community. It has been applied across a wide range of domains both commercial and non-commercial. Collaborative groups are usually formed by some clustering method, which finds the similarities across the ratings data for a number of attributes. In Figure 2.5 a new user is shown to have the same attributes as a particular group. This group likes a particular set of items so these items are recommended to the new user. While the type of domain still effects the type of associations between user and object, the main area of adaptation has been in the type of mathematical method used to form the collaborative groups.

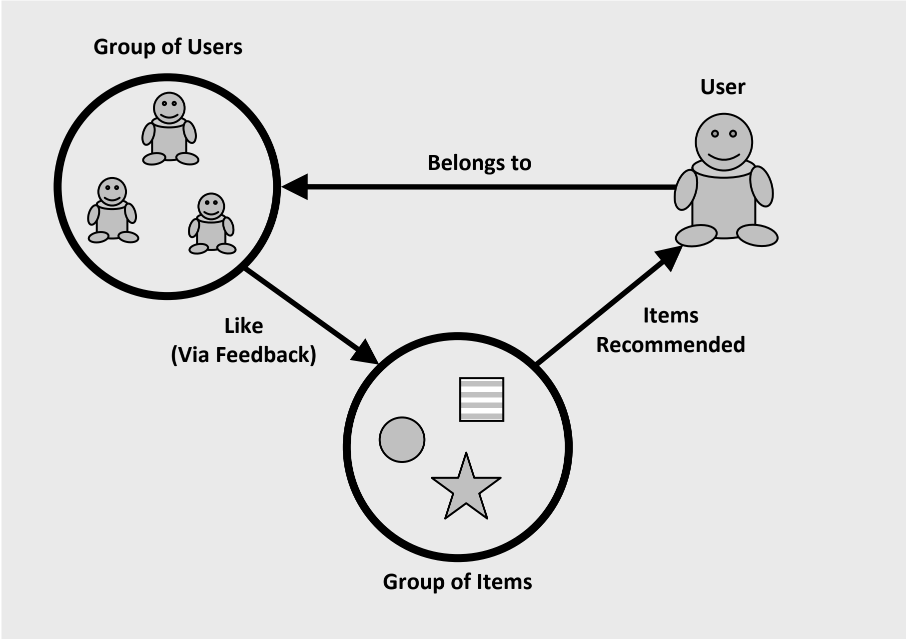


Figure 2.5 Collaborative profiling

Tapestry (Goldberg et al. 1992) is one of the earliest known implementations of a collaborative technique to an IR problem for the purpose of recommending (Im and Hars 2007; Griffith and Riordan 2002). Built as a filtering application for email mailing lists, it uses other users' interests in particular documents as the profile. Users often identify with other readers that share similar interests. Tapestry enables the user to identify other users that they have similarities with. The type of response that these *friends* have towards any particular item is then used as a filtering mechanism on the main user's incoming mail and news items.

The GroupLens system *is similar in spirit to Tapestry* (Resnick et al. 1994) but extends the idea by correlating ratings data across users. When a user views a particular article they are encouraged to rate how much they liked the article from one, low, to five, high. Using the heuristic that *people who agreed in the past are likely to agree again* they correlate the ratings, and use this to assign weights for each user from the perspective of a single user. To predict any single user's level of interest in a document, a weighted average of all the previous ratings for this document is calculated using the user-to-user weightings.

One of the problems perceived with the content filtering method by Shardanand and Maes (1995) is its focus on textual content and the use of keywords. They also indicate that this method fails to capture any concept of quality (Yu 1999) or make serendipitous recommendations. Ringo (Shardanand and Maes 1995) is an early email based music recommendation system, developed in response to these problems. Using ratings data, this system provides some semblance of agreed quality across multiple ratings.

However, there are several fundamental issues with the use of ratings data. It has been noted by many that users are not often consistent in this process. This particular phenomenon is also present in content filtering systems that require users to rate recommendations or purchases on various scales. Ratings based systems can also suffer from a lack of interest or trust from users. As discussed in 2.1.2, it is hard to encourage users to make the contributions, particularly, if the benefits of contributing are not clear, they may perceive it as not worthwhile or even intrusive.

Ringo (Shardanand and Maes 1995) also aimed to solve the problem of serendipitous recommendations, something content filtering fails to achieve. By adjusting the degree of similarity within the cluster process and encouraging fringe elements in all profiles. When clusters contain profiles with too high a degree of similarity the novelty aspect becomes very thin and recommendations will become stagnant for those individuals that like to experience a change in direction.

There are several problems still to be resolved with collaborative filtering. The cold start problem is widely known (Griffith and Riordan 2002; Lee et al. 2001; Ackerman et al. 1997; Zigoris and Zhang 2006; Cohen and Fanb 2000, Umyarov and Tuzhilin 2011) and applies both to new users and new objects added to the recommendation set. As CF is reliant on explicit ratings data by users, items added to the recommendation set have no ratings and thus are never recommended. Some systems have solved this by *borrowing information* from other users to create initial pseudo ratings to kick-start the process (Zigoris and Zhang 2006). Others ask new users to rate a standard set at the point of subscription (Shardanand and Maes 1995; Lee et al. 2003), but again this can be off-putting for the user given issues of trust and time.

Cacheda el al. (2011) recently evaluated a number of different collaborative algorithms suggesting that scarcity of data is one the main issues for these methods.

One of the least discussed, but most obvious problems with systems based on similarities between users or their ratings is the one-of-a-kind problem (Griffith and Riordan 2002). Like a lot of things in life the majority often reap the benefits at the expense of the minorities. For those with highly specific tastes or viewpoints and those with almost randomly broad perspectives, similarities with the general populace are few and far between. While CF sounds inclusive it is actually fairly exclusive to those who follow the popular trends and habits.

A vast amount of research has been undertaken with the view that CF is the solution to the personalisation problem (Griffith and Riordan 2002). Many variations on the basic idea of correlation between profiles have been tried. Methods from IR and AI such as association rule mining (Lee et al. 2001), fuzzy associate memory (Lee et al. 2003), and self-organising map with case based reasoning (Roh et al. 2003). As well as statistical and probability based methods such as Pearson Correlation Coefficient (Shardanand and Maes 1995), mean square difference (Shardanand and Maes 1995) and Bayesian hierarchical methods (Zigoris and Zhang 2006). These methods are applied at various stages in the recommendation and are often combined as discussed in 2.1.5.

The success of CF is highly dependent on the domain it is applied to, which restricts the generated profiles ability to migrate to another domain. Several papers make note of this phenomena including (Lee et al. 2001; Ralph and Parsons 2006; Yu 1999). Im and Hars (2007) investigates this using two different domains, research papers (about 2,000 abstracts) and movies (492 cross genre). They conducted two experiments with university students to identify whether or not the accuracy of CF as a function of the number of users was greater for knowledge domains (research papers) or consumer domains (movies). Their results suggest that the domain of research papers is more heterogeneous than movies, and is more likely to produce accurate recommendations.

Im and Hars (2007) research also suggests that the context of the user's task is important particularly in reference to item ratings. They examined the impact of different search modes, scanning mode (none goal orientated) and probabilistic mode (goal orientated) (El Sawy, 1985 as cited by Im and Hars 2007), on ratings and how this difference transferred itself to the quality of recommendations. The results across both domains show that ratings generated in one search mode and then used to generate recommendations in the other are less accurate than when the mode is the same for both phases. Other factors such as mood, external aspects and context have been suggested by Pennock et al. (2000) as noise that needs to be taken into account when dealing with ratings data.

More recent work on CF has seen a move away from ratings data. The use of text analysis, such in the movie ratings system LET that utilises the review content (Kawamae 2011) as the profile. Liang et al. (2010) is extracting tag data extracted from a folksonomy and utilises a taxonomy to manage the connection between profile and recommendation set. With the increase in smart phone technology recommenders can also consider context and location data within the user profile. Yong et al.'s (2011) recommender system is concerned with delivering relevant multimedia to users on the move. They apply ontology to represent the user preference data and the item data, in this case multimedia information relating to TV, and the context data generated from the mobile location. The purpose here is to recommend TV to user in relation to their context. However, no detail is given concerning how the aspects of the users' location contribute to the recommendation process at a conceptual level.

The issues outlined above show that CF is still very much in its early stages of development (Griffith and Riordan 2002). While many have focused on changing and adapting the clustering method some research has examined how these problems can be surmounted by combining CF with the other filtering methods.

2.1.5 Hybrid systems

CF is most often combined with content filtering (Griffith and Riordan 2002) as they both negate each other's weaknesses to some degree as discussed by Montaner et al. (2003). The lack of subjectivity (user preference) and novelty recommendations exhibited by content systems are neatly countered by CF's use of user ratings and user clustering respectively. The lack of ratings for new items (the sparsity problem) introduced to the recommendation set in a CF system can be countered by applying content filtering.

Basu et al. (1998) present an early movie recommendation system that incorporates both collaborative and content features. The collaborative aspect is viewed as two sets of ratings data. Those movies liked by any single user and those users that have rated a movie the same. This data is then used to generate rules using the inductive learning system, Ripper (Cohen 1995; 1996 as cited by Basu et al. 1998). The content aspect is extracted from an Internet movie database and includes data such as actors, directors, keywords, genres, and title. These are combined with the collaborative aspects and a new set of rules are generated using the Ripper system. Their aim is to use the content features to inform the collaborative grouping of users. Instead of a group being defined just because they like a particular movie, the group is defined because they also like movies of a particular genre.

This created four distinct methods for comparison: basic recommender using simple collaborative filter; CF with Ripper (rule generation); Ripper with simple content added to the CF; and Ripper with content informed CF. Their results show that adding rule generation and simple content, reduced precision of the recommendations, whereas the content informed CF improved precision markedly. Good et al. (1999) and Weng and Liu (2004) show similar improvements in precision and recall for combined methods.

More recently the content aspect of these combined methods has been enhanced by the use of context related data in the form of ontology (Anand et al. 2007). Effectively the system takes into account the purpose of a users visit and constructs a profile using data only from contextually related visits. This reduces the noise of using all the data within a user profile, which often lowers the accuracy of recommendations. They also discovered that this technique countered the sparsity problem that more traditional CF methods suffer from.

Massa et al. (2010) have used content mining of user blogs, to analyse user interests, and inform the CF aspect of their recommendation system. The resulting profile is then applied to both news articles and television news. Although closely related these are effectively two unique domains which they have successfully used a single profile for the recommendation process. They do note the success for individual users is still reliant on the heterogeneity across blog data and news data.

While these works indicate that there are improvements in the measures of precision and recall to counteract some of the issues with content and CF filtering, they by no means solve all the problems faced by filtering methods.

2.1.6 Personality within recommenders

Hu (2010) has developed a collaborative profiling technique based on the FFM for music recommendation that has had some degree of success. The evaluation process is focused on user perceptions and the findings suggest this method of profiling is more amenable to a domain novice and medium user than experts. It also suggests that the profile works well under sparsity conditions. The future work is looking to see how this can be applied across domains. Golbeck (2011) has also been applying FFM model but this time in the context of social media. Here it is the personality that is being predicted from the demographic data pulled from Facebook pages.

Another way the concept of personality has been used within profiling systems is within group recommendation (Recio-Garcia, 2009). This recent development recommends items using CF to groups of users. Quijano-S´anchez et al. (2010) have used personality, represented here by the Thomas-Kilmann Conflict Mode Instrument (Thomas and Kilmann 1974 cited by Quijano-S´anchez et al. 2010) to define roles within the group dynamics of identifying a movie suitable for all. Although this instrument represents some psychological aspects it is not a true representation of personality.

2.1.7 Summary of profiling techniques

There has been a variety of techniques used to develop and improve the filtering algorithms in recommendation systems. From simple content based methods to sophisticated data mining and rule generation processes. Initially only descriptions of products or users were used. This developed into the CF method using the similarities in both users and products. The importance of context and the use of observed user behaviour were noted in one of the earliest systems developed (Lieberman 1995) and has recently become a renewed avenue of exploration (Mobasher 2007). Im and Hars (2007) suggests that purpose has a high effect on ratings data and thus without capturing this ratings may lead to false recommendations, emphasising the need for more sophisticated user, context and item profiling. Also more recent developments are moving towards the use of psychological aspects such as music recommendation system presented by Hu (2010).

Across all the methods discussed here there are various unresolved issues. The following summarises those identified in this review and the filtering methods they are most often associated with:

Implicit vs. Explicit profile data generation (All)

- The benefits and disadvantages between these are a source of disagreement. Implicit infers current behaviour or attitude and requires no action on the part of the user. Whereas explicit requires the user to perform some input but is generally viewed as a more accurate representation of the user’s perceptions.

Encouraging users input (All)

- Linked to the explicit aspect of data generation, input requires considerable commitment and time on the part of the user. This is often cited as a barrier to the success of ratings based systems.

Trust and security (All)

- All personal data, whether implicit or explicitly generated, is subject to user mistrust and threats of identity theft, and misuse.

Changes over time (All)

- This is more often than not attributed to content and demographic as they do not have a natural user feedback mechanism like ratings based CF. However some systems do have degradation method for old profile data where others retain all data to use all the time.

Over specialisation (Content)

- Content filtering is limited to recommending only the most similar items, which is perceived as over specialising in specific attributes of items.

Lack of serendipitous (All)

- Linked to over specialisation, content systems cannot recommend novelty items that may be related by tenuous links.

Objective vs. Subjective (Content/Demographic)

- Content and demographic systems are perceived as being objective, in that they use actual user or item attributes. Whereas collaborative systems are based on subjective data that is perceived to be the opinions of the users. The lack of subjectivity in the former is criticised for not representing the user accurately.

Using Experts (Demographic/Stereotypes)

- The development of stereotypes may involve the use of experts to define user classes. This is time consuming and costly, although they often produce more accurate groupings than clustering methods.

Consistency in user ratings (Collaborative)

- One of the most criticised aspects of CF is the reliance on individual users being consistent in their ratings data. Mood and task context have been shown to have a strong influence on how users rate items.

Cold start (Collaborative)

- Although CF is perceived by many as the way forward one of the biggest barriers it faces is the fact that it takes time for a profile to be built and become of use to any particular user.

Sparsity of data for new items (Collaborative)

- Similar to the cold start problem, which is from the users perspective, is the sparsity of ratings in growing recommendation collections, resulting in unrated or low rated items never being recommended.

Heterogeneity of domain (Collaborative)

- Research has shown that different domains, recommendation collection, have different levels of cohesion between individual items from the perspective of user interest. This implies that some domains are just not suitable for CF application.

Cross-domain compatibility (All)

- Profiles developed in one domain have little or no utility when applied to another domain. This is generally to do with service-selected attributes but also suggests that a user's preferences don't work the same way in different domains.

2.2 Personalisation within eLearning

Section 2.1 considered profiling that was content or product driven. In highly competitive markets the need to target audiences more accurately makes good business sense. However, financial gain is not usually the primary motivator for good educational practice, and there must be other more demanding factors to the use of personalisation within this area.

In 2005, the London Knowledge Lab invited a group of people to examine the rationale behind the need for personalisation (Freitas and Yapp 2005). The publication resulting from this workshop, *Personalising Learning in the 21st Century*, presents a variety of views and issues across all levels of education. However, the most intriguing piece is the foreword written by Tim Brighouse. He describes those actions that, in the physical world, are perceived as being pertinent to a successful and intuitive teacher applying personalisation. What is emphasised is that it is not just the act of tailoring actions to individuals, but the act of making those actions personal as opposed to impersonal.

This perspective highlights that personalisation within education is applicable on several levels. It is not something solely attributed to the use of technology but is something that has been on the fringe of curriculum and content development for some time. Changes in policy, teaching methods and organisational structures are all required to develop a truly personal and personalised experience for the learner. How these changes impact on learners and educators is something discussed at length in the papers presented in Freitas and Yapp (2005). Technology is just one route towards the goal of personalised learning, one which has the scope to retain mass delivery while focusing on the individual.

For decades it has been well known that a one-to-one process between student and teacher is four times more effective than the traditional classroom (Bloom, 1984 as cited by Chou et al. 2003). Couple this with the ability for those same students to learn anything, anywhere and anytime and the vision of life-long-learning steps closer to reality. However, there is much that needs to be researched and while many educationalists still perceive technology as a burden to teaching they find little time to explore the potential benefits.

The next part of this review examines how educational technologies have implemented personalised learning via the application of a user profile. Personalisation of both content and learning paths have been investigated in the development of intelligent tutor systems (ITS), explored in 2.2.1, and learning companions in 2.2.2. Both of these areas have contributed to online developments and the more extensive area of research Adaptive Educational Hypermedia (AEH), reviewed in 2.2.3. More recent work has taken onboard the methods and profile techniques developed within recommendation systems discussed in 2.2.4. Key texts consulted during this review included: Li et al. (2008); Stephenson (2001); Garrison and Anderson (2003) alongside those cited.

2.2.1 Intelligent tutor systems

Intelligent tutor systems (ITS) aimed to replace the tutor and predominantly used methods from AI to control the learning process. Many such systems contained models of the user that are not dissimilar to those used within the recommender systems discussed in 2.1.1.However ITS models tended more towards the use of user related attributes such as: personal characteristics (Cheung et al. 2003); learning goals (Cheung et al. 2003; Kabassi and Virvou 2004); and learning history (Marin et al. 2004; Cheung et al. 2003).

Carbonell (1970 as cited by Chou et al. 2003) developed the Socratic tutor system which simulated the Socratic Method of learning via debate (Halff 1988). It focused on the learning of declarative knowledge relying on a large store of information, in this case modelled within a semantic net, from which to pose questions to the learner. Other systems supported the learning of procedural skills and knowledge (Andriessen and Sandberg 1999).

The quality of learning from these systems is often criticised. Learning is rarely a solo affair and ITSs provide no means for peer collaboration or alternative student tutor roles. This lack of practical application highlighted by Andriessen and Sandberg (1999) contributed too many moving away from this area of research.

SmartTutor (Cheung et al. 2003) attempted to address some of the concerns raised by early ITS research, by embedding identified support processes used by tutors when teaching. They also use subject tests as a means of profile updating, which then inform the advice given to students regarding their next best course of learning action. In this case, the learner model is generated using both explicit data from surveys, and implicit data from interactions. Their evaluation of the SmartTutor application shows that it was generally well received and helpful to the majority of learners.

Another system, Web Intelligent Trainer (Web-IT) (Kabassi and Virvou 2004) reasons about a user's goals, and assesses whether or not the users current actions are different from the systems expectations. To establish the effectiveness of this system they compared the decision process to that of ten human tutors. Their results show that 68.81% of the decisions matched the human counterpart.

The personalisation aspect of Web-IT is applied using the *relevance principle* of Sperber and Wilson (1986 as cited by Kabassi and Virvou 2004), which states that humans pay attention only to information that is relevant to them. This is applied in two ways. Firstly via utilising the data within the user profile to decide what and when the information should be presented. Secondly, by dynamically using the user's current environment for the file and folder layout on screen, as the example. These methods clearly capture a broad picture of personalisation, firstly by considering timeliness for the user and secondly by considering the learning context.

One of the attributes that Web-IT utilises is that of age, which is assumed to have a significant effect in the user's ability to remember. This attribute determines the frequency for re-visiting the learning information. Psychological research suggests that this is not necessary true for all and is less of an impact than generally assumed (Merriam et al. 2001; Tennant 2006). This highlights a common tendency in many profile-based systems to make generalised assumptions about the attributes that they capture and oversimplifying their impact on the system's decision process.

A more recent example of ITS is SQL-Tutor (Mitrovic and Martin 2004) which focuses on adaptive problem selection based on a learner specific knowledge. Their research compares two methods for selecting problems for learner's to undertake. One method calculates the difficulty of the problem for the learner based on their learning history. The other uses a simple difficulty rating assigned by an expert. The two methods were tested with 59 students; the results indicated that the first method suited a much broader range of learner abilities than the second. This suggests that attribute values that are derived from an individual's interaction with the system, are more useful than those values assigned to attributes from outside the system. This in turn suggests that the use of such methods as CF (2.1.4) and ratings data with educational systems might not be beneficial.

Even though research into intelligent tutors is still undertaken, initial enthusiasm has waived. Researchers questioned the feasibility of this direction, as the all-knowing tutor informing the student repeatedly, is a perspective of education that had long since been displaced. By the mid 1990's researchers in eLearning were looking for a much broader and supportive style, which lead to the development of learning companions, discussed next in 2.2.2. However this review has uncovered some of the issues with user modelling and the types of attributes selected. Mitrovic and Martin's, (2004) work also suggests that the use of attributes generated from users other than the profile user are less effective, which may rule out methods such as CF for educational purposes. This is further discussed in 2.2.4.

2.2.2 Learning companions

ITSs utilised a simplistic model of the tutor-student relationship, modelling the domain knowledge, student learning history and a restrictive set of learning strategies. As mentioned previously these systems are a solo affair and fail to provide the student with any collaborative support which is important from a pedagogical perspective (Andriessen and Sandberg 1999). To rectify this research began to investigate the idea of modelling a companion for the student to accompany them on their learning journey. These systems are referred to as Learning Companions (LC) and can take on a variety of forms and roles, including that of tutor.

One of the first developed was in 1998 by Chan and Baskin (as cited by Chou et al. 2003) which comprised of three characters: the student, the computer based learning companion and a computer tutor. Their idea is to provide the student with a competitor, something to collaborate with, as well as something to make suggestions, as in an ITS. They also put forward the suggestion that the human student could take on the role of a tutor and teach the learning companion.

Learning companions have simulated some interesting relationships within the learning environment. As AI based agents they enable researchers to create companions with a variety of characteristics. Hietala and Niemirepo, (1998), developed a system named EduAgents where the agents have varying levels of expertise in the topic. Those that are knowledgeable are classed as strong companions and those that are less knowledgeable as weak companions. Their research showed that interaction with the different types of companions varied over six interaction sessions. Weak agents were preferred in the earlier sessions and stronger ones were preferred towards the end. There was also a preference for the learning companions based on the student being introverted, preferring strong companions, and extroverted preferring weaker companions, particularly as the tasks got harder.

Others have given the LC the characteristics of a troublemaker (Frasson et al. 1996 as cited by Brusilovsky and Peylo 2003) as a support strategy. (Vizcaino and Boulay 2002) have used a LC as a simulated student in a collaborative workspace to encourage students to remain focused on the task and give advice. This system was fairly successful although there were some instances of misinterpreting the student's behaviours. Betty's Brain developed by Viswanath et al. 2004 utilises a learning by teaching approach. Students teach Betty via a visual drag and drop concept map. They can then pose questions to the companion, which reasons out an answer with the information given to it by the students.

Learning companions are a more intuitive extension of the intelligent tutor systems. However, as their focus has moved to a more flexible approach, based on interaction rules, there has been less emphasis on modelling the user within these environments. Thus while they encompass a better range of modern pedagogical practices they have moved away from the idea of profile based personalised learning.

2.2.3 Adaptive educational hypermedia

Hypermedia systems have been utilised in education for some time, but generally followed the one-size-fits-all (Cristea 2004, Hella and Krogstie 2011) attitude that did not complement pedagogical ideals. The adaptation techniques developed within ITS (2.2.1) and LC (2.2.2) promised to enhance the flexibility of hyperlink and create a more responsive learning environment. The combining of these areas spawned a bewildering range of Adaptive Hypermedia within an educational context. Brusilovsky and Peylo (2003) devised two taxonomical models (classic and modern) of adaptive and intelligent web based technologies, which have been combined and presented in Figure 2.6. This maps the five main adaptive web-based technologies being explored by eLearning research (rectangles) to their primary influencing technologies (ovals).

The rationale for presenting this taxonomy here is quite simply a matter of time. Prior to the advent of the WWW, ITSs (2.2.1) and LCs (2.2.2) were designed to be used by individuals from a single terminal. This perspective delayed the move to using adaptation within hypermedia until recently (Cristea et al. 2006). However, the move to develop web based systems refreshed interest in utilising AI-based research. For example, curriculum sequencing was developed by ITSs to guide the learner through a body of deliverable content. This process is as applicable to a web-based system as it is to a stand-alone application.

Figure 2.6 highlights the range of techniques that are currently being employed in the web-based arena by educationalists. Quite often several of these will be combined within a single system making it difficult to precisely place any one application. Having already explored aspects of intelligent tutoring via ITS (2.2.1) and intelligent collaborative learning via learning companions (2.2.2), these areas will not be re-visited here. Rather the focus is on the use of adaptive hypermedia (AH) for presentation and navigation, and the learner modelling used to achieve this. Adaptive information filtering is discussed in 2.2.4.

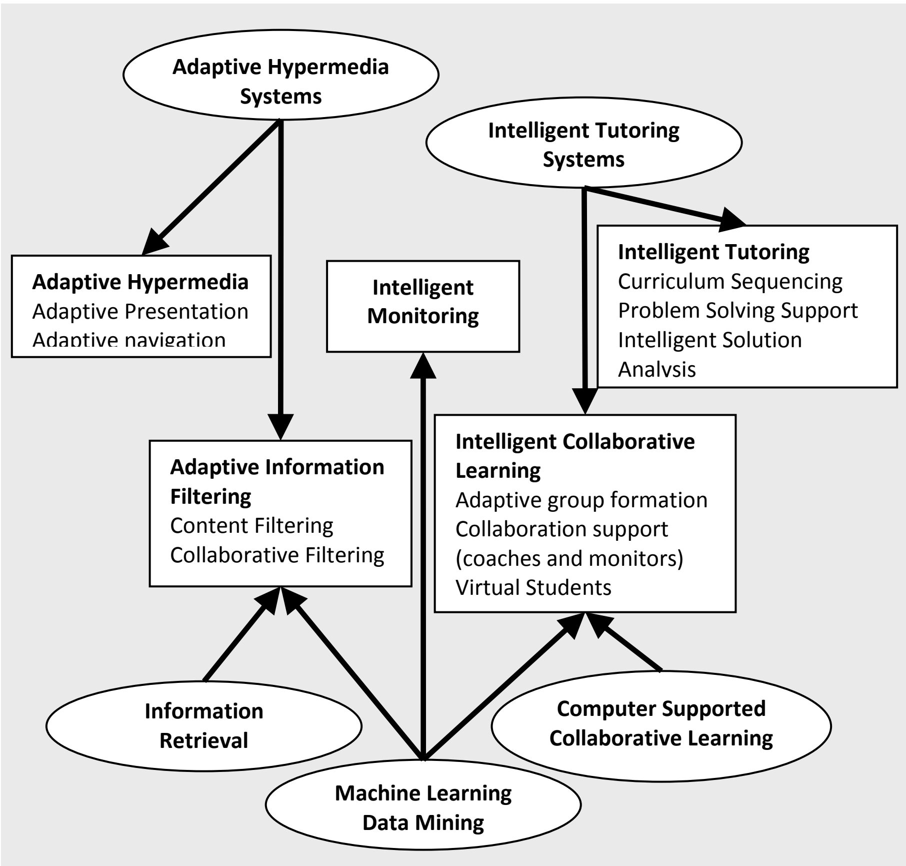


Figure 2.6 Taxonomy of Adaptive and Intelligent Education systems adapted from Brusilovsky and Peylo 2003

AH within education focuses on three areas of adaptation (Wolf 2003; Keenoy et al. 2004): navigation adaptation; content adaptation; and tool adaptation. The first adapts hyperlinks to guide the user through course content or to present alternative tools and materials. The second creates content that is specifically tailored to some aspect of the user's profile. This can be done on the fly by assembling various components or by selecting from a predefined set of alternative text and media. Content adaptation can also be applied to quizzes, suggested further reading, advices and warnings (Xu et al. 2002). The final area is that of tool adaptation, which has only recently been explored in the context of hypermedia but has frequently been applied in more traditional educational applications.

In a review of these processes Keenoy et al. (2004) identifies a number of methods used to adapt navigational links to the user: activity history, observed behaviour (Grigoriadou et al. 2001); test results (Grigoriadou et al. 2001) or other artefacts of understanding; and course prerequisites. For content adaptation several methods are used: what is new to the learner; the next item within a sequence; additional content within the same topic; or variants of the same content dependent on the learner's skill level or learning style (Wolf 2003; Abdullan and Davis 2005). Tool adaptation can be based on all the aforementioned methods as well as more general user defined preferences.

An example of the application of these techniques is iWeaver (Wolf 2003). This system adapts both content and navigation to the learner's learning style. The Dunn & Dunn learning style (Dunn et al., as cited by Wolf 2003) inventory is used to establish the user's learning style which is used to tailor the content availability from the interface. However, unlike many other learning style systems (Stash et al. 2004; Grigoriadou et al. 2001), iWeaver does not assume that the learning style is permanently fixed or preferred for all learning experiences. The interface enables all tools and content variations to be available to all learners via the use of expand buttons. Their rationale for initially presenting a limited set is in consideration for the effect of cognitive overload on the user.

In addition to learning styles Inspire (Grigoriadou et al. 2001) also considers the learners' ability and observed behaviour to tailor the learning path. Like many systems (Xu et al. 2002, Ruelland

and Brisebois 2002) it uses online quizzes to assess the current ability of the learner. In comparison to iWeaver, Inspire uses the Honey and Mumford (as cited by Grigoriadou et al. 2001) learning style. This use of explicit data gathering does not appear to generate the same concerns that are prevalent with recommender systems (2.1.1). They appear to be quite accepted within an educational context.

Within Inspire (Grigoriadou et al. 2001) content is pre-structured using three levels of performance, *remember, use and find* that support learning from initial contact to application in broader scenarios. For each learning concept the user is graded as *insufficient, rather sufficient, almost sufficient and sufficient* according to their test results. These grades then determine which of the three performance levels content should be presented from. The learning styles dictate the flow of content rather than the type of content, so the same material can be presented to learners with differing styles. This negates the need for the variety of media representations as used in (Wolf 2003). Effectively this process determines the learning path of the learner. When compared to the attributes captured within recommender systems (2.1.1) there is a notable increase in the level of complexity.

Adaptation using learning styles has become a key area of research (Cristea 2004). The two systems outlined above show how this method of user representation can be used within an AH system. Stash et al. (2004) reviews a further six systems that utilise learning styles as part of their adaptation method. From this they draw out two particular problems. Firstly the use of learning style inventories assigns learners to stereotypical groups that are not updated during interaction with the system. Secondly the systems reviewed utilise one particular learning style theory, of which there are many (Alkhalifa 2005), and do not enable tutors to choose the style system they wish to use.

Like recommender systems (2.1.1) the driving force behind Adaptive Educational Hypermedia (AEH) development is the model of the user (Cristea 2004). Learning styles is not the only attribute to be captured, as is evident in INSPIRE the example system presented above. Others suggest: goals (such as grade required or knowledge to achieve) (Stash et al. 2004; Keenoy et al. 2004); current knowledge level in specific domains (Zarraonandia et al. 2005; Stash et al. 2004; Keenoy et al. 2004); user preferences (Stash et al. 2004); educational background (Zarraonandia et al. 2005; Stash et al. 2004; Keenoy et al. 2004); interests (Stash et al. 2004; Keenoy et al. 2004); hyperspace experience (Stash et al. 2004); location and language (Keenoy et al. 2004); other physiological and psychological characteristics (Zarraonandia et al. 2005).

Feedback processes vary widely and frequently fail to update the learner model. Particularly with early systems the focus tends to be on the current session and visited content. The main mechanism for feedback is testing the learner's knowledge. A criticism of learning style systems is the assumption that once identified the style does not change (Wolf 2003) although some do enable the user to change this for a given session (Grigoriadou et al. 2001; Abdullan and Davis 2005). Zarraonandia et al., (2005) suggests that even an ideal set of characteristics captured in the learner model will fail if time and environment are not taken into consideration.

As with some recommender systems there are issues regarding cross-domain application due to the need for large knowledge bases. Access to the WWW and open educational resources (JISC 2010b) is combating this to some extent. Another criticism is the lack of standards used in many of these systems. Zarraonandia et al. (2005) reviews this problem across eight systems and finds that only a single system applies standards to the domain and user adaptation models. While standards do promote interoperability part of the lack of uptake is due to no single standard expressing the depth of learner attributes that is utilised by AEH systems.

Compared to recommender systems AEH learner models capture a much broader range of user characteristics, some of which are psychological in nature. Observation of interaction is common to both but how this is interpreted is quite different. Recommenders treat a visited link as a preference and AEH treat it as a step along a learning pathway.

2.2.4 Recommender methods within eLearning

Many of the systems reviewed so far have made some form of recommendation to the student. However the interest in utilising the methods developed within the recommender community has been slow to take up. This might be due to a difference in perceptions, evident from the type of attributes in AH and AEH systems. Alternatively, recommender attributes have as yet not found their niche within these systems. Within education the concern is with individual differences where as in recommenders the focus is on finding broad similarities.

In Brusilovsky and Peylo's (2003) review of hypermedia technologies in 2003 only two educational systems, one content based, and the other collaborative, were identified as using a recommender filtering technique. Since then interest has increased and several other developments produced, such as described in Chen et al. (2005); Chen et al. (2004); Zaïane (2002); Yang et al. (2005); Liang et al. (2006); Wang and Li (2006). Wolf (2003) has also indicated an interest in moving in this direction with the iWeaver development, by expressing an intention to explore both content and collaborative profiling techniques with the use of ratings data in future work.

Filtering methods are generally applied within an educational context for recommending course materials. Chen et al. (2005) uses CF based on ratings data generated course materials. In this instance, the filtering method has been adapted by taking into account the learner's current ability, which is evaluated at the same time as the ranking data. The course materials are each assigned a difficulty level which is matched to the current ability of the learner. This is an interesting way of combining methods from recommendation and AH systems, although this does produce a narrow representation of the learner. The authors also concur with research in recommender systems that it takes many learners' ratings for the system to become stable and recommendations reliable.

Zaïane (2002) disregards CF as a means of course recommendation on the grounds that course materials are generally not accompanied by ratings data. However, they utilise a similar process by mining weblog access patterns of many users, to define rules for specific situations. The example given is a recommendation for the learner to read some particular material before undertaking a test. This rule has been devised from the previous action of other learners. It is assumed that the success rate, in this instance, of previous learners on the test, is associated with their previous actions. Again, this reduces the learner representation to a minimal set, only useful within this scenario.

Liang et al. (2006) uses content and collaborative methods to generate two lists of top five recommendations for the learner. The first list is most popular courseware within their area of study based on ratings data inferred from user behaviour while viewing learning material. The second is generated from the learner's similarity to other learners. What is interesting here is rather than just using the ratings data to identify users with similar interests they have used demographic and course topic information as well.

More recently a number of developments have used the personalisation aspects of recommenders to create more aware systems. Within education the focus is very different to that of commercial systems and profiles have broader and often more psychological based attributes. Gasparini et al. (2010) is utilising recommender technology to produce personalised links to learning content. The user profile contains data relevant to the users' technological, pedagogical and cultural requirements as well as other personal aspects. By modelling a user's current scenario across these areas the system can adapt the web interface for navigation to suitable content.

Brut and Sèdes (2010) uses a hybrid recommender using content and CF to capture student's navigation through a document set. This is then represented as ontology based profile to for future recommendations. Also concerned with content navigation and the use of ontology based profiling Zhuhadar and Nasraoui (2010) have enabled the students to provide feedback on the relevance of content that is then added to their profile.

Buder and Schwind (2012) consider the role of recommenders from a psychological perspective attributing them the qualities of collective responsibility, collective intelligence, how they preserved user autonomy, providing guidance, and personalisation. Their idea is that these qualities sit well with education, but warn against the transfer of recommenders to a learning environment on a one-to-one basis. They noted that there is a growing concern about the focus on the technical evaluation of recommenders rather than more user-orientated approaches, such as the psychological aspect taken by their work. They examine a number of systems and express some concerns over the issue of directly transferring these commercial systems to an educational setting. One particular point was the suggestion that while commercially based recommenders try to reduce the amount of user interaction with the system by using implicit methods, educational systems should actively encourage users to engage directly in the process.

The view that educationalists have of individual learners is far more detailed than that of general recommender systems. There is a danger here that the use of recommendation techniques will markedly reduce the complexity of learner models that have been developing in AH systems. However, some systems are effectively combining the basic process of filtering methods with the complexities of AH based user models, as described above in Chen et al. (2005) and in Liang et al. (2006). These may as yet feedback into recommender research and enrich their profile development.

As discussed in 2.1.7 filtering techniques have many counteracting problems which are not alleviated with their application to educational systems (Chen et al. 2005). Others suggest (Im and Hars 2007) that a homogenous corpus is required in order for CF to work effectively. Using CF within an educational system may require learning content to be subdivided to such an extent that it becomes unfeasible for delivery to learners. This and other problems have yet to be fully explored within eLearning applications.

2.2.5 Summary of personalisation within eLearning

Intelligent Tutor Systems (ITSs) and Learning Companions (LCs) have both made important contributions to the development of personalised learning. While as stand-alone systems very few are still being developed the lessons learned in these areas have fed directly into today's primary development area that of AEH.

While AEH systems often present recommendations to learners they differ markedly from the recommender systems discussed in 2.1.1 as follows:

- AEH systems encompass a greater variety of user attributes that are far more descriptive of the user. This increase in complexity is often reflected in the adaptation process with the use of AI techniques and extensive domain modelling.
- AEH have moved beyond just providing a selection of recommendations to focus on personalising the sequencing of content. For example the use of learning styles these systems can choose from a variety of content types about the same topic or present the same content in different sequences.
- Another difference is that recommenders draw from an existing collection of objects or content, whereas within AEH, content is often specially designed which adds further costs to the development.
- Explicit data entry is less of an issue in AEH systems than within recommender systems. Although Keenoy et al. (2004) highlights that there are the some issues regarding explicit and implicit data. They suggest that a mix is the best option. This enables users to supply as little or as much as they want.

Although there is a great deal of personalisation applied within eLearning using a wide range of user attributes, none of the systems explored truly capture the whole individual. As with recommender systems the tendency is to consider the domain and functionality first in defining the user's attributes.

2.3 Identifying the problem

From the review of profiling within recommenders (2.1) several issues were identified and presented in 2.1.7. These issues are currently limiting the profiles generated to a single domain and groups of users with common characteristics. Neither of these situations is acceptable when considering profiling from a broader perspective, such as education. Speculation at this stage suggests that the use of a personality-based profile will account for a broader range of user variation as well as being able to accommodate multiple domains.

The review of personalisation within eLearning systems (2.2) shows that educational systems are currently exploring the use of psychology related aspects, such as learning styles, goal orientation and abilities, with a focus on individual differences. This supports the idea of exploring psychologically related variables, such as personality, further. Section 2.2.4 examines the means in which recommender techniques have been applied within an educational context; these have had some degree of success.

From this, this thesis puts forward the notion that to truly apply a profile within a complex domain, such as that of education, those basic traits and tendencies that make an individual who they are, need to be captured. To achieve this it is suggested here that the most likely place to identify such a profiling method is within the research and theories of personality psychology. It is proposed that the concept of personality is better suited to create the foundations of a cross-domain profiling method that can account for both individual differences and group similarities. The primary idea being that the root of an individual, their personality, can be linked to a set of most likely actions within a given context. In reverse this puts forward the notion that personality can be used to predict the most likely behaviour within a specific context.

Chapter 3
Modelling personality

The aim of this research is to develop a generic model of personality that can be used across multiple domains to recommend many different objects or services in different software environments. To that end this chapter seeks to explore those elements described within theories of personality that can be represented in such a generic computational model. The process of identifying such elements of personality involved the review of several prominent psychological theorists who were identified throughout the process of enquiry. As each theorist was reviewed the concepts and elements of personality they described were considered from a computational perspective and whether or not they added value to the developing model.

The structure of this chapter follows the process undertaken in developing the model, identifying several key decision points, presented in 3.2. The method of selecting the psychological theories was an intuitive one determined by those concepts or relationships described by the current theorist under review. Starting with Freud, who is perceived as the father of modern psychology, the literature review moved through a number of different psychological schools of thought as described in Schultz and Schultz (2005).

There are a number of ways in which the literature could have been selected and reviewed, some of which are more systematic than others. A single school of thought could have been reviewed, but this may have limited the underlying understanding of personality psychology as a whole resulting in a less adaptive model, or lead to a dead end with no computational model. A range of criteria could also have been developed to determine which theories were most likely to contain computational elements, but again this may have constrained the developing model, not allowing those concepts with computational potential to be identified from within more holistic theories.

The rationale for presenting the chapter in this way was simply to ensure that these key decision points and the impact they had on the developing model were explicitly discussed and related to the relevant literature. By presenting the current state of the model at the end of each of these key points, the reader can readily track the addition, movement and adaptation of each element. As the selection process for each psychologist reviewed was intuitive, different decisions could have resulted in the generation of a different style of model. Although on reflection it is highly unlikely that the core elements identified, such as the concept of dynamic and static elements, would have been overlooked given a different research path. It is also possible that the research path that was followed has overlooked elements that could be included and could be of benefit to computational developments. Given the range of personality psychology research available and currently being undertaken, it is inevitable that this review be constrained in some way. For this reason it was decided that the process was as equally important to present as the resulting models.

3.1 What is personality?

As there are many variations on the concept of personality, there are also many definitions of personality. The word personality comes from the Latin Persona meaning mask, what is shown in public but also implying that some things remain hidden behind it (Hergenhahn 1980). Another way of looking at it is as an organised, active force, which affects the body in consistent patterns (Carver and Scheier 2004). It is concerned with both individual differences and the similarities across individuals (Pervin 1989). With so many different aspects to cover it is difficult to identify a single definition. Many theorists tend to develop working definitions that just captures what they are currently exploring (Reber 2001).

The following are three working definitions. The first is very general and has been adapted from Gordon Allport, who resides in the dispositional perspective. The second is a more focused definition, also from the dispositional perspective, by Raymond Cattell, which shows the differences theorists can have even when their aims and assumptions are similar. The final one is from the cognitive perspective, focusing on how an individual is the sum of their knowledge and how they use it.

> *Personality is the dynamic organisation within the individual of those psychophysical systems that determine his characteristic behaviour and thought.*
>
> *(Allport 1937 cited in Carver and Scheier 2004).*

> *A trait may be defined as that which defines what a person will do when faced with a defined situation.*
>
> *(Cattel, 1979 cited in Pervin 1989)*

> *We place social intelligence at the centre of this personality theory and define it as the concept, memories, and rules -- in short, the knowledge -- that individuals bring to bear in solving personal life tasks.*
>
> *(Cantor and Kihlstrom, 1987 cited in Pervin 1989)*

These definitions start to highlight some of the structures and processes that different theories utilise to explain personality. The first one highlights the idea that personality encompasses both mental and physical aspects with the use of psychophysical. It also indicates how differences in personality can be judged, via behaviour, thoughts and feelings. The second definition is more focused on a particular aspect of personality, the trait. This is used to identify individual concepts linking behaviour to particular situations. The final one more focused still, suggesting only that which we store in the brain as being linked to the ways in which we interact in a given situation.

For the purpose of this research personality is defined as: *those psychological elements that can be described in terms of having a specific attribute or attributes that can be linked to particular type of behaviour*. For example Freud's concept of the subconscious is not described in specific terms in which it can be associated with individual behaviour. Whereas Allport's concept of attitudes are defined as having a positive or negative response to a person, place or thing that influences the behaviour while interacting with said person, place, or thing.

3.2 Modelling psychological theories

When considering what personality is and how elements of it can be identified, it becomes clear that behaviour is the observable aspect. The link between personality and behaviour has been established by many theorists (Carver and Scheier 2004). The question is which elements influence which aspects of behaviour. As behaviour is the observable aspect it may be worth considering how behaviours differ from one individual to the next or from one scenario to the next. What are the questions that need to be asked of behaviour, which may lead to elements of personality? In considering this the following questions were initially used to examine the elements of personality identified within psychological theories:

1. **What** will the action be?
2. **Where** (place) will the action happen?
3. **When** will the action take place?
4. **Who** (person) will be involved in the action?
5. **Which** (object/s) will be involved in the action?
6. **Why** will the action be performed?
7. **How** (the process) will the action be performed?

If a computational model of personality can answer all these questions, it theoretically should be able to determine all manner of behaviour, in any given situation. For example: *Why?* suggests a reason for the behaviour, perhaps the motivation behind the behaviour; *When?* suggests that time is an aspect of behaviour, possibly indicating some aspect that dictates when a behaviour occurs; *Which?* and *Who?* indicates that the behaviour interacts with objects other than the self as well as a choice or preference to be made. Whether you are observing a complex behaviour pattern, such as an individual's process of learning a new topic, or a more simplistic preference for one product over another, all these questions come into play, even if the answer to them is none.

During the review the questions stated above are used to consider if and how an element of personality can contribute to a computational model. The first stage (3.2.1) of development examines elements relating to why a particular action/behaviour is performed. The second stage (3.2.2) looks for the source of personality. Is it based in nurture? Are the answers to these questions found in the sum of our experiences? Alternatively, is it based in nature? Are the answers linked to some physical internal constructs? The third stage (3.2.3) considers how cognition may be interacting with personality and how it maybe effecting the answers to these questions. The final stage considers trait and type style theorists (3.2.4) in more detail and how some of the concepts around these theories have a computational structure more suitable to this development than other theory types.

A number of personality theorists were analysed during this review. Where appropriate direct works were reviewed from; Henry Murray (Murray 1938), Gordon Allport (Allport 1955, Allport 1973), Hans Eysenck (Eysenck 1952; Eysenck 1967; Eysenck 1970, and Barrett et al. 1998), Raymond Cattell (Cattell and Child 1975, Cattell and Kline 1975, Cattell 1971, Cattell et al. 1985, and Cattell 1992), George Kelly (Kelly 1963) and Paul Costa and Robert McCrae (McCrae and Costa 1999; McCrae and Costa 2003; McCrae and Jr. 2004, Wiggins 2003 and McCrae and P.John 1992). The following: Sigmund Freud, Alfred Adler, Carl Jung, Albert Bandura, Carl Rogers, Abraham Maslow, Burrhus Skinner were reviewed from a selection of personality psychology texts including Malim and Birch (1998), Schultz and Schultz (2005), Carver and Scheier (2004), Ewen (1993), and Cooper (1998).

3.2.1 Identifying the why

Why do people do what they do? Is this because of their personality or is it something separate that influences the link between behaviour and personality? Very few personality theorists have not included some rationale of purpose to human behaviour within their theory.

Sigmund Freud (Schultz and Schultz 2005, Carver and Scheier 2004) believed that the basic element of personality was instinct, the driving force behind all behaviour. Instincts are internally generated and are transformed in the mind into wishes. These wishes then motivate behaviours that satisfy the wish aiming to reduce internal tensions. It is the reduction of internal tensions that is the primary goal of behaviour.

Alfred Adler (Schultz and Schultz 2005) focused on this concept of tension, but as a positive force, one that needs to be increased not reduced. Adler believed that tension indicates the level of personal striving within an individual. Without this striving people cannot achieve perfection, perceived as the ultimate goal. However, this ultimate goal is a thing of motivation only and cannot be attained according to his theory.

Carl Jung (Schultz and Schultz 2005, Carver and Scheier 2004) viewed personality as a psychic energy, which individuals invest in various interests and activities. This investment reflects and influences the individuals' life. To explain this process, Jung borrowed three principles from physics and applied them as follows: opposition where all positive energy has a negative, creating conflict which motivates behaviour and produces more energy; equivalence conserves energy which is distributed among interests and behaviours; and entropy where opposing energies tend towards balance. This process is cyclic and the continuous aim is to achieve balance. As with Adler's theory this state of perfection is unattainable.

Freud, Adler and Jung all promote some form of motivation in driving behaviour. None of these theories provide a clearly defined and identifiable element to include in a computational model of personality. Tension levels and psychic energy are internally based and conceptual with no ready means of capturing or measuring them. Two theorists that provide a more concrete perspective are Gordon Allport and Henry Murray.

Allport's theory has a central focus on motivation, which he believed to be an essential and difficult part of personality (Schultz and Schultz 2005). Referred to as Functional Autonomy, it defines a process by which behaviours that were once guided by a particular motivation, become intrinsically compelling and detach from that original motivation. The motivation moves from being a means to an end to becoming an end in itself (Ewen 1993, Biesta 2006). There are two levels of Functional Autonomy. The first referred to as Perseverative Functional Autonomy are motives that have become habitual, but do not gain any personal reward for the individual (Hergenhahn 1980, Schultz and Schultz 2005). The second level termed the Propriate Functional Autonomy, are behaviours that were fuelled by motivation but have evolved into the motivation themselves. These behaviours become intrinsically part of the self, or as Allport referred to it the Proprium (Hergenhahn 1980, Ewen 1993). They are pursued entirely for themselves, and are essential to understanding and unique to individual (Schultz and Schultz 2005).

Murray, like Freud and Adler believed that tension is a major motivator. But unlike Freud he believed it is for the satisfaction of reducing it, that it is created. Like Adler he believed that a tension free state is not the ultimate goal. This tension is created from needs, of which Murray identified a list of twenty, as presented in Table 3.1. He classified needs in several ways: environmental or internal; primary, such as food; or secondary, those stimulated by the environment; reactive, those that are in response to objects in the environment or proactive, those that are spontaneous. Several needs can be satisfied by a single behaviour, or one need may be activated by another, forming patterns of needs and behaviours. He perceived that not all people experience all the needs he identified and some would only be experienced at particular points in life. For Murray it is the dispositional tendency towards certain needs, and the strength of those needs that define personality. For example the *order* need is described by Schultz and Schultz (2005) *as to put things in order, achieve cleanliness, arrangement, organisation, balance, neatness and precision*. An individual with a predisposition towards this need is motivated to cleaning, and organisational behaviours, when objects in the environment are unclean or out of place. The disordered environment creates tension within the individual and their behaviours would seek to satisfy that tension.

Abasement	Achievement	Affiliation	Aggression	Autonomy
Counteraction	Defendance	Deference	Dominance	Exhibition
Harmavoidance	Infavioidance	Nurturance	Order	Play
Rejection	Sentience	Sex	Succorance	Understanding

Table 3.1 Henry Murray's list of needs

from (Schultz and Schultz 2005)

Allport and Murray both describe processes where motivation for behaviour is readily identifiable. While Allport does not identify a list of needs like Murray, it is clear to see how the behaviours he is describing translate into the behaviours that can be observed in everyday situations. Murray presents an even clearer picture, attributing behaviour to both internal needs and external influences, which he termed as Press (Ewen 1993).

Figure 3.1 presents the first step in the personality modelling process. On the left had side are the questions that need to be answered in order to define an action. On the right hand side are the internal processes that have so far been identified and their link to corresponding questions.

Mood and emotion are included here from a brief review of current models of computational personality. A number of synthetic agent systems are utilising psychological aspects to better model believable behaviour in interactive characters and human personas (Gratch et al. 2002). One of the most prominent models of emotion implemented is that of Ortony Clore and Collins (OCC) (Ortony et al. 1988). The Oz project at Carnegie Mellon University (Marsella and Gratch

2003) is a development that implements the OCC. Kshirsagar (2002) extends this idea by including a third aspect, that of mood. In this case mood acts as the linking layer between personality and emotions.

In Figure 3.1the elements mood and emotion are assumed to be aspects of personality, where as Murray's concept of needs are depicted as influencers generated from both internal and external sources. These then link to the concept of motivation, as described by Allport and theoretically provided the answer to why a particularly behaviour is performed.

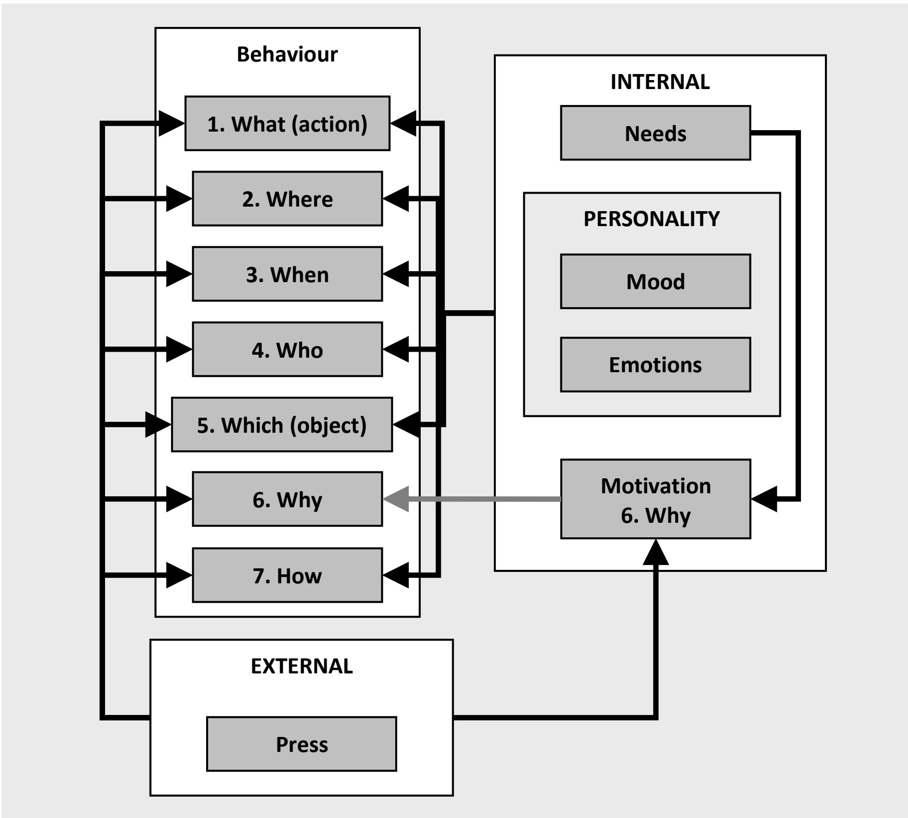


Figure 3.1 Positioning motivation

At this stage it is not entirely clear how these elements interact, or whether needs reside external or internal to personality. Given Murray's list in Table 3.1 and his theory of how these needs relate to a dispositional tendency, there are most likely to be some influence between needs and personality. The one aspect that this step clearly identifies is that, while personality may reside internally, the external environment has some contribution to make in answering these questions.

3.2.2 The source of personality

One of the most predominate debates in personality psychology is over the source of personality. Whether individuals are only the sum of their experiences, nurture, or whether personality is inherited from an individual's parents through genetics, nature. The answer to this question might also provide the answer to the variations within behaviour and expose some aspects of personality.

The question of personality's source, being either nature or nurture, is inherently tied up in questions about personality development and whether there is a point at which growth stops, personality stability. According to Schultz and Schultz (2005), Freud, Adler and Murray both take a fifty, fifty view on the nature versus nurture issue, while Jung perceives nurture as being more influential. Freud's theory focuses on childhood experiences as being the key to personality development and advocates mid-childhood as the point of stability. The other theorists consider personality development as ongoing throughout an individual's life. The development and stability of personality is an important issue as it determines where or not a computational model needs to take account of change.

Other researchers provide much clearer evidence on this subject. Eysenck, who has a biological based perspective on personality, contributed a great deal to answering these questions. He conducted many studies that involved the comparison of twins. He found that Monozygotic (MZ) identical twins were more likely to share characteristics than Dizygotic (DZ) fraternal twins (Carver and Scheier 2004). He also studied adopted children and found they were more like their biological parents than adopted parents (Schultz and Schultz 2005). This supported his belief that traits have a strong genetic component and he stated that *genetic factors* contribute something like two-thirds of the variance in major personality dimensions (Eysenck 1982 as cited by Biesta 2006). He did not rule out the effects of the environment but believed them to be minimal. His studies also suggested that personality is a fairly stable entity throughout childhood and adult life.

Costa and McCrae consider personality to be relatively stable throughout life, but suggest that it does not mature until much later than suggested by other theorists, around the age of thirty. Their theory is based on the Five Factor Model (FFM) (McCrae and Costa 2003, McCrae and P.John 1992) of traits which is discussed later in 3.2.4. For them these traits are genetically rooted and many of the changes that are observed are due to the natural changes that occur via the genetic clock. As yet there is not sufficient evidence of this (McCrae and Costa 2003) but there is enough to speculate that experience has a limited effect on trait development. The only problem with this perspective is the assumption that personality only consists of traits. Again 3.2.4 looks at this issue in regard to their Five Factor Theory and their model of the personality system.

Allport, like Eysenck, had a very strong biological stance, but this did not mean he disregarded the influence of environmental factors. His dislike of the predominant theories in stimulus-response psychology encouraged him to think of the environment as a learning process and genetics as supplying the raw materials (Schultz and Schultz 2005). He believed that both nature and nurture have a hand in personality development with current experiences and growth being a human's natural process.

Allport's concept of genetically based building blocks being adapted by individual experience (Schultz and Schultz 2005) suggests that there are two parts to personality, as shown in Figure 3.2. The first represents structure of personality and the second personality's effects on the processes linking to behaviour. Personality structure has a core formed from genetic influences, and experience is fed in, making some variations resulting in the effects of personality. This produces some result, as yet undefined that filters through mood, emotion, attitude and possibly other attributes to produce behaviour.

With the consideration for genetics, the concept of physiology comes into play. Murray's descriptions of needs included both psychological and physiological although in Figure 3.1 there was no clear need to define this. Now the model identifies that needs can be of either type.

Given that both experience and Press have the same source, environment, these have been remodelled as influences from the environment that effect personality and motivation respectively.

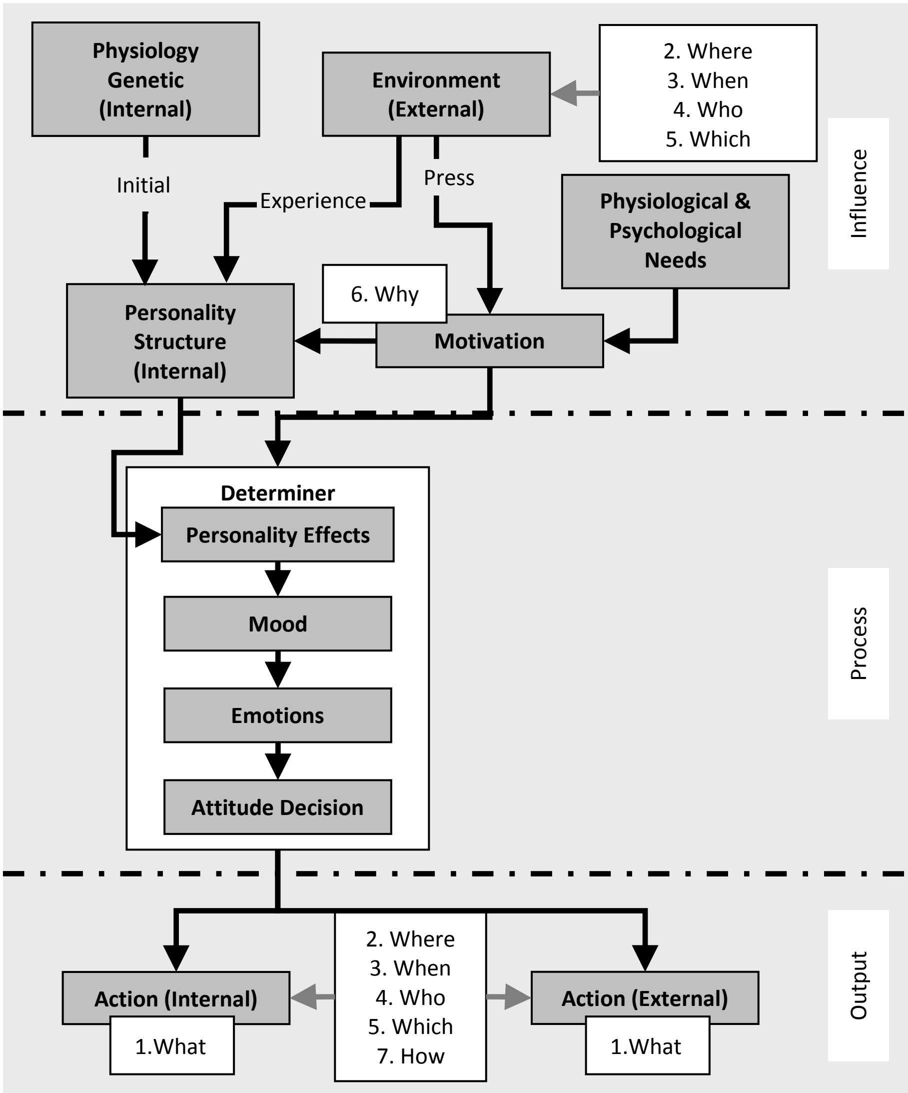


Figure 3.2 Introducing physiological aspects and attitude

At this stage the model (Figure 3.2) has been divided into three parts to distinguish the broad role of various elements. Those elements that influence the personality process, the elements linked to determining behaviour, the main process, and the outputs from that process. To accommodate this personality has two representations, one indicating the structure that is being influenced, updated, and another that is an individual instance of that structure, a snap shot, used within the process at a specific moment in time.

External action has already been identified as an output of this process, but are there any other outputs to consider? Jung talks about the concept of psychic energy, personality, being directed internal or externally. He describes this process as having an introverted attitude or an extraverted attitude respectively (Carver and Scheier 2004). Each individual has the capacity for both these attitudes, but only one can remain dominant, and in the conscious mind at a time. The other resides in the unconscious mind influencing behaviour in less obvious ways. Therefore personality directs behaviour internally or externally depending on the current attitude within the conscious mind.

To account for this, a second output has been introduced to the model (Figure 3.2), representing the concept of the internal action. The element labelled Attitude Decision, indicates the point at which the decision to act internally, or externally is made. It is not clear at this point, where within the determiner this should occur, nor what constitutes internal behaviours.

3.2.3 The cognitive aspect of personality

On reading a number of the psychological theories one particular aspect stands out, that of the mind. Much of the debate around personality aspects is focused on the effects of the conscious and subconscious mind. This was particularly prevalent in Freud's work who divided the mind into three levels: the conscious, the preconscious and the unconscious. The conscious part of the mind interacts with the everyday world. The preconscious stores memories and thoughts that individuals are not currently consciously aware of, but are easily accessible. The largest part of the mind is the unconscious mind and the source of instincts, Freud's drivers to personality.

Later, he extended his theory to include three structures: the Id; ego; and superego. These control the interactions between the levels of personality (Schultz and Schultz 2005). Id represented impulsive behaviour and is fuelled by instincts. Superego encapsulated morality and social rules by which individuals learn to live in relative harmony together. Ego represents the rational aspects and strived to find balance between Id and superego.

Freud's main focus was the unconscious aspects of the mind, which he believed to be the most important aspect of personality. From the perspective of developing a computational model the concept of the unconscious mind is limited. By its very nature it is not directly accessible, even by the individual to whom it belongs. This presents a problem when considering how to capture and represent this aspect computationally. While some behaviour's may be recognised in hindsight as having been performed unconsciously, the process of untangling these from those performed consciously would be complex, possibly involving the individual to engage in some form of dialogue to identify the differences. With no clearly defended structure or ready means of capture Freud's concept of the unconscious aspects of personality cannot feasibly be modelled.

A brief review of cognitive psychology suggests that there are distinct basic abilities that all normal humans are born with. These abilities are discussed within the context of the mind's physical structure and are not attributed to personality. If anything it could be suggested that personality is an emergent factor, which fits with the theories of George Kelly (Kelly 1963). According to Robinson-Riegler and Robinson-Riegler (2004) cognitive processes include:

- Memory – of which there are several forms for example: sensory working, autobiographical and long-term.
- Attention – focusing on incoming information.
- Pattern recognition – such as written language.
- Problem solving – working within constraints, utilising resources and achieving a goal.
- Decision making – organisation and evaluating choices.
- Language – reading, speaking, understanding and writing.
- Knowledge representation – enable access and assimilation.

To represent these, the element labelled cognitive processes has been introduced into the model in Figure 3.3. It is uncertain at this stage whether or not this element has more or less influence than personality on the nuances of behaviour, for this reason cognitive process have been placed between two instances of personality effects, with the personality structure remaining distinctly separate from this.

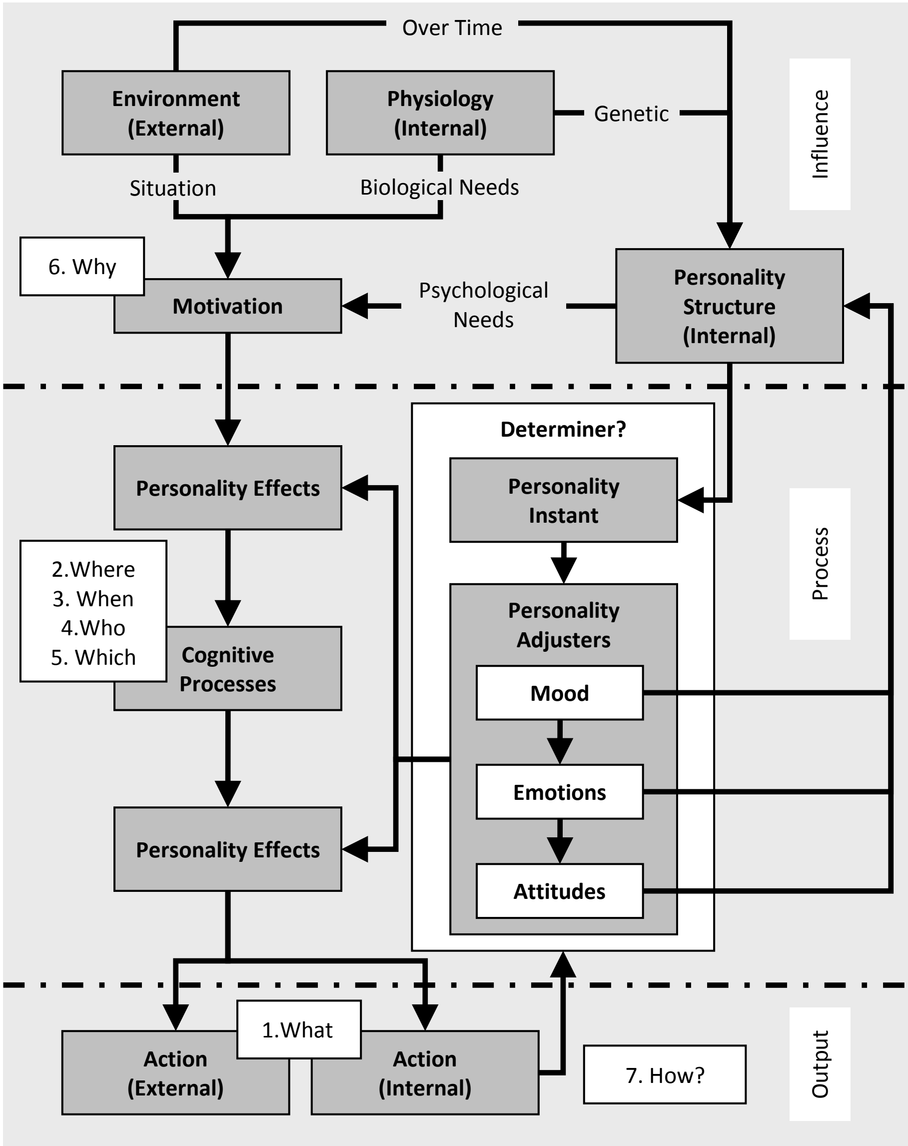


Figure 3.3 Introducing cognitive processes

At this point the model was also reframed to remove some of the duplications present in Figure 3.2 and to create a more cyclic process, demonstrating that actions that change the environment are feedback as potential influencers. As physiology has more than one role within the model a single element has been created in Figure 3.3, with the roles being: a genetic effect on personality; and a needs effect on motivation. The personality structure has been given the dual role of representing both personality and other psychological aspects of the model. This has resulted in Murray's needs becoming a directional element from personality to motivation. This role could have been assigned to the cognitive element, but this did not capture the personalised nature of desires and needs. There was also a need to keep the cognitive element simple, in the sense of representing common basic abilities.

Further investigation of the term attitudes suggests that this is more an aspect of personality than a process of directing behaviour, internally or externally as described by Jung. Allport suggests that attitudes are significant but distinct from personality traits. He describes them as being focused towards a particular person, place or object. They also have an element of evaluation, of whether the directed focus is positive or negative (Hergenhahn 1980; Schultz and Schultz 2005). This description firmly places the attitude as part of personality and not as a type of behaviour.

At this point it is perceived that the cognitive process may be the point at which most of the questions about behaviour are answered. Effectively this makes the cognitive process element the main point of determination rather than personality. The question of how something is done is still left unanswered, as indicated by its direct link to action. This suggests that this is the question that needs to be answered in order to determine the elements of personality.

3.2.4 Reviewing trait and type structures

The exploration of several trait and type theories was undertaken next. From this process a single theory was selected to represent this aspect of personality. During the process several other non-trait personality aspects were identified and included in the model. The following gives a brief review of the trait type theories proposed by selected personality theorists.

Jung and MBTI

As already discussed in 3.2.2, Jung identified two types of traits, Introversion and Extroversion. He also identified four psychological functions: thinking, feeling, intuiting and sensing (Carver and Scheier 2004). The first two are focused on the organisation of experience and lie in opposition to each other. The final two again are in opposition but are not rational in nature.

These he combined with the two directions of psychic energy to produce eight distinct personality types (Schultz and Schultz 2005). These personality types went on to be become one of the most successful personality assessment inventories. Developed in the 1920's by Katharine Cook Briggs and Isabel Briggs Myers, the Myers-Briggs Type Indicator (MBTI) is often used by organisations for employee selection (Schultz and Schultz 2005).

Eysenck

Eysenck also developed a system of traits and an inventory for identifying them. From the literature he noticed that there were similarities between descriptions of personality. The traits Extraversion and Neuroticism appeared in Jung's personality theories and were evident in works as far back as the ancient Greek philosophers, Galan and Hippocrates (Hampson 1988). These formed the basis to his ideas which he then proceeded to spend the next fifty years testing (Cooper 1998). He used the process of correlation to identify high order traits such as Extroversion. This he supplemented with rigorous experimental testing to confirm his theories (Carver and Scheier 2004). The combination of these two dispositions results in four distinct types, as shown in Table 3.2. These types also fit well with the types identified by Gallen and Hippocrates (Schultz and Schultz 2005, Hampson 1988) that Eysenck discovered in literature.

	Emotional Stable		**Emotional Unstable**	
Introvert	Passive Careful Thoughtful Peaceful Controlled Reliable Even-tempered Calm	Phlegmatic	Quiet Pessimistic Unsociable Sober Rigid Moody Anxious Reserved	Melancholic
Extrovert	Sociable Outgoing Talkative Responsive Lively Carefree Leaderly	Sanguine	Active Optimistic Impulsive Changeable Excitable Restless Touchy	Choleric

Table 3.2 Eysenck's personality dimensions

from Eysenck 1975 as cited by Carver and Scheier 2004

Later Eysenck added a third disposition, psychoticism, which included those traits not present in the other dispositions (Hampson 1988). This was identified from his work with patients suffering from personality disorders, and identified between schizophrenics, borderline and normal individuals (Cooper 1998). High scores on this scale include characteristics such as: aggressive, antisocial, egocentric, cruel, insensitive and hostile (Schultz and Schultz 2005). These were added to Eysenck Personality Questionnaire (EPQ) developed in 1975 (Eysenck and Eysenck cited in Hampson 1988). He was very aware of the flaws inherent in using questionnaire data, particularly the tendency for people to refrain from being truthful. To counteract this he included a lie scale within the questionnaire that detected fake responses (Biesta 2006).

Cattell

Cattell's research into personality started with work already done by Allport & Odbert (1936 cited in Hampson 1988) who identified 18,000 English words associated with behaviour. In which he narrowed these words down to 4,500 specific aspects (Cooper 1998, Allport & Odbert (1938) as cited by Cattell and Kline 1975). Next he removed all the synonyms leaving 171 words (Cattell and Kline 1975). This was still too large a number to be useful, so these were inter-correlated producing thirty six clusters, to which ten more were added from psychiatry research. There is some disagreement in whether the traits added from research, were added prior (Hampson 1988) or after (Cattell and Kline 1975) clustering, this could seriously affect the nature of the results. This resulted in about forty six traits which he termed as the standard reduced personality sphere (Cattell and Kline 1975).

This initial list was termed as surface traits, and deemed by Cattell, as of little use until their source traits had been identified. In order to identify the source traits, this initial list was used as scales in ratings studies (Hampson 1988, Cattell and Kline 1975). Results were correlated using the technique of Factor Analysis revealing twelve to fifteen factors, or source traits (Cooper 1998).

Cattell's work resulted in a personality inventory referred to as the 16PF (Cattell and Kline 1975) as shown in Table 3.3. At the time there was great debate on the terminology used within personality psychology. To avoid controversy Cattell made up labels for each of the sixteen source traits, which are shown in the right hand columns of Table 3.3. The left hand columns show the labels used today which provide better understanding. Cattell was convinced that his research would eventfully lead to the prediction of behaviour from personality.

Modern labelling				Original Description			
1	Reserved	V	Warm	**A**	Sizia	V	Affectia
2	Concrete-reasoning	V	Abstract-reasoning	**B**	Low Intelligence	V	High Intelligence
3	Reactive	V	Emotional stable	**C**	Low Ego Strength	V	High Ego strength
4	Deferential	V	Dominant	**E**	Submissiveness	V	Dominance
5	Serious	V	Lively	**F**	Desurgency	V	Surgency
6	Expedient	V	Rule-conscientious	**G**	Weaker Superego	V	Stronger Superege
7	Shy	V	Socially bold	**H**	Threctia	V	Parmia
8	Utilitarian	V	sensitive	**I**	Harria	V	Premsia
9	Trusting	V	Vigilant	**L**	Alexia	V	Protension
10	Practical	V	Imaginative	**M**	Praxernia	V	Autia
11	Forthright	V	Private	**N**	Artlessness	V	Shrewdness
12	Self-assured	V	Apprehensive	**O**	Untroubled Adequacy	V	Guilt Proneness
13	Traditional	V	Open to change	**O^1**	Conservativism	V	Radicalism
14	Group-orientated	V	Self-reliant	**O^2**	Group Adherence	V	Self-sufficiency
15	Tolerates disorder	V	Perfectionist	**O^3**	Low self-sentiment	V	High Self-sentiment
16	Relaxed	V	Tense	**O^4**	Low Ergic Tension	V	High Ergic Tension

Table 3.3 Cattell's 16PF

from Carver and Scheier 2004 and B064

Traits are generally seen as the basic building blocks of personality. They are described as having properties such as; being stable (Schultz and Schultz 2005) and representing a pattern of regularity across time and situation (Biesta 2006). Cattell identified various methods of categorizing them. The first of which was mentioned above with defining traits as surface or source. Another of which is in agreement with Allport's views (Schultz and Schultz 2005) describes traits as either common or unique (Schultz and Schultz 2005 Biesta 2006). Common traits are those that all individuals possess to some degree or other (Hergenhahn 1980). They can be derived from a genetic basis, or an environmental one such as social or cultural groups. Unique on the other hand are those that only appear in certain individuals, and reflect their interests and attitudes (Schultz and Schultz 2005).

Different traits, according to Cattell, were of different types. In order for behaviour to be explained Cattell identified two types of dynamic (motivational) traits. These he labelled ergs, innate driving force, and sentiments, a pattern of learned attitudes in relation to specific aspects of life (Schultz and Schultz 2005). He then goes on to describe how these interact to produce specific attitudes (behaviours), via a structure termed the dynamic lattice represented in (Hergenhahn 1980). The process that links these elements was the concept of subsidiation whereby attitudes are subordinate to sentiments, and sentiments are subordinate to ergs.

Allport

Allport did extensive work on the concept of traits but he never really did much towards identifying them (Biesta 2006). Much of his early work was focused on defining the concepts and terms used within personality research. His definition of personality in (Allport 1973) was the result of studying fifty of the then current definitions. He went on to define other concepts in great detail most notably that of traits. His definition of traits is as follows:

> *... neuropsychic structure having the capacity to render many stimuli functionally equivalent, and to initiate and guide equivalent (meaningfully consistent) forms of adaptive and expressive behaviour.*
>
> *(Allport 1973)*

From this he defined five distinct characteristics that traits possess, as follows (Allport 1947 cited Schultz and Schultz 2005):

- Personality traits are real and exist within each of us. They are not theoretical constructs or labels made up to account for behaviour.
- Traits determine or cause behaviour. They do not arise only in response to certain stimuli. They motivate us to seek appropriate stimuli, and they interact with the environment to produce behaviour.
- Traits can be demonstrated empirically. By observing behaviour over time, we can infer the existence of traits in the consistency of a person's responses to the same or similar stimuli.
- Traits are interrelated; they may overlap, even though they represent different characteristics. For example, aggressiveness and hostility are distinct but related traits, and are frequently observed to occur together in a person's behaviour.
- Traits vary with the situation. For example, a person may display the trait of neatness in one situation and the trait of disorderliness in another situation.

He defined traits as being both individual and common. To keep these distinct he referred to common traits as traits and unique traits as personal dispositions (Schultz and Schultz 2005, Ewen 1993 and Hergenhahn 1980). Each trait has the properties of frequency, intensity and range. He did not link single traits to behaviour, but suggested that a group were responsible for any specific behaviour (Biesta 2006). To add to this complexity, Allport suggested that traits are often aroused in one situation and not in another (Allport 1937 as cited by Carver and Scheier 2004).

Traits also varied in their levels of influence on behaviour. Cardinal dispositions are all-pervasive and may dominate an individual. Not everyone possesses such dispositions but when they do they are present in almost all behaviour. Central dispositions are less influential and an individual may possess only a small amount. They are observed in a more limited range of situations, and are identified as the consistencies across an individual's behaviour. Finally, secondary dispositions have even less influence and consistency, and respond to a narrow range of stimuli. This includes such things as minor preferences, idiosyncrasies which are similar in concept to habits and attitudes (Schultz and Schultz 2005, Ewen 1993, Biesta 2006 and Hergenhahn 1980).

Although Allport was primarily focused on traits he also identified and defined other aspects of personality that were not traits. His concept of attitudes has already been presented in 3.2.3. He also distinguished between traits and habits, suggesting that traits were linked to specific habits, but habits were specific behaviours. For example, the trait cleanliness is linked to brushing teeth and cleaning hands, but they are clearly behavioural descriptions (Hergenhahn 1980, Schultz and Schultz 2005).

Values were another aspect of personality that Allport researched. He identified six types of values that he believed all individuals possessed to some degree (Ewen 1993). They are the source of strong motivations and interests within individuals, and represent more personal traits. He developed a self-report questionnaire that identified the strength or relative importance of each to an individual (Allport, Vernon and Lindzey 1931/1960 as cited by Ewen 1993)

Costa and McCrae

Cattell developed his 16PF personality inventory from the process of lexical analysis. While his later work could not be replicated he had reduced Allport and Odbert's (John and Srivastava 1999) 4,500 trait terms to thirty five. This created much interest by other researchers. Some used this to reconstruct more simplistic models, such as Frisk (1949 as cited by John and Srivastava 1999). Norman extended Allport and Odbert's work by re-categorising the lexical terms and proceeding from there. Across these and other works five main factors were independently and consistently discovered. These were coined the big five by Goldberg (1981 cited in John and Srivastava 1999).

From the 1970's to the mid 80's there was a drop in interest in this area. Many researchers had replicated the big five but there were still several problems. The names used to describe each of the factors varied, as shown in Table 3.4 along with the number and type of traits they encompassed. A range of scales had been developed but none were completely satisfactory.

Extraversion and Neuroticism showed the greater stability, but the others often had overlaps and were often measured on different scales.

	1	2	3	4	5
Fiske (1949)	Social adaptability	Conformity	Will to achieve	Emotional control	Inquiring intellect
Norman (1963)	Surgency	Agreeable-ness	Conscientious-ness	Emotionality	Culture
Borgatta (1964)	Assertiveness	Likeablity	Responsibility	Emotionality	Intelligence
Digman (1990)	Extraversion	Friendly compliance	Will to achieve	Neuroticism	Intellect
Cost & McCrea (1985)	Extraversion	Agreeable-ness	Conscientious-ness	Neuroticism	Openness to experience

Table 3.4 Labels for the FFM
from Carver and Scheier 2004

In 1976 Costa and McCrae became involved in the Big Five development, starting with the analysis of data from Cattell's 16PF (McCrae and Costa 2003). From this they originally devised a three factor model consisting of Neuroticism, Extraversion and Openness to Experience. During the 1980's while examining Norman's five factors they discovered strong similarities between their Neuroticism, Extraversion and Norman's Emotional Stability and Surgency respectively. From this they suggested that if their Openness to Experience and Norman's Culture matched, the three factor model would fit within Norman's five factor model. They went on to develop scales for the remaining two factors Agreeableness and Conscientiousness and published the NEO Personality Inventory, NEO-PI. This became known as the Five Factor Model.

The FFM consists of five broad factors under each of which six traits or facets are related. This structure is shown in Table 3.5. Although there is some controversy over the inclusion of exactly six facets to each factor, the NEO-PI is well established and has been tested across gender, age and culture. According to McCrae and Costa (2003) it has been translated into many languages. It has been used in a wide range of studies with the aim of predicting behaviour in particular circumstances.

Neuroticism	Extraversion	Openness to Experience	Agreeableness	Conscientious-ness
Anxiety	Friendliness	Imagination	Trust	Self-Efficacy
Anger	Gregarious-ness	Artistic Interests	Morality	Orderliness
Depression	Assertiveness	Emotionality	Altruism	Dutifulness
Self-Conscious-ness	Activity Level	Adventurous-ness	Cooperation	Achievement-Striving
Immoderation	Excitement-Seeking	Intellect	Modesty	Self-Discipline
Vulnerability	Cheerfulness	Liberalism	Sympathy	Cautiousness

Table 3.5 Five Factors and their facets

from McCrae and Costa (2003)

In 1996 McCrae and Costa extended the FFM with the presentation of a Five Factor Theory (FFT). The theory is based on six postulates, which are summarised as follows:

Basic tendencies – consist of hierarchical traits, shown in Figure 3.4 that are endogenous and influence an individual's pattern of thoughts, feelings and behaviours. They develop during childhood and become stable within cognitively intact adults.

Character adaptations – are changeable aspects of personality influenced by biology, the environment and intervention. They evolve patterns that are consistent with their personality. They are not always optimal with personal goals or social rules.

Objective Biography – individuals have goals that organise action over time. Experience effects those character adaptations that are invoked by the environment.

Self-concept – individuals have a cognitive view of themselves which is selectively created and is constant with an individual's personality.

External influences – social and environmental interactions shape character adaptations consistent with basic tendencies. Perception and influence of the environment is constant with basic tendencies.

Dynamic processes – are the interactions between all these elements. Some are universal and some are pertaining to basic tendencies.

The interaction model depicted in Figure 3.4 shows the five main elements interacting via dynamic processes, which are currently unknown. The rectangular elements are primary to the system whereas the elliptical act on the periphery. The environment has an influence on Objective Biography and Characteristic Adaptations; whereas Basic Tendencies are seen to have a biological base. Both Biological Bases and External Influences are inputs to the system. Objective Biography is an output, the accumulative life history of thoughts, feelings and actions.

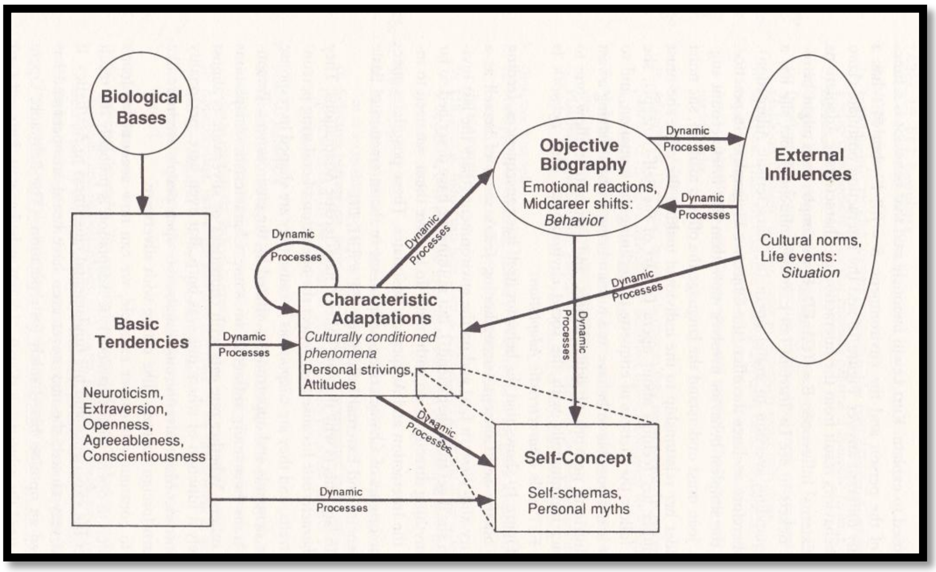


Figure 3.4 Model of the FFT

from McCrae and Costa (1996) as cited by McCrae and Costa (2003)

The crux of this model is the difference between Characteristic Adaptations and Basic Tendencies. Characteristic Adaptations represent the learned concrete structures developed over time via interaction with the environment. These include such things as interests, habits, beliefs, attitudes, roles and the self-concept. Basic Tendencies are abstract concepts that remain stable over time. For McCrae and Costa (2003) their research focus is within Basic Tendencies and the concept of traits, but it also contains other psychological concepts such as cognitive abilities, talents and sexual orientation. Between these elements various dynamic processes create interaction. These are not clearly defined but relate to such as reasoning, planning, perception, coping, etcetera. Form McCrae and Costa (2003) perspective these details will have to be supplied by specialists in such fields as: cognitive psychology; and neuropsychological.

3.2.5 Defining the personality structure

Allport's work on traits is extensive, only a small sample is expressed in 3.2.4, but he never synthesised a list of potential traits or an inventory to identify key aspects. However, some of his ideas around qualities of traits and those non-trait aspects have potential within a computational model of personality. The most prominent idea is the way in which he describes traits as being both common and unique. Here the term traits refer to all potential elements of personality not just those aspects used in an inventory system. If there are traits that can be classed as common, which individuals all carry, but possibly to varying degrees, this fits well with similar concepts in CF techniques. Pairing this with a second set of unique traits would add that individualised effect, and possibly create serendipitous output when crossed with the common aspects.

Cattell also discusses categorising traits as common and unique but, as with much of his work, there is no clear definition or indication of how this fits with the 16PF. Costa and McCrae also suggest this sort of structure, via biologically derived basic tendencies and environmentally influenced characteristic adaptations (Figure 3.4). This suggests common, genetically based aspects and individual experience derived aspects.

Given this evidence, Figure 3.5 shows the addition of common and unique aspects as part of the personality structure. The common aspects provide the personality instant and the unique aspects the personality adjusters, forming a layered process. The unique aspects being experience driven receive feedback from the environment. As such a process is time dependent the unique aspects are further divided into short and long term elements to account for this.

Returning to Allport's work he suggests that all traits possess certain properties such as frequency, intensity and range. Given this it is possible to examine those aspects that he suggests and define some values that represent them. He identified four unique traits, personal dispositions as he later referred to them, as attitudes (3.2.3), habits, values and preferences. He defines a habit as behaviour, one that is usually a response to a particular situation, making them relatively inflexible in nature. Considering the common definition of a habit, it can be assumed that this deposition can be said to have frequency, and that it is active over the long term, as presented in Figure 3.5.

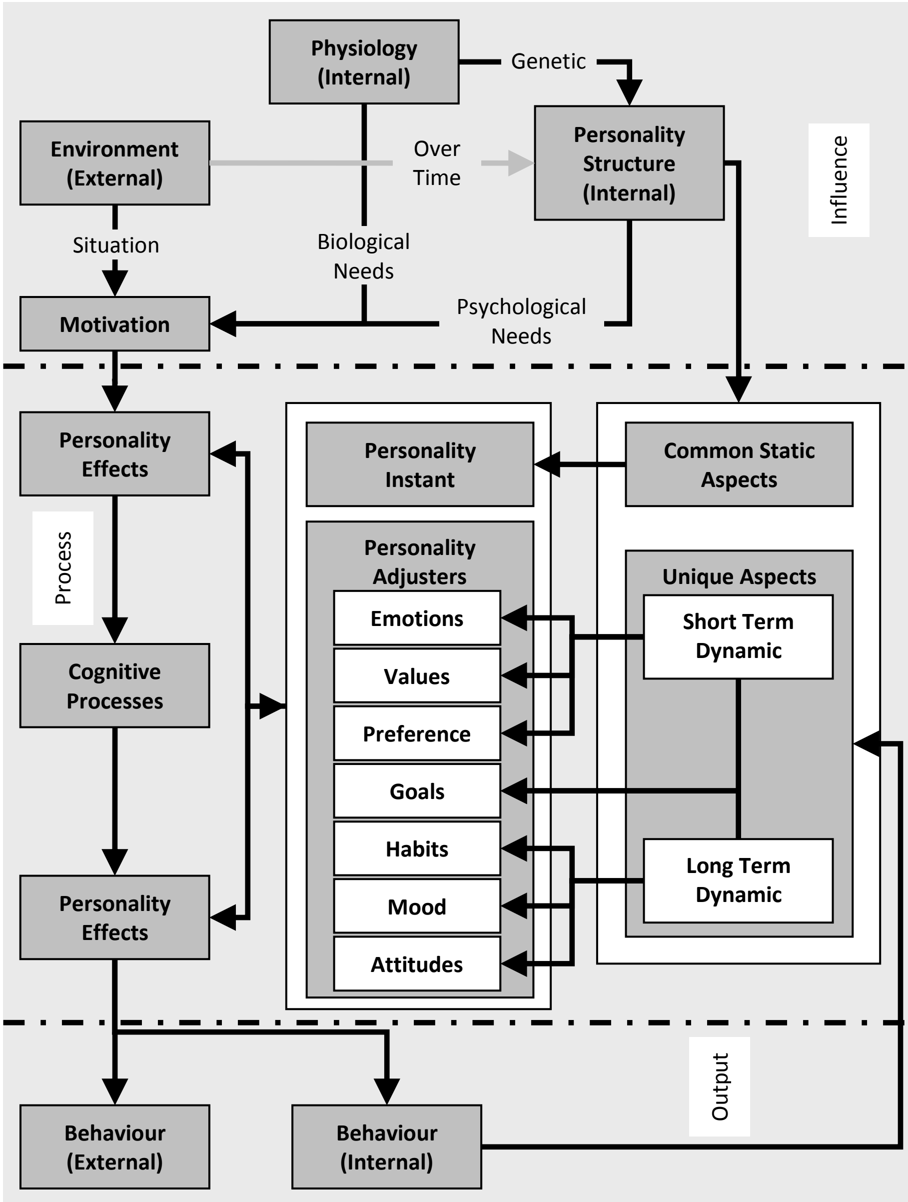


Figure 3.5 Adding dynamic and static personality aspects

Although Allport spent some time examining values, defining a list of six key values based on Eduard Spanger's work, and developing an evaluation tool for them (Ewen 1993), there is little else to define them. Referring to the generalisation he made regarding disposition properties, frequency does not fit with the common definition of the term value. However, both intensity and range seem to sit fairly well. How much an individual values something can be interpreted as the intensity of that value. While range is more ambiguous, it could be used to define the boundary of the value, indicating those situations within which the value is applied. For example, an individual may value companionship, but only in certain circumstances.

Preferences are the least defined of the dispositions mentioned. Allport considered them to be relatively minor in comparison to other aspects of personality. Again the generalisation can be applied endowing them with intensity, frequency, and range. Range again suggests the limitations of application within a situation. Frequency and intensity could be used to compare preferences when more than one can be applied to a given situation.

Goals are a concept that many of the trait theorists refer to in passing, primarily to distinguish their perspective on personality, from the internal instinct driven work of Freud and followers. However, as an element within a computational model they offer a means of focusing the profiling process. Figure 3.5 shows goals as being part of both long and short-term unique aspects. Generally individuals have a range of goals. In the case of a profiling system they are usually fairly immediate and short term, such as searching for information on a topic. In order to have a more complete picture of an individual, long-term goals may need to be taken into account. The former information goal maybe linked a more mid-term goal such as completing an essay, which in turn is linked to a long-term goal of passing a course.

The unique aspects could be populated with a whole range of applicable aspects, like goals, attitudes and habits. This could also be determined by the type of system the profile is being embedded in. Some aspects maybe more relevant or produce better predictions in different domains. Not all the unique aspects described are depicted in Figure 3.5 due to space limitations. In the next phase of development a reconstruction of the diagram enables them all to be represented (Figure 3.7).

3.3 Behavioural framework and model of personality

Types and traits provided the clearest structure for populating the common aspects of the personality structure. There is a potential link to genetics, as suggested by several of the theorists reviewed. Allport coupled with Costa and McCrae indicates that traits are the most likely elements to be consistently present across individuals, and stable within individuals. Having made this decision to include such, the only question that remains is which type or trait theory to apply?

Personality type systems like the MBTI initially looked as though they could be used to define the structure of personality within the model. However, there is no real difference between this system, and the ones the profiling methods used in 2.1. It is primarily a categorisation system, a typology (Pervin and John 1999) method. Each category on the MBTI relates to a set of characteristics or tendencies. The four dichotomies: Extraversion and Introversion; Sensing and Intuition; Thinking and Feeling; Judgment and Perception; giving a total of sixteen possible types. This is a small number of possible personality types to predict behaviour from.

Cattell's 16 PF and Eysenck's big three are both well thought of trait inventories. However, there are several issues with them. Cattell's has been criticised for lacking rigor first from his addition of extra terms from psychiatry at the beginning of the process, and second there are concerns for the use of manual factor analysis in identifying the higher order traits. The fact that no one has managed to replicate his trait set suggests that there are problems with it. Eysenck's EPQ is frequently used today and has a strong following, although there is some scepticism over his addition of the third high order trait, it has not had quite as detrimental effect as Cattell's additions. As there are issues other than the expected disagreements with both these systems they were put aside from the model development.

This leaves Costa and McCrae's work for consideration. Built on a long history, in psychology terms, the five factor model (FFM) has perhaps had the most attention of the trait theories. The addition of the five factor theory (FFT) and the six postulates that support it, presented in

McCrae and Costa 2003, adds weight to the use of this trait model. It is not without its critics, but what is surprising is how much of Figure 3.4 mirrors this development, as shown in Figure 3.5.

Comparing Costa and McCrae’s model in Figure 3.4 to the development so far in Figure 3.5 there are some marked similarities. Both models consider the impact of biology, labelled physiology in Figure 3.5, and the environment, external influences in Figure 3.4. Characteristic adaptations and objective biography which are concerned with individual experience, in Figure 3.4, contain many of the personality aspects that are assigned to the unique aspects of personality in Figure 3.5. The main difference being that the dynamic processes in Figure 3.4 are as yet undefined, whereas the mechanisms for linking personality to behaviour in Figure 3.5 have started to take shape. Considering how well this two models match, a number of the Costa and McCrae elements were considered for inclusion within the structure of personality, in Figure 3.7.

The first stage in this process was to clearly identify the behaviour personality process and the structure of personality. In order to produce a parsimonious view the structure of personality was extracted from the behavioural framework and developed into a separate model. This enabled room to clearly identify the key aspects of personality within it rather than allude to their presence. The behavioural framework presented in Figure 3.6, provides a process within which personality resides. Personality is represented as its structure (central) and as an instance of those aspects that are in effect at a particular point in time during the process.

The first change to the overall behaviour framework was to transfer the linear form of the three phases: influence, process and output, into a cyclic process like that depicted by Costa and McCrae’s model in Figure 3.4. This better represents the nature of how an individual’s behaviour changes the environment, and thus changes the situation that individual personality reacts to, forming a natural feedback loop. A clear boundary between internal and external elements is defined, leaving behaviour and the environment as external elements. The internal elements were redistributed in a cyclic pattern, so each loop around the system can be easily identified.

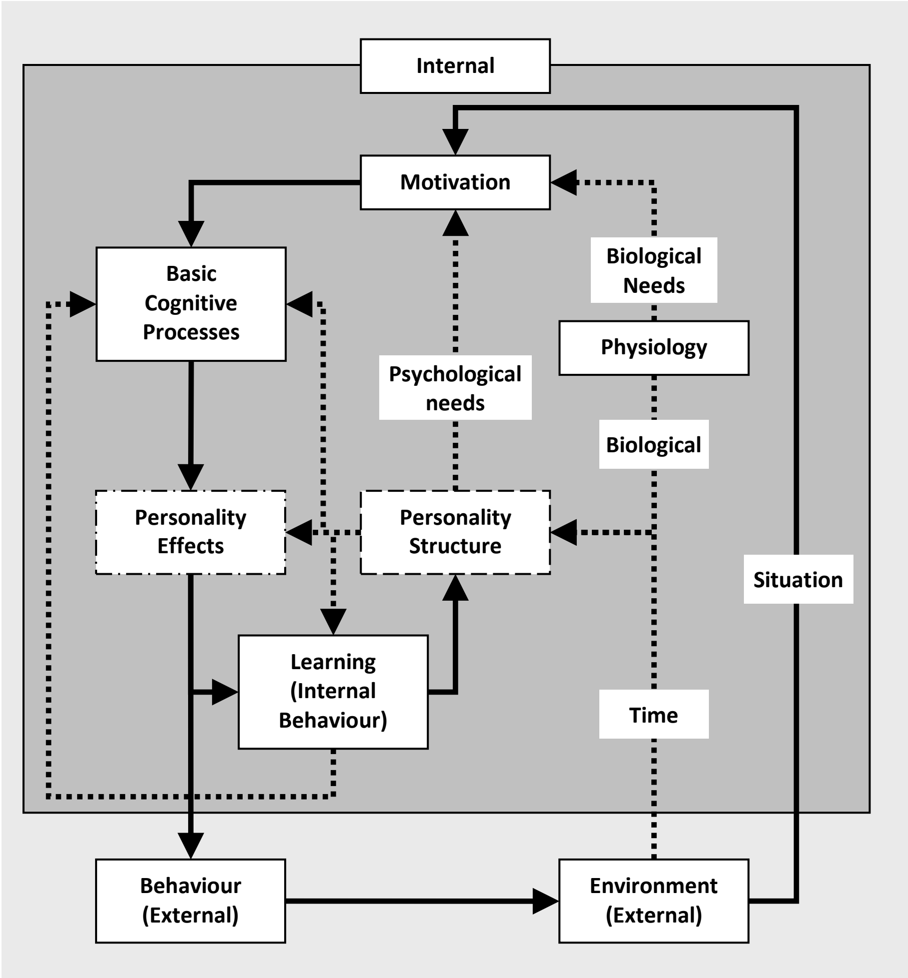


Figure 3.6 Behavioural framework

As mentioned throughout this development, personality can shape different aspects of behaviour. In 3.2.3 with the introduction of cognitive processes to the model it was discussed how this may or may not be effective on individual preferences. Given this, the potential effect of personality on the other elements within the system needs to be considered. From Figure 3.5, motivation receives input from: the environment of the current situation; physiology via biological needs; and personality via psychological needs. All of which have been retained in Figure 3.6. Personality still retains the two primary influences of genetics, from physiology and time from the environment. Figure 3.5 emphasises the effects of personality on cognition via sandwiching cognition between two instances of personality effects. This structure presents two problems: firstly, it is fixed, and does not allow for those times where there may be no personality based influences on the choice of cognitive tools, or how they are applied; secondly, updates to personality may have occurred between the two instances being extracted, possibly creating discrepancies in the process. To enable a more flexible approach effects have been transformed into a link, rather than a distinct step in the process. This link may or may not be activated, depending on the situation.

The final change was the identification of learning as the feedback mechanism into both cognitive processes and personality. Of all the elements with this model the two most likely to require this are cognitive process and personality. Motivation is perceived as capturing the essence of the current situation after the learning has taken place.

Figure 3.5 depicts that there is a feedback process, labelled internal behaviour, but does not elaborate. This is not learning in the sense of human learning but learning in the sense of machine learning, an up-dating process, the mechanism by which change is enabled. Although it is directed at the whole of the personality structure it is more likely to be applied to the dynamic rather than the static aspects. The static aspects would be better captured direct from user input as there is little change and this method is more accurate. The dynamic aspects on the other hand can be created and updated from the observation of interaction within a system using the personality representation profile.

As noted above the structure of personality has been extracted out of the behavioural process presented in Figure 3.6. Figure 3.7 shows all the elements from Figure 3.5 have been included, along with several from Costa and McCrae’s model in Figure 3.4. The rationale for focusing on Costa and McCrae’s model was initiated during the search to find a suitable trait or type based model to provide the static aspects of personality. Alongside traits, Costa and McCrae also put forward talents and orientations as other possible biologically generated aspects.

On the unique dynamic aspect side the list has been extended to include belief, ability and the self-concept. Beliefs and the self-concept have been included from Costa and McCrae’s work.

The idea of the self-concept is an important aspect in personality psychology. It is defined in many psychologists' work including Freud's (Schultz and Schultz 2005), Carl Rogers and Abraham Maslow's (Carver and Scheier 2004). It has not been explored in any depth in relation to this research, as it is a term that alludes to several different ideas given the theorist perspective. From a computational view, it is also something that would be prohibitively challenging to capture and monitor, as it is very much an internal aspect of any individual. However, it has been included here because of the prominence it is given with the FFT.

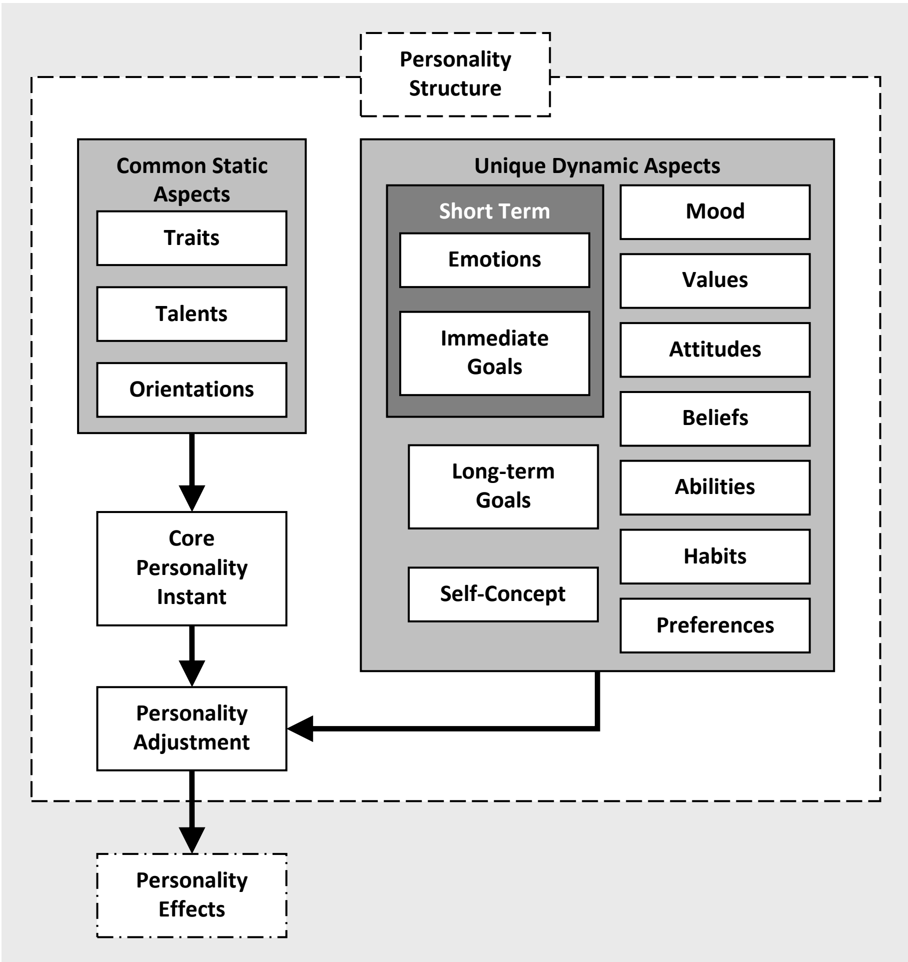


Figure 3.7 Model of personality

Ability is an aspect that may actually belong with cognitive process, but it is an attribute that has already been used in several educational systems as a means of monitoring student progress (2.2). There are several other attributes that could be included from profiling and personalisation systems, such as interests. However, as mentioned previously the dynamic aspects could be used as a means of capturing personality aspects that are directly relevant to the purpose of the application. In the case of this research the application domain is eLearning, hence abilities is one of the more relevant aspects.

Before exploring how these models can be used as a profiling system, there is one final adjustment to explain in Figure 3.7. In Figure 3.5 each of the dynamic aspects are labelled as short or long term. On reflection the attribute of time in this way is subjective and difficult to assess. The only aspects where it adds value is that of goals and emotions. Goals are usually time orientated and may last as long as a few seconds or a whole lifetime. From the perspective of this model the reason for having both short and long term goals is the need to differentiate between those things that need to be done within the time frame of an applications use and those that need to be retained form session to session. The long-term goals are more likely to be personal to the user, while the short term linked to the task at hand within an application. To account for this change, emotions and immediate goals are labelled as short term and all the rest are time independent, to be determined by the learning process of the system.

This concludes the review of psychological theories and the process undertaken to develop the final models presented in Figure 3.6 and Figure 3.7. Throughout this development process there has been discussion about how various elements could interact or how they could be represented. The aim of these models is to act as a generic model to developing a personality representation (PR) profiling system.

3.4 Contribution

To recap, some of what has already been discussed in 3.3, these models contribute to the field of recommender systems specifically and more generally to the fields of profiling and user modelling. First the behavioural framework in Figure 3.6 provides a holistic view of human interaction, and how personality resides within this process. It also identifies those aspects of human interaction that may not be derived directly from personality, the key concern of this thesis, but may contribute in some form to a process of personalisation.

Secondly the model of personality, Figure 3.7, identifies the types of personality attributes that can be used to capture personality and theoretically give the process of prediction more depth of understanding. Currently the majority of recommender systems focus on a simple relationship between user and product or product. This is also the case with those using personality implementations such as Hu (2010). This is either a dichotomous or scaled variable representing whether the individual likes or dislikes the product or product attribute. The model of personality gives greater scope for representing this relationship. For example a user may value a particular product attribute above another but not have the ability to utilise that attribute. A system that understands the difference between these personality attributes will recommend products that do not have this attribute or compensate for the lack of ability in some way, but select products the user is able to utilise. Moving beyond a simple *user likes* relationship will enable recommender systems to be more effective in discerning between similar products and provide the user with a more comprehensive and relevant selection of products or services.

One of the concerns within the recommender community was the reliance on ratings data and the use of the collaborative filtering (CF) method. This raised a number of issues as highlighted in 2.1.7. To combat these problems more recent systems have been looking at combining other techniques with CF as discussed in 2.1.5. While this has had varying degrees of success, they do highlight the need to utilise more than one source of data. The personality model in Figure 3.7 echoes this need by demonstrating that personality consists of aspects that are both common across individuals and unique to individuals. These two aspects are also linked to how the data is gathered, either explicitly and implicitly, and whether it is objective or subjective. The implications for this are discussed in 4.3 where the models are theoretically applied to an educational system.

The final contribution these models make is the identification of the environment (situation or context) as an important element in generating a prediction as well as the need for a feedback mechanism. This is something that has been previously recognised in related fields of inquiry such as search engines and to some extent in educational systems (2.2). These models show the degree to which the environment influences behaviour and how without context there is no true comprehension of a users request or actions as demonstrated by (Huang, W., et al 2006). This interaction with different environments also changes future behaviours, requiring some way to update and learn from these experiences by keep some form of user history, related to specific contexts.

Educational systems generally generate a profile with more depth than recommenders, utilising such aspects as ability. However profile attributes are generally selected from the perspective of the systems functionality. By utilising a more complete picture of a user's characteristics educational systems will be able extend their services more readily. The following chapter examines how these models could be applied within an educational system and their potential for finding possible solutions to the issues raised in 2.1.7.

Chapter 4
Theoretical application of the developed models

This chapter presents a theoretical application of the models developed in Chapter 3 within the context of educational support. The aim here is to demonstrate how each of the models' elements can be used within a profiling system and how they may have the potential to overcome some of the issues that were raised in 2.1.7.

The behavioural framework (Figure 3.6) models the interaction process between an individual and the environment. It shows the key elements that are applied in the process of producing behaviour. As the aim here is to predict behaviour these same elements can be used within a recommender style system to generate that prediction. Section 4.1 demonstrates how the behavioural framework maps onto the main elements of a recommender system, as shown in Figure 4.1, and discusses how this process can be applied to generate predictions or recommendations.

The model of personality (Figure 3.7) is discussed in 3.3 highlighting those elements that can be more realistically captured given today's level of technology. For those elements that are less feasible to capture, it is speculated on how they could be used within a profiling system given the right resources. At the end of 4.2 those elements that are more feasible are identified and taken forward into the theoretical system presented in 4.3.

Section 4.3 brings the behavioural framework and personality model together in a discussion on how they can be applied within a large scale cross-institutional system that is able to support students in a variety of ways. The potential benefits, limitations and ethical considerations of such a system are explored. The final section of this chapter, 4.4, identifies from this discussion what has been tested in the context of this work, and how this was to be implemented.

4.1 Mapping the behavioural framework

From an educational context the aim of the models presented in 3.3 is to predict behaviour in order to provide personalised support as discussed in Wood and Ishaya (2005). This is similar to the way recommender systems operate. Using

Figure 2.2 as a guide, some comparisons can be drawn between the two that show how the behavioural model can function. The recommender process starts, effectively, with the user, who from a behavioural perspective is both part of the environment and the internal behavioural process, shown in Figure 3.6. From a system perspective the user is part of the environment. The system itself represents the user's internal process in order to predict behaviour as shown in Figure 4.1.

A recommender system deals with two types of input, implicit and explicit. The same can be used with the behavioural framework, as labelled in Figure 4.1. The environment as a whole is monitored for implicit input, such as detecting a user's current task. Explicit input is provided directly by the user, such as the initial population of parts of the profile via a psychological instrument. Implicit input can influence both the learning mechanism and the main internal process, whereas explicit input is fed directly into the learning mechanism for personality updating, see Figure 4.1.

The selection process, as used by a recommender, is represented by the behavioural process as depicted in Figure 3.6. The first step, *motivation*, takes the initial input identifying what the user is trying to achieve. This is done by understanding the context, and the motivation for the current task. Next the *cognitive* element identifies all the steps required to achieve the task, along with any specific tools required. This can take the form of both human cognitive process as well as digital and non-digital elements. Having identified the scope of the task, the system can use personality to predict the most likely behaviours given the *personality effects* drawn from the profile.

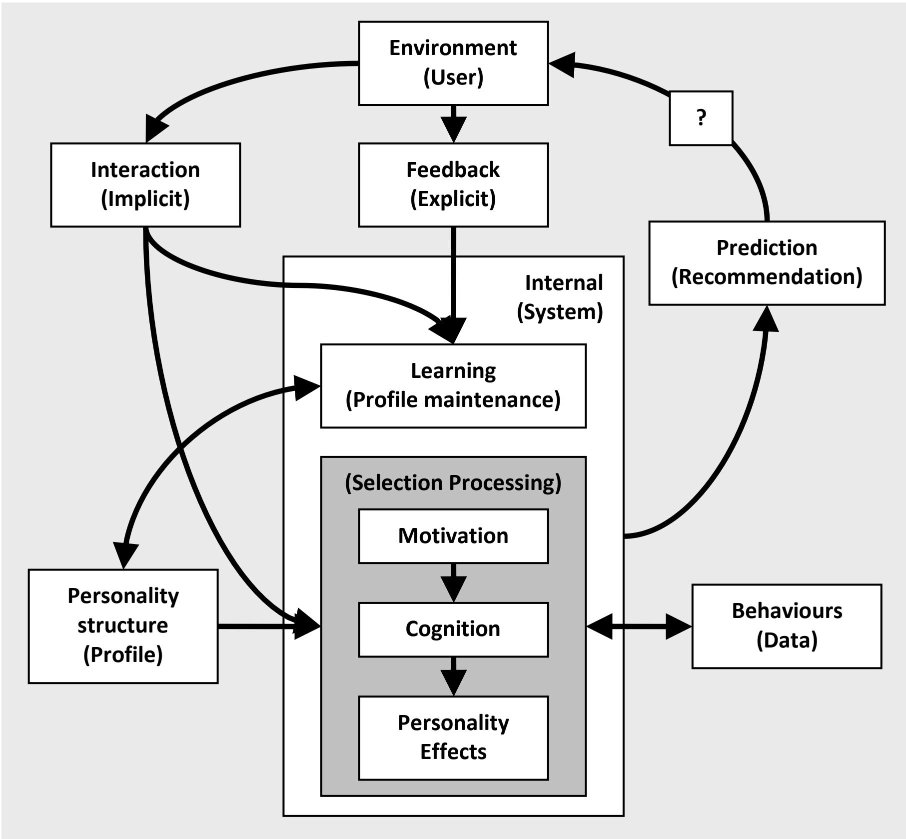


Figure 4.1 Behavioural framework as a prediction (recommendation) process

There are three distinct differences between this process and that of a recommender. The personality structure, user's profile, can influence all three stages of the selection process. Other than the profile the primary data is *behaviours*, which again are used at all three stages of the process. The prediction is not the final output; it is up to the system developer to decide how the predicted behaviour can be used. In the case of an educational system the predicted behaviour is used to define and tailor support for the user. In much the same as the *buddy* entity described in 1.1 supports *Jeff* through his revision process.

By adapting the behavioural framework in this way it is possible to see how it can be applied as a profiling mechanism. In the context of book or music recommendation (2.1) it is speculated that this process will enable the systems to apply the profile across domains and to individuals who do not follow popular trends or are looking for something different. The reason for this is that personality adds the context of the self to the process.

4.2 Personality representation

The personality model in Figure 3.7 is initially divided into two parts. Those elements belonging to all individuals, referred to as the common static aspects, and those belonging to individuals derived from their personal experience, referred to as the unique dynamic aspect. Together these two parts describe personality in a broad and general manner and constitute the top level of personality representation (PR) based profile.

The elements that reside within these two aspects are those identified during the review of psychological theories (Chapter 3) may not necessarily be applicable to use in all personalisation contexts. However they do fit closely with two types of data often used within recommender systems. The common static aspects represent data that can be captured across groups of individuals, focusing on what they have in common, similar to collaborative filtering as discussed in 2.1.4. The unique dynamic aspects focus on those elements that all individuals have but with a wider range of variables and values, of which many combinations are unique. This is less similar but closely related to demographic and content data. The assumption here is that the dynamic aspects will generate greater granularity than the distinctiveness between demographic groups or product descriptions.

4.2.1 Common static aspects

The common static aspects are the core aspects of personality representation (PR). They intend to represent those aspects that individuals are generally born with. They are different from the other half of PR in that they change very little over time and they are generally common to everyone. Talents and orientations are included in Figure 3.7 for completeness and at present there is no clear indication of how they are linked to behaviour, although their capture by survey instruments would be straightforward. The important aspect of these two attributes is that like traits they represent large groups of people, but unlike traits they are not pertinent to everyone.

The focus for the common static aspects is on traits. As discussed in 3.2.4 the use of traits as part of a PR profiling system is beneficial because they can be used to identify similarities across groups. They can act in a similar way to collaborative profiling (2.1.4) by creating statistical clusters of users attributes that have similarities.

To populate this aspect of the profile, Costa and McCrae's Five Factor Model (McCrae and Costa 2003) was selected from the trait and type theories reviewed in 3.2.4. One of the advantages of selecting this particular model is that it has a mature assessment instrument, which can be used to collect the users' score on each of the five factors and thirty facets as listed in Carver and Scheier (2004). However, depending on the requirements of a system other trait or type inventories could be used. Correlation data is also available between the more popular trait and type systems that would enable profiles to be interchangeable regardless of the trait theory used.

Once populated via a personality inventory the system can begin the process of using correlation techniques and regression to create connections between, input situations, the user's personality and observed behaviour. To start with this part of the profile may suffer from similar problems to collaborative profiling, presented in 2.1.7. An alternative is to pre-populate the critical areas from actual observations of user behaviour, in the same way as used to develop the prototype system in Chapter 5. However, like the process of using experts to define stereotypes in 2.1.3, this process is neither simple nor inexpensive.

Given the successful population of situation, personality and behaviour links, the system will then be able to present candidate behaviours, for a given situation and a given user, in an ordered list with the most likely first. This is the results obtained from the *core personality* element as shown in Figure 3.7. The next stage is to further personalise this prediction by drawing on the unique dynamic aspects of the user. For example, if the top three most likely behaviours in response to the situation *revise for exam* are *read core text, review notes, discussion with peers*, for a given individual's static aspects. These can be fine-tuned if there is evidence in the user's dynamic aspects that they *value* their peer discussions or they have a negative attitude towards reading. This could change the predicted order of preference depending on the effect each of these dynamic aspects have in relation to the ranking. The method of applying dynamic aspects will be explored during the prototype development phase in Chapter 5.

4.2.2 Unique dynamic aspects

This part of the PR profile effectively represents the recorded interaction history of the user with the current or multiple systems depending on how the profile is situated. The important facts about the values stored in this part of the profile is that they are: unique, not all attributes are required to be populated by all systems or all users; dynamic, the values the attributes hold change over time, either increase with interactions or deprecate with lack of use, they are not fixed.

As with the static aspects some of these attributes are part of the personality model (Figure 3.7) for completeness only, and as yet no suitable method of capturing the data required to populate them has been found. Emotions and mood could be indicated by the user, possibly through a series of icons, but this is not a satisfactory method. There may be inventories to use in psychology that capture this type of information, but given the frequency of change in these areas may not be practical to apply prior to each new session. They would also fail to capture changes in both these attributes throughout the session. An automated method would be most suited to this type of data capture. Hormonal levels are linked to some aspects of mood and emotion (Carver and Scheier 2004) and could be used as a detection process, but being physically linked to your hardware could be distressing as well as potentially unethical. Body temperature, eye movements and facial expressions are other possible means of capturing this data without being intrusive or expensive.

The other attribute that is complex for capture is that of the self-concept. Again there may be instruments in psychology that capture this data, but they are reliant on the user being of an open and honest character. This is not to say that the majority of people are not open and honest, but the self-concept is very integral and personal, and the possibility of being open to observation and potential criticism is again an ethical issue. Even in those who are willing it is not an easy aspect of the human psyche to express.

From the remaining attributes in Figure 3.7 each has a particular structure that reflects how they affect the behavioural and prediction process. This structure has been drawn from the descriptions given by various psychologists, most notably Allport and Cattell as discussed in 3.2.5. These are as follows:

Values: describes actions or objects and a value that indicates the importance level to the user. For a learning context a user may value presentations more than reading books but less than watching a video. To represent this presentations are given a value of fifty, from a possible one hundred, reading books forty and videos sixty-five. However values are distinct from preferences in that they stand alone and are not compared to each other. This attribute is also likely to be utilised at the motivation point within the behavioural framework.

Attitudes: are people, places or objects to which the user has a positive or negative reaction, attitude, towards. For example, a user may have a positive attitude towards books, they think they are interesting, enjoyable and a negative attitude towards lectures, they think they are boring, uninteresting. Again this attribute is most likely to be utilised at the motivation phase in the behavioural framework.

Beliefs: are actions, concepts or objects that the user holds to be true. Those things that are not identified as being true are not classed as not true, they just hold no special significance for the user. Beliefs are added weights to other attributes holding the same concepts, actions or objects. For example, the user may hold the belief that: *I cannot do mathematical calculations*. If they have a negative attitude towards mathematics or a low value on the ability of calculations, the belief adds weight to these other attributes. Some types of belief, such as limiting self-belief can be focused on by a learning system in particular as areas for extra support.

Abilities: skills, learned actions, knowledge. This type of attribute is graded indicating a level of competence on the part of the user. This can be represented by formal competencies within a learning system, or particular skills that the system has identified and can monitor the progress of. This attribute is also likely to have an influence at the cognitive stage in the behavioural framework.

Habits: are patterns of behaviour that are triggered given very specific contexts. As prediction is the aim of the system habits can add real weight to matching potential behaviours, increasing their chance of being correct.

Preferences: are ordered lists of objects or activities that are clustered according to a common situation or context. For example given the context of *learning from* the user may have a list of preferred activities which include watching a video, watching a presentation, reading a book. The ordering indicates the preference only, and is not necessarily linked to how the user values the same activities, or how they are ordered given a different context.

Goals: are behaviours that the user is aiming to achieve. They can be long or short term, hierarchical one goal attributing to another, or time limited. They can be personal (internal) or they can be given (external). Goals can have a precedence order depending on their aspects and the user's personality. Goals are likely to be used to enhance the motivation phase in the behavioural process.

There are two ways to populate this aspect of the PR, via a questionnaire gathering system-specific data, or over time given observation of the user interactions. Unlike the trait aspects, the dynamic values can be populated either way. It would be more effective to populate some of the key aspects of the system via a questionnaire as the user signs up, as well as enable users to adjust the values of the generated values to better reflect their self-understanding. Unlike linking traits to behaviours, in 5.2.3, a questionnaire to capture core values, attitudes, goals is a far more practical solution for quick and cheap population of the profile.

Unlike the static aspects, containing traits, the dynamic aspects of the PR do not in themselves predict behaviour that is done by the traits alone. The dynamic aspects add weight to the predictions given an individual's experiences.

4.3 Application to an educational context

One of the motivations behind the concept of a personality based profiling system is enabling the user to retain control over their profile. At the moment user profiles reside in the hands of the organisations that provide the personalised service. This is partly due to the requirements of some of the methods used to generate personal recommendations, such as collaborative profiling, and partly due to the fact that such data is a valuable commodity in its own right. When considering this in the context of an educational establishment and the type of data they may collect, there is a need for a higher standard of privacy and ethical consideration. This in itself is one of the reasons for selecting education as the application case study.

The argument is that given personality is central to a users behaviour capturing it should enable the predication of behaviour across all domains. If this is the case, this should enable the user to retain control over their personality details and grant access to their profile when requiring personalised services. The model presented in Figure 3.7 enables this to be the case. By providing both group similarities (common static aspects), and individual differences, (unique dynamic aspects) the profile can be divided and stored in two places.

As depicted in Figure 4.2 the organisation retains the behaviours that apply to their domain context linked, via correlation and regression (as discussed in 5.1.1) to the factors and facet values of the FFM (McCrae and Costa 2003). When a user wishes to use their personalised service, whatever that maybe, they simply allow access to their static aspect values. Using the method described in 5.1.1 the system can predict the most likely behaviour within this context for this user. Access to the dynamic aspects can also be granted to give a more personalised service, depending on the level of trust between the user and the organisation. In return the organisation retains further statistical data to improve the behaviour prediction and through observation of the user's interactions their dynamic aspects are updated within the users system.

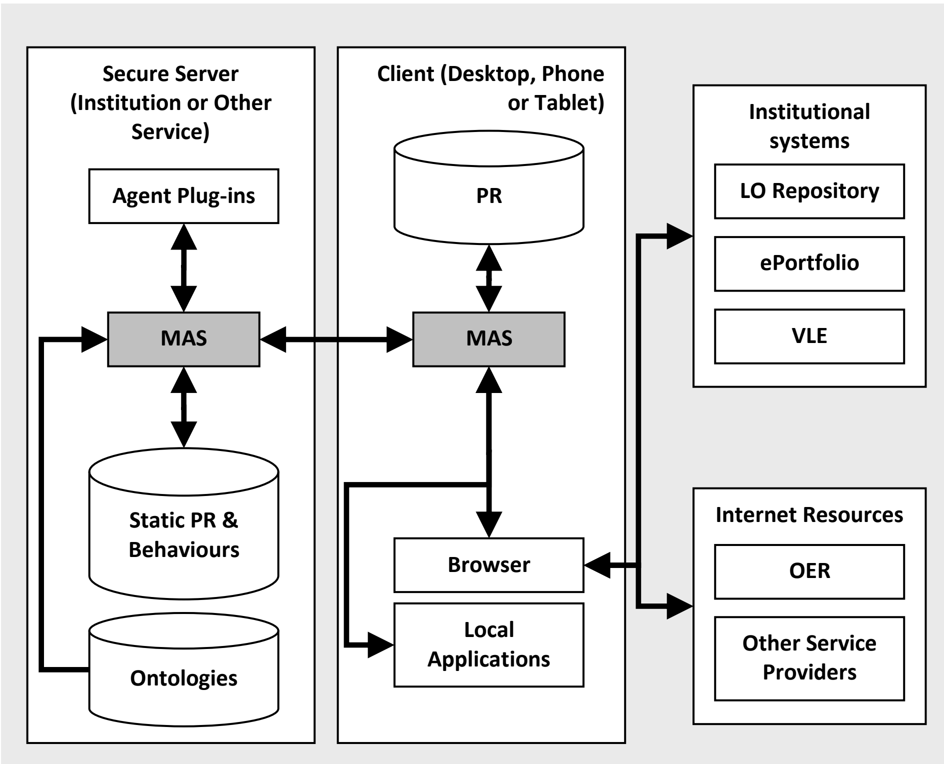


Figure 4.2 Theoretical system architecture

From the context of an educational system a number of personalised services could be provided to support the user through their learning. It is most likely that this type of support would be focused around the achievement of specific tasks. The system would be required to monitor and track such tasks, possibly connecting directly with institutional systems such as a repository of learning objects (LOs), (McGreal, 2004, IEEE 2004)), an ePortfolio (Sutherland and Brotchie 2011) of the students work, and content provided through a virtual learning environment (VLE). The internet in general will also be a valuable source for resources to support learning. Having identified the task the system would need to determine whether it is a) supportable in general terms, b) supportable in relation to the user's requirements, and c) supportable in terms of institutional requirements. Once this has been determined the best means of support needs to be generated.

Personalisation can be applied to a number of teaching and learning activities, similar to those described in the eLearning scenario (1.1) at the beginning of this thesis and in the review of eLearning in 2.2. For example support may consist of one or more of the following:

- Action advice – advising the user to look for other content, use a specific application, communicate with peers or tutors, presenting a learning path through specific content or activities.
- Content presentation– finding and presenting alternative methods of presenting an idea or concept.
- Organisation – scheduling and breaking down of tasks, including encouraging break times.
- Encouraging ideal behaviour – comparing a user's previous behaviour to ideal behaviour pattern for task completion and suggesting support activities.
- Motivation – strength and weakness analysis of tasks performance and providing encouragement activities.

The theoretical system presented in Figure 4.2 utilises a multi agent system (MAS) as the platform for delivering personalised services. There are a number of reasons why this particular method has been selected. First within the review of educational systems (2.2.3) a number of them are already utilising agent technologies. The review of recommenders also identified that a number of techniques developed within artificial intelligence (AI) are being applied to varying degrees of success. MAS frameworks, such as JADE (Bellifemine et al. 2005), are designed with a number of these techniques available to the agents for use in decision making and processing data. This is not to say that these techniques are not available to other platform types. However it is assumed here that on a large scale with many potential behaviour predictions to select from such techniques, particularly decision logic, will be required to utilise the user's dynamic aspects and identify the ideal predication. The agent's ability to pick up utilise and discard such techniques, without having them permanently programmed in makes this a flexible and light weight option.

The second reason for suggesting the use of a MAS is that individual agents are able to move between platforms. This is particularly critical when ensuring that any device the user has, desktop or mobile phone, can utilise the PR profile. This means that agents residing on low powered devices can move, with the user's data, to the service platform, perform the requested actions, monitor the user's interaction and return data to the user PR, all by utilising the more substantial processing power on the service providers end.

The final reason for looking to a MAS is the way individual agents can be assigned a particular task to process, monitor or negotiate resources for. The decision-making ability of agents enables the system to drop activities that are no longer relevant given the users current interactions with the system. Agents can be designed to have several processes run in parallel and can monitor each other's progress using a standardised communication language that can be accessed by external systems if required. This makes for a flexible and potentially lightweight system that can access and integrate effectively with currently existing systems

4.4 Proposed implementation and testing

The aim of this research is to determine the feasibility of the concept of personalisation via a personality based profiling mechanism. The models developed in 3.3 demonstrate the types of processes and elements required to generate a personality based profile. The previous sections of this chapter discuss the theoretical feasibility of this concept in light of the issues currently faced by recommender systems (2.1.7) within the domain of education. The next stage of this research is to evaluate this concept within a prototype system. To do this, this research proposes to examine a single case study that demonstrates how the models can be applied as a profiling system within a real world scenario, that of educational support. This case study will evaluate both the technical (system perspective) and operational (user perspective) feasibility of Personality Representation (PR) to be able to personalise support for individual users. The first stage in developing this case study is to identify the essential and practical elements of the two models for implementation within a profiling system.

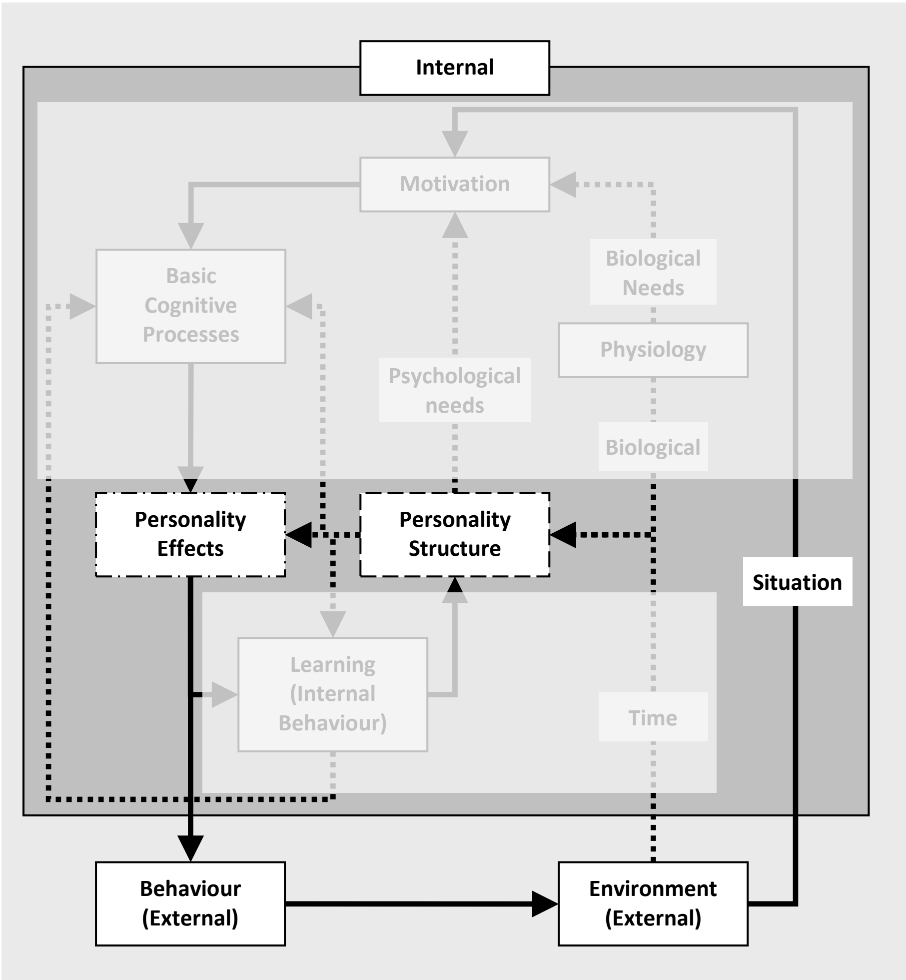


Figure 4.3 Proposed implementation of behavioural framework

The behavioural framework contains a number of elements that while linked to the concept of personality lie beyond the scope of this research. As the focus here is on how a representation of personality can be used for personalisation the prototype implementation will focus only on those elements directly related to this. These elements are personality, behaviour and the environment, as depicted in Figure 4.3. It is this core that will be implemented and tested within an educational support system. A personality representation (PR) will be developed and used to predict behaviour within a given environment (situation). The environment will be monitored and fed directly into the Personality Effects bypassing the other elements at this stage. This will enable the impact of using a PR profile to be evaluated directly without interference from the other elements.

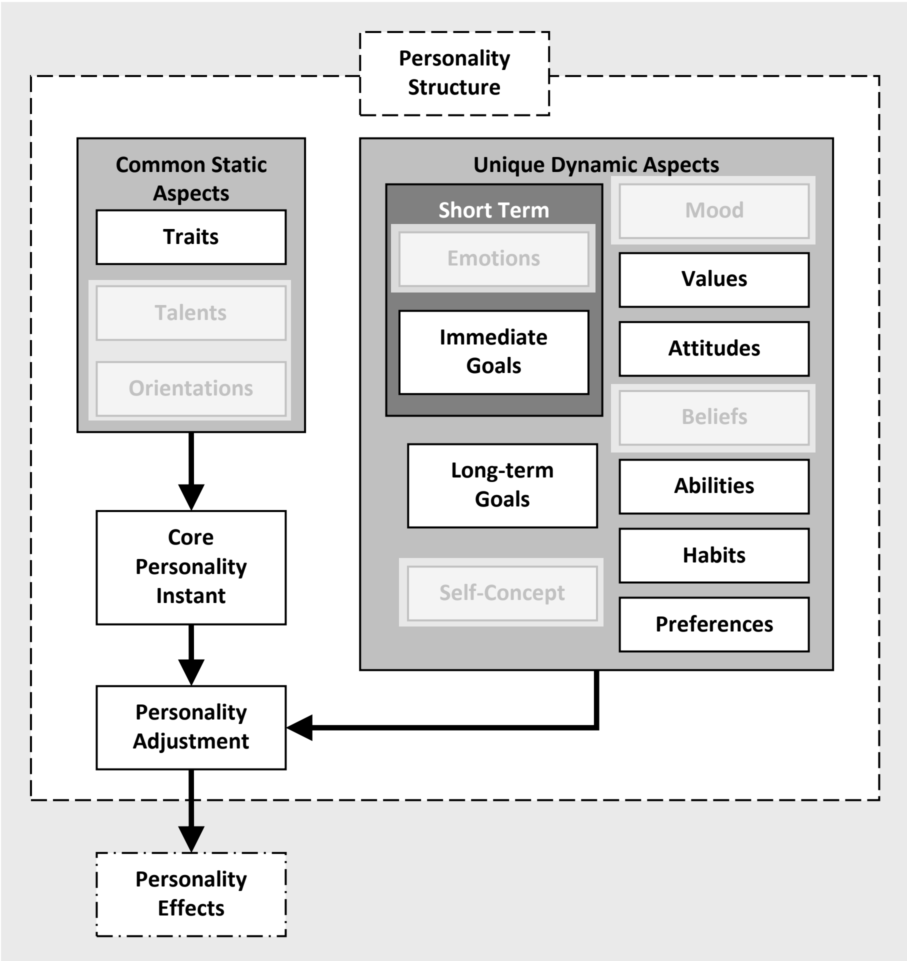


Figure 4.4 Proposed implementation of personality model

From the discussion in 4.2 it is evident that it is not feasible at this time to implement all the personality elements presented in Figure 3.7. To evaluate the feasibility of a PR based profile at this early stage it will also be beneficial to restrict the number of different variables that are tested in order to better assess their impact on the prediction process and the resulting personalised support. For these reasons a single element, that of traits, was selected to represent the common static aspects of personality and a number of elements to represent the unique dynamic aspects, as depicted in Figure 4.4.

Using a case study

The rationale for using a case study to evaluate the feasibility of the models developed in 3.3 is that personality data is unique to individuals and the process of simulating that data, through randomisation or other methods, would not reflect this uniqueness. This is particularly important in the evaluation of the interaction between dynamic and static aspects, and determining whether their respective predictions reflect observed behaviour. In addition, this is an experimental development and at this stage it is uncertain how various aspects of the personality model or the process connecting them will be implemented. Using a case study will provide the development with a genuine implementation context.

There are number of other benefits to using a case study over more traditional approaches, such as precision and recall (2.1), to evaluate a personalisation system. Case studies provide a holistic view of the impact of the system on the user as well as functionality of the system. Understanding that a system can predict or recommend to a certain level of accuracy does not necessarily mean that the user perceives the results in the same way. It does not measure how personalised the results were to the user. By taking a case study approach there is scope to identify to what degree the PR profile can personalise educational support and what impact that has on the user. As part of this approach there is also an opportunity to take into account the practicalities of developing such a system, how effectively the PR can be applied to a real world scenario. These are discussed in 5.3.

However there are also a number of limitations. First by not using traditional measurements there will be no precise way to measure whether the developed system technically performs better than any other. The use of a case study, in this case the development and evaluation of a system, adds a number of layers on top of the model, making it more difficult to precisely test individual elements of the model, or attribute particular outcomes directly to some aspect to the models design. For this research the advantage of using a case study approach is to examine the feasibility of the concept from a number of perspectives. Returning to the scenario described in 1.1 how possible is this level of personalisation and is the use of a PR profile a step in the right direction or not?

What will be evaluated

From the perspective of evaluating the feasibility of the PR profile as a means to personalise services there are three aspects of the model that will be evaluated:

- The effectiveness of the prediction process generated by the PR profile.
- The perceived personalisation of the support by participants.
- The effectiveness of the system to produce support.

The predictive effectiveness of the system will be evaluated from three perspectives. Firstly how accurate the system can predict the behaviour of the user given the same context. Secondly how accurate the system can predict the behaviour of the user from one scenario to another. Finally how well the system can predict from a known user to an unknown user via the static aspects alone. This will contribute to the overall technical feasibility of the concept. The methods used to evaluate this aspect are presented in 6.1.1 and the results in 6.2.1.

It can be reasonably argued that the primary evidence for evaluating the personalisation of any service or system can only come from the individual to whom that personalisation pertains. This is no trivial task as asking the user, *how personalised was that support,* relies on them having had previous experience of other support systems or similar systems without personalised support. However the user can be asked to evaluate the support they have received from the perspective of their current context and whether the support was relevant or not.

To evaluate the users perceived level of personalisation the nature of what is being personalised needs to be considered. In this research the case study, is presented in Chapter 4 and Chapter 5, was developed around the concept of learning support. Therefore, it is the generated support that needs to be considered by the users. Again there is no straightforward means of evaluating the support from the context of personalisation, as the user maybe more focused on the quality of the support. To overcome this four qualities were identified that focused on the evaluation of the context of the support, rather than the support itself.

These qualities are understandable, relevant, helpful and timeliness. As in the support should be *understandable* within the current context, *relevant* to what the individual is doing at the time; it should be supportive, *helpful*, of the current context and it should be presented at the *time* it is required.

Although not derived from, these qualities are in the spirit of the Technology Acceptance Model (TAM) and the construct of *perceived usefulness*, as demonstrated in (Hella and Krogstie 2011 and Hu 2010). Similar measures are also used in HCI evaluation of the personalisation of interfaces and artefacts (Dı´az et al 2008) and to assess users' perceived quality of recommendations in (Cremonesi et al. 201l). It is put forward here that in the context of personalising support to an individual one of the main factors that determine whether it is perceived as personal is whether it is perceived as useful within the current context. This perceived personalisation data can then be compared to a number of aspects within the case study to determine if there is any relationship between them as outlined in 6.1.2. This part of the evaluation process will contribute to the overall operational feasibility of the models. Do the models enable to some degree the personalisation of some service? The methods used to evaluate this aspect are presented in 6.1.2 and the results in 6.2.2.

The evaluation of the system to generate support is a means of determining the general quality of the system development and how this impacts on the evaluation case study as a whole. This will be assessed from the perspective of the level of success in generating support and the users' evaluation of particular aspects of the system, such as the interface and editor. The support produced by each of the dynamic and static aspects will be analysed individually and will give some indication as to the impact of each of these on the concept as a whole. However this will not give a complete picture, as there are a number of other system elements that may also have impacted on this process as well. This will contribute to the overall technical feasibility of the concept. The methods used to evaluate this aspect are presented in 6.1.3 and the results in 6.2.3.

Having identified what can be implemented and tested the following chapter discusses the key points in the development and implementation of the Personality Representation Support System (PReSS) prototype. This highlights a number of issues encountered during the implementation and how they impact on the assumptions made in 4.3 concerning the potential benefits of using personality representation as a mechanism for profiling.

Chapter 5
Case study: PReSS prototype design and implementation

This chapter presents the Personality Representation Support System (PReSS) that was developed as a case study to test the models in 3.3 as described in 4.4. As shown in 4.4 the prototype is focused only on the relationship between personality and behaviour within a given context, in order to ensure that the effects of personality on personalisation of services can be clearly identified. The PR profile that the prototype has utilised focused on those elements of the personality model, shown in Figure 4.4, that at this stage are assumed to be most readily computational. This PR profile includes both dynamic and static aspects in order to test how they can be utilised and how feasible they are within such a system. The purpose of this development is ultimately to personalise a service. In this case the service is learning support.

The prototype brings together a number of elements discussed in Chapter 4. These include the process of prediction demonstrated in Figure 4.1, the discussion in 4.2 on how to represent the aspects of personality and builds on some of the ideas presented in the discussion of a theoretical system in 4.3. The aim of this prototype is to capture a user's personality and utilise it to predict behaviour in order to provide personalised support from a pool of generic support resources. The process of developing this prototype also enabled this research to evaluate the technical feasibility of implementing a personality based profiling system which is discussed at the end of this chapter in 5.3.

The context of this case study is the support of students learning programming concepts using the Java (Oracle, 2011) programming language. There were a number of reasons for selecting this context. Firstly this research was being conducted within the Centre for Internet Computing (CIC) which initially made a pool of potential Java programmers available for the evaluation process. Although as stated in 1.6 by the time this research reached the evaluation stage this pool did not include first year undergraduates. Secondly the researcher has experience of supporting practical programming labs on Java giving some initial insights into the types of issues and problems experienced by those learning to programme.

When starting to consider this development it became apparent that to capture the almost infinite patterns of behaviour for any given context was beyond the resources of this research. So it was decided to identify a context that had natural limitations. It was assumed that since a programming language has a limited set of constructs and instructions there would also be a restricted set of potential behaviours. This assumption was to prove unfounded during the development and as discussed in 5.3 is one of the issues identified with the practical applications of models (Figure 3.6 and Figure 3.7).

There are also limitations with selecting such a structured context area, in that it does not necessary follow that the results will be applicable to less or unstructured context areas. Also programming is a practical activity and thus the findings may also not be relevant to more theoretical based contexts. However the process of developing such a system should reveal initial issues and/or benefits that can be addressed in future developments within different context areas.

Before embarking on the development of the prototype some initial groundwork needed to be in place. First the types of behaviour patterns students exhibited while programming need to be identified and a suitable representation devised (5.1.1). Secondly what types of problems do students face while learning to program (5.1.2)? The following section focuses on gathering these requirements by conducting an observational study of student's programming and interviewing a number of Internet Computing tutors about the issues they have observed.

The development phase of the prototype is presented in 5.2. This includes a review of agent platforms (5.2.1), the primary functions of the systems (5.2.2), how behaviour is predicted from the personality attributes (5.2.3), and how the support objects are constructed (5.2.4). During this phase of the development there was a significant change to how the static and dynamic aspects of personality interacted with each other. This changed the interaction between them from a parallel process to serial process, with each producing a separate prediction rather than the static prediction being weighted by the dynamic aspects as discussed in 4.2.1. This is highlighted during the development phase but discussed in more detail in the final part of this chapter (5.3) which reviews the whole implementation process in the context of how practical and technically feasible the concept of PR profiling is.

5.1 Requirements gathering

From the perspective of implementing part of the behavioural framework, as indicted in 4.4, the prototype was required to focus on the relationship between personality and behaviour within a given context, in this case programming tasks. To achieve this understanding of user behaviours within this context was required as well as the identification of users' personality. To gather this information an observation study was undertaken to identify student behaviour patterns in regard to completing programming tasks. During this study the participants were also asked to complete two surveys to gather data on their static and dynamic aspects. How the behaviour patterns were analysed from the observations is discussed in 5.1.1. Their link to personality and the process of prediction was developed later during the implementation phase and is discussed in 5.2.3.

The purpose of predicting behaviour from personality is to be able to better personalise some service. In this case the service is that of programming support during practical lab work. In order to do this an understanding of the types of problems students encounter was required as well as how this could be supported. The observation study provided initial insight into some of the issues. To support these observations and identify any other potential areas for support a second study involving a group discussion with the teaching staff of the CIC was undertaken.

The results are presented in relation to the requirements they pertain to, with 5.1.1 drawing exclusively on the observation study to identify behaviours and their method of representation and 5.1.2 drawing on both the observation study and the group discussion to identify areas of support. The methods used during these studies are described as follows:

Observation Study

The observation study took place in three phases. In the first phase participants were required to complete a third-party online personality questionnaire (Johnson 2000; Johnson 2005), the tabulated results are presented in Appendix F.2. The data from this instrument was then entered into the statistical analysis tool SPSS (SPSS 2006; Leech et al. 2005) for correlation with the behaviour pattern labels. This process is described in 5.2.3.

The observation phase consisted of a simulated lab experience with individual participants. Each participant was required to perform a set of the programming tasks, Appendix A.5. Their interaction with the computer was captured using screen capture software Morea. This recorded keystrokes, voice and video for each participant. They were able to access all the resources normally available in a lab situation. The researcher undertook the role of lab officer, giving assistance when asked for, and advice in the same manner that is used in labs. This enabled the researcher to observe less intrusively.

The final phase was undertaken by participants after the observation phase. This consisted of a self-evaluated programming preference questionnaire shown in Appendix A.2. This questionnaire was designed to capture the dynamic aspects of personality that are relevant to learning and programming. It is similar in style to how explicit data is gathered by the recommender systems discussed in 2.1.1.

However there are potential limitations with this approach, which are common to such surveys. There is no guarantee that the participant has responded honestly, understood the question as intended or that the question relates directly to the dynamic aspect it is being related to within this context. In addition to these issues, is the fact that at the point of gathering this data it was not clear how it would be represented within the personality profile or how it could be used within the support objects? Therefore, it was decided to generate a standard survey structure and make adjustments with the data later. However when it came to applying this data within the dynamic aspects of the profile they were not directly transferable and several methods were adopted to translate the values into something meaningful for the system. This transfer process is detailed in Appendix A.4. This process may have impacted on the performance of the dynamic aspects as discussed later in 5.3.

The results of this questionnaire are presented in Appendix A.3. This data was used to populate the following dynamic aspects *ability, values, attitudes, goals* and *preferences* and uses the explicit method of data capture. The dynamic aspect *Habit* was populated directly from the observation of repeated behaviours during the study, simulating the implicit method of data capture.

The application of both surveys and their link to behaviours is discussed in 5.2.3. Invitations to participate were sent via email to all CIC students that had completed the first year module on Java programming. This included both undergraduate and postgraduate students at various levels of programming ability. Participation was entirely voluntary.

Group discussion

In the group discussion the participants were invited informally to attend the session. Each was asked to complete a consent form and received an explanatory statement. The researcher posed a series of questions to the group and encouraged discussion in an exploratory manner. The discussion was kept fairly open and flexible with the questions acting as a guide. The focus questions were as follows:

1. How do you identify a student has a programming problem?

2. Given the following tasks what sort of problems would you expect to occur? (Participants shown the two task sets selected for use with this research, as shown in Appendix A.5 and Appendix B.2).

3. From the problems identified how have you or how would you go about supporting them?

4. What none programming specific learning do you expect to occur from the process of learning to programme?

5. What programming specific learning occurs from outside the programming module?

Questions one to three are specifically looking at support issues regarding teaching programming. Four looks at which activities or concepts within programming are reinforced from other modules. Question five explores whether support for learning programming occurs externally to the programming module.

The resulting discussion was recorded and transcribed. The transcription was then analysed and thematically coded. This focused on the identification of student programming difficulties and support methods that had been used or considered. The coded text was then grouped into common themes under the heading of each question. Questions four and five were combined as responses tended to indicate both perspectives.

5.1.1 Identifying patterns of behaviour

Before analysis of the observation study data could begin a method of behaviour description needed to be identified. Hierarchical Task Analysis (HTA) (Dix et al. 1997) a method in human computer interaction (HCI) was used to capture user interactions with both physical and digital systems in order to identify how best to develop interfaces for software processes.

Hierarchical task analysis

The square boxes, in Figure 5.1, represent a single behaviour within the behaviour pattern. The solid lines above and below each row of behaviours indicate their groupings and where anyone behaviour is linked to another sub-behaviour pattern via the vertical lines. Each behaviour pattern is consecutively numbered so that it is easy to identify their order and relate sub-behaviour patterns.

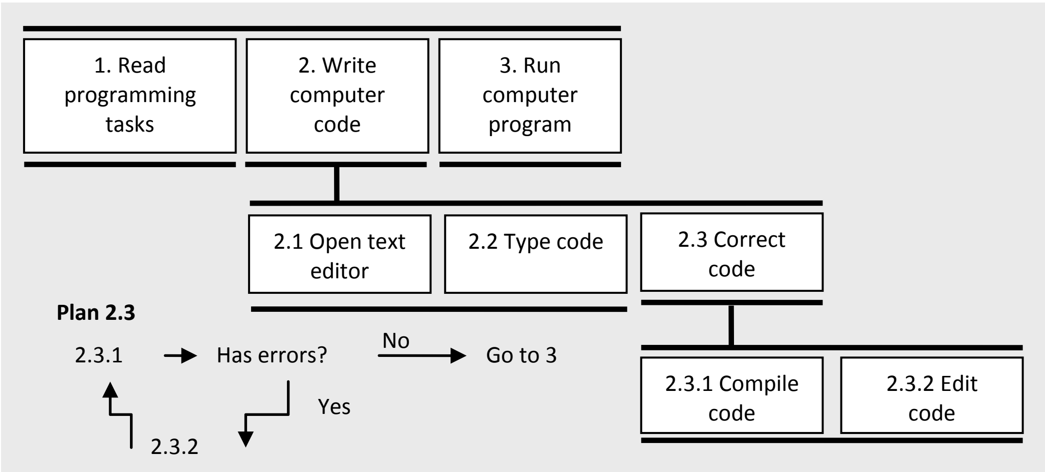


Figure 5.1 Example HTA for undertaking a programming task

Sometimes the linking of behaviour patterns alone does not accurately capture the picture of what is occurring. To assist, action plans are noted alongside the relevant behaviours. In Figure 5.1 the process of correcting code is iterative until there are no compile errors. To indicate this plan 2.3, relating to behaviour 2.3, shows the iterative process and its conditions. If there are compilation errors the process of editing continues, if not the user jumps up the hierarchy to behaviour 3 and runs their program.

Adapting HTA

Behaviours in the structure are described using a short phrase. However this phrase does not necessarily follow any particular protocol and can be as long or as short as the developer sees fit. As these descriptions will eventually be read and compared by a piece of software it was decided that this required a more simple structure. Thus the long descriptions in Figure 5.1 were condensed into a simple `action:focus` pair. For example, if a participant was found to be reading one of the tasks in the browser, this was shortened to `read:task`. The action being: reading was represented by the word: read; and the focus of the action, the set of tasks, represented by the word: task. The colon is used as a separator so each keyword of the statements could be easily identified.

This `action:focus` pair is the basic behavioural unit used throughout the analysis process to describe all the students' interactions with the system. As no behaviour generally happens in isolation, these pairs are strung together to form behaviour patterns. For example the high-level view of the HTA in Figure 5.1 is represented as:

```
read:tasks, write:code, run:programme
```

Dividing a single long behaviour up into sub-behaviours provided two positive outcomes from the perspective of the prototype. Firstly, it reduces the amount of storage required as short behaviour patterns are more likely to repeat more often than longer ones. Secondly, the system is able to use the high-level view in situations where detail adds no value to the current actions and predictions.

Data Analysis

For each participant observed the screen capture software produced four types of data. The participant was recorded via web-cam, capturing their facial expressions and eye movement giving an indication of which part of the screen they were currently focused on. An audio recording captured queries asked by participants and the lab officers responses. The screen interactions were recorded giving a visual representation of the participants' interactions with various applications. The participants' physical actions, keystrokes, mouse clicks were also captured and linked to the individual applications they were using

During the observation the researcher was able to make notes, via a remote observation computer, of any other external behaviour. This was done using the set of pre-arranged markers for the identification of such activities as: uses books, user asks question, and writes notes. The focus during the observation was to note all those actions that would not be easily identified later from the recording. These included: when the participant referred to books or personal notes; when the participant wrote pseudo-code or other notes; any external environment factors that may impact on the participants' attention; and any unexpected actions such as computer problems.

After the observation each recording was reviewed. Markers for the activities that were observable from the screen capture, audio and web cam were added. This process adhered to a strict set of pre-defined rules so that the application of these markers was consistent across all the participants. An example rule is: Statements that are on multiple lines should be left and counted as multiple lines, regarding the ratio of white space to code in defining the quality of code formatting. The screen capture software was then used to produce a single comma separated file for each participant. This file contained the markers and keystrokes in chronological order. This was then imported into a spreadsheet for the next stage of the analysis.

TIME	EVENT	KEY	MODIFIER	NOTES	APPLICATION
03:19.6	Keystrokes	/		** type comment "Task 1"	SciTE.exe
03:19.8	Keystrokes	/			SciTE.exe
03:20.2	Keystrokes	Shift	Shift		SciTE.exe
03:20.2	Keystrokes	Spacebar			SciTE.exe
03:20.4	Keystrokes	T	Shift		SciTE.exe
03:20.6	Keystrokes	a			SciTE.exe
03:20.6	Keystrokes	s			SciTE.exe
03:20.8	Keystrokes	k			SciTE.exe
03:21.0	Keystrokes	Spacebar			SciTE.exe
03:21.4	Keystrokes	1			SciTE.exe
03:21.8	Keystrokes	Enter			SciTE.exe
03:22.2	Keystrokes	Enter			SciTE.exe

Table 5.1 Analysis spreadsheet

Table 4.3 shows an excerpt from the analysis spreadsheet. All the columns bar the one labelled notes are produced by the screen capture software. The *Time* column indicates the time the event occurred from the start of the recording, the *Event* column indicates the type of event that occurred. In this example all events are of the type Keystroke. Key indicates the primary keystroke made by the participant with modifier indicating a secondary keystroke. The second highlighted area shows that the key T was pressed with the shift key to produce the capital T in the column. Code and description are used to identify the marker event which is not present in this example. Application gives the name of the application within which the event occurred.

Each participant produced between 2,000 and 5,000 keystrokes, and markers. A large percentage of these were duplicate keys like the last highlighted section of Table 4.3. These are produced when the key is held down for a period of time. All the duplicates were removed leaving a single occurrence of each. Then each word or command was tagged with a string describing it and its function. The first highlighted area in Table 5.1shows that the participant typed a comment starting with // that contained the text *Task 1*. Once this description was in place the keystrokes that belonged to it could be deleted. When this process was finished the following keystrokes were also removed: *Alt, Capslock, Ctl, Del, Down Arrow, End, Enter, Left Arrow, Right Arrow, Shift, Spacebar, Tab, and Up Arrow.*

The next stage was to change the keystroke's textual description and markers into a standard set of behaviours using the `action:focus` pair described earlier in this section. Taking several iterations to ensure that the same behaviours were described using the same tags. At this stage the level of granularity that would be required for statistical analysis was uncertain. In order to retain some uniqueness between similar behaviours extra information was added between parentheses. This included information such as: whether the code statement was complete or incorrect; where copied code had originated; and variable, class and method names for tracking.

Appendix A.1 shows an excerpt from a completed behavioural analysis for one of the participants. This shows the transformation of the section in Table 4.3 to its tag description `type:comment` in the highlighted area. The textual description originally contained more information, namely the text within the comment. This has been discarded from the tag as it is unlikely that many participants would write exactly the same text, making correlations less likely to occur between personality and behaviour. This might not be the case if the number of participants were a great deal larger.

The highlighted groups in Appendix A.1 have a high-level behaviour (usually the first line of the block) and several sub-behaviours that it is composed of. For example, the class declaration tag is stated as `type:class declaration (complete)`. This is followed by `type:modifier public` and `type:class`. Both are required to make a complete statement. Then the participant stops coding and `read:tasks` to find the class. This is then `copy:class name (tasks)` and `paste:class name (AverageMark)`. They then study previous coding examples `thinking:sample code (ValidTriangle1.java)`, before completing the class declaration by `type:brace pair (class)`. This group is a behaviour pattern.

Taking all the high-level descriptions, high-level behaviour patterns can be produced. These are coding tags such as `type:class declaration (complete)` or tags with no sub-behaviours such as `read:tasks`. From Table 4.4 the following behaviour pattern can be produced: `open:new file, type:comment, thinking:sample code (ValidTriangle1.java), type:class declaration (complete), type:comment, paste:main method declaration (complete).`

If required behaviour patterns can be re-tagged to form a new high-level pattern. For instance if search:files, open:file thinking:sample_code is a frequently occurring pattern across participants. It can be re-tagged as search:code for example. The previous pattern becomes the sub-behaviour pattern.

Using this process common behaviour patterns can be identified across participants and linked to instructions within the tasks. Identifying which behaviours belong to each task is obvious from observation. All the participants completed them in a sequential order, always completing one before proceeding to the next. However, connecting the behaviour patterns to the task and individual instructions is more difficult. This is achieved by using the support area as a focus to what should and should not be considered as discussed in 5.2.4.

5.1.2 Identifying areas for support

The issues around learning to program are not a new research area; but there are still a number of research groups looking into this, such as (PPIG 2011). It has already been mentioned in 1.6, that the area of adult learning and specifically that of learning to program are not areas considered central to this thesis. Although to test the models in 3.3 this case study is focused on the support of learning to program, the focus is on the issue of personalisation and not the effectiveness of that support. However in order to ensure that the case study presents a balanced assessment of the feasibility of the PR concept it was necessary to give some consideration to the nature and quality of support the system was to provide. As a review of the current research in learning to program was beyond the scope of this research it was decided to utilise the observations made during the observation study and support them by a short study into examining the experiences of the then current CIC teaching staff.

One reason for selecting to investigate the programming issues experienced by the CIC students directly was to ensure that the type of support developed would be directly relevant to these evaluation participants. There was the possibility that by drawing on externally identified issues they may not have been relevant and thus the system would fail to be effective. However the reverse of this is that by only drawing on this department's experiences the system would be bias towards success. Drawing on both aspects would have given the system a balanced approach, but as discussed in 5.2.4 there were a number of difficulties with finding and adapting relevant external support objects.

Three sources were used to identify potential areas for support. Two sources were drawn from the observation study described previously in 5.1. The programming tasks undertaken by the participants were marked in terms of their key programming concepts and the types of errors experienced by the participants was analysed. The final source was a group interview with the CIC teaching staff drawing on their experience of teaching and supporting the learning of programming.

Results from observations

Two measures were taken from the observation study. Firstly a very simple marking scheme was applied to tasks one to three in Appendix A.5. One mark was given to each of the main elements of the task. Appendix A.6 shows the marks allocated to each participant. Secondly the types of compilation error were analysed as shown in Appendix A.7. The testing of code via compilation to discover how and if it works correctly is part of the process of programming, and frequently applied during the learning process. Here the analysis examined those compilation errors that took participants by surprise or caused problems during the correction process.

In task one only three students succeeded in producing a correct answer for finding an average. This tested the participants understanding of Java types. Those that did not produce a correct answer used integer types rather than floating points.

Task two showed two distinct problems. While the question requires a mathematical calculation the focus is on producing the results in a specified format. This requires some forethought and planning. To achieve this pseudo-code was requested as part of the question. Table 4.1 indicates that only three participants wrote pseudo-code successfully prior to coding the program. The half marks were allocated to those who initially started coding then noticed the question asked for pseudo-code, completing that before continuing with code. The rest of the participants either did not write any, or wrote it in hindsight. This is considered bad practise when learning to programme, as students need to be encouraged to design before they code. The second problem was in getting the output format as requested. This was a matter of using tabs and spaces to align the text in the output strings. While some attempted to do this there were minor mistakes that resulted in a zero mark being allocated. It should be noted that in lab sessions students often do not give a task all their attention, unless it is an assessed piece of work. This may have been the case for both these problems.

Two participants did not get as far as Task 3, so the results are based on only eight participants. The two problems were, writing an integer function and getting the maths correct were interlinked. Maths appears to cause some students a great deal of difficulty. To avoid having to understand the problem some participants searched for a Java method that would calculate a power for them. While this sort of thinking is not discouraged, those that used Math.pow from the Java API failed to wrap the method within an integer function, thus failing to complete part of the question. Those that attempted to write a function to calculate a power simply did not understand the concept. It was noted that while pseudo-code was not explicitly asked for here, only a few participants used it to explore the problem they were trying to solve.

The most common areas of complication error across these tasks were attributed to participants missing out symbols that are part of the codes syntax, such as the end of line symbol represented by a semi-colon or the pair of braces that define blocks within code. A third of these mistakes were made by one participant who cut and pasted sections of code from previous programmes, but did not check if the content was correct. Another third was produced by a participant who persistently forgot to close parenthesis or add end of line to their code.

More participants made capitalisation errors than any other error type. Java is case sensitive, but does follow a format convention. Class names are always capitalised and method names are in lowercase. There are very few packages in the Java API that don’t follow this. However, it is unlikely that any of the participants would be aware of them and they were not required during the observation study.

Three of the participants were experienced programmers and used several languages. The types of errors they exhibited were focused around capitalisation, operator symbols and class/method names. This was most likely due to lack of recent experience with the Java language rather than a misunderstanding of programming concepts.

Exception handling errors were exhibited by four participants. This was missing throws clauses or try-catch blocks. Those participants that cut and paste code from previous work did not suffer from this mistake if their code contained the try-catch block. They did if their previous code used a throws statement, as it is placed separately from the code they copied.

Mistakes with global variables contributed to the problems with the main method declaration. Those that used global variables did not understand why the program compiled correctly but then had run-time errors. As the error was focused on the keyword static within the main method, participants would then go and remove this. This is a conceptual misunderstanding of how variable modifiers are used.

There were several instances where participants used the wrong Java types as parameters or with incompatible operators. Some of this may have been due to differences in programming languages as explained previously. The other group in Appendix A.7 consists of those errors that occurred infrequently or it was not clear whether they were due to misunderstanding the use of a particular concept. These include: file and class names being labelled differently; understanding class package relationship; general typing errors; forgetting how to declare and assigning to an array.

Results from group discussion

There was a general consensus that there are no hard and fast rules regarding the identification of student support requirements. Some students are embarrassed by their difficulties and try to hide their work. They do not seem to appreciate that getting stuck is part of the process, often keeping a physical distance between themselves and the teacher. This behaviour can also be mistakenly attributed to problems. The student may just be thinking about the problem, needing time and space, or they could simply find arriving on time in the mornings difficult.

It was indicated by some of the group that supervised laboratory exercises are particularly good for identifying student difficulties. They also provide feedback on the success of the teaching methods used in relation to the students needs. This provides an opportunity to talk to students, giving them the chance to air their concerns and ask questions. Attention is given to all students in this manner so that none feel singled out or left out.

Assessment marks are also an indicator of poor comprehension, but are not always directly related to programming difficulties. Other issues such as understanding the tasks or writing skills may be a contributing factor. It was noted that sometimes a student whose performance is good in practical sessions can often have low marks, indicating that the subject itself may not be the problem.

The way students lay their code out is another indicator of their confidence and understanding. It particularly highlights whether they understand programming concepts as well as programming commands. For example it is usually obvious if a student does not understand how to construct a loop in Java or does not understand the need to repeat something. This is linked to the development of understanding algorithms and problem solving. One of the issues raised was the current trend to hack at code until something happens. Rather than thinking about solving the problem, and then producing the code. The majority of the group agreed that this was a major problem in teaching programming.

This idea about programming concepts is important in the process of teaching programming. Unlike other subjects, the language taught is in some ways irrelevant. It is used as a vehicle for teaching the concepts of programming, like looping, which can then be applied in any language. It is getting the balance between concepts and commands that is difficult as students often focus on the command aspect in their first year. As the course progresses this produces problems with more advanced ideas, as the students have missed the point of the first year programming concepts.

The next question put to the group was focused on the task sets (Appendix A.5 and Appendix B.2) that were being used in this research to gather initial data, and later evaluate the effectiveness of the prototype system. There was a general consensus that problems start with students not understanding the question. They indicated that there was nothing particularly wrong or ambiguous about the questions. Students just have difficulty in extracting the necessary information and data to complete the task. Those students that are not confident will often focus on one aspect that they understand. Like if the task asks for output. Then they hack the rest of the programme so there is something to output. This links into their ability to problem solve and build algorithms.

It was estimated that as much as 80% of students do not start with pseudo-code as a method of finding a solution. The majority take on a trial and error approach often cutting and pasting code from lecture examples and previous work. This often results in compilation errors that they, the students, have no idea how to solve. As they have not understood the things they have added to their code.

In both task sets, task one asks the students to find the average of a set of values. Some of the group suggested that many would not understand what an average is. Some would be due to difficulties with maths. Others, such as mature students, may have used averaging so often that they cannot explain how it works, thus cannot apply it to the concept of programming.

Other suggested problems with task one, are focused on the use of variables. Some students will find the idea of representing values with variables difficult in the early stages of learning. Then having to think of the variables as different types makes the task more difficult. It was suggested that a common answer to the problem would look something like this:

```
int total = 56 + 75 +40 + 86 +32 / 5
```

This would result in an incorrect answer for two reasons. First the values to be averaged need to be aggregated separately to the division process. In this example only 32 would be divided by 5, the result being added to the rest of the numbers. The second problem is that the variable has been defined as an integer (int) number whereas the result of division is most likely to require a floating point number. In Java this results in the numbers after the decimal point being truncated. So 32.6 would become 32. Understanding how different types are used in programming is an important aspect of modern high level object orientated languages.

For both task sets, task two is focused on obtaining user input, performing some calculation and output the results in a particular format. This requires the programmer to consider a number of aspects at the same time: how to catch exceptions; the format of the output; and how to implement a loop.

The task three in both sets is focused on producing a function that the main method uses. One group member suggested that students do not understand the difference between a function that returns a value and one that prints a value. This would result in the function not returning something to the main method, thus making its use less generic from some perspectives. Another concern here was the focus on maths again. The second task set asks for a function that raises one value to the power of another. There was a consensus that most students would not know what a power was and a simple multiplication would most likely be used instead.

Throughout the discussion several support methods had already been mentioned. These included: the use of practical laboratory based sessions; tutorial sessions; encouraging a research habit as self help; talking to students individually; and weekly feedback on lab reports. This part of the discussion focuses on a few methods that had been tried or debated during module development.

One area that was debated was that of peer support. There is a general understanding that those students that discuss and assist each other as a group, generally do better than those who do not. As noted by one member of the discussion group, this is a hard thing to quantify. Particularly when most lecturers may have only two hours of direct contact with students a week.

To encourage the use of this apparently advantageous support the master’s level programming module employed pair programming. Each student is paired to another of similar ability so they can work through tasks together. This was not successful, and many of the pairs split. The students themselves felt it did not help them at all. This technique was also used by second year students within their group work. They reported it to be quite productive. The difference between these two instances, may have been related to the fact that the method was imposed on the masters students, but self-imposed by the second year groups.

Another method that had been discussed by the department was the use of tool support by using development environments. There are many products available, such as Eclipse, initially developed by IBM (Eclipse Foundation Inc 2008) and the Sun Microsystems sponsored Netbeans (2008). These tools assist and to some degree write code enabling students to focus more on solving the problem than tackling the programming language. The main reason that this has not been adopted at first year level was the concern that the learning focus would switch to the tools functionality. This could add another layer of issues and difficulties for the students to overcome. The group felt it would be more advantageous to introduce tools at second year level, when basic concepts had been learnt.

On a computer science based course many of the concepts or abilities learnt in programming appear in other modules, and vice versa. These are referred to as transferable skills. The group identified three transferable skills that programming teaches: logical reasoning; problem solving; and learning according to rules. Basic computer skills are also boosted by practicing programming. Such things as file and folder organisation, typing skills and common function quick keys.

Other modules assist in the learning of programming in various ways. Learning about computer architectures and memory usage, help some students understand variables better. Different variable types are stored using different numbers of bits, which often help in understanding how and why there are integer types and floating point types. Concepts like static variables, polymorphism and arrays are often comprehended better when the student learns about programming memory.

One aspect that was perceived by some members of the group, to be an important contribution from other modules, was motivation. In learning about how programming fits into the bigger picture of Internet Computing, motivation to learn to programme increases so they are able to build more complex and integrated systems, such as a database driven website. This requires both knowledge of database design and access, but also knowledge of programming.

Understanding the bigger picture can often provide incentive to learn something that a student finds irksome.

Throughout the discussion students' individuality was often referred to as a reason for needing various support methods. From the perspective of identifying students with difficulties, their behaviour is often used as an indicator. It was noted in the previous discussion that a student who distances themselves from the lecturer is often hiding their problems. This is not a hard and fast rule, as in other individuals this may indicate a desire to tackle the problem themselves or that they are applying what they have learnt to more complex problems.

As students are different they require different approaches. This includes the method of teaching, how to support them when difficulties arise as well as the communication of feedback. It was commented that it may not be the lecturer's teaching methods that are incorrect more the lecturer's understanding of an individual's personality and how that effects what they require.

One of the reasons that were given as to why the pair programming, discussed previously, was unsuccessful was a clash of personalities. If the pair consisted of an introvert who dislikes sharing, and an extrovert who persistently gives, an imbalance may be perceived between the two.

Even though personality was discussed as a source of the differences between individuals, and thus their learning requirements, there were many references made to behaviours that were attributed to many individuals. Like the tendency not to write pseudo-code, or proceed in a logical manner. These two views, that there are aspects that are common among students learning programming and aspects that are unique to each individual, provides support for the decision to divide personality within the model in 3.2.5.

Another aspect that is referred to throughout this discussion is the differences in student abilities. Students who were perceived as struggling with particular programming concepts were also perceived as having lower abilities in logical reasoning and problem solving. Again, this lends support to the personality model, Figure 3.7, where the dynamic aspects of personality contain the attributes of ability. While these students initially have a difficulty with programming and low reasoning and problem solving abilities, through practise these aspects can be improved. This supports the need, as discussed in 3.2.5, for the dynamic aspects of personality, so these types of changes can be tracked and adjusted over time.

Support area	From Scores	From Errors	From Discussion Group
How to break up the process of finding an average	Seven students failed to arrive at the correct result		Decomposition of the problem was seen as a major hurdle in programming
Keeping calculations separate from output process			Students included as much as possible within a familiar statement type
Remembering to name the class and file names the same	Seven instance across three tasks failed to name the file correctly		Reading and understanding the question
Developing a habit of using comments within code			Best practice that is often ignored
Formatting code for easy understanding			Best practice that is often ignored
Developing the habit of importing new types		Eight compilation errors	
Using pseudo-code to understand a problem	Three students failed to write any and four wrote after the fact		Best practice often ignored or done in hindsight
Understanding that the main method never changes		Eight compilation errors	
How to use global variables correctly		Five compilation areas. 80% who used them had errors	
When to use different primitive types	Main reason for failure with finding an average		Understanding when to use different primitive types
Formatting output as requested	Nine failed to output correctly for task two, complex formatting		Pointed out there would be issues with the formatting and use of loops
How to handle exceptions using try-catch or throws		Nine compilation errors	

Table 5.2 Rationale for support areas selected

Combining the sources

From these three sources a compilation of twelve potential areas of support were identified and are presented in Table 5.2. How these areas were supported and how they were linked to behaviour prediction is discussed in the next section on the implementation process.

5.2 Implementation

The implementation followed an experimental approach as a number of the concepts and their interactions were only understood from a theoretical perspective and there was a need for continuous adaptation. To ensure that the development would be flexible an agile development (Beck et al. 2001; Ambler 2004) approach was used. The process involved the testing of small conceptual elements, integration into the whole and further testing. This iterative process enabled the development to quickly respond to those concepts that failed or worked together in unexpected ways.

Although this is a relatively new approach to software engineering, similar in some instances to rapid prototyping, it fitted well with the experimental approach of this work. More traditional methods such as the waterfall model and rapid application development (RAD) (Pressman 2000) are heavily design focused and reliant on stakeholder input and testing. While the end user was considered during the development process there were no opportunities for system testing directly with potential users. However the profiling data gathered in the observation study was used to test the various elements and the system as whole throughout the implementation process.

The prototype is based on the theoretical system discussed in 4.3. However it was not feasible to develop the whole of the system as there were a number of resources such as access to the university's VLE that were not available. There were also some concerns at the time about the process of evaluation and enabling student participants to access the prototype through their standard accounts. To ensure that access would not be an issue the system was designed to run on a single computer without network access as a minimum. Due to these limitations and time restraints the development focused on those critical elements that would enable the testing of models as discussed in 4.4. These elements were the development of the client-side multi-agent system and both the client and server-side personality representation (PR) databases.

As discussed in 4.3 there are a number of advantages to using a MAS. They have also been used in a number of recommender and educational systems such as (Hietala and Niemirepo, 1998 and Godoy and Amandi 2003) providing various forms of personalisation. However this was not the only platform available to this prototype development. During the design process other options were considered, including the Universities virtual learning environment (VLE) which was Blackboard (2006) at the time. There were a number of advantages to utilising this platform such as: student's already being familiar with environment, and potential data sources such as existing content and student input into discussion boards. Unfortunately the development kit for Blackboard was beyond the financial resources of this research and the University was in the early stages of considering its purchase. It was decided to continue with the agent platform with a view to being able to build on this in the future and further test some of the ideas presented in 4.3.

5.2.1 Reviewing agent platforms

Agent platforms are ideal for dealing with today's distributed and dynamic environments particularly that of the world wide web (WWW). Many recent eLearning applications are utilising agent and multi-agent systems as a means of providing their systems with mobility (Triantis and Pintelas 2004), enabling collaborative social environments (Vassileva et al. 1999) and providing learners with competitive peers (Viswanath et al. 2004). Section 2.2 provides a review of various eLearning applications, some of which utilise agent based systems. The key advantage of using an agent system was to be able to monitor, create and kill multi-threaded processes on the fly. The inherently module structure and cross platform abilities of some agent systems would also enable this work to be reused in future developments.

There are a wide range of platforms, tools and architectures available. These include: ZEUS, RESTINA, JACK (Luck et al. 2004), OAA(Anonymous 2004b), MAST (Anonymous 2004a) ABLE (Bigus et al. 2002, IBM 2000) Aglets (Jones 2002, IBM 2002, Bigus and Bigus 2001) and JADE (JADE 2004). RESTINA is a very mature framework but only recently developed as a multi-agent architecture and support was limited.

Of these ABLE, Aglets and Jade were tried and tested with a preliminary test development. ABLE and Aglets, either supported or developed by IBM, had had no recent development, whereas JADE was being updated frequently. ABLE is based on a system of pluggable processing modules, such as a backward chaining inference engine, decision trees and self organising maps. While this enabled quick development of logical agents it did not make for easy implementation of more generic algorithms. The Aglets platform provided a much more open architecture, and follows Java's object orientated style of development. However, like ABLE it has seen little recent development.

Of the systems reviewed JADE presented an adaptable framework that is standards compliant with FIPA's (2003) ACL (Agent Communication Language) (Fipa 2010). There is a range of support documentation and an active support community. Several extensions enable the platform to be used on mobile devices and integrate with web services which would be of benefit for future developments. The test development was fairly easy to implement after a few issues with getting the environment to run. During testing it was discovered that the communication process was able to use Java Objects with the message payloads. Although writing the more complex Jade `MessageTemplates` to identify incoming packages took time, this, made fine-tuning and tracking errors far easier than using complex formatted strings. Given this it was decided that this platform offered the most potential for current and future developments.

5.2.2 Overview of the system

The basic function of the system is to monitor the user's interactions within a coding editor and present support, referred to here as hints. Support is provided both when the user is predicted to find a particular concept or task difficult and when the user is predicted to perform behaviours that move towards successful completion of the task. This was done in order to provide balanced support, similar to that of a tutor within a practical environment, rather than just focusing on correcting. The nature of this support is discussed in 5.2.4.

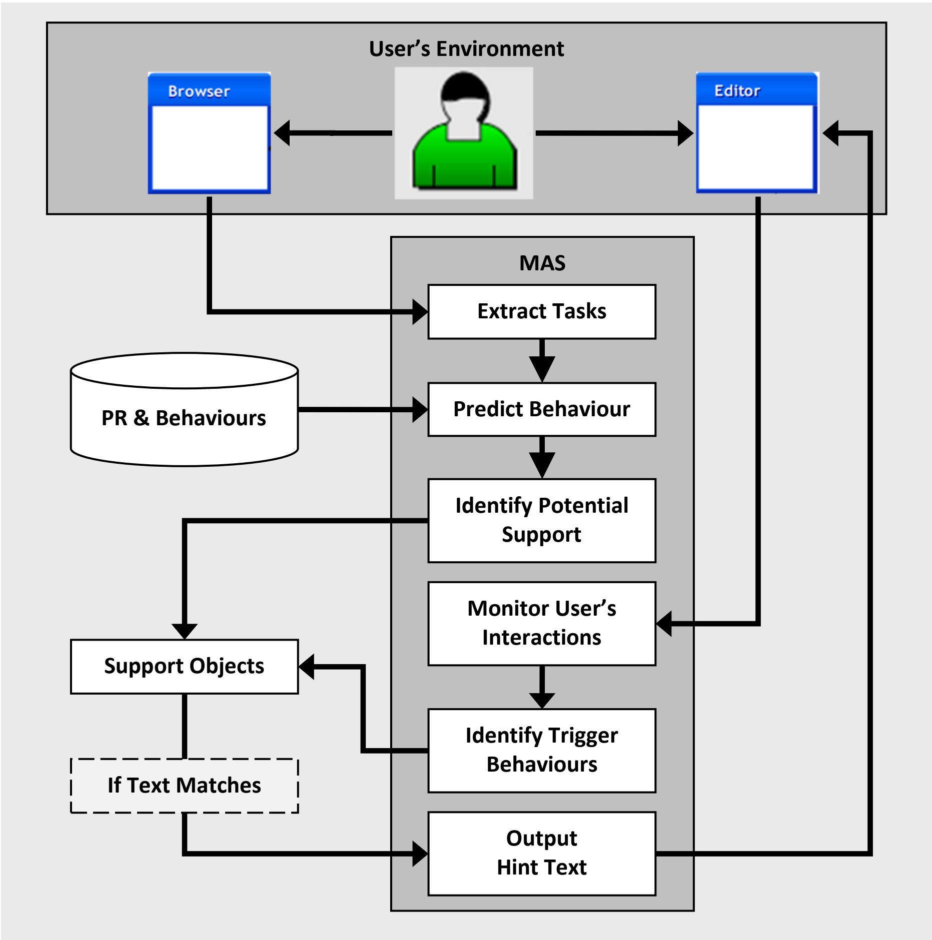


Figure 5.2 PReSS main processes

How this system achieves this is shown in Figure 5.2. Once the user has logged into the system, which enables the system to identify the user, they are presented with the programming tasks within a browser like environment. From here they can choose when to open the editor to practice the tasks displayed. Users are requested to select which task they are going to undertake before starting to code, using a series of dynamically generated buttons within the editor.

Images of these two interfaces are presented in Appendix E.2. Once the system has extracted the tasks that are going to be undertaken from the browser, it starts the process of generating predictions using the user PR profile and identifies the relevant support objects for those predicted behaviours.

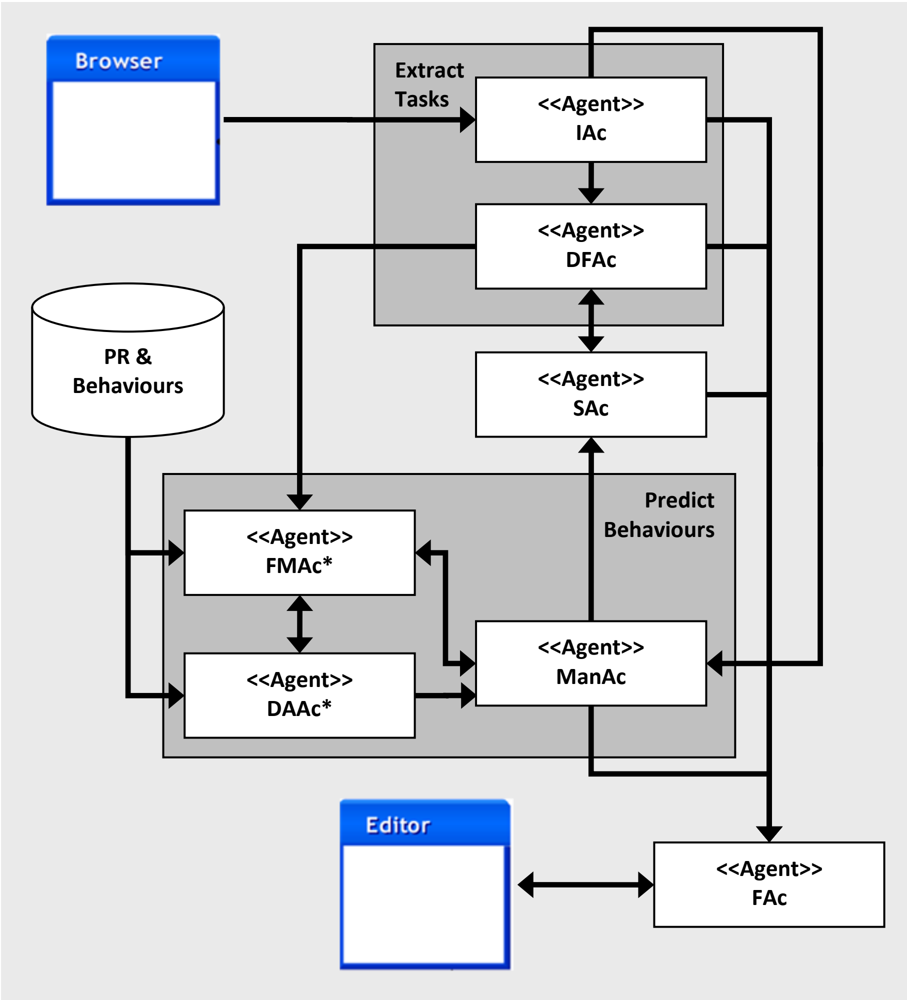


Figure 5.3 Overview of prototype agent system

From this point forward, unless the task set is changed, the system is focused on monitoring the students' interactions within the systems editor and browser. This monitoring process is looking for behaviours that match the trigger behaviours within the support objects. If a trigger is found this may result in either outputting a support hint or starting a checking process within the support object against a sequence of predicted behaviours, which may result in a number of hints being outputted from a single support object.

To achieve this the MAS system, shown in Figure 5.3, allocated the necessary processes to individual agents as follows:

Manager agent (ManAc)

As with all the agents in this system the ManAc's first task is to register with the local directory service. As the manager of the whole process this agent is one of five permanent agents within the system. The other four are created by the ManAc which also sends each a list of the local designations for all permanent agents

The ManAc has three main tasks. The first is to monitor the number of on-the-fly agents that are created. The second to establish if all the personality and behavioural data has been gathered for each task set and trigger the calculation process for the predictions. The final task is to create a backup file and a tracking file. The backup file enables the ManAc to retrieve all the session information thus far, in case of failure. Prior to this point it is more efficient to regenerate from the database. The tracking file is a human readable version that allows the process to be tracked manually.

Interface agent (IAc)

IAc agent is responsible for the login and browser interface. It has access to the database for checking the user login details and extracting the users profile data. Once login is complete the browser is pane is opened and the tasks displayed Two menus are available within the browser: File and View. File->Exit exits the system and View->Editor opens the test editor for coding. The functions have been kept to a minimum for the prototype as the evaluation process (Chapter 6) will only utilise a single task set. It was intended that the browser support internet access, but after issues with monitoring the change from tasks to web page this was put aside for future development. Images of all these interfaces are in Appendix E.2. This agent passes the URL for the tasks in the browser to the ManAc and DFAc along with the user's profile.

Dynamic formatting agent (DFAc)

This agent is responsible for extracting the necessary data from the task page (HTML document) presented in the browser. To enable this process customised tags have been embedded in the HTML document. These tags identify keywords in the text that represents each instruction required to satisfy the task. The instructions take a high level view and focus on those aspects essential to task completion. An example of this tagging is shown in Appendix E.1. Once the instruction list for each task has been produced they are used to create *instruction packages* within a *task package.* Once created the *task packages* are forwarded to FFMAc's, and the DFAc has no further responsibilities unless a new file is opened in the browser.

Five Factor Model agent (FFMAc)

The FFMAc's primary purpose is to retrieve all the *behaviour patterns* associated with the current user's static personality aspects and the *instructions* from the database. There may be zero to many *behaviour patterns* for each *instruction* within a *task*. Each *behaviour pattern* is accompanied by the statistical data, discussed in 5.2.3, linking it to both user and *instruction*. Again this information is packaged up into the Task package for later processing. For each *instruction package* a new DAAc is created to process it. When the DAAc has finished processing, the *instruction package* is returned back to the initiating FFMAc, which then forwards an updated and complete *task package* to the ManAc and self-terminates.

Dynamic Aspects Agent (DAAc)

The DAAc uses the data within the *instruction package* to retrieve dynamic personality data from the database. All the dynamic aspects, as listed in 5.2.3, (bar *habit* which acts at the *behaviour pattern* level) act at the *behaviour* level. So for each *behaviour pattern* within each *instruction* there are many individual *behaviours* to check. This process is achieved by using internal Jade agent behaviour `ParallelBehaviour`. This data is repackaged into the original *instruction* which is returned to the parent FFMAc, and the agent terminates. After having informed the ManAc that it is doing so.

Support Agent (SAc)

When the SAc is created by the ManAc at application start-up, its first task is to upload the SO data from a flat file system and create Java objects. This is basically a parsing exercise using a series of custom tags; an example is given in Appendix C.3, to identify the various attributes of the Java object.

The SAc receives a complete *page package* from the ManAc with all the calculated predications for each *behaviour pattern*, for each *instruction*. The *behaviour pattern* that scores the highest for each *instruction* is selected. For each instruction there is an associated SO which contains rules regarding how to support each possible *behaviour pattern*. The SAc identifies the most likely *behaviour pattern* with the SO, and uses the associated static-hint (Appendix C.1) and dynamic-hint (Appendix C.1) rules, to edit the *behaviour pattern*. This data is then stored within a *prediction* and all the predictions for each task are stored in a *prediction set*. This process is repeated for each *task package* and the results are repackaged into the *page package* before being passed to the FAc. This concludes the SAc contribution to the process.

Feedback Agent (FAc)

The FAc has the most prolonged activity of the agent group. Its primary function is to monitor the user's activities within the editor and provide textual support from the *predictions* embedded in the *page package*. The editor shown in Appendix E.2 is opened by the user from the browser using *View->Editor*. The editor consists of three panes: the response pane (at the top of the editor) is where the FAc posts the *hints* and other instructions; the coding area (central pane) for users to write and edit their code; and finally the compile pane (bottom of the editor) where the results of code compilation are displayed.

The three menus on the top bar contain standard file and edit commands as shown in the interface image of the editor in Appendix E.2, with a compile command available within tools. The second bar displays the current user's alias and the current task being coded. Notification of a new *hint* within the response pane is also presented here, as shown in Appendix B.1 instructions for the participants.

The central line of buttons is dynamically generated from the number of tasks sent by the DFAc, which are used by the user and communicate the current task to the system. It was originally intended for task identification to be done via the monitoring system. However, this proved too complex at this early stage of development. As programming generally follow a similar starting pattern to could be some time before any unique features appeared and could be used to identify which task was being undertaken. This resulted in much of the support being passed over and the task being almost complete.

As the user enters code into the editor the FAc identifies each statement type and generates a model of the code. Each time more code is added a new model is created and compared to the old one. Those statements that are new or have changed are used to build a model of the order in which the code was written or edited. Each statement or change identified as new is compared to the *behaviours* within each of the SOs. If the checked *behaviour* has a *hint* attached this is sent to the response pane of the editor. The text `*New Hint*` appears on the information bar at the top of the application. This remains for a short period of time before being removed.

Database

All the personality and behaviour patterns data was stored in a single database, rather than implementing the server client division discussed in 4.3. Appendix E.5 presents the entity relation diagram for the database design and the final table structure is shown in Appendix E.8. There were no major changes from the initial design through to implementation via a Postgres (2006) platform. A single Java class was developed to provide all the agents with access to the database as well as to enable the population of data from a single file.

5.2.3 Linking PR to behaviours

This is perhaps the most critical aspect of the system implementation and contributes substantially to evaluating the feasibility of using a PR profile for the purpose of personalising service. It is the action of predicting behaviour from the PR profile that will determine what type of support a user will receive and ultimately how personalised the support service is.

The model of personality in Figure 3.7 shows how personality is divided into two parts: the dynamic aspects; and the common static aspects. In Figure 3.7 the static aspects, represented here by traits alone, provide an initial core personality instant. From the perspective of prediction this is the statistical prediction formed from the correlation between traits and behaviour. The model then goes on to adjust this initial prediction using the dynamic aspects. However this process proved to be unrealistic when applied within the system. When the weighted behaviour predictions were compared back to the observed behaviour of the participant's they were generated from they failed to predict the original observed behaviour. It became apparent that the static aspects did not require weightings to predict back to the original behaviour, they were working effectively without this. For this reason the static and dynamic aspects were implemented as two parallel systems. Each aimed at providing support, but from two different personality aspects. The impact of this is further discussed in 5.3. How each of these processes was then implemented is as follows:

Static personality aspects

The main focus of the static aspects is that of traits, and their representation using the five factor model (McCrae and Costa 2003). These are represented within the prototype at both the factor and facet levels and are described in Appendix F.1 and the results of the third-party online questionnaire (Johnson 2000; Johnson 2005), are presented in Appendix F.2. This data is stored as several tables in the profile database as shown in Appendix E.5.

The main issue here is the linking of this personality data to the behaviour patterns given different tasks or instructions. This needs to be achieved in such a way that the personality data can be used in similar situations or with users that the system has not yet encountered. The means of linking also needs to enable the system, in future developments, to link new behaviour patterns to factors and facets.

Two statistical methods were used to provide this link. The first is by analysing linear relationships between behaviour and personality through Pearson's correlation coefficient. This takes a set of values and determines whether there is any relationship, negative or positive, between them. For each user there is a set of values relating to their personality type as described in Appendix F.1. For each instruction, there is a set of behaviour patterns that satisfies it, each with a high level label. By applying Pearson's correlation the degree of relationship between the factors or facets and the behaviour pattern label can be determined.

Appendix D.1 shows the correlation table produced for the instruction `use:primitive types` and its set of potential behaviour patterns: `use:incorrectly`, `use:correctly`, `use:correctly after error`. From this table those factors or facets that show significance, the highlighted values, are selected. For the behaviour pattern `use:incorrectly` there are two significant values: A5 (Trust) at -0.607* and O6 (Liberalism) at -0.584*.

These values can then be used with the second statistical method, regression analysis. This produces a second set of values that can be used to calculate the probability of each behaviour pattern given the personality values. Appendix D.2 shows the results of regression analysis for the behaviour pattern `use:incorrectly`. Regression analysis calculates the line of best fit for each independent variable, in this case, the factors or facets that showed a significant correlation with the behaviour pattern in question. It also calculates a constant; also known as the intersection of that line (Miles and Shevlin 2001). These are then used in the following regression equation, where y is the resulting probability, b is the slope for each factor or facet, x is the original user's value for the factor or facet and c is the constant.

$y = b1x1 + b2x2 + c$

The values for b and c for each behaviour pattern are stored within the database linked to the behaviour pattern label. Once the instruction set has been identified, the behaviour patterns and their corresponding regression values, for each, are stored along with the user's factor and facet values within the *page package*. The ManAc then triggers the calculation process that uses those values with the above formula to produce a prediction value for each behaviour pattern.

Dynamic personality aspects

The dynamic aspects of personality were gathered using the self-assessed questionnaire presented in Appendix A.2, the tabulated results are shown in Appendix A.3. Figure 4.4 shows that this implementation is to focus on six of the dynamic aspects: values, attitude, ability, preferences, short goals and habits. The questionnaire assessed the first five dynamic aspects and the sixth, habits, was to be extracted from the observation study by observing small units of repeated behaviour for each participant. The analysis of the observation data revealed insufficient repeating behaviours to form Habits. For example, those participants that hand coded the first program copied and pasted common elements into the second, a different behaviour pattern. Discussion of why this may have been so and the impact on the overall system is presented at the end of this chapter in 5.3.

Three of the aspects, ability, value and goal are represented on a scale of one to one hundred and are linked to a single behaviour, `action:focus` pair. The higher the value associated with the behaviour the stronger this aspect is. Attitude is represented simply by three values: positive (3); indifferent (2); and negative (1). Preferences are constructed from a descending list of behaviours, with position one being occupied by the behaviour that is most preferred from the set, within a given context. Appendix A.4 lists those questions from the self-assessed questionnaire that were used to populate each of the dynamic aspects and the method used to convert that data to data within the PR profile.

Unlike the static aspects the dynamic aspects do not directly predict behaviour. Originally the dynamic aspects were intended to contribute to the prediction by adding further weight.

However with discovering that this was not feasible, as discussed at the beginning of this section (5.2.3) the question was how else could the dynamic aspects add to the process personalisation? The answer lay in the fact the data they contained could be directly used to provide support. For example: if a user had a *value* element such as `write:comments`=20, meaning they valued the action of writing comments as low, (given a scale of 1-100). That in itself could be used to directly generate some form of support. In this case by making suggestions to write more comments or indicating that this was best practice in programming. Therefore, by generating a series of if… then statements the dynamic aspects could contribute directly to the personalisation process.

5.2.4 Developing the support objects

The concept behind the support objects (SOs) is to provide a logical set of support *hints* for each type of *instruction* that is required to complete a programming task. This concept was derived from ideas around experimental learning and reflective practice (Kolb 1984, Moon 2004). Rather than give the students direct instruction or correction the system aims to provide a *hint* that prompts the student to consider their code and work out for themselves what a better action might be. The *hints*, particularly those attached to the dynamic aspects, also provided positive reinforcement when the student was predicted to apply best practice.

When the concept of SOs was being designed a number standards were consider for providing a framework within which to build the process. Learning objects (McGreal 2004) are packages of learning content that are described by metadata, data about data (Friesen et al. 2004) to enable their sharing and reuse. A number of standards are available for constructing this metadata such as the IEEE Learning Object Metadata (LOM) (IEEE 2004) and ADL's (Advanced Distributed Learning) SCORM (Sharable Content Object Reference Model) standard (ADL 2004a; ADL 2004b). Of particular interest was the work of Rumetshofer and Wöß (2005) who suggest that the IEEE LOM (IEEE 2004) structure should be extended to include more common factors associated with adaptive systems. Such as a section labelled psychological, containing values such as cognitive styles, learning strategy, learning modality and skills. While this idea of adapting LOM is moving in a direction that could of benefit to this research. At the time of the prototype development the SOs that were being developed were simple support statements and did not require the level of description that is presented in this standard.

Another standard that was considered at this time was the use of XML and the development of a DTD (Document Type Definition) to fully describe the different aspects of an SO package. The benefits of doing this would have been the generation of a template that could be utilised repeatedly and parsed by the system relatively quickly. However as the development of the SOs was very much an experimental process, a DTD would have required considerable time investment to keep up to date, as well as redoing the mark-up with the SOs themselves. There was also the concern that investing time in the development of more fully formed support objects would steer the research away from its core concern that of personalising a service and implementing a personality based profile.

The SOs have been developed for the purpose of applying personalisation and not in themselves a major contribution of this research. For this reason a simple labelling mechanism was used in order to enable the system to extract the relevant data from a flat file and generate Java objects. This enabled changes to be made quickly to the contents of the files and the system would automatically generate new objects on being rerun. These labels are as follows:

`<INSTRUCTION>` - The name of the *instruction* this SO supports which is used to link this SO to the instructions retrieved from the task set.

`<IDEAL>` – This contains a special *behaviour pattern* that represents the ideal behaviour a student can perform to complete an instruction. This *behaviour pattern* is used for comparison with the predicted *behaviour pattern*. It is also used as the edited *behaviour pattern* when the `<STRICT>` tag is set to false. The strict tag affects two key processes within the SO. The first is which *behaviour pattern* is edited, either the predicted behaviour or the ideal behaviour. The second is which monitoring process is applied to the edited *behaviour pattern*. This is either an ordered process, where the behaviours are checked off in the same order that they appear in the *behaviour pattern*, or an unordered process, where they are checked off in the order the user performs them. The reason for using two methods is that some *tasks* contain *instructions* where the order of coding is irrelevant to completing the *instruction*.

`<TRIGGERS>` – This indicates the specific *behaviours* that need to be identified for the SO to be activated.

`<HINTS>` – This section states the rules used with the static aspects of the PR profile. A list of the rules that can be used here is given in Appendix C.1.

`<DYNAMIC>` – This section states the rules used with the dynamic aspects of the PR profile. A list of the rules that can be used here is given in Appendix C.2.

`<LEXICON>` - The lexicon section enables the SO to recognise when a *behaviour* can be legitimately replaced by other *behaviours* that will also complete the *instruction*. For example, all floating point numbers, `type:float_dec`, are represented as `type:double_dec` within the SO `use:primitive_types`. In this case either a float or a double will achieve the *instruction* of using primitive types within the specific task. So being able to accept both is an important function of the SO. To write individual rules for all variations would be a long and incomplete task. The lexicon simply enables the *behaviour* within a rule to be replaced by an equally variable behaviour and the rule activated.

`<OPTIONAL>` - This is similar to the `<LEXICON>` tag above by enabling a single rule to be adapted to equivalent but different *behaviour pattern*. For example, the parser will identify all the steps in the declaration and instantiation of an array. However, considering the `use:primitive_types` SO a declaration of five ints can also be done as an array of ints. So if the user is monitored as having created an array this is translated, by this SO, into five int declarations so as to match the *behaviour patterns* that it uses within the *hint* rules.

`<END>` - Signifies the end of the SO file.

For each *instruction* there is an associated SO which is used to edit the predicted *behaviour pattern* with *hints*. A full list of *hints* is shown in Appendix C.4. The editing is controlled by a set of *hint* rules for each SO (see Appendix C.1 and C.2). The predicted and the ideal *behaviour patterns* are compared using these rules to determine when and where a *hint* is added.

5.3 Issues and impact

During the development of the PReSS prototype a number of issues were identified with the implementation of the personality profile and how it was linked to behaviours and tasks. The first occurred during the process of linking the behaviour patterns to the static aspects of personality.

During the analysis of the behaviour data from the observation study (5.1) a great deal of detail was produced. This resulted in many long and complex behaviour patterns and sub behaviour patterns. When it came to apply the statistical link to personality, in 5.2.3, this detail resulted in no statistical significance being produced when correlated with the factors or facets of the FFM (McCrae and Costa 2003). To solve this the behaviour patterns of the participant group were examined as a whole for similarities in response to specific aspects of the programming tasks. The resulting groups used high-level behaviour patterns rather than the detailed ones identified in the study.

Although the idea behind the static aspects (represented by the FFM) was to look for commonalities across behaviour, it was assumed that the behaviours would be described at a much lower level. However this was not the case. This is most likely linked to the low participant numbers in the observation study. With fewer participants there is less chance that complex behaviour patterns would reoccur, so less complex higher-level patterns were used. Given this it could be reasonably expected that with a higher number of users the group similarities would appear at higher granularities of behaviour. Nevertheless, this does demonstrate that by adapting the level of granularity the system can respond to the low numbers expected at the initiation of the system. This is something that was not considered at the start of this development, but may prove beneficial.

Also during the process of linking the behaviours to personality, a more critical change to the model of personality was required. As discussed in 5.2.3 the initial concept of the dynamic aspects adding weight to the static aspects, as shown in Figure 3.7, proved to be ineffective and a new approach was required. This resulted in the dynamic and static aspects working in parallel producing two types of prediction and providing two sets of support. This change in the model is represented in Figure 5.4 that shows the two result sets feeding into the personality effects. Again this required change may be the result of a low number of participants and is also likely linked to the change in granularity of the behaviours, as described above. It could be speculated that with a larger number of users the number of strong predictions for each user for a given task will be greater. Enabling the dynamic aspects to contribute in the prediction process in some way, in order to narrow that set down. However it is suspected that this would not be the case and that these two aspects may actually complement each other more successfully as parallel processes rather than serial.

Both the above problems impacted on the development of the SOs resulting in a reduction in the number of support areas from the indicated twelve in Table 5.2 to nine. This in turn will impact on the scope of the evaluation study and its ability to demonstrate a range of personalisation. The dynamic aspect habits was not able to be implemented within any of the profiles as already discussed in 5.2.3. During the observation study none of the participants exhibited repeating behaviours. This again could be due to the granularity of the behaviour patterns, although several levels were tried, or the small number of tasks that were observed for each user. It could be speculated that with a larger number of interactions these repeating behaviour patterns will be identified. However this does suggest that this particular element of the dynamic aspects could be susceptible to the cold start problem experienced by profiling systems (discussed in 2.1.7). Figure 5.4 has been adjusted to reflect this change in implementation.

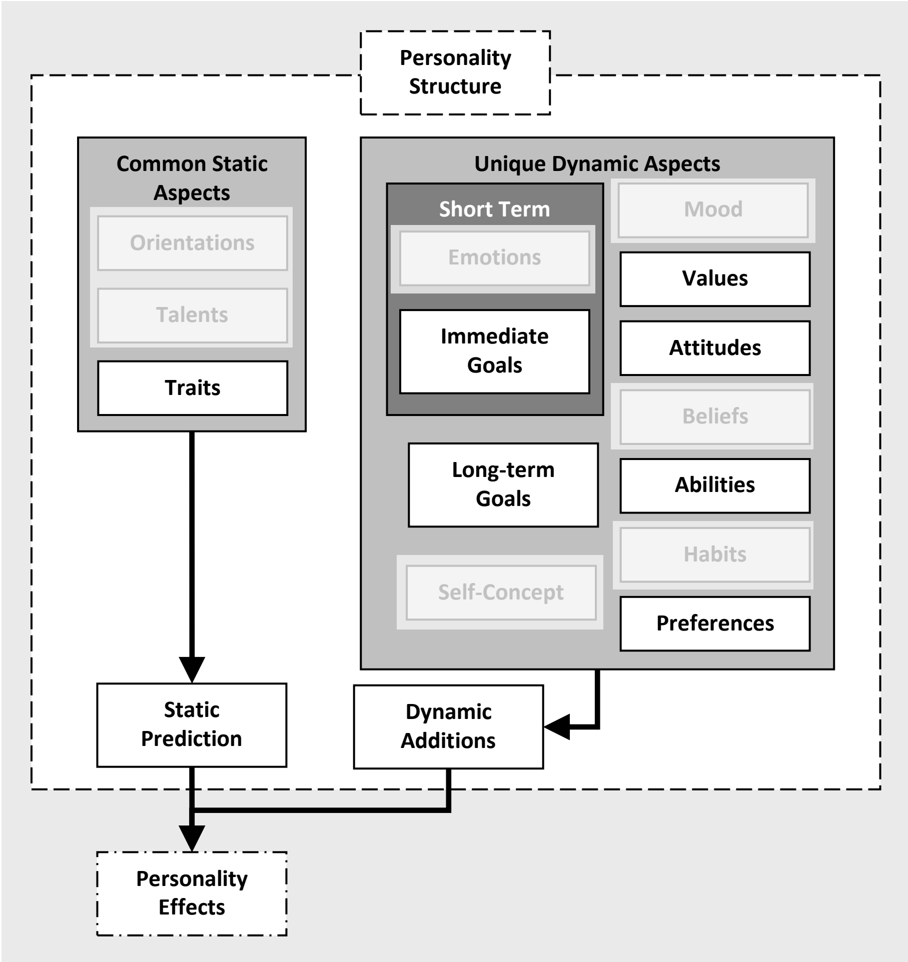


Figure 5.4 Adjustments to the implementation of the personality model

Already discussed in 5.1 is the issue around the self-assessed survey used to gather the dynamic aspects of personality. The transfer of this survey data to scale data within the profiling system may limit the effectiveness of the dynamic aspects and should be taken into consideration during the evaluation stage. However these issues do highlight the difficulties experienced with implementing the PR profile with a software system. The complexity of the behaviours and low number of links to personality within the relevant support areas start to indicate that this concept is not as technically feasible as first thought. This may be due to the approach taken or the range of new concepts being implemented or simply down to the fact that a PR profile is not a feasible approach to profiling. The following chapter presents' the evaluation of the PReSS system and evaluates its ability to generate personalised support.

Chapter 6

Case study: PReSS prototype evaluation

This chapter presents the second half of the case study used to assess the feasibility of developing a personalisation system using the models developed in 3.3. During this stage of the case study the PReSS prototype was used and evaluated by participants within a simulated lab environment similar to that used in the observation study (5.1). The main difference being that the participants used the PReSS prototype rather than a text editor and system browser. Those users that participated in the observation study were required to complete an in-session and post-session evaluation surveys, while those who were new to the research also completed the FFM personality instrument prior to the session. An invitation to participate was sent to all the students enrolled at the Centre for Internet Computing. This included second and third year undergraduate students as well as postgraduates.

Section 6.1 presents the methods used to evaluate the areas identified in 4.4. Section 6.2 presents the results of the evaluation study and some initial discussion. The final section (6.3) of this chapter revisits issues identified during the implementation stage (5.3) and ascertains if and how they have impacted on the evaluation study.

6.1 Evaluation methods

For the evaluation of the PReSS prototype an observation study was used, similar in format to that presented in 5.1. Participants were asked to undertake a set of programming tasks within a simulated lab scenario, using the PReSS prototype as their browser and code editor. They had access to a lab assistant (the researcher), the Java API and any books or code samples they saw fit to bring with them. As with the first study, recording software was used to capture the participant's interactions with the computer, as well as audio and video capture

For participants the evaluation process consisted of the following steps:

1. Complete the online third party personality inventory (Johnson 2000; Johnson 2005) prior to the evaluation study, if they had not participated in the previous observation study.

2. Participate in the simulated lab exercise using the PReSS prototype.

3. Evaluate the support outputted by the system using the quick-sheet evaluation as shown in Appendix B.1.

4. Complete the online evaluation questionnaire for the PReSS prototype after the observation study shown in Appendix B.4.

At the start of the observation study participants were asked to sign a consent form and read the introductory text about the PReSS prototype's functionality (Appendix B.1). Due to the extra time required for familiarising participants with the prototype and the evaluation of support throughout the study, only two tasks were presented to the participant rather than the four presented in observation study one 5.1.

Task one of this set (Appendix B.2) is identical in its solution to task one presented in the first observation study, with the question being phrased differently. The aim of this task was to observe if participants from the first study would use the same behaviour patterns they exhibited previously to develop the solution to the task.

Task two was designed to only retain similarities in program concepts. In study one participants were asked to output the times table from a user entered value, in a specific format. Whereas in study two they are asked to generate a count for each of the different characters within a user given sentence and output the results in a given format. Algorithmically both require a loop, user input and formatted output, but require assembling in a different way. The aim here was to provide the system with behaviour patterns it had previously encountered, but applied within a different context.

6.1.1 The predictive ability

These measures evaluate the degree to which the PR profile can predict behaviour via the static aspect and all contribute to the technical feasibility of using a PR profile to generate personalisation.

Prediction to the same task for the same user

This measure examines whether or not the predicted behaviours for task one were the same as the observed behaviours for task one for each participant.

The predicted behaviour was extracted from the file *prediction.txt* that is generated by the system for each user. The observed behaviour was extracted from the observation data for the evaluation study, using the same process as that described in study one in 5.1. The observed behaviour was selected by identifying the point at which the participant was performing the instruction relevant to the prediction. If they exhibited the same behaviour as the prediction this was coded as a positive, else it was coded as a negative. This was done for all the predicted behaviours regardless of whether the support process was activated or the participant received any support associated with the behaviour. Where the participant did not attempt a task or instruction the prediction was removed from this measure.

The number of correct and incorrect predictions for each task were collated and totalled. This is then presented as a percentage of the total possible predictions, minus those associated with none-attempted instructions.

This measure intends to evaluate the effectiveness of the prediction process from the static aspects of personality given the same user and the same context. If the PR fails to predict for the same users performing the same tasks, it has failed in its primary function as a profiling mechanism.

Prediction to a different task for the same user

This measure examines whether or not the predicted behaviours for task two were the same as the observed behaviours for task two

The process of comparison here is the same as described for the previous measure. Task two bases its predictions on behaviours observed in study one, in a conceptually similar task, but are applied here in study two within a different programming context.

This measure intends to assess how effective the static aspects are at applying prediction from one context to another. This process is reliant on the idea that a different context is determined by the words used to describe a task and not the underlying concepts.

Prediction to a different user for both tasks

This measure examines whether or not the predicted behaviours of new participants were the same as the observed behaviours of the new participants.

The process of comparison here is the same as described for the previous two measures. This measure was only used with participants who had not participated in study one. New participants' profile data only consists of the static aspects and not the dynamic aspects. As discussed in 4.2 and 4.3 the static aspects of the profile are intended to represent group similarities and as such act in similar ways to that of collaborative profiling.

This measure intends to assess how effective the static aspects are at applying prediction from a group of participants to individuals not within that group.

6.1.2 Personalisation

These measures are examining the degree to which personalisation can be determined within the case study by analysing users ratings across four qualities used to evaluate the context of the support produced. The degree of perceived personalisation is then compared to the predictive ability of the system to determine whether or not there is any relationship. These measures will contribute towards considering the overall operational feasibility of the use of a PR profile for personalisation.

Perceived personalisation by users

To ascertain the perceived personalisation of the support the participants' data was gathered during the quick-sheet evaluation (Appendix B.1) after each item of support was presented. Participants were required to rate the support on a scale of one to five, one being strongly disagree and five strongly agree. The ratings were applied to each of four qualities: understandable, relevant, and helpful and timeliness. The rationale of using these qualities is discussed previously in 4.4.

An aggregated mean was calculated for each participant across the four qualities. This was calculated across all the hints generated for:

1. The dynamic and static aspects of the personality model.
2. Each support object.
3. Each task set.

These data sets are intended to determine if there is any significant difference perceived by participants in the personalisation generated for each group. A significant difference in the first set, dynamic and static, may suggest that there is a difference in the effectiveness of each aspect in the generation of personalisation. As the support objects used a number of different methods to link the predicted behaviour to the support, a significant difference here may indicate the relative effectiveness of the methods used. The final set may indicate how effective it is to use behaviour data generated in one context, within another.

It is expected that there will be no significant differences between the dynamic and static aspects. Both should deliver a similar degree of personalisation. There is expected to be some discernable difference between task one and task two as the predictions for task one deliver more specific support than those for task two. The more generic nature of the support for task two may make it more difficult for participants to associate it with their current context. It is also expected that there will be differences across the support objects. Again particularly those related to task two, but also because the methods used to generate the support differ in some of the objects which may impact on their ability to be presented accurately in relation to the users context.

This approach does not take into account other factors that may have influenced the participants' ratings, such as the quality of the support or the difficulty of the task. The qualities used within these instruments have not been validated as directly relating to perceived personalisation. While there are assessment instruments available for the evaluation of various aspects of a software application, such as usability and interface personalisation (Dı´az 2008). None were identified that specify examined the perception of personalisation from a support context.

This measure intends to determine the overall perceived personalisation of using a PR profile.

Is there any relationship between the perceived personalisation of each support object and its predictive ability?

This measure brings together two values already generated by the evaluation process, that of predictive ability and perceived personalisation. Focusing on the data generated from the quick-sheet evaluation, as described previously, the perceived personalisation scores were compared to whether or not evaluated support was generated from correct behaviour prediction, as defined in 6.1.1. The aim here is to determine whether or not the perceived personalisation was generated from the prediction made by the PR profile, and thus the effectiveness of the PR profile to generate personalisation.

As the behaviour patterns determine at which point the support will be output within the context of what the user is doing. If the user perceives this as appropriate, measured by the four qualities (understandable, relevant, helpful, timeliness), in the context of what they are currently doing. It can be assumed to some degree that the prediction process has contributed to the support process, by positioning that support within the context of the user's activities. Since the prediction cannot be generated without the PR profile, it can be inferred that the PR profile has generated the correct prediction for the user's context, and thus generated personalisation.

The logic for relationship between these two values is as follows:

- If the prediction accuracy is low and the perceived personalisation is high it can be inferred that the system has failed to personalise support via the prediction of behaviour from personality and participants gained benefit only from the nature of support itself.
- If the prediction accuracy is high and the perceived personalisation is low it can be inferred that while the profiling mechanism functions the participants perceived no benefit from the personalisation process, and personalisation has limited or no positive effect within this context.
- If the prediction accuracy is high and the perceived personalisation is high then it can be inferred that the PR profile has to some extent provided personalised support via predicting behaviour.
- If prediction accuracy is low and the perceived personalisation is low then it can be inferred that the lack of PR generated predictions has had a negative effect on the perceived personalisation.

The limitations stated in the measures for perceived personalisation and predictive ability will also be valid here. This is not intended to be a determinate measure, but an indication towards the potential feasibility of personalisation via PR profile.

6.1.3 The process of support

These measures are examining the general functionality of the system in order to identify if there were any significant failings or difficulties experienced by the users that may have impacted on the evaluation process and the measures described in 6.1.1 and 6.1.2.

Level of successful generated support

This measure examines the number of successful output support hints across all participants for each of the two tasks.

This data was generated from the *prediction.txt* file generated for each user. The focus here is on determining success or failure from system errors not prediction errors. For this reason support hints that were associated with tasks not attempted by participants were removed. Along with any support hints that were not triggered because of changes in participant behaviours (prediction error). This measure will contribute to evaluating the technical feasibility of the concept of personalisation via a PR profile.

This measure intends to indicate the reliability of the system and the various processes that instigate the output of a support hint and contributes to evaluating the technical feasibility.

User perceptions on the quality of support

This measure examines the participants' perceptions on the quality of the support hints they received during the evaluation study.

Along with the individual evaluation of the support received within the study the participants were asked to evaluate their whole experience of personalised support on the post-evaluation survey (Appendix B.4.). These questions are aimed at generating a holistic view of their experiences and are focused on assessing the level of support, the relevance of support, and the helpfulness and timeliness of the support. The mean was calculated across the qualities and across the participants. This measure will contribute to the operational feasibility of using a PR profile to generate personalisation.

This measure intends to indicate the general quality of the support objects and assess if any significant issues generated by them.

User perceptions on ease of use and the concept of personalised support

This measure examines the participants' perceptions on the general quality of the prototype system and user interface.

This is done via a number of questions in the post-evaluation questionnaire (Appendix B.4). Two of which assess the participants' perception on the layout (interface) and functionality of the prototype individually. Another six evaluate the surveys used to capture the personality data to generate the profile. The final six evaluate the participants' views on the concept of personalised support generally. Participant responses were given on a five-point scale from one, very poor, to five very good. This data was collated and the mean and mode calculated for each question across all participants and for each participant across the two questions.

The first eight questions intend to indicate the general quality of the prototype environment and assess if there were any significant issues for particular participants. The final six intend to indicate the general views of the participants in relation to personalisation of support. Both of these will contribute towards the operational feasibility of the concept of personalisation.

6.2 Evaluation results

Eight participants took part in the evaluation study. Seven of those had previously participated in the observation study (5.1) and had full profiles (using both dynamic and static aspects). The single participant who was only involved in the evaluation phase had a partial profile consisting only of the static aspects. This new participant was required to complete the personality survey from study one prior to evaluation. All participants completed task one of the programming task set (Appendix B.2). Task two was attempted by five of the participants, of which only one completed. Three of the participants felt that task two was beyond their programming abilities, and preferred not to attempt it.

The demographics for this group were generated from the post-questionnaire data, tabulated in Appendix B.5. This group can be described as predominantly male undergraduates aged twenty-five and under with a mixed educational background. The majority indicated that they have had at least six years programming experience, and at least two years Java experience. Their last programming experience was less than six months from the study, and it was generally not using the Java programming language. All participants that took part in the evaluation phase of this research were known to the researcher from the support of their programming labs or group tutorials.

During the evaluation participants were also given the opportunity to make comments on the support they received as shown in Appendix B.3 and B.5. While these are not analysed here directly they do provide anecdotal evidence to their perception of personalised support and are drawn later in the discussion of these results in Chapter 7.

The following sections present the results and initial discussion in relation to the methods described in 6.1. Throughout this presentation examples will be given, drawn from the qualitative comments made by the participants, which have particular relevance to the context of the tabulated results.

6.2.1 The predictive ability

The first area of evaluation is the predictive ability of the PReSS prototype to determine how effective the PR profile is at predicting behaviour. Four measures were calculated here, prediction for the same user in a similar task, prediction for the same user in a dissimilar task and prediction to a new user for both a similar and dissimilar task. The methods used to generate these results are described in 6.1.1.

To determine whether or not the observed behaviour was the same as the predicted behaviour, only behaviours exhibited up to the point where the predicted behaviour could, given any system errors, output support were examined. If the behaviour was observed during this period then it was determined that the prediction was correct. After this point the prediction was deemed incorrect.

Prediction for the same user across both tasks

Data from all eight participants was used to generate the values given in Table 6.1. All participants attempted task one whereas only five attempted task two with only one completing it. This means that the results for task two are drawn from a smaller data set than that of task one, making them not directly comparable. The number of system errors (discussed further in 6.2.3) generated during task two was higher than those generated during task one, further reducing the number of potential predictions. These errors are primarily attributed to the difficulties experienced during the implementation process and the means of applying behaviours generated in one context to an alternative context.

	Task 1	Task 2	Overall
Total Predictions	35	21	56
Correct Predictions	65.71%	47.62%	58.93%

Table 6.1 Predictions matching observed behaviour for each task

Table 6.1 shows that a number of predictions are performing as expected, although not to a high degree. Given the reduction in potential predictions for task two, Table 6.1 does suggest that the predictability of the system is less effective when applying the process across tasks. This reflects the difficulties experienced by profiling systems in general in applying profiles across domains (2.1.7), although this case study is on a much smaller scale. Other factors may have contributed to these relatively low scores such as the high level of granularity regarding the behaviour patterns, as discussed in 5.3, and the general difficulties with implementing the process.

However some prediction has occurred across both tasks suggesting that there is some merit in the concept if not necessarily in the current method of implementation. From the perspective of feasibility it is difficult at this point to determine if the level of results warrant the overheads.

Prediction to a different user across both tasks

This result was only obtained from a single new user. The evaluation survey was undertaken during the second semester of the university year and very few students were willing to participate. Of the three new participants acquired for the evaluation study only one was able to attend the evaluation itself. The others completed the personality inventory and then withdrew from the research. There was some doubt as to include these results here; however it was decided that in keeping with the holistic perspective of the case study they may have something to contribute to the discussion. More unfortunately was the fact this participant was one of the students who did not attempt task two, so only data for task one can be considered.

	Task 1	Task 2
Total Predictions	5	-
Correct Predictions	60%	-

Table 6.2 Predictions for a new user matching observed behaviour for each task

If the result in Table 6.2 is compared to the task one in Table 6.1 there is only a 5% difference between the two. This suggest that while not a high percentage of predictions, some prediction is taking place and, in this case, is almost as effective with a new user. Examining both tables it could be said that the system is relatively successful at predicting to new user but not at predicting to new tasks. However the low number of participants in both groups should be emphasised and these results only suggest some potential but by no means indicate that this process is an effective means of generating prediction.

6.2.2 Personalisation

This section examines the data gathered by the quick-sheet (Appendix B.1) evaluation produced during the evaluation study. This data captures the perceived personalisation of the participants for each of the support hints they experienced. They evaluated each hint as it was presented to them across four qualities: understandable, relevant, and helpful and timeliness. The rationale of using these qualities is discussed previously in 4.4.

Perceived personalisation by users

The perceived personalisation data was examined for each user across, personality aspects; tasks one and two; and the support objects for task one. Table 6.3 shows the results across personality aspects and tasks. Comparing the two personality aspects there is hardly any difference between the perceived personalisation. Similarly no significant differences are perceived between the two tasks. This is also emphasised when examining the scores for each user across the four areas. This is surprising for the personality aspects as the method of generating support was distinct for each as discussed in 5.2.3, suggesting that there is merit in both approaches to some degree. It is not surprising across the two tasks as there were no differences between the methods used, although there was a different balance of static to dynamic methods.

Quality	Static Aspects	Dynamic Aspect	Task 1	Task 2
PA02	3.50	-	3.50	-
PC01	3.85	3.93	3.93	-
PC02	2.67	2.58	2.63	-
PB01	2.55	2.83	2.46	2.00
PB02	3.13	3.75	3.21	-
PA01	4.25	4.33	4.25	3.85
PC04	4.69	4.50	4.67	4.56
PD01	4.63	-	4.63	-
Across Participants	3.66	3.65	3.66	3.47

Table 6.3 Perceived personalisation for each personality aspect and each task

Table 6.4 show the perceived personalisation for each participant across all the support objects that generated support in task one. This shows a larger degree of variance for perceived personalisation suggesting that the nature of the support had a greater impact than either the personality aspects or the tasks directly. Comparing scores across the support objects for each participant they generally stay within a close range of values, except for PA01. This may suggest that this participant perceived very little difference between the support they received in terms of its personalisation or they tended towards allocating consistent values. Either way this suggests that the personalisation for each participant was perceived as fairly uniformed.

Task	1	1	1	1	2	2	2
Support Object	**CA**	**MO**	**PT**	**WC**	**OV**	**WC**	**WP**
PA02	-	5.00	4.50	-	-	-	-
PC01	4.50	5.00	3.50	4.00	-	-	-
PC02	2.38	-	2.42	3.75	-	-	-
PB01	3.25	3.25	2.50	3.25	-	-	2.00
PB02	3.00	2.75	3.25	3.75	-	-	-
PA01	4.50	3.25	-	4.63	1.00	4.63	4.50
PC04	4.00	5.00	-	5.00	4.75	3.5	5.00
PD01	-	4.75	4.50	4.75	-	-	-
Across Participants	3.61	4.14	3.45	4.16	2.88	4.07	3.83

Table 6.4 Perceived personalisation for each SO

Participant PA01 scored one particular support hint distinctly low compared to the rest of their scores. On reviewing the comments made on the quick-sheet scores and later on the post-evaluation survey, participant indicated that they didn't *notice it at the time. When I did I couldn't understand what it meant in my context* (Appendix B.3 and B.5 for participant comments). This is an interesting point in case as it highlights when personalised support can be ineffective, even detrimental. A number of comments from participants highlighted their concern with noticing or missing the support notification. Some suggesting that a popup system would be preferable. This maybe something that needs to be considered in future developments, but is considered here as something that could impact on the perceived benefits of personalisation in the context of this research.

Is there any relationship between the perceived personalisation of each support object and its predictive ability?

Here the perceived personalisation values are compared to the percentage of correct predictions for each task. The rationale for this measure is discussed in 6.1.2 and intends to demonstrate that there is some link between personalisation and the PR profile. Only those predictions that apply to the support scored by participants were used in this case. Table 6.5 and Table 6.6 show the number of predictions and the number of correct predictions used to calculate the percentage.

Participants	PA02	PC01	PC02	PB01	PB02	PA01	PC04	PD01
Perceived personalisation	4.75	4.25	2.85	3.06	3.19	4.13	4.67	4.67
Num. Predictions	2	4	3	4	3	3	3	3
Num. Correct Predictions	2	4	1	2	0	3	2	2
Percentage of correct predictions	100%	100%	33%	50%	0%	100%	67%	67%

Table 6.5 Perceived personalisation score compared to predictions for task 1

Participants	PA02	PC01	PC02	PB01	PB02	PA01	PC04	PD01
Perceived personalisation	-	-	-	2.00	-	3.38	4.42	-
Num. Predictions	-	-	-	1	-	3	2	-
Num. Correct Predictions	-	-	-	0	-	1	1	-
Percentage of correct predictions	-	-	-	0%	-	33%	50%	-

Table 6.6 Perceived personalisation score compared to predictions for task 2

Following the logic in 6.1.2 five out of eight participants demonstrate the high-to-high relationship between personalisation and predictions for task one. This suggests that there is some relationship and that the PR profile is generating personalisation. This is coupled with the inverse for the remaining three participants, a low-to-low relationship, with the three lowest scores for perceived personalisation match the number of least successful predictions.

As expected, considering the difficulties experienced with generating support for task two, Table 6.6 presents weaker evidence of this relationship. However there is still a trend of low-to-low and high-to-high if less pronounced.

This measure has provided initial evidence that the PR profile generates personalised to some degree support when the prediction process is successful. However there are a number of unsuccessful prediction events signifying a degree of inconsistency with the method as a whole. This may just be mirroring the cold start problem experienced by recommenders in general (2.1.7). However given the low number of participants and the narrow context for the support generation, it could also signify some more fundamental difficulty with the concept or the way in which it has been implemented within this case study. The following section examines the system to try to identify any serious flaws in the support process.

6.2.3 The support process

This section seeks to determine the general effectiveness of the PReSS prototype and the processes used to generate prediction and personalisation. As previously mentioned in 6.2.1 there were a number of technical issues with the system that may have impacted on the generated results. The first part of this section intends to determine the nature of those errors and where they may have potential impact. The second part of this section examines the overall perspective of the participants on the key elements of the system and the concept of personalised support.

Level of successful generated support

Table 6.7 shows the number of hints outputted successfully and the number that failed due to a system error. From the *prediction.txt* file, generated for each participant, a total of one hundred and fourteen were identified across the two tasks. From this the hints associated with participants that did not attempt task two were removed along with hints associated with behaviours that never occurred. This left fifty-three and twenty-two potential hints for task one and two respectively. These hints were removed from the count as the focus here is on failure due to system error not due to error in predicting behaviour.

All the hints were removed from this count for the SOs, `use: global variables`. This was because no participant attempted to use global variables in this study. During study one it was observed that the creation of global variables created a number of difficulties for the participants. This may have been the reason none attempted this during study two. However, it was because of these difficulties that this object was selected for development.

	Total number of potential hints	After none-attempting participants removed	After Interaction changes	Successfully outputted	Failed due to system error	Percentage of successes
Task 1	64	64	53	42	11	79.25%
Task 2	50	32	22	12	10	54.55%
Overall	114	96	75	54	21	72.00%

Table 6.7 Percentage of successful hints across tasks

The eleven failed hints on task one, were due in some cases to a mismatch between the labels used to describe behaviour by the parser, and in the support objects (SOs) lexicon. Other failures were due to the limitations of the Java parser. There was also an issue with the SO `use: primitive types` which contained a logical error that checked off too many behaviours at once and effectively jumped over the output process.

Task two had more problems than task one. Table 6.7 shows that just over half the hints were successfully presented. Most of these issues were again due to discrepancies between the parser labels and the SOs lexicon. This was more pronounced in task two as the SOs designed for this task were more general in nature as the behaviour data they used was generated from another task. This resulted in greater reliance on the parser descriptions and lexicon being matched. One particular triggering method also caused of some of the issues. The trigger behaviour would be identified and then the participant would perform the remaining behaviours in a different sequence to that defined in the SO. This was the case with the SO `output: message`.

The number of technical failures here reflects both on the complexity of the process and the breadth of behaviour patterns that can be used to complete even the smallest element of a task.

The use of the lexicon was intended to overcome behaviour variance, but clear was not completely effective.

From the perspective of the participants the system was relatively reliable and there were no serious technical problems observed for the participants who would have been unaware of the reduction in support. This may have been a bonus to some extent in that given the number of hints and the average time participants spent on task one, if they had all been output this would have meant approximately one every four minutes.

The technical difficulties with task two may have contributed to its performance and why a number of participants failed to complete it. This will have impacted on the perceived personalisation data for this task, as there were a reduced number of hints to evaluate. However this did not impact on the prediction data, as this is not reliant on the system's ability to output hints.

User perceptions on the quality of support and ease of use

The overall quality of support was evaluated by participants responding to Q3 to Q6 in Table 6.8. The scores for each of the four questions do not appear to signify any particular difficulties in the support statements themselves. Nor do they suggest that the support was particularly strong, suggesting a very mediocre response to them, which is contrary to some of the comments made by participants as shown in Appendix B.5.

Q1 and Q2 in Table 6.8 present the participants response to the quality of the PReSS prototype in terms of the editors layout and functionality. There were concerns that the limited functionality of the system and unfamiliarity would cause difficulty with programmers used to a preferred system. However given the result shown there is no significant negative response. There was also no comments regard this made by students on the quick-sheets Appendix B.3 or the post evaluation survey Appendix B.4.

	Q1. The Editor's Layout	Q2. The Editor's Functionality	Q3. The Level of Support Provided	Q4. The Promptness of the Support Statements	Q5. The Relevance of the Support Statements	Q6. The Helpfulness of the Support Statements
Mode	4.00	4.00	4.00	4.00	4.00	3.00
Mean	4.25	3.63	3.75	3.38	3.50	3.50

Table 6.8 Evaluation of the PReSS prototype and Support

(High values = positive)

	Q1. Personality survey: ease of use	Q2. Personality survey: shortness	Q3. Personality survey: acceptability of length	Q4: Java survey: ease of use	Q5. Java survey: shortness	Q6. Java survey: acceptability of length
Mode	2.00	3.00	1.00	1.00	3.00	1.00
Mean	2.00	2.75	1.38	2.00	2.88	1.38

Table 6.9 Evaluation of the survey instruments used

(Low values = positive)

Table 6.9 shows that participants found both surveys relatively easy to use. The scorings for the length of the surveys indicates that they were not perceived as excessively long from the perspective of completion. There appear to be no particular issues arising from this aspect of the evaluation, which does not concur with issues identified with other research (2.1.7).

User perceptions of concept of personalisation

Participants were asked to consider the concept of a personalised support system such as PReSS in relation to three scenarios (Q1-Q3 in Table 6.10) and reflecting back to their experiences of learning programming (Q4-Q6 in Table 6.10). These figures demonstrate a generally high response to the idea of personalised support. Also from some of the comments made by participants (Appendix B.4), they felt it was particularly appropriate for those at the beginning of their learning journey. There was some indication that it would be less appropriate for more experienced programmers.

	Q1. Scenario: practicing outside of labs	Q2. Scenario: programming after prolonged absence	Q3. Scenario: learning new language work	Q4. Hindsight: easier to practise	Q5. Hindsight: more motivated	Q6. Hindsight: useful in labs
Mode	5.00	5.00	5.00	5.00	5.00	5.00
Mean	4.25	4.38	4.50	4.00	4.00	4.13

Table 6.10 Evaluation of the concept of a personalised support system

(High values = positive)

6.3 Evaluation of the case study method

Using a case study approach this research has presented a holistic view of the process of implementing and evaluating of the two models (Figure 3.6 and Figure 3.7) within a specific domain. The first phase of the case study the application to a real scenario demonstrated a number of issues with the processes of implementation as discussed in 5.3. The application process also highlighted an incorrect assumption made within the model about the relationship between the dynamic and static aspects, which was adjusted and implemented to some degree of success.

The observation study (5.1) provided an invaluable opportunity to examine and analyse the relationship between task and behaviour and how this was linked to personality. However much of this data was discarded during the implementation process as it was irrelevant to the support areas selected or was too detailed to form a correlation with personality. The small number of participants in this study also contributed to the difficulty with finding correlations.

The second phase of the case study the evaluation of the implementation within a simulated lab environment was less successful. The low number of participants and in particular the single new participant resulted in small data sets that were difficult to draw any strong conclusions from. Much of the qualitative data was generated with the observations and participant comments that added very little to evaluating the feasibility of using a PR profile to personalise support. This was due to the post-evaluation questions being more focused on the effectiveness of the support hints. As already discussed in 4.4 asking participants to evaluate the level of personalisation they experienced is no easy task without some basis for comparison. The questions were intended to capture contextual comments but generated responses focused more on the content of the support, rather than the relevance of the support to the current task. Although this data did provide a deeper understanding as to why participants scored the support hints the way they did.

Chapter 7
Reflection, conclusions and future work

This, the final chapter in this thesis brings together the findings of this research and reflects on them in the context of the research aims and research question. Throughout this thesis a number of points have been raised in relation to various aspects of this experimental development. The models developed in 3.3 were discussed in the context of a theoretical application to an educational scenario, highlighting a number of speculative benefits in 4.3. The case study developed to evaluate particular aspects of the models, as defined in 4.4, brought out a number of issues in regard to their practical implementation, discussed in 5.3. The evaluation of the PReSS prototype in Chapter 6, has further highlighted these practical concerns, but has also provided some initial evidence of the PR profiles ability to generate personalisation, the core theme of this research.

Section 7.1 views this research from the perspective of operational (user) feasibility. Drawing on the evidence presented during the evaluation phase of the case study of users' perceived personalisation and their evaluation of the PReSS prototypes functionality from the results of the evaluation study (6.2.2 and 6.2.3). Following this the technical feasibility will be considered in 7.2. Again this draws on the results presented in the evaluation study (6.2.1 and 6.2.3) and also considers the development difficulties of the prototype highlighted in 5.3. Section 7.3

reflects on the evidence from the case study as a whole, in the context of the feasibility of the models presented in 3.3 as a mechanism for providing personalisation of services. The conclusions to this research are presented in 7.4 followed by a discussion on potential future work in 7.5.

7.1 Operational feasibility

The operational feasibility of some system is essentially the users' perceptions of that system evaluated from aspects such as usefulness and ease-of-use. This section reflects on the case study and the results of participants' evaluation of the PReSS prototype. This includes the perception of the personalisation provided (6.2.2), the quality of the support provided, the functionality of the system and their perceptions on the usefulness of personalised support (6.2.3). During the evaluation participants were also given the opportunity to make comments on the support they received as shown in Appendix B.3 and B.5. While not directly relevant to the evaluation process defined in 4.4, some of these comments provided anecdotal evidence to their perception of personalised support and are drawn on here where appropriate.

Perceived Personalisation

The measure devised to assess the users' perceived personalisation (discussed in 4.4 and 6.1.2) indicates that neither the tasks nor the two aspects of personality impacted on this perception. Table 6.3 shows very little difference between the aggregated means for each of the users. This indicates that there is no difference and the support hints provided for each user were all regarded by the user in the same way in respect to the context of their current activity. It may also suggest that there simply was not a large enough group of support objects for any difference to be detected. As it stands there is currently too little evidence to make a definitive judgment, but reflecting on the processes the PReSS uses to personalise the output of support, it is unlikely that a difference between the aspects and tasks would be perceived.

The perceived personalisation of the support objects, Table 6.4, presents a different story. Both across users and across support objects there is a range of values, suggesting that individuals perceived the level of personalisation very differently for each piece of support presented. This may suggest at this point that the perceived personalisation measure is more focused on measuring some other aspect of the support, for instance its content, rather than its context to the user's current activity. Alternatively it could also suggest that for particular individuals some of the support failed to be presented within the correct context for that user. Considering some of the technical difficulties experienced by the system (5.3) and the fact that the predictive accuracy was not 100% (Table 6.1) this suggests that support was not presented within the correct context for a number of participants.

To further test this Table 6.5 and Table 6.6 compare the overall aggregated mean of perceived personalisation to the number of correct predictions for task one and task two respectively. These values demonstrate a strong relationship between both the positive and negative case for this measure (6.1.2). This would suggest that perceived personalisation is a fairly accurate measure in this context. However further testing would be required considering the size of the participant group for this evaluation process.

If this measure is correct, and at this stage the evidence presented here goes someway to supporting this, this suggests that the operational feasibility of the PR profile to predict behaviour for personalisation is fairly high. As the predictions used in this measure were generated using the FFM aspect of the PR profile, further evidence of its ability to generate good predictions can also be found in recent literature. (Hu 2010) indicates that the FFM can be used as a collaborative profiling method to some degree of success. The fact that there are more fundamental measures, such as precision and recall, of the FFMs personalisation ability, also may lend support to the accuracy of the perceived personalisation measure, but this is very tentative and would require further testing and comparison.

Quality of the support provided

Table 6.8 (Q3-Q6) shows the results from the post-evaluation survey by participants in response to questions concerning the quality of the support provided. Again this measure is concerned with identifying potential issues with the application not measuring some aspects of personalisation. Overall these results are generally positive but not particularly high. This suggests that some aspects of the support were not satisfactory. From some of the comments made by participants there is evidence that the support text was not always appreciated or well understood, which is contrary to the scores given to the qualities used to generate perceived personalisation. This suggests that the user perceived they understood the support in the context of what they were doing, marked high on the quick sheet (mean of 4.4 across all participants and support), but possibly disagreed with the statement itself, giving negative comments on the quick sheet and the post-evaluation. However this does need to be considered in light of the perceived personalisation scores used above and suggests that there may be limitations to what those scores are suggesting. This also indicates that there was some level of impact on the evaluation process itself from the quality of support.

Functionality of the system

Participants were asked to evaluate the functionality of the system in order to give some indication as to whether or not it had an impact on the evaluation study. The elements selected for evaluation were the layout and functionality of the editor and the impact of completing the profile surveys (the FFM (Johnson 2005) and Appendix A.3). The results presented Table 6.8 (Q1-Q2) show that there were no serious issues, both results displaying a positive trend towards the general functionally. This was an area for concern given the nature of programmers to prefer their own editor for coding. Neither were the results significantly high particularly where the editor was concerned. From comments made by participants the main area for concern was the hint notification method.

One participant compared the prototype to the Eclipse IDE (Eclipse Foundation Inc 2008) often used by Java developers. They suggested that they have become accustomed to its prompts¸ that of the IDE, (PA01) which they felt may have been the reason why they often missed the support statements as they were *looking at the main editor of the prototype*. It is assumed that they meant being accustomed to ignoring them, and for them it was the interaction with the coding process that is their focus. Other participants also commented that hint appearance was often not clear, going on to make suggestions for improvement such as popup boxes, flashing icons or background audio as a means to enhance hint indication.

The hints were output in plain text within a dedicated part of the interface area and with a switching label, notifying users of a new hint as shown in the study's introductory text in Appendix B.1. During the prototype development it was decided that further enhancements may interfere with the user's workflow producing the opposite criticism of too much notification. The review of recommender systems in 2.1 highlighted this as a potential issue with feedback systems and warned against creating too much interference. However, the suggestions made by participants could be built into the system in such a way as to be personalised by individual users depending on the level of hint indication that they feel is appropriate for them. In this way the user remains in control and has the option to choose what is most suitable for them.

As noted previously (2.1.7) automated systems that utilise a profiling mechanism have to make the decision between generating the profile automatically or inviting the user to submit the information into the system. One of the issues raised by Zigoris and Zhang (2006) was that users tend not to participate in the submission of personal information to automated systems. There are several reasons for this: length of time it takes to complete such a task, ambiguity of the questions asked, issues of privacy and data protection.

The evaluation results of the surveys used in this case study are presented in Table 6.9. This shows that participants generally found both surveys easy to use and that they did not feel they were too long to complete. There appears to be no particular issues arising from the use of the surveys which contradicts what some of the research has previously suggested. However, it has been noted during the period of this research that these early concerns are possibly becoming less of an issue across many internet-based applications such as social networking and commercial sites, for example Facebook and Amazon (Amazon 2009) who have built their success on an individual's willingness to share their data in one form or another.

Perceptions on the usefulness of personalised support

The results for this are presented in Table 6.10 which show a relatively high score for all the support scenarios presented (Appendix B.4), generally indicating a positive attitude towards the concept. However participant comments did raise a few points in regard to their concerns about artificial support systems. One commented that they would much rather deal with a live lab assistant, or have code examples to learn from. Preferring the human connection is something very personal. This could possibly be simulated by the use of multi-media support hints, such as video clips, or enabling the system to ring-up a remote but live assistant for those with this preference. This participant also indicated the need to turn off the automation so that they are not bothered by the statements output. Turning off the output statements could be presented as an interface preference, however, turning off the behavioural monitoring would lead to gaps in the systems knowledge about the users learning and current behaviours.

7.2 Technical feasibility

The technical feasibility of a system considers how effective the system performs the functions it has been designed to use and how straightforward these were to implement given resource available. Thus this section reflects on the results presented in 6.2.1 for the predictive ability of the system, 6.2.3 for the degree of successfully presented support, and the process of implementing such a system described 5.2. From this reflection the intention is to draw out the main aspects of the system implementation that were successful and those issues that need to be reconsidered for future work.

Predictive ability

The predictive ability of the system was evaluated across four specific perspectives. Its predictive ability was calculated for the same user and similar task, to the same user and dissimilar task, to a new user and similar task, and a new user and dissimilar task. Table 6.1 shows the results for the same user and Table 6.2 for a new user, across both tasks.

For participants who had a complete profile (utilising both dynamic and static aspects) and undertaking task one, a reasonable quantity of support was produced, which according to the results presented in 6.2.2 was perceived as personalised to their current context. This was more successful for some participants than others. Table 6.1 shows the number of total predictions made by the system across all participants against the percentage of predictions that matched the behaviour actually exhibited by the participant during the evaluation study. For task one this was 66% and for task two a much lower 48%. This measure was selected in order to evaluate the feasibility of applying the profile to different scenarios, with a view to inferring the potential for cross-domain use. These values suggest that either something is impacting on task two or that the means for generating the predictions is not as effective when applied outside of the context they originated from. Giving some indication that applying the static aspects across domains may not be as viable as first perceived.

There are some other factors that may have impacted on task two. The generic support structures required a larger lexicon for translating specific behaviour patterns into more generic ones. Being reliant on a larger lexicon increased the potential for errors. There were also a number of participants who did not attempt task two and most never completed it. This reduced the scope for support output, meaning that the percentage was generated from a much smaller group than for task one. Given these problems further testing is required to asses the degree of potential cross domain application if any, which is discussed further in 7.5.

Table 6.2 shows the results for the new user who joined the evaluation study. Taking the results across both groups for task one there appears to be some similarity in predictive ability of the system. However this is not a substantial result considering that the new group consisted of one participant. These results are not as complete as intended and are inconclusive, partly due to the low number of participants, but also because of technical difficulties experienced with the functionality of the support process for task two.

Degree of success in providing support

This measure is examining the basic functionality of the prototype system to provide support. Table 6.7 shows the number of successful hints outputted to participants for task one and task two. From this there is a clear difference in the success rate of task one to task two. As already mentioned there were a number of technical issues concerning the application of the generic support generated by task two. This was primarily an issue with the lexicon that grouped detailed coding labels into more generic behaviours. This issue was more prominent in task two, as it required more translation due to its more generic nature. The use of ontology as suggested by Anand et al. (2007) in relation to hybrid profiling systems in 2.2.5 would most likely overcome some of these issues, with careful planning.

On examination of the frequency of hint appearance, the system errors may actually have been to the advantage of the evaluation process, as it reduced the number of hints produced. The output of more may have seriously detracted from the users' experience. Regardless of this some users did indicate that the hints were a distraction as discussed in 6.3.3. For task one an average of five hints were presented to each participant in the space of approximately thirty minutes. Whereas if no errors had occurred, then an average eight hints would have been produced, approximately one every four minutes. This suggests that the system needs to monitor this and become more selective about what it outputs within a given period of time. Trying to provide reflective support for all the students' misconceptions all of the time may leave students feeling disorientated or demoralised.

Reflecting on implementation

Section 5.3 discusses a number of issues relating to the prototype development. These are predominantly focused around the process of linking the aspects of personality to behaviours within the context of an area of support. There was also a major change to the model in respect to the relationship between the dynamic and the static aspects resulting in them becoming two parallel processes rather than a single serial process as original envisaged as described in 5.3.

To develop the support objects the initial idea was to use the behaviour patterns directly from the observation study analysis. As mentioned in 5.2.3 and 5.3 there were a number of problems with this: lack of statistical significance with the static-aspects of personality; the selected behaviour patterns did not identify if they satisfied the task requirements and so did not relate to issues of support; non-coding behaviours, such as read:notes, effectively caused interference within the statistical process. This lack of statistical significance could be attributed to the size of the participant group in the observation study. With a greater number of participants more detailed behaviour patterns could have been used and the amount of variance accounted for. Many of the more obvious behaviour patterns could not be used as a base for support because of the lack of statistical significance.

This lack of statistical significance may have been due to the fact the behaviour data was gathered during the observation study without consideration for the intended areas of support. If this had not been the case the descriptions used may have been better informed and more natural correlations revealed during the support object development. However, this is a cyclic argument as a number of the support areas were identified during the analysis process and would most likely have still required the second process of behaviour description adjustment in order to create a correlation with the static aspects of personality.

The process of developing the support objects was not as straightforward as anticipated and some adjustments had to be made to the overall process. However despite these difficulties, the core principle of linking behaviour to personality given a specific instruction was implemented. The main lessons learned here are: how best to describe behaviour patterns within the context of the support the system is designed to offer; and how those descriptions inform the groups used in the statistical process.

The final aspect of the prototype implementation that was not as successful as intended was that of the dynamic aspects. The personality model in Figure 3.7 identifies a number of different elements of the dynamic aspects. Section 4.4 selects from the dynamic aspects those that are more practical to implement given the resources of this research. In the theoretical application (4.3) these aspects are generated by the system gathering data from the interactions of users, with the option to supplement this with explicitly gathered data if required. For the implementation it was decided, primarily due to resource restraints, to gather the data solely using the explicit method via the survey in Appendix B.4. This data was gathered as part of the observation study in 5.1. This survey had not been previously tested but was expected to gather some very basic data regarding the dynamic aspects of the participants.

As with the observation data used with the static aspects, the dynamic aspect data was gathered with no reference to the types of support that would be produced. When the support objects were generated very little of the survey data gathered was relevant to the support contexts identified. This had an impact on the number of dynamic aspect related support developed, only ten statements across four support objects, and also on the support's ability to be outputted. This resulted in the direct evaluation of the dynamic aspects, and their comparison to the static aspects not being possible. The number of support objects that were generated by the dynamic aspects was small with some participants receiving none. On reflection the dynamic aspects data may have served this research more effectively if it had been produced after the support areas had been identified and more target data gathered.

7.3 Feasibility of the models

The models in 3.3 were derived from a review of psychological theories of personality. The review focused on those theories that were from a dispositional perspective and used the concept of traits as a key structure within the theory. In particular the work of Costa and McCrae and the five factor theory they present influenced the final personality model's structure and process as discussed in 3.2.4. The use of a trait based theory as the static aspects of personality within the PReSS profiling system enabled the system to predict behaviour from trait values. The decision to use the five factor model as opposed to Eysenck's three factor model or Cattell's 16 PF, was proven effective as predictions were generated for a number of participants across both programming tasks as discussed in 6.2.1 and from this support was personalised as suggested in 6.2.2. Whether the other instruments would be as effective is an issue for future work.

The dynamic aspects were generated from a number of personality theories drawing on the work of Allport, Cattell and Eysenck to mention a few. Their implementation was initially better defined than the static aspects as described in 4.2, with clear indications of the structure of their attributes and values. However as already discussed in 7.2 the generation of the data to implement them was not as clear-cut as first perceived. This was due to a number of issues including the difficulty of adapting the evaluation data and the change to how they would interact with the static aspects. The impact of these difficulties resulted in only a small number of dynamic aspects being implemented in relation to the support.

The impact these two aspects had on the case study was very different. The static aspects produced enough predictions and personalisation to be able to evaluate it individually, where as the dynamic aspects produced one or two support hints for most of the participants. From this perspective the dynamic aspects may appear less feasible, but without them the model of personality would be incomplete and reliant solely on the collaborative process of the static aspects.

Referring back to the operational feasibility of the PReSS prototype in 7.1 and 6.2.2 there is no discernible difference between the perceived personalisation of either aspect. This may suggest that regardless of the low input of the dynamic aspects into the personalisation process their impact was no better or worse than that of the static aspects. What is the real problem is the method used for implementation and the means by which the dynamic aspects were connected to the support. The change from these aspects working together in serial to working separately in parallel was unexpected. This meant reconsidering how the dynamic aspects would contribute to the process of personalisation. The method implemented (5.2.3) may not have been the most effective way for that contribution to occur and other methods are presented in 7.5.

The technical feasibility discussed in 7.2 highlights a number of issues with the process of implementing the models. The complexity of describing the behaviour patterns required by both static and dynamic aspects was time consuming and difficult to judge in regard to the level of granularity that would work best. The difficulty of relating these behaviours to the relevant tasks via correlation suggests that it is not technically feasible to implement these models.

7.4 Conclusion

The main theme of this thesis is the personalisation of services. To better understand how it is to provide personalised services from a number of aspects, a review of literature was undertaken focusing on profiling techniques and personalisation within an educational context. This identified a number of issues around the process of personalisation as well as a range of techniques currently being applied to make the process more effective across a number of domains. This motivated an investigation into the use of personality, as defined in psychology, in defining a model of personalisation, from that perspective.

This thesis elaborates and analyses the idea that it is feasible to capture and represent personality within a software system for the purpose of personalising services. To gauge the feasibility of this idea and its potential as a personalisation mechanism the thesis identified four research aims (1.3).

The first aim was to generate a generic profiling model from psychological theories of personality that has the potential to find solutions to current profiling issues. This has been met by the development of the models in 3.3.These models have been developed with no intended application area in mind and have drawn only on elements related to the psychological aspects of individuals. The inclusion of both static and dynamic aspects brings together the idea of group similarities and individual differences that can address some of the issues profiling methods are facing.

The second aim has been to consider the practicalities and implications of this approach to personalisation within an educational support context. The behavioural framework is applied to a recommendation process demonstrating how it can be used for prediction in Figure 4.1.This framework and the model of personality are considered in the context of an educational setting putting forward a number of practical ideas as to their implementation in 4.3.

The third aim has been to determine the feasibility of this approach through prototype experimentation within the domain of educational support. A PReSS prototype system was developed (Chapter 4) given a number of development issues (5.3) and has been evaluated in 6.2. The implementation approach taken in the case study is deemed to be unsuccessful particularly in regard to the development of the support objects, which are the key link between a prediction and the support the users receive. While the concept of support appears to be generally well received by the participants (6.1.3) the methods used to track behaviour (Appendix C.1 and C.2) within the support object failed on a number of occasions. There was some confusion from the participants in regard to the support text they received and concern over how the system indicated that support had been provided.

The methods used to capture and implement the dynamic aspects of personality were less successful than those used for the static aspects. Although some support was derived from the dynamic aspects they failed to provide support specific to individuals as intended and discussed in (4.2). Some of this failing can be attributed to the questionnaire used to gather the values for the dynamic aspect attributes. Also the assumption made during the model development (3.2.5) and theoretical application (4.2) of how the static and dynamic aspects could interact in a serial fashion proved incorrect during the implementation stage (5.2.3). This change from serial to parallel places a different emphasis on how they interact and what they can contribute to the personalisation process. This is something that has not been explored fully in this research and needs reconsideration before progressing further, as discussed in 7.5.

For these reasons this implementation is considered not to be technically feasible which has impacted on the ability of the evaluation to generate significant results. Future research is required to consider other means of linking static and dynamic aspects to elements of support as discussed in section 7.5.

The final aim was to identify the potential scope and the limitations of this approach to a cross-domain profiling system. This aim has not been met successfully, as the effectiveness of the case study is somewhat limited and it was unable to generate sufficient data to infer whether the models have the potential for use in a cross-domain context.

In answer to the research question: Can a representation of personality be used to predict behaviour within a software system, in such a way, as to be able to personalise support? There is a small amount of evidence that suggests that some parts of the model have the potential to personalise support. That is the static aspects, which to some extent has already been put forward by recent research (Hu 2010). These have predicted behaviour in a limited number of cases. The dynamic aspects have contributed to the prediction process but only in a couple of cases. The degree of personalisation is high from the user perceptions and appears to have some relationship to the predication process. Without further corroborating evidences it is uncertain at this point how accurate this measure is.

Overall the case study has presented some evidence towards the operational feasibility of utilising a PR profile for personalisation. However the technical difficulties of modelling behaviour and linking to an appropriate support process suggest that the approach used in the evaluation case study is not feasible. The complexity of capturing and correlating behaviours manually resulted in a number of errors during the operation of the PReSS system making results difficult to determine. Whether this process could be better achieved through a different approach is subject to further research.

In light of all this the main contributions of this thesis are:

- the development of a set of profiling models from psychological theories which provides a framework for behaviour and a model of personality accounting for both individual differences and group similarities, as a means of personalising services
- the theoretical application of the profiling models to an educational context demonstrating how they can be utilised for the personalisation of services on a large scale and across multiple domains
- the concept of perceived personalisation as defined in this thesis has some degree of relationship to the level of prediction generated by the system, which requires further investigation to validate this finding
- the initial implementation and evaluation of the profiling models within a prototype personalised support system demonstrating that:
    - the static aspects of personality can predict behaviour to some degree from user to user and less effectively from task to task (Table 6.1 and Table 6.2), the former supporting current research such as (Hu 2010)
    - although there were no perceived difference in personalisation between static and dynamic aspects from the perspective of participants the dynamic aspects provided little additional individual personalisation within the current implementation. This was predominantly due to technical difficulties and further work (7.5) is required to consider how these aspects can contribute more effectively alongside the static aspects
    - the process of generalising behaviour patterns identified in one situation to another is complex and produces less accurate predictions within small participant groups
    - the concept of a support object can be utilised to link behaviour predictions to support for a given task. However the current methods used within the support object are susceptible to error due to their complexity and do not adapt effectively

to changes in user behaviour, requiring further consideration in future work as discussed next.

7.5 Future work

This research provides the foundation for further investigation into how aspects of personality can be utilised as a profiling system for the personalisation of services. Some of which have already been presented in Huang et al. (2006) embedding PR profiles in a semantic framework and Eze et al. (2007) via utilising PR profile for multimedia semantics. This section presents areas for potential development building on the results, questions and issues raised throughout this thesis.

Focusing on the static aspects

Evidence has been presented in this thesis (6.2.1) that it is possible to predict behaviour from the static aspects both between users and across tasks. As this was the most successful aspect of the prototype it would be beneficial to rerun the evaluation process focusing solely on this aspect. Using a larger participant group and running the process over a number of sessions will enable further understanding of how the static aspects generate personalisation over time.

This would provide an opportunity to further test whether or not this aspect can predict outside the user group that generated the behaviours. From the discussion in 6.2.1 only a single participant within the evaluation phase was not part of the observation study which gave only an initial indication of whether this aspect of profile as effective in this manner. This could be done by dividing the larger group into two by random selection, with the second group taking part only in the later sessions.

Whether this is done using the current implementation or after a redevelopment phase needs to be assessed. There were a small number of support objects that showed no serious issues and some that only require corrections to the lexicon and internal logic. Those support objects reliant on dynamic aspects would be removed. There were also several suggestions made by participants in this research for improvement of the interface (7.1). It would be advantageous to implement these at least prior to secondary testing. The aim here is that by focusing on the static aspects baseline data can be generated for later comparison when the dynamic aspects are reintroduced.

It would also be advantageous to focus on the static aspects in regard to whether or not they are able to predict to new tasks or scenarios given data generated from other sources. This would require extending the range of support objects and consideration of how the behaviour patterns can be utilised in new interactions, which is discussed later in this section. By focusing solely on this aspect baseline data can be generated for later comparison with a system operating with both static and dynamic aspects later on.

Re-considering the dynamic aspects

One aspect of this research that did not achieve what was intended was the development and implementation of the dynamic aspects of the personality profile. The initial concept, as discussed in 5.2.3, was for the dynamic aspects to add weight to the static aspects, which proved to be unfeasible. It was then decided to apply the dynamic aspects as a parallel process to the static aspects, with the intention of providing support unique to the individual. How these two processes provide support together needs further consideration. What can the dynamic aspects add that the static aspects cannot? The answer to this may require the dynamic aspects to take a secondary role in the personalisation process, such as providing personalisation when static data is sparse. In this instance the system could utilise the dynamic aspects captured from other learning situations. For example if the user has shown to have poor organisational skills support could focus on this in relation to the new learning area where relevant. It may also be beneficial to link these aspects to a different type of support, possibly one that captures a more personal conversation style, for example motivational statements for goal-based attributes.

There were also a number of issues with the process of identifying dynamic aspects and assigning values to them. Again this process needs to be carefully reconsidered. The attributes selected for populating the dynamic aspects were based on assumptions drawn from the observational study and the discussion group (5.1). These then proved difficult to align with the development of the support objects. Given this difficulty it may be beneficial to identify the attributes required by the support objects first and then populate the values form either explicitly implicit data or a combination of both. Over time the user’s dynamic aspects profile would gather attributes from the various support objects they had interacted with. When these same attributes were reencountered the values would already be available for use. These values may then be adjusted depending on the outcomes of that interaction, generating the dynamic part of these aspects.

The model in Figure 3.7 identifies a number of dynamic aspect elements such as attitude and goals. Some of these elements may be more relevant to the context of educational support than others. While this research tried to generate data for all these elements this may not actually be required. Again this comes back to re-examining the question of how the dynamic aspects add to the personalisation process. These elements and the attributes selected to represent them needs further testing too clearly identify how they can be linked to support within the given domain.

On reflection it is quite clear now how the dynamic aspects can provide a very different role to that of the static aspects. They now appear to be more general in nature than first assumed, with the potential to be applied across a much broader range of domains. Where the focus in this research has been on the static aspects providing cross domain support it may well be the role the dynamic aspects, given their more generic nature, to fill that role as part of the complete profile.

Using a combination of dynamic and static aspects as a means of generating a profile supports the idea of using both group data and individual data. This is central to the concept of using a personality based profile and a number of different approaches may need to be generated and tested to identify the optimum means for these to work effectively together. The potential advantages of developing a two part profile are highlighted in Chapter 4 benefiting both users and service providers.

Developing automated behaviour capture

One particular area of difficulty as discussed in 5.3 was the generation of correlations for low level behaviours. This proved difficult for two reasons: firstly because of the small participant group in the observation study; and secondly the manual process used for extracting and assigning the behaviour patterns. It is assumed that the use of a larger participant group will overcome some of these problems and the remaining issues can be solved through the development of a self-extraction process for the prototype system. This process needs to be able to identify behaviours within the context of the task the user is performing. This will require the development of some form of language structure, possibly using the action:focus pair already implemented and a means of semantically linking similar actions. The development of this structure needs to be undertaken in relation to the redevelopment of the support objects, discussed further on. Once behaviour patterns have been identified the system can then correlate with the participants' static aspects automatically. If the user is then unsuccessful in performing the task it can be assumed that these patterns are related to that outcome. This will be fairly straight forward to develop at the task level, but will be more complex at the instruction level as used in the current implementation. This needs to be considered when redeveloping the support objects.

Re-developing the support objects

The least effective area of the prototype system was the support objects and in particular the hints, that were presented to the users. While this aspect of the prototype was not initially considered a major part of this research they have proven to have had a significant impact on generating results. During the research process these were developed after the tutor discussion group in 4.4. It would be valuable at this stage to receive feedback on the statements from the tutor perspective, or as a working group to develop new statements, as this was a weak point in the researcher's knowledge and experience. Alternative statement types need to be considered as many of the ones developed were intended to encourage self-reflection but proved ambiguous to a number of participants. Also it would be an advantage for the support to be presented in other forms such as images, videos and alternative learning resources such as OERs (Open Educational Resources).

The methods used to link and monitor behaviour with the support objects proved to be subject to error and did not adapt well to changes in participants' real time behaviour. They were also complex and time consuming to assemble. Future developments need to consider a mechanism that overrides the prediction when the participant has learned from previous mistakes and is exhibiting the correct behaviour pattern. This focuses on reconsidering how the monitoring process works in relation to the support objects and which interactions are more important to react to.

Redevelopment of the support objects would also provide an ideal opportunity to update the improvised lexicon with a semantic ontology (Gómez-Pérez et al. 2004 and Davis et al. 2003). Enabling a much more accurate identification and understanding of which programming terms relate to each other which was cause for some of the errors experienced within the evaluation process.

Overall the concept of the support object needs to be considered more carefully and may require some significant changes particularly in regard to changes in the application of the dynamic aspects. It may be more beneficial to move away from this concept and consider methods of linking directly to OERs and other existing educational support mechanisms.

Perceived Personalisation

The results in 6.1.2 demonstrate some correlation between the participants' perceived personalisation and the number of successful predictions. This measure requires some further testing to ascertain how strong this relationship is and whether or not it is suitable for comparing personalisation systems. Further research into similar measures would enable a fuller understanding of what this one has to offer and how it may compliment existing comparison processes. It would also be beneficial to use this measure with other similar prediction and personalisation systems to generate comparative data for future development of the PReSS system as well as to further validate the measure.

Considering alternative platforms

While the multi agent system has performed well in this development it may be necessary to evaluate other types of platform for future developments. At this time none of the more popular educational systems appear to be considering providing the support required for agent interactions. In order for the concept of personality profiling to be provided on a large scale it will be required to work on multiple platforms and interlink with other educational systems. Recent developments on the internet, mobile phones and tablet devices suggest that the use of cloud computing, service orientated architectures and more light weight applications, known as apps are the way forward, although some agent platforms such as JADE (2004) are developing in these areas as well.

Application to other domains

Looking further afield there are two particular areas of investigation that would also benefit and develop the system. The first would be to look at providing support via the PR profile to one or more subject areas outside that of programming. It would be particularly interesting to look at its application to less practical subject areas such as business law or history of art for example. This would possibly require a second observation study for each area to identify the type of description that is best for such activities. This would require providing a new semantic ontology for the subject area.

The second area of investigation would be to apply profiles developed within one domain to an unrelated domain. This could be done in two ways. The first would be to apply the profiles already developed to other subject areas within the domain of computer science. Second would be to take those profiles that have been developed within computer science and apply them to subject areas such as business studies, which are often taught together as part of a cross discipline course.

References

Abdullan, N. A. and Davis, H. C. (2005) A real-time personalization service for SCORM. *5th International Conference on Advanced Learning Technologies (ICALT 2005)*, 5th-8th July 2005 Kaohsiung Taiwan. IEEE, 61-63.

Ackerman, M., Billsus, D., Gaffney, S., Hettich, S., Khoo, G., Kim, D., Klefstad, R., Lowe, C., Ludeman, A., Muramatsu, J., Omori, K., Pazzani1, M., Semler, D., Starr, B. and Yap, P. (1997) Learning probabilistic user profiles: Applications to finding interesting web sites, notifying users of relevant changes to web pages, and locating grant opportunities. *AI Magazine*, 18 (2), 47-56.

ADL (2004a) SCORM® sharable content object reference model: SCORM 2004 2nd edition overview. Advanced Distributed Learning (ADL). Available from: http://www.adlnet.org/downloads/index.cfm [Accessed 15th June 2005].

ADL (2004b) SCORM® sharable content object reference model: SCORM content aggregation model version 1.3.1. Advanced Distributed Learning (ADL). Available from: [Accessed 15th June 2004].

Alkhalifa, E. M. (2005) Investigating the interaction between learner cognitive styles and two multi-media teaching styles. *5th International Conference on Advanced Learning Technologies (ICALT 2005)*, 5th-8th July 2005 Kaohsiung Taiwan. IEEE, 531-535.

Allport, G. W. (1955) *Becoming. Basic considerations for a psychology of personality.* New Haven, USA: Yale University Press.

Allport, G. W. (1973) *Pattern and growth in personality.* USA: Holt, Rnehart and Winston.

Amazon (2009) Amazon: Low prices in electronics, books, sports equipment & more. Available from: http://www.amazon.co.uk/ [Accessed 21st September 2008].

Ambler, S. W. (2004) *The object primer: Agile model-driven development with UML 2.0.* 3rd ed. Cambridge University Press.

Anand, S. S., Kearney, P. and Shapcott, M. (2007) Generating semantically enriched user profiles for web personalization. *ACM Transactions on Internet Technology,* 7 (4), Article 22.

Andriessen, J. and Sandberg, J. (1999) Where is education heading and how about AI? *International Journal of Artificial Intelligence in Education,* 10, 130-150.

Anonymous (2004a) Multiagent systems tool description. Available from: http://www,gsi.dit.upm.es/~mast/whatis.html [Accessed 16th January 2004].

Anonymous (2004b) The open agent architecture. Available from: http://www.ai.sri.com/~oaa/main.html [Accessed 16th January 2004].

Baeza-Yates, R. and Ribeiro-Neto, B. (1999) *Modern information retrieval.* Harlow, UK: Addison-Wesley.

Barrett, P. T., Petrides, K. V., Eysenck, S. B. G. and Eysenck, H. J. (1998) The Eysenck personality questionnaire: An examination of the factorial similarity of P.E.N. and l across 34 countries. *Personality and Individual Differences 25*, 805-819.

Basu, C., Hirsh, H. and Cohen, W. (1998) Recommendation as classification: Using social content-based infromation in recommendation. *15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, Madison Wisconsin USA. American Association of Artificial Intelligence Press, 714-720.

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001) Manifesto for agile software development. Available from: http://agilemanifesto.org/ [Accessed 07/08/2010].

Bellifemine, F., Caire, G., Trucco, T. and Rimassa, G. (2005) Jade programmer's guide (jade 3.3). Available from: http://sharon.cselt.it/projects/jade/ [Accessed 14th June 2005].

Biesta, G. J. J. (2006) *Beyond learning. Democratic education for a human future.* Boulder, USA: Paradigm.

Bigus, J. P. and Bigus, J. (2001) *Constructing intelligent agents using java.* 2nd ed. New York: Wiley.

Bigus, J. P., Schlosnagle, D. A., Pigrim, J. R., Mills, W. N. and Diao, Y. (2002) Able: A toolkit for building multiagent autonomic systems. *IBM Systems Journal: Applications of Artificial Intelligence,* 41 (3), 350-371.

Blackboard (2006) Welcome to blackboard! Washington: Blackboard. Available from: http://www.blackboard.com/us/index.aspx [Accessed 13/04/2006].

Bogers, T. and Bosch, A. V. D. (2007) Comparing and evaluating information retrieval algorithms for news recommendation. *ACM Conference on Recommender Systems (RecSys'07)*, 19th-20th October 2007 Minneapolis MN USA. ACM.

Brusilovsky, P. and Peylo, C. (2003) Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education,* 13, 156-169.

Brut, M. and Sèdes, F. (2010) Ontology-Based Solution for Personalized Recommendations in E-Learning Systems. 10th IEEE International Conference on Advanced Learning Technologies, 469-471.

Buder, J. and Schwind, C. (2012) Learning with personalized recommender systems: A psychological view. Computers in Human Behaviour, 28, 207-216.

Burns, R. B. (2000) *Introduction to research methods.* Australia: Pearson Education.

Cacheda, F., Carneiro, V. I., Ndez, D. F. and Formoso, V. (2011) Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems. ACM Transactions on the Web, 5 (1), Article 2 1-33.

Carver, C. S. and Scheier, M. F. (2004) *Perspectives on personality.* 5th ed. Boston, USA: Pearson Education.

Cattell, R. B. (1971) *Abilities their structure, growth, and action.* USA: Houghton Mifflin Company.

Cattell, R. B. (1992) The relevance of fluid and crystallized intelligence concepts to nature-nurture investigation. *The Mankind Quarterly*, 32 (4), 359-375.

Cattell, R. B. and Child, D. (1975) *Motivation & dynamic structure.* London: Holt-Blond.

Cattell, R. B. and Kline, P. (1975) *The scientific analysis of personality and motivation.* London: Academic Press.

Cattell, R. B., Rao, D. C. and Schuerger, J. M. (1985) Heritability in personality control system: Ego strength (c), super ego strength (g), and the self sentiment (q3); by the mava model, q-data, and maximum likelihood analysis. *Social Behaviour and Personality,* 13 (1), 33-41.

Chen, C.-M., Duh, L.-J. and Liu, C.-Y. (2004) A personalized courseware recommendation system based on fuzzy item response theory. *IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)* IEEE.

Chen, C.-M., Lee, H.-M. and Chen, Y.-H. (2005) Personalized e-learning system using item response theory. *Computers & Education,* 44, 237-255.

Cheung, B., Hui, L., Zhang, J. and Yiu, S. M. (2003) Smarttutor: An intelligent tutoring system in web-based adult education. *The Journal of Systems and Software,* 68, 11-25.

Chou, C.-Y., Chan, T.-W. and Lin, C.-J. (2003) Redefining the learning companion: The past, present, and future of educational agents. *Computers & Education* 40, 255-269.

Cohen, W. W. and Fanb, W. (2000) Web-collaborative filtering: Recommending music by crawling the web. *Computer Networks,* 33 (1-6), 685-698.

Cooper, C. (1998) *Individual differences.* London: Arnold.

Corbin, J. and Strauss, A. (2008) *Basics of qualitative research.* 3rd ed. Thousand Oaks, USA: Sage Publications.

Cremonesi, P., Garzotto, F., Negro, S. and Papadopoulos, A. (2011) Comparative Evaluation of Recommender System Quality. Proceedings of the International Conference on Human Factors in Computing Systems (CHI 2011), 7th–12th May 2011 Vancouver, Canada. 1927-1932.

Creswell, J. W. (2003) *Research design. Qualitative, quantitative and mixed methods approaches.* 2nd ed. California, USA: Sage Publications.

Cristea, A. (2004) Authoring of adaptive and adaptable educational hypermedia: Where are we now and where are we going. *Web-based Education*, 16th-18th February 2004 Innsbruck, Austria.

Cristea, A., Stewart, C. and Sirmakessis, S. (2006) Adaptivity, personalization, and the semantic web. *APS'06*, 23rd August 2006 Odense, Denmark. ACM, 1-2.

Davis, J., Fensel, D. and Harmelen, F. V. (eds.) (2003) *The semantic web. Ontology driven knowledge management* Chichester, UK: John Wiley & Sons.

Dı´Az, A., Garcı´a, A. and Gerva´S, P. (2008) User-centred versus system-centred evaluation of a personalization system. Information Processing and Management, 44, 1293-1307.

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1997) *Human -computer interaction.* 2nd ed. Essex, UK: Pearson Education.

Eclipse Foundation Inc (2008) Eclipse.Org home. Ottawa, Ontario: Eclipse Foundation. Available from: http://www.eclipse.org/ [Accessed 12th October 2008].

Encyclopaedia Britannica (2003) Encyclopaedia Britannica 2003 ultimate reference suite. Encyclopaedia Britannica.

Ewen, R. B. (1993) *An introduction to theories of personality.* 4th ed. Sussex, UK: Psychology Press.

Eysenck, H. J. (1952) *The scientific study of personality.* London: Routledge & Kegan Paul Limited.

Eysenck, H. J. (1967) *The biological basis of personality.* Illinois, USA: Charles C Thomas.

Eysenck, H. J. (1970) *The structure of human personality* 3rd ed. UK: University Paperback.

Eze, E., Ishaya, T. and Wood, D. H. (2007) Contextualizing multimedia semantics towards personalised eLearning. Journal of Digital Information Management, 5 (2), 61-66.

Fan, W., Gordon, M. D. and Pathak, P. (2005) Effective profiling of consumer information retrieval needs: A unified framework and empirical comparison. *Decision Support Systems,* 40 (2), 213-233.

Fipa (2003) Welcome to the foundation for intelligent physical agents. FIPA. Available from: http://www.fipa.org/ [Accessed 6th May 2004].

Fipa (2010) Agent communication language specifications. FIPA. Available from: http://www.fipa.org/repository/aclspecs.html [Accessed 20th April 2010].

Freitas, S. D. and Yapp, C. (eds.) (2005) *Personalizing learning in the 21st century.* Stafford, UK: Network Education Press.

Friesen, N., Hesemeier, S. and Roberts, A. (2004) Cancore. Guidelines for learning object metadata. *In:* McGreal, R. (ed.) *Online education using learning objects.* Oxon, UK: Routledge, 225-235.

Garrison, D. R. and Anderson, T. (2003) *E-learning in the 21st century. A framework for research and practice.* London: Routledge Falmer.

Gasparini, I., Pimenta, M. S., Oliveira, J. P. M. D. and Bouzeghoub, A. (2010) Combining Ontologies and Scenarios for Context-Aware e-Learning Environments. The 28th ACM International Conference on Design of Communication, 27th-29th September 2010 São Carlos-São Paulo, Brazil. 229-236.

Gillham, B. (2000a) *Developing a questionnaire.* London: Continuum.

Gillham, B. (2000b) *The research interview.* London: Continuum.

Godoy, D. and Amandi, A. (2003) A user profiling architecture for textual-based agents. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial,* 21, 27-36.

Godoy, D., Schiaffino, S. and Amandi, A. (2004) Interface agents personalizing web-based tasks. *Cognitive Systems Research*, 5, 207-222.

Golbeck, J., Robles, C. and Turner, K. (2011) Predicting Personality with Social Media. Proceedings of the International Conference on Human Factors in Computing Systems (CHI 2011), 7th-12th May 2011 Vancouver, Canada. ACM, 253-262.

Goldberg, D., Nichols, D., Oki, B. M. and Terry, D. (1992) Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35 (12), 61-70.

Gómez-Pérez, A., Fernández-López, M. and Corcho, O. (2004) *Ontological engineering.* London: Springer-Verlag.

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J. and Riedl, J. (1999) Combining collaborative filtering with personal agents for better recommendations. *16th National Conference on Artificial Intelligence*. American Association of Artificial Intelligence Press, 439-446.

Google (2009) Google UK. Available from: http://www.google.co.uk/webhp?hl=en [Accessed 24th October 2009].

Grigoriadou, M., Papanikolaou, K., Kornilakis, H. and Magoulas, G. (2001) Inspire: An intelligent system for personalized instruction in a remote environment. *Third Workshop on Adaptive Hypertext and Hypermedia, User Modelling.* 31-40.

Halff, H. M. (1988) Curriculum and instruction in automated tutors. *In:* Polson, M. C. and Richardson, J. J. (eds.) *Foundations of intelligent tutoring systems.* Lawrence Erlbaum Associates, 79-108.

Hampson, S. E. (1988) *The construction of personality an introduction.* 2nd ed. New York: Routledge.

Harper, D. (2001) Online etymology dictionary. Wikipedia. Available from: http://www.etymonline.com/index.php?search=profile&searchmode=none [Accessed 28th December 2009].

Hella, L. and Krogstie, J. (2011) Personalisation by Semantic Web Technology in Food Shopping. Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS'11) 25th-27th May 2011 Sogndal, Norway. ACM, Article 34.

Hergenhahn, B. R. (1980) *An introduction to theories of personality.* Essex, UK: Prentice-Hall.

Hietala, P. and Niemirepo, T. (1998) The competence of learning companion agents. *International Journal of Artificial Intelligence in Education,* (9), 178-192.

Hu, R. (2010) Design and User Issues in Personality-based Recommender Systems. RecSys '10 Fourth ACM Conference on Recommender Systems, 26th-30th September 2010 Barcelona, Spain. 357-360.

Huang, W., Webster, D., Wood, D. H. and Ishaya, T. (2006) An intelligent semantic e-learning framework using semantic web technologies. British Journal of Educational Technology: Semantic web for e-learning, 37 (3), 351-373.

IBM (2000) Agent building and learning environment. Available from: http://www.alphaworks.ibm.com/tech/able [Accessed 10th April 2010].

IBM (2002) Aglets USA: IBM. Available from: http://www.trl.ibm.com/aglets/ [Accessed 18th September 2005].

IEEE (2004) UK learning object metadata core, draft 0.2. New York: IEEE. Available from: http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html [Accessed 21st July 2005].

Ilson, R., Crystal, D., Wells, J. C. and Long, T. H. (eds.) (1987) *Universal dictionary*. London: The Reader's Digest Association Limited.

Im, I. and Hars, A. (2007) Does a one-size recommendation system fit all? The effectiveness of collaborative filtering based recommendation systems across different domains and search modes. *ACM Transactions on Information Systems*, 26 (1), Article 4.

Ip, R. W. L., Lau, H. C. W. and Chan, F. T. S. (2000) An intelligent internet information delivery system to evaluate site preferences. *Expert Systems with Applications,* 18 (1), 33-42.

Jade (2004) Java agent development framework. Available from: http://sharon.cselt.it/projects/jade/ [Accessed 14th June 2004].

JISC (2010b) Open educational resources infokit. What are open educational resources? JISC. Available from: https://openeducationalresources.pbworks.com/w/page/24836860/What-are-Open-Educational-Resources [Accessed 04th December 2010].

John, O. P. and Srivastava, S. (1999) The big five trait taxonomy: History, measurement, and theoretical perspectives. *In:* Pervin, L. A. and John, O. P. (eds.) *Handbook of personality theory and research.* 2nd ed. New York: The Guilford Press, 102-138.

Johnson, J. A. (2000) Web-based personality assessment. Available from: http://www.personal.psu.edu/~j5j/papers/web.html [Accessed 8th September 2008].

Johnson, J. A. (2005) IPIP-NEO. Available from: http://www.personal.psu.edu/faculty/j/5/j5j/IPIP/ipipneo300.htm [Accessed.

Jones, M. T. (2002) Java mobile agents & the aglets SDK. *Dr. Dobb's Journal: Software Tools for the Professional Programmer,* 27 (1), 42-48.

Kabassi, K. and Virvou, M. (2004) Personalised adult e-training on computer use based on multiple attribute decision making. *Interacting with Computers,* 16, 115-132.

Kawamae, N. (2011) Predicting Future Reviews: Sentiment Analysis Models for Collaborative Filtering. WSDM’11, 9th–12th February 2011 Hong Kong, China. ACM, 605-614.

Kay, J. (2006) Scrutable adaptation: Because we can and must. *In:* Wade, V., Ashman, H. and Smyth, B. (eds.) *Adaptive hypermedia and adaptive web-based systems. Fourth international conference, ah 2006, Dublin, Ireland, June 2006 proceedings.* Berlin: Springer-Verlag, 11-19.

Keenoy, K., Freitas, S. D., Levene, M., Jones, A., Brasher, A., Waycott, J., Kaszás, P., Turcsányi-Szabó, M. and Montandon, L. (2004) Personalise trails and learner profiling within e-learning environments. Available from: http://www.dcs.bbk.ac.uk/trails/results.html [Accessed 15th June 2005].

Kelly, G. A. (1963) *A theory of personality. The psychology of personal constructs.* New York: W.W. Norton and Company.

Kolb, D. A. (1984) *Experiential learning: Experience as the source of learning & development.* New Jersey, USA: Prentice Hall.

Krulwich, B. (1997) Lifestyle finder. *AI Magazine*, 18 (2), 37-45.

Kshirsagar, S. (2002) A multilayer personality model. Symposium on Smart Graphics, 11-13 June 2002 New York. ACM.

Lee, C.-H., Kim, Y.-H. and Rhee, P.-K. (2001) Web personalization expert with combining collaborative filtering and association rule mining technique. Expert Systems with Applications, 21 (3), 131-137.

Lee, D.-S., Kim, G.-Y. and Choi, H. (2003) A web-based collaborative filtering system. Pattern Recognition, 36 (2), 519-526.

Leech, N. L., Barrett, K. C. and Morgan, G. A. (2005) *SPSS for intermediate statistics: Use and interpretation.* 2nd ed. New Jersey, USA: Lawrence Erlbaum Associates.

Li, F., Zhao, J., Shih, T. K., Lau, R., Li, Q. and McLeod, D. (eds.) (2008) *Advances in web based learning - ICWL 2008 7th international conference Jinhua, China.* . Berlin: Springer-Verlag.

Liang, G., Weining, K. and Junzhou, L. (2006) Courseware recommendation in e-learning system. *In:* Liu, W., Lau, R. W. H. and Li, Q. (eds.) *Advances in web-based learning - ICWL 2006. Fifth international conference, Penang, Malaysia, July 2006 proceedings.* Berlin: Springer-Verlag, 10-24.

Liang, H., Xu, Y. and Li, Y. (2010) Mining Users' Opinions based on Item Folksonomy and Taxonomy for Personalized Recommender Systems. IEEE International Conference on Data Mining Workshops, 1128-1135.

Lieberman, H. (1995) Letizia: An agent that assists web browsing. *14th International Joint Conference on Artificial Intelligence (IJCAI'95)*. San Mateo CA USA: Morgan Kaufmann, 924-929.

Luck, M., Ashri, R. and D'inverno, M. (2004) *Agent-based software development.* Norwood USA: Artech House.

Lycos (2009) Lycos. Go get it. Available from: http://www.lycos.co.uk/ [Accessed 24th October 2009].

Malim, T. and Birch, A. (1998) *Introduction psychology.* Basingstoke, UK: Macmiklan Press.

Marin, B.-F., Hunger, A., Werner, S., Meila, S. and Schuetz, C. (2004) A generic framework for an interface tutor agent within a virtual collaborative learning environment. *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 30th August-1st September 2004 Joensuu Finland. IEEE, 31-35.

Marsella, S. and Gratch, J. (2003) Modelling coping behaviour in virtual humans: Don't worry, be happy. 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03), Melbourne Australia. ACM Press, 313-320.

Massa, R. D., Montagnuolo, M. and Messina, A. (2010) Implicit News Recommendation Based on User Interest Models and Multimodal Content Analysis. 3rd International Workshop on Automated Information Extraction in Media Production, 29th October 2010 Firenze, Italy. 33-38.

McCorduck, P. (2004) *Machines who think.* Massachusets, USA: A K Peters.

McCrae, R. R. and Costa, P. T. (1999) A five-factor theory of personality. *In:* Pervin, L. A. and John, O. P. (eds.) *Handbook of personality theory and research.* 2nd ed. New York: The Guilford Press, 139-153.

McCrae, R. R. and Costa, P. T. (2003) *Personality in adulthood.* 2nd ed. New York: The Guilford Press.

McCrae, R. R. and Jr., P. T. C. (2004) A contemplated revision of the neo five-factor inventory. *Personality and Individual Differences,* 36, 587-596.

McCrae, R. R. and P.John, O. (1992) An introduction to the five-factor model and its applications. *Journal of Personality,* 60 (2), 175-215.

Merriam, S. B., Caffarella, R. S., Wlodkowski, R. J. and Cranton, P. (2001) *Adult learning: Theories, principles and applications.* USA: Jossey-Bass.

Miles, J. and Shevlin, M. (2001) *Applying regression & correlation. A guide for students and researchers.* London: Sage.

Mitrovic, A. and Martin, B. (2004) Evaluating adaptive problem selection. *In:* Bra, P. D. and Nejdl, W. (eds.) *Adaptive hypermedia and adaptive web-based systems.* Berlin: Springer-Verlag, 185-194.

Mobasher, B. (2007) Recommender systems. *Kunstliche Intelligenz, Special Issue on Web Mining,* 3, 41-43.

Montaner, M., López, B. and De-La-Rosa, J. L. (2003) A taxonomy of recommender agents on the internet. *Artificial Intelligence Review,* 19 (4), 285-330.

Moon, J. A. (2004) *A handbook of reflective and experiential learning: Theory and practice.* Oxon, UK: Routledge Falmer.

Moukas, A. (1996) Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. *Conference on Practical Applications of Agents and Multiagent Technology,* London.

Mulwa, C., Lawless, S., Sharp, M., Arnedillo-Sanchez, I. and Wade, V. (2010) Adaptive Educational Hypermedia Systems in Technology Enhanced Learning: A Literature Review. Special Interest Group for Information Technology Education 2010, 7th-9th October 2010 Midland, Michigan, USA. 73-84.

Murray, H. A. (1938) *Explorations in personality: A clinical and experimental study of fifty men of college age.* New York: Oxford University Press.

Nanas, N., Uren, V. and Roeck, A. D. (2003) Building and applying a concept hierarchy representation of a user profile. *26th Annual ACM on Research and Development in Information Retrieval (SIGIR' 03)*, Toronto, Canada. ACM, 198-204.

Netbeans (2008) Welcome to Netbeans. Available from: http://www.netbeans.org/ [Accessed 12th October 2008].

Oppenheim, A. N. (1992) *Questionnaire design, interviewing and attitude measurement.* London: Continuum.

Oracle (2011) Oracle Technology Network. Available from: http://www.oracle.com/technetwork/java/index.html [Accessed 12th November 2011].

Ortony, A., Clore, G. L. and Collins, A. (1988) *The cognitive structure of emotions.* Cambridge University Press.

Pennock, D. M., Horvitz, E., Lawrence, S. and Giles, C. L. (2000) Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. *16th Conference on Uncertainty in Artificial Intelligence (UAI-2000)*. Morgan Kaufmann, 473-480.

Pervin, L. A. (1989) Personality Theory and Research. 5th ed. Sussex, UK: John Wiley & Sons.

Pervin, L. A. and John, O. P. (eds.) (1999) *Handbook of personality theory.* New York: The Guilford Press.

PPIG (2011) Psychology of Programming INTEREST GROUP. Available from: http://www.ppig.org/ [Accessed 22nd October 2011].

Pressman, R. S. (2000) Software Engineering: A Practitioners Approach - European Adaptation. 5 ed. McGraw Hill.

Quijano-S´Anchez, L., Recio-Garc´ıa, J. A. and D´ıaz-Agudo, B. E. (2010) Personality and Social Trust in Group Recommendations. 22nd International Conference on Tools with Artificial Intelligence, IEEE, 121-126.

Ralph, P. and Parsons, J. (2006) A framework for automatic online personalization. 39th Hawaii International Conference on System Sciences (HICSS-39), 4th-7th January Kauai, Hawaii, USA. IEEE.

Reber, A. S. and Reber, E. (2001) The penguin dictionary of psychology. 3rd ed. London: Penguin Group.

Recio-Garcia, J. A., Jimenez-Diaz, G., Sanchez-Ruiz, A. A. and Diaz-Agudo, B. (2009) Personality Aware Recommendations to Groups. 3rd ACM Conference on Recommender Systems, 23rd-25th October 2009 New York, New York, USA. 325-328.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994) Grouplens: An open architecture for collaborative filtering of netnews. ACM 1994 Conference on Computer Supported Cooperative Work, North Carolina United States. ACM Press, 175-186.

Robinson-Riegler, G. and Robinson-Riegler, B. (2004) *Cognitive psychology. Applying the science of the mind.* Boston, USA: Pearson Education.

Roh, T. H., Oh, K. J. and Han, I. (2003) The collaborative filtering recommendation based on SOM cluster-indexing CBR. *Expert Systems with Applications,* 25 (3), 413-423.

Rubin, J. (1994) *Handbook of usability testing.* USA: John Wiley & Sons.

Ruelland, D. and Brisebois, A. (2002) An electronic performance support system for the eLearner. *International Conference on Computers in Education (ICCE'02),* 3rd-6th December 2002 Auckland New Zealand. IEEE, 1072-1077.

Rumetshofer, H. and Wöß, W. (2005) Xml-based adaptation framework for psychologically-driven e-learning systems. *Educational Technology & Society,* 6 (4), 18-29.

Russell, S. and Norvig, P. (2003) *Artificial intelligence a modern approach.* 2nd ed. Upper Saddle River, USA: Pearson Education.

Schultz, D. P. and Schultz, S. E. (2005) *Theories of personality.* 8th ed. Belmont, USA: Thomson Wadsworth.

Shapira, B., Shoval, P. and Hanani, U. (1997) Stereotypes in information filtering systems. *Information Processing & Management,* 33 (3), 273-287.

Shardanand, U. and Maes, P. (1995) Social information filtering: Algorithms for automating "Word of mouth". *SIGCHI Conference on Human Factors in Computing Systems*, Denver Colorado USA. ACM Press, 210-217.

SPSS (2006) Statistical analysis at SPSS data analysis, SPSS, data management, statistics software. Chicago: SPSS. Available from: http://www.spss.com/spss/index.htm?source=homepage&hpzone=ad_box [Accessed 15/04/2010].

Stash, N. V., Cristea, A. I. and Bra, P. M. D. (2004) Authoring of learning styles in adaptive hypermedia: Problems and solutions. *13th international World Wide Web conference*, 17th-22nd May 2004 New York. ACM Press, 114-123.

Stephenson, J. (ed.) (2001) *Teaching & learning online. Pedagogies for new technologies.* London: Kogan Page.

Sutherland, S. and Brotchie, J. (2011) Pebblegogy: Ideas and activities to inspire and engage. Telford, UK: Pebble Learning.

Tennant, M. (2006) *Psychology and adult learning.* 3rd ed. Oxon, UK: Routledge.

The Nielsen Company (2010) Nielsen Claritas. New York: Nielsen. Available from: http://en-us.nielsen.com/tab/product_families/nielsen_claritas/prizm [Accessed 2nd January 2010].

Umyarov, A. and Tuzhilin, A. (2011) Using External Aggregate Ratings for Improving Individual Recommendations. ACM Transactions on the Web, 5 (1), Article 3 1-39.

Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudga, C. and Grant, S. (1999) A multi-agent design of a peer-help environment. *Artificial Intelligence and Education (IAIED'99)*, Le Mans France. IOS Press, 38-46.

Vizcaino, A. and Boulay, B. D. (2002) Using a simulated student to repair difficulties in collaborative learning. *International Conference on Computers in Education, ICCE 2002*, New Zealand. IEEE Computer Society, 349-353.

Wang, X. and Li, B. (2006) Intelligent knowledge recommendation system based on web log and cache data. *In:* Liu, W., Lau, R. W. H. and Li, Q. (eds.) *Advances in web-based learning - ICWL 2006. Fifth international conference, Penang, Malaysia, July 2006 proceedings.* Berlin: Springer-Verlag, 48-56.

Weng, S.-S. and Liu, M.-J. (2004) Feature-based recommendations for one-to-one marketing. *Expert Systems with Applications*, 26 (4), 493-508.

Wiggins, J. S. (2003) *Paradigms of personality assessment.* New York: The Guilford Press.

Wolf, C. (2003) iWeaver: Towards 'learning style'-based e-learning in computer science education. *Australasian Computing Education Conference (ACE2003)*.

Wood, D. H. and Ishaya, T. (2005) Representing learners' personality within eLearning agent systems. Cognition and Exploratory Learning in Digital Age (CELDA 2005), 14th-16th December 2005 Porto Portugal.

Xu, D., Wang, H. and Su, K. (2002) Intelligent student profiling with fuzzy models. *35th Hawaii International Conference on System Sciences (HICSS'02)*, Hawaii. IEEE.

Xuan, Z., Xia, H. and Miao, J. (2011) A Personalized Recommender Algorithm based on Semantic Tree. Fourth International Joint Conference on Computational Sciences and Optimization, 15th-19th April 2011 Yunnan, China. IEEE, 1304-1308.

Yahoo (2009) Yahoo! UK & Ireland. Available from: http://m.uk.yahoo.com/ [Accessed 24th October 2009].

Yang, F., Han, P., Shen, R. and Hu, Z. (2005) A novel resource recommendation system based on connecting to similar e-learners. *In:* Lau, R. W. H., Li, Q., Cheung, R. and Liu, W. (eds.) *Advances in web-based learning - ICWL 2005. Fourth international conference, Hong Kong, china, July/August 2005 proceedings.* Berlin: Springer-Verlag, 122-130.

Yong, S. J., Lee, H. D., Yoo, H. K., Youn, H. Y. and Song, O. (2011) Personalized Recommendation System Reflecting User Preference with Context-awareness for Mobile TV. Ninth IEEE International Symposium on Parallel and Distributed Processing, 26th-28th May 2011 Busan, Korea. IEEE, 232-237.

Yu, P. S. (1999) Data mining and personalization technologies. *6th International Conference on Database Systems for Advanced Applications (DASFAA '99)*, 19th-22nd April 1999 Hsinchu Taiwan. IEEE, 6-17.

Zaïane, O. R. (2002) Building a recommender agent for e-learning systems. *International Conference on Computers in Education (ICCE'02)*, 3rd-6th December 2002 Auckland New Zealand. IEEE, 55-59.

Zarraonandia, T., Fernández, C., Díaz, P. and Torres, J. (2005) On the way of an ideal learning system adaptive to the learner and her context *5th International Conference on Advanced learning Technologies (ICALT 2005),* 5th-8th July 2005 Kaohsiung Taiwan. IEEE, 641-643.

Zhang, X. and Han, H. (2005) An empirical testing of user stereotypes of information retrieval systems. *Information Processing and Management,* 41 (3), 651-664.

Zhuhadar, L. and Nasraoui, O. (2010) Personalized Search based on a User-cantered Recommender Engine. ACM International Conference on Web Intelligence and Intelligent Agent Technology, 200-203.

Zigoris, P. and Zhang, Y. (2006) Bayesian adaptive user profiling with explicit & implicit feedback. *Conference on Information and Knowledge Management (CIKM'06)*, 5th–11th November 2006 Arlington Virginia USA. ACM Press, 397-404.

Bibliography

Abercrombie, N., Hill, S. and Turner, B. S. (2000) The penguin dictionary of sociology. 4th ed. London: Penguin Books.

Abreu, A. D. and Kurze, M. (2002) Visualizing user's personality and information in 3d environments, a step beyond avatars. 6th International Conference on Information Visualisation (IV'02), 10-12 July 2002 London. IEEE, 303-308.

Advanced Distributed Learning (2009) Adlnet - advanced distributed learning home page. USA: Available from: http://www.adlnet.gov/Pages/Default.aspx [Accessed 18th January 2009].

Alonso, E., Kudenko, D. and Karzakov, D. (eds.) (2003) *Adaptive agents and multi-agent systems.* Berlin: Springer-Verlag.

Ambler, S. W. (2005) The elements of UML 2.0 style. New York: Cambridge University Press.

Amichai-Hamburger, Y. (2002) Internet and personality. Computers in Human Behaviour, 18, 1-10.

Amiel, T. and Sargent, S. L. (2004) Individual differences in internet usage motives. Computers in Human Behaviour 711-726.

André, E., Klesen, M., Gebhard, P., Allen, S. and Rist, T. (1999) Integrating models of personality and emotions into lifelike characters. *Workshop on Affect in Interactions - Towards a new Generation of Interfaces in conjunction with the 3rd i3 Annual Conference*, Siena Italy. 136-149.

André, E., Klesen, M., Gebhard, P., Allen, S. and Rist, T. (2000) Exploiting models of personality and emotions to control the behaviour of animated interactive agents. *Workshop on "Achieving Human-Like Behaviour in Interactive Animated Agents" in conjunction with the 4th International Conference on Autonomous Agents*, Barcelona Spain. 3-7.

André, E., Rist, T. and Muller, J. (1997) Webpersona: A life-like presentation agent for the world-wide web. *Workshop on Animated Interface Agents: Making them Intelligent (IJCAI-97)*, 25th August 1997 Nagoya, Japan.

Angehrn, A., Nabeth, T. and Roda, C. (2001) Towards personalised, socially aware and active e-learning systems: Illustrated with the agent-based system k-inca. Available from: http://www.calt.insead.edu/eis/documents/K-InCA-whitepaper.pdf [Accessed 24th February 2005].

Anghel, C., Godja, C., Dinsoreanu, M. and Salomie, I. (2005) Jade based solutions for knowledge assessment in eLearning environments. Available from: http://jade.tilab.com/papers/EXP/Anghel.pd [Accessed 14th June 2004].

Anwar, A. and Franklin, S. (2003) Sparse distributed memory for 'conscious' software agents. Cognitive Systems Research, 4, 339-354.

Argyle, M. (1991) *Cooperation the basis of sociability.* London: Routledge.

Askar, P. and Halici, U. (2004) E-learning as a catalyst for educational innovation. *In:* Ghaoui, C. (ed.) *E-education applications: Human factors and innovative approaches.* London: Information Science, 196-206.

Ayersman, D. J. and Minden, A. V. (1995) Individual differences, computers and instruction. Computers in Human Behaviour, 11 (3-4), 371-390.

Badler, N. I., Reich, B. D. and Webber, B. L. (1997) Towards personalities for animated agents with reactive and planning behaviours. In: Trappl, R. and Petts, P. (eds.) Creating personalities for synthetic actors. Towards autonomous personality agents. Berlin: Springer -Verlag, 43-57.

Batson, T. (2002) The electronic portfolio boom: What's it all about? Syllabus Media group. Available from: http://www.campus-technology.com/article.asp?id=6984 [Accessed 9th June 2005].

Bazzan, A., Bordini, R., Andrioti, G., Vicari, R. and Wahle, J. (2000) Wayward agents in a commuting scenario (personalities in the minority game). 4th International Conference on MultiAgent Systems (ICMAS-2000), 10-12 July 2000 Boston USA. IEEE, 5-12.

Bellifemine, F., Caire, G., Trucco, T., Rimassa, G. and Mungenast, R. (2005) Jade administrator's guide. Available from: http://sharon.cselt.it/projects/jade/ [Accessed 14th June 2005].

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) The semantic web. USA: Scientific American. Available from: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&pageNumber=1&catID=2 [Accessed 6th May 2004].

Blöchl, M., Rumetshofer, H. and Wöß, W. (2003) Individualized e-learning systems enabled by a semantically determined adaptation of learning fragments. *14th International Workshop on Database and Expert Systems Applications (DEXA'03)*. IEEE, 640-646.

Bradshaw, J. M. (1996) Kaos: An open agent architecture supporting reuse, interoperability, and extensibility. 10th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96), 9th-14th November 1996 Banff, Canada. 14-17.

Brainov, S. (2000) The role and the impact of preferences on multiagent interaction. In: Jennings, N. R. and Lesperance, Y. (eds.) Intelligent agents vi. Berlin: Springer-Verlag, 349-363.

Brooks, R. (2001) The relationship between matter and life. *NATURE,* 409 (1), 409-411.

Brooks, R. A. (1990) Elephants don't play chess. *Robotics and Autonomous Systems,* 6 (1-2), 3-15.

Brooks, R. A. (1991) Intelligence without representation. *Artificial Intelligence,* 47, 139-159.

Brusilovsky, P. (2004) Knowledgetree: A distributed architecture for adaptive e-learning. *13th international World Wide Web conference* 17th-22nd May 2004 New York. ACM Press, 104-113

Busetta, P., Howdenn, N., Ronnquist, R. and Hodgson, A. (2000) Structuring BDI agents in functional clusters. In: Jennings, N. R. and Lesperance, Y. (eds.) Intelligent agents vi. Berlin: Springer-Verlag, 277-289.

Caire, G. (2003) Jade tutorial - jade programming for beginners. Available from: http://sharon.cselt.it/projects/jade/ [Accessed 14th June 2004].

Caire, G. and Cabanillas, D. (2004) Jade tutorial application-defined content languages and ontologies. Available from: http://sharon.cselt.it/projects/jade/ [Accessed 14th June 2005].

Caspi, A., Chajut, E., Saporta, K. and Beyth-Marom, R. (2005) The influence of personality on social participation in learning environments. *Learning and Individual Differences*, ARTICLE IN PRESS.

Cawsey, L. (1998) *The essence of artificial intelligence.* Essex, UK: Prentice Hall.

Chamorro-Premuzic, T., Furnham, A., Dissou, G. and Heaven, P. (2005) Personality and preference for academic assessment: A study with Australian university students. *Learning and Individual Differences* (15), 247-256.

Choe, H., Bae, Y., Kim, T. and Lee, T. (2004) Work in progress - the study of web-based adaptive feedback based on the analysis of individual differences. 34th ASEE/IEEE Frontiers in Education Conference, 20th–23rd October 2004 Savannah GA. IEEE, 25-26.

Cordon, O., Herrera-Viedma, E., Lopez-Pujalte, C., Luque, M. and Zarco, C. (2003) A review on the application of evolutionary computation to information retrieval. International Journal of Approximate Reasoning, 34, 241-264.

Cole, J. (2005) *Using Moodle. Teaching with the popular open source course management system.* Sebastopol, USA: O'Reilly.

Cornelius, B. (2001) *Understanding java.* Harlow, UK: Pearson Education.

Coronel, R. (2000) *Database systems. Design, implementation and management.* 4th ed. Cambridge: Course Technology.

Cotterill, S., McDonald, T., Drummond, P. and Hammond, G. (2004) Design, implementation and evaluation of a 'generic' ePortfolio: The Newcastle experience. *ePortfolio 2004*, La Rochelle.

Cox, B. J., Borger, S. C., Taylor, S., Fuentes, K. and Ross, L. M. (1999) Anxiety sensitivity and the five-factor model of personality. Behaviour Research and Therapy, 37 (7), 633-641.

CRA (2006) E-portfolios – pre 19 a survey of what is around – February 2006. Available from: http://www.recordingachievement.org/downloads/e-Portfolios_Pre19_Survey_Feb06.pdf [Accessed 14th March 209].

CRA (2010) Centre for recording achievement. UK:CRA. Available from: http://www.recordingachievement.org/ [Accessed 4th December 2010].

Dall, S. R. X., Houston, A. I. and McNamara, J. M. (2004) The behavioural ecology of personality: Consistent individual differences from an adaptive perspective. Ecology Letters, 7, 734-739.

Daniel, B. K. and Mohan, P. (2004) A model for evaluating learning objects. *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 30th August-1st September 2004 Joensuu Finland. IEEE, 56-60.

Davis, D. N. (2000) Minds have personalities - emotion is the core. *Symposium on How to Design Functional Mind at AISB2000 Convention*, 17th-20th April 2000 University of Birmingham, UK.

Davis, D. N. (2003) Architectures for cognitive and a-life agents. In: Plekhanova, V. and Wermter, S. (eds.) Intelligent agent software engineering. IDEA Group.

Decker, K. and Sycara, K. (1997) Intelligent adaptive information agent. *Journal of Intelligent Information Systems,* 9, 239-260.

Decker, K., Sycara, K. and Williamson, M. (1997) Middle-agents for the internet. 15th International Joint Conference on Artificial Intelligence (IJCAI), 25th August 1997 Nagoya, Japan. 578-583.

Deitel, H. M. and Deitel, P. J. (1999) *Java how to program.* 3rd ed. New Jersey, USA: Prentice-Hall.

Derntl, M. and Motschnig-Pitrik, R. (2004) Patterns for blended, person-centred learning: Strategy, concepts, experiences, and evaluation. *ACM symposium on Applied computing*, 14th-17th March 2004 Nicosia Cyprus. ACM Press, 916-923.

DfES (2005) Harnessing technology transforming learning and children's services: Summary version. Nottingham: DfES Publications. Available from: http://www.dfes.gov.uk/publications/e-strategy/docs/e-strategysummary.pdf [Accessed 3rd March 2006].

Dickinson, I. and Wooldridge, M. (2003) Towards practical reasoning agents for the semantic web. *2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03)*, 14th-18th July 2003 Melbourne Australia. ACM Press, 827-834.

Doran, J. E., Franklin, S., Jennings, N. R. and Norman, T. J. (1997) On cooperation in multi-agent systems. The Knowledge Engineering Review, 12 (3), 309-314.

Dornan, T., Lee, C., Stopford, A., Hosie, L., Maredia, N. and Rector, A. (2005) Rapid application design of an electronic clinical skills portfolio for undergraduate medical students. *Computer Methods and Programs in Biomedicine,* 78, 25-33.

Downes, S. (2004) Learning objects. Resources for learning worldwide. *In:* McGreal, R. (ed.) *Online education using learning objects.* Oxon, UK: Routledge, 21-31.

Du, J., Zheng, Q., Li, H. and Yuan, W. (2005) The research of mining association rules between personality and behaviour of learner under web-based learning environment. *In:* Lau, R. W. H., Li, Q., Cheung, R. and Liu, W. (eds.) *Advances in web-based learning - ICWL 2005. Fourth international conference, Hong Kong, china, July/august 2005 proceedings.* Berlin: Springer-Verlag, 111-121.

Egges, A., Kshirsagar, S. and Thalmann, N. M. (2003) A model for personality and emotion simulation *7th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES2003)* 3rd- 5th September 2003 Oxford UK. Springer-Verlag.

Ekdahl, B. (2001) Can computers be social? 5th International Conference on Computing Anticipatory Systems (CASYS 2001), 13th-18th August 2001 Liege Belgium. 95-106.

El-Nasr, M. S., Ioerger, T. R., Yen, J., House, D. H. and Parke, F. I. (1999) Emotionally expressive agents. *Computer Animation 1999*, 26-28 May 1999 Geneva Switzerland. IEEE, 48-57.

Elab (2007) Door. Lungano, Switzerland: eLab. Available from: http:///door.sourceforge.net [Accessed 25th October 2007].

Etzioni, O. and Weld, D. S. (1995) Intelligent agents on the internet: Fact, fiction and forecast. IEEE Expert, 10 (3), 44-49.

Fjell, A. M. and Walhovd, K. B. (2004) Thinking styles in relation to personality traits: An investigation of the thinking styles inventory and neo-pi-r. *Scandinavian Journal of Psychology,* 45, 293-300.

Florea, A. M. and Kalisz, E. (2001) Anticipatory attributes of agent behaviour in MAS. 5th International Conference on Computing Anticipatory Systems (CASYS 2001), Liege Belgium. 359-364.

Flynn, T. (2004) Personality tests and tools. Available from: http://similarminds.com/personality_tests.html [Accessed 4th July 2005].

Franklin, S. and Graesser, A. (1996) Is it an agent, or just a program? A taxonomy for autonomous agents. *3rd International Workshop on Agent Theories, Architectures, and Languages, published as Intelligent Agents III*, 12th-13th August 1996 Budapest, Hungary. Springer-Verlag, 21-35.

Franklin, S. P. (1995) *Artificial minds.* Cambridge: The MIT Press.

Fruyt, F. D., McCrae, R. R., Szirmák, Z. and Nagy, J. (2004) The five-factor personality inventory as a measure of the five-factor model: Belgian, American, and Hungarian comparisons with the neo-pi-r. Assessment, 11 (3), 207-215.

Georges, T. M. (2003) Digital soul. Boulder, USA: Westview Press.

Ghanea-Hercock, R. (2003) *Applied evolutionary algorithms in java.* New York: Springer-Verlag.

Gillham, B. (2000) The research interview. London: Continuum.

Gollwitzer, P. M. and Bargh, J. A. (eds.) (1996) The psychology of action. Linking cognition and motivation to behaviour. New York: The Guilford Press.

Graham, J. R. and Decker, K. S. (2000) Towards a distributed, environment- centred agent framework. *In:* Jennings, N. R. and Lesperance, Y. (eds.) *Intelligent agents vi.* Berlin: Springer-Verlag, 290-304.

Granta, T. and Dajeeb, K. (2003) Types of task, types of audience, types of actor: Interactions between mere presence and personality type in a simple mathematical task. Personality and Individual Differences, 35, 633-639.

Gratch, J., Rickel, J., André, E., Cassell, J., Petajan, E. and Badler, N. (2002) Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems*, 54-63.

Gratch, J. and Marsella, S. (2004) A domain-independent framework for modelling emotion. Cognitive Systems Research, 5, 269-306.

Greenberg, G. (2003) Electronic portfolio white paper. Version 1.0. ePortConsortium Available from: http://www.eportconsortium.org/Uploads/whitepaperV1_0.pdf [Accessed 31st January 2006].

Hammond, K. J., Converse, T. M. and Grass, J. W. (1995) The stabilization of environments. Artificial Intelligence, 72 (1-2), 305-327.

Hare, M. and Deadman, P. (2004) Further towards a taxonomy of agent-based simulation models in environmental management. *Mathematics and Computers in Simulation,* 64 (1), 25-40.

Hatala, M., Richards, G., Eap, T. and Willms, J. (2004) The interoperability of learning object repositories and services: Standards, implementations and lessons learned. 13th international World Wide Web conference, 19th-27th May 2004 New York. ACM Press, 19-27.

HEC (2003) Time2learn. D2.2 needs assessment for future professional training in Europe. HEC. Available from: http://www.time2learn.org/images/PagesUn/T2L_D2_2_final_EC.pdf [Accessed 13th June 2005].

HEFCE (2010) Higher education funding council for England. UK:HEFCE. Available from: http://www.hefce.ac.uk/ [Accessed 4th December 2010].

Henninger, A. E., Jones, R. M. and Chown, E. (2003) Behaviours that emerge from emotion and cognition: Implementation and evaluation of a symbolic-connectionist architecture. *2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03),* Melbourne Australia. ACM Press, 321-328.

Hess, B. (1999) Graduate student cognition during information retrieval using the world wide web: A pilot study. Computers & Education, 33 (1), 1-13.

Hexmoor, H., Huber, M., Muller, J. P., Pollock, J. and Steiner, D. (2000) On the evaluation of agent architectures. In: Jennings, N. R. and Lesperance, Y. (eds.) Intelligent agents vi. Berlin: Springer-Verlag, 106-116.

Hills, H. (2003) *Individual preferences in eLearning.* Aldershot, UK: Gower.

Huang, D. W., Diefes-Dux, H., Imbrie, P. K., Daku, B. and Kallimani, J. G. (2004) Learning motivation evaluation for a computer-based instructional tutorial using arcs model of motivational design. 34th ASEE/IEEE Frontiers in Education Conference, 20th–23rd October 2004 Savannah GA. IEEE, 30-36.

Huang, Z., Eliens, A., Van-Ballegooij, A. and De-Bra, P. (2000) A taxonomy of web agents. 11th International Workshop on Database and Expert Systems Applications (DEXA'00), 6th-8th September 2000 London. 765-769.

Illeris, K. (ed.) (2009) *Contemporary theories of learning. Learning theorists ... In their own words.* Oxon, UK: Routledge.

Internet Mail Consortium (2009) Personal data interchange. Available from: http://www.imc.org/pdi/ [Accessed 18th January 2009].

Isbister, K. and Doyle, P. (2002) Design and evaluation of embodied conversational agents: A proposed taxonomy. *1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '02)*, 15th-19th July 2002 Bologna, Italy. 15-19.

Isbister, K. and Nass, C. (2000) Consistency of personality in interactive characters: Verbal cues, non-verbal cues, and user characteristics. *International Journal of Human-Computer Studies*, 53, 251-267.

Ishaya, T. and Wood, D. H. (2005) Enhancing effective ePortfolios through agents. *5th IEEE International Conference on Advanced Learning Technologies (ICALT2005)*, 5th-8th July 2005 Kaohsiung Taiwan. IEEE, 409-413.

Isozaki, H. and Katsuno, H. (2000) Observability-based nested belief computation for multiagent systems and its formalization. In: Jennings, N. R. and Lesperance, Y. (eds.) Intelligent agents vi. Berlin: Springer-Verlag, 27-41.

Java (2006) Java technology. Santa Clara: Sun Microsystems. Available from: http://java.sun.com/ [Accessed 19th April 2006].

Jennings, N. R. and Lesperance, Y. (eds.) (2000) *Intelligent agents vi*. Berlin: Springer-Verlag.

JISC (2006) E-portfolios. JISC. Available from: http://www.jiscinfonet.ac.uk/InfoKits/effective-use-of-VLEs/e-portfolios/printable_version.pdf [Accessed 14th March 2009].

JISC (2010a) JISC: Inspiring innovation. UK:JISC. Available from: http://www.jisc.ac.uk/ [Accessed 4th December 2010].

Joachims, T., Mitchell, T., Freitag, D. and Armstrong, R. (1995) Webwatcher: Machine learning and hypertext. Beiträge zum, 7.

Kakas, A. and Moratis, P. (2003) Argumentation based decision making for autonomous agents. 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03), Melbourne Australia. ACM Press, 883-890.

Kapetanakis, S., Kudenko, D. and Strens, M. J. A. (2003) Reinforcement learning approaches to coordination in cooperative multi-agent systems. *In:* Alonso, E., Kudenko, D. and Karzakov, D. (eds.) *Adaptive agents and multi-agent systems.* Berlin: Springer-Verlag, 18-32.

Kazi, S. A. (2004) A conceptual framework for web-based intelligent learning environments using scorm-2004. IEEE International Conference on Advanced Learning Technologies (ICALT'04), 30th August-1st September 2004 Joensuu Finland. IEEE, 12-15.

Kim, E. B. and Schniederjans, M. J. (2004) The role of personality in web-based distance education courses. *Communications Of The ACM*, 47 (3), 95-98.

Kim, W., Kerschberg, L. and Scime, A. (2002) Learning for automatic personalization in a semantic taxonomy-based meta-search agent. Electronic Commerce Research and Applications, 1 (2), 150-173.

Klusch, M. (2001) Information agent technology for the internet: A survey. *Data & Knowledge Engineering*, 36, 337-372.

Kornør, H. and Nordvik, H. (2004) Personality traits in leadership behaviour. Scandinavian Journal of Psychology, 45, 49-54.

Kreijns, K., Kirschner, P. A. and Jochems, W. (2003) Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: A review of the research. Computers in Human Behaviour, 19, 335-353.

Kuo, Y.-F. and Chen, L.-S. (2001) Personalization technology application to internet content providers. *Expert Systems with Applications,* 21 (4), 203-215.

Lang, T. B. (1999) Software agents in education. Available from: http://www.computerteacher.org [Accessed 4th March 2004].

Lange, D. B. and Oshima, M. (1998) Programming and developing java mobile agents with aglets. Massachusetts, USA: Addison Wesley Longman.

Lee, J. (2000) Reactive-system approaches to agent architectures. *In:* Jennings, N. R. and Lesperance, Y. (eds.) *Intelligent agents vi.* Berlin: Springer-Verlag, 132-146.

Lee, M. C., Tsai, K. H., Ye, D. Y. and Wang, T. I. (2006) A service-based framework for personalized learning objects retrieval and recommendation. *In:* Liu, W., Lau, R. W. H. and Li, Q. (eds.) *Advances in web-based learning - ICWL 2006. Fifth international conference, Penang, Malaysia, July 2006 proceedings.* Berlin: Springer-Verlag, 336-351.

Leide, J. E., Large, A., Beheshti, J., Brooks, M. and Cole, C. (2003) Visualization schemes for domain novices exploring a topic space: The navigation classification scheme. Information Processing and Management, 39, 923-940.

Li, G., Hopgood, A. A. and Weller, M. J. (2003) Shifting matrix management: A model for multi-agent cooperation. Engineering Applications of Artificial Intelligence, 16 (3), 191-201.

Lin, N. H., Shih, T. K., Hsu, H.-H., Chang, H.-P., Chang, H.-B., Ko, W. C. and Lin, L. J. (2004) Pocket SCORM. *24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04)*. IEEE.

Little, B. R., Lecci, L. and Watkinson, B. (1992) Personality and personal projects: Linking big five and PAC units of analysis. *Journal of Personality,* 60 (2).

Liu, H. and Maes, P. (2004) What would they think? A computational model of attitudes. 9th International Conference on Intelligent User Interfaces (IUI'04), Madeira Funchal Portugal. ACM Press, 38-45.

Loehln, J. C. (1968) *Computer models of personality.* New York: Random House.

Luck, M., McBurney, P. and Preist, C. (2003) Agent technology: Enabling next generation computing. A roadmap for agent based computing. Liverpool: AgentLink. Available from: http://www.agentlink.org/admin/docs/2003/2003-48.pdf [Accessed 14th June 2004].

Martin, P. and Bateson, P. (1993) Measuring behaviour: An introductory guide. 2nd ed. Cambridge University Press.

Martínez-Miranda, J. and Aldea, A. (2005) Emotions in human and artificial intelligence. Computers in Human Behaviour 21, 323-341.

Meyer, J.-J. C. (1998) Agent languages and their relationship to other programming paradigms. In: Muller, J. P., Singh, M. P. and Rao, A. S. (eds.) Intelligent agents v, agent theories, architectures and languages. Berlin: Springer-Verlag, 309-330.

Moffat, D. (1997) Personality parameter and programs. *In:* Trappl, R. and Petts, P. (eds.) *Creating personalities for synthetic actors. Towards autonomous personality agents.* Berlin: Springer -Verlag, 120-165.

Moodle (2006) Moodle - a free, open source course management system for online learning. Moodle. Available from: http://moodle.org/ [Accessed 13/04/2006].

Morimoto, Y., Ueno, M., Yonezawa, N., Yokoyama, S. and Miyadera, Y. (2004) A meta-language for portfolio assessment. *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 30th August-1st September 2004 Joensuu Finland. IEEE, 46-50.

Mukaidono, M. (2001) *Fuzzy logic for beginners.* Singapore: World Scientific.

Muller, H. J. (1997) Towards agent systems engineering. *Data & Knowledge Engineering,* 23, 217-245.

Muller, J. P., Singh, M. P. and Rao, A. S. (eds.) (1998) *Intelligent agents v: Agent theories, architectures, and languages.* New York: Springer.

Mylonas, P., Tzouveli, P. and Kollias, S. (2004) Towards a personalized e-learning scheme for teachers. IEEE International Conference on Advanced Learning Technologies (ICALT'04). IEEE.

National University of Ireland (2002) Non-traditional collaborative filtering techniques. Galway National University of Ireland, NUIG-IT-121002.

Nemani, S. S. and Allan, V. H. (2003) Agents and the algebra of emotion. *2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03)*, Melbourne Australia. ACM Press, 1076-1077.

Oliveira, E. and Sarmento, L. (2003) Emotional advantage for adaptability and autonomy. *2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03)*, Melbourne Australia. ACM Press, 305-312.

Olver, J. M. and Mooradian, T. A. (2003) Personality traits and personal values: A conceptual and empirical integration. Personality and Individual Differences, 35, 109-125.

Omicini, A., Zambonelli, F., Klusch, M. and Tolksdorf, R. (eds.) (2001) *Coordination of internet agents: Models, technologies and applications.* Berlin: Springer-Verlag.

Ortony, A., Clore, G. L. and Collins, A. (1988) *The cognitive structure of emotions.* Cambridge University Press.

Ossowski, S. and García-Serrano, A. (1998) Social structure in artificial agent societies: Implications for autonomous problem-solving agents. *In:* P. Müller, J., Singh, M. P. and Rao, A. S. (eds.) *Intelligent agents v: Agent theories, architectures, and languages.* New York Springer, 133-148.

Panayiotou, G., Kokkinos, C. M. and Spanoudis, G. (2004) Searching for the "big five" in a Greek context: The NEO-FFI under the microscope. Personality and Individual Differences, 36, 1841-1854.

Panteleyev, M. G., Puzankov, D. V., Sazykin, P. V. and Sergeyev, D. A. (2002) Intelligent educational environments based on the semantic web technologies. IEEE International Conference on Artificial Intelligence Systems (ICAIS'02). IEEE.

Paranagama, P., Burstein, F. and Arnott, D. (1998) Adaptor: A personality-based adaptive DSS generator.

Paretti, M. C. (2004) Work in progress: Using e-portfolios to assess communication skills. *34th ASEE/IEEE Frontiers in Education Conference*, 20th–23rd October 2004 Savannah GA. IEEE, 26-27.

Paris, M. (2004) Reuse-based layering: A strategy for architectural frameworks for learning technologies. *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*. IEEE, Perlman, B. and Varma, R. (2001) Bridging the cultural chasm: Improving collaboration and cooperation between the computer and social sciences. International Symposium on Technology and Society (ISTAS'01), 06-07 July 2001 Stamford USA. IEEE.

Petta, P. and Trappl, R. (1997a) Personalities for synthetic actors: Current issues and some perspectives. *In:* Trappl, R. and Petts, P. (eds.) *Creating personalities for synthetic actors. Towards autonomous personality agents.* Berlin: Springer -Verlag, 207-218.

Petta, P. and Trappl, R. (1997b) Why to create personalities for synthetic actors? *In:* Trappl, R. and Petts, P. (eds.) *Creating personalities for synthetic actors. Towards autonomous personality agents.* Berlin: Springer -Verlag, 1-8.

Pitt, J., Mamdani, A. and Charlton, P. (2001) The open agent society and its enemies: A position statement and research programme. Telematics and Informatics, 18, 67-87.

Plaza, E. and Ontanon, S. (2003) Cooperative multiagent learning. *In:* Alonso, E., Kudenko, D. and Kazakov, D. (eds.) *Adaptive agents and multi-agent systems: Adaption and multi-agent learning.* Berlin: Springer-Verlag, 1-17.

Pollock, J. L. (2000) Rational cognition OSCAR. In: Jennings, N. R. and Lesperance, Y. (eds.) Intelligent agents vi. Berlin: Springer-Verlag, 71-90.

Polsani, P. R. (2004) Why learning objects? *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*. IEEE.

PostgreSQL (2006) PostgreSQL: The world's most advanced open source database. PostgreSQL Global Development Group. Available from: http://www.postgresql.org/ [Accessed 15/04/2006].

Pountain, D. (2001) The new penguin dictionary of computing. London: Penguin Books.

Power, G., Davis, H. C., Cristea, A. I., Stewart, C. and Ashman, H. (2005) Goal oriented personalisation with SCORM. *5th International Conference on Advanced learning Technologies (ICALT 2005),* 5th-8th July 2005 Kaohsiung Taiwan. IEEE, 467-472.

Raghu, T. S., Kannan, P. K., Rao, H. R. and Whinston, A. B. (2001) Dynamic profiling of consumers for customized offerings over the internet: A model and analysis. Decision Support Systems, 32 (2), 117-134.

Riera, A., Lama, M., Sánchez, E., Amorim, R., Vila, X. A. and Barro, S. (2004) Study of communication in a multi-agent system for collaborative learning scenarios. *12th Euromicro Conference on Parallel, Distributed and Network-Based Processing (EUROMICRO-PDP'04)*. IEEE, 223-230.

Rizzo, P., Veloso, M., Miceli, M. and Cesta, A. (1997) Personality-driven social behaviors in believable agents. *Fall Symposium Series: Socially Intelligent Agents AAAI-97*. 109-114.

Ronchetti, M. and Saini, P. (2004) Knowledge management in an e-learning system. *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*. IEEE, 365-369.

Rose, J. and Hensher, D. A. (2004) Modelling agent interdependency in group decision making. Transportation Research Part E: Logistics and Transportation Review, 40 (1), 63-79.

Rousseau, D. and Hayes-Roth, B. (1996) Personality in synthetic agents. Stanford: Stanford University. Available from: http://citeseer.ist.psu.edu/rousseau96personality.html [Accessed 26th January 2006].

Rousseau, D. and Moulin, B. (1997) Mixed initiative in interactions between software agents. AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction, 24-26 March 1997 Stanford USA. Stanford University, 135-137.

Saad, A. (2005) A multi-agent spreading activation network model for online learning objects. Available from: http://julita.usask.ca/mable/saad.pdf [Accessed 24th February 2005].

Saggino, A. (2000) The big three or the big five? A replication study. Personality and Individual Differences, 28, 879-886.

Salmon, G. (2002) E-tivities. The key to active online learning. London: Kogan Page.

Salmon, G. (2002) E-moderating, the key to teaching and learning online. 2nd ed. London: Kogan Page.

Santally, M. I. and Senteni, A. (2005) A learning object approach to personalized web-based instruction. *European Journal for open, distance and eLearning (EURODL)*.

Schillo, M. (2002) Self-organization and adjustable autonomy: Two sides of the same coin? Connection Science, 14 (4), 345-359.

Sengers, P. (2002) Schizophrenia and narrative in artificial agents. LEONARDO, 35 (4), 427-431.

Sharafi, P., Hedman, L. and Montgomery, H. (2004) Using information technology: Engagement modes, flow experience, and personality orientations. Computers in Human Behaviour Article in press.

Sharon, T., Lieberman, H. and Selker, T. (2003) A zero-input interface for leveraging group experience in web browsing. 8th International Conference on Intelligent User Interfaces, Florida USA. 290-292.

Sloman, A. (1997) What sort of control system is able to have a personality. *In:* Trappl, R. and Petts, P. (eds.) *Creating personalities for synthetic actors. Towards autonomous personality agents.* Berlin: Springer -Verlag, 166-208.

Song, C.-R. (2002) Literature review for hypermedia study from an individual learning differences perspective. British Journal of Education Technology, 33 (4), 435-447.

Sosteric, M. and Hesemeier, S. (2004) A first step towards a theory of learning objects. *In:* McGreal, R. (ed.) *Online education using learning objects.* Oxon, UK: Routledge, 32-42.

Stergioulas, L. K., Minetou, C., Koskinen, T., Klein, M., Dang, J., Denual, Y., Kamtsiou, V. and Makropoulos, K. (2004) Roadmap for the future of eTraining/eLearning technologies in Europe. IEEE International Conference on Advanced Learning Technologies (ICALT'04). IEEE, 460-464.

Stone, P. (2000) *Layered learning in multiagent systems.* Massachusetts, USA: Massachusetts Institute of Technology.

Stuurman, K. and Wijnands, H. (2001) Intelligent agents: A curse or a blessing? A survey of the legal aspects of the application of intelligent software systems. Computer Law & Security Report, 17 (2), 92-100.

Supjarerndee, S., Temtanapat, Y. and Phalavonk, U. (2002) Recruitment filtering with personality-job fit model. International Conference on Information Technology: Coding and Computing (ITCC'02), 08-10 April 2002 Las Vegas USA. IEEE, 46-51.

Sutherland, S. (2005) E-portfolios: A personal space for learning and the learner voice. *In:* Freitas, S. D. and Yapp, C. (eds.) *Personalizing learning in the 21st century.* Stafford, UK: Network Education Press, 79-82.

Sycara, K. (2001) The intelligent software agents lab -the robotics institute. Pittsburgh: Carnegie Mellon University. Available from: http://www-2.cs.cmu.edu/~softagents/index.html [Accessed 14th June 2004].

Trappl, R. and Petts, P. (eds.) (1997) Creating personalities for synthetic actors. Towards autonomous personality agents. Berlin: Springer -Verlag.

Triantis, A. and Pintelas, P. (2004) A mobile multi agent architecture for web based learning. *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, 30th August-1st September 2004 Joensuu Finland. IEEE, 51-55.

Turing, A. M. (1950) Computing machinery and intelligence. *Mind*, 59, 433-460.

Valderrama, R. P., Ocaña, L. B. and Sheremetov, L. B. (2005) Development of intelligent reusable learning objects for web-based education systems. *Expert Systems with Applications* 28, 273-283.

Vel, O. D. and Nesbitt, S. (2004) A collaborative filtering agent system for dynamic virtual communities on the web. Available from: http://citeseer.nj.nec.com/158278.html [Accessed 12th July 2004].

Viswanath, K., Adebiyi, B., Leelawong, K. and Biswas, G. (2004) A multi-agent architecture implementation of learning by teaching systems. *IEEE International Conference on*

Advanced Learning Technologies (ICALT'04), 30th August-1st September 2004 Joensuu Finland. IEEE, 61-65.

Wallace, S. A. and Laird, J. E. (2000) Toward a methodology for AI architectures evaluation: Comparing soar and clips. In: Jennings, N. R. and Lesperance, Y. (eds.) Intelligent agents vi. Berlin: Springer-Verlag, 117-131.

Webber, C., Bergia, L., Pesty, S. and Balacheff, N. (2001) The Baghera project: A multi-agent architecture for human learning. Available from: http://julita.usask.ca/mable/webber.pdf [Accessed 24th February 2005].

Webct (2006) Webct.Com. Washington: Blackboard. Available from: http://www.webct.com/ [Accessed.

Welle, M., Pegler, C. and Mason, R. (2005) Use of innovative technologies on an e-learning course. *Internet and Higher Education,* 8, 61-71.

Woelk, D. and Lefrere, P. (2002) Technology for performance-based lifelong learning. International Conference on Computers in Education (ICCE'02), 3rd-6th December 2002 Auckland New Zealand. IEEE, 92-96.

Wood, D. H. and Ishaya, T. (2004) A taxonomy of agents for global use. *PREP 2004*, Hertfordshire UK. University of Hertfordshire, 164-165.

Wooldridge, M. (2000) *Reasoning about rational agents.* Cambridge: The MIT Press.

Wooldridge, M. (2002) *An introduction to multiagent systems.* Chichester, UK: John Wiley & Sons.

Wooldridge, M. and Jennings, N. R. (1994) Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10 (2).

Woszczynski, A. B., Roth, P. L. and Segars, A. H. (2002) Exploring the theoretical foundations of playfulness in computer interactions. Computers in Human Behaviour, 18, 369-388.

Wray, R., Chong, R., Phillips, J., Rogers, S. and Walsh, B. (1994) A survey of cognitive and agent architectures. USA: University of Michigan. Available from: http://ai.eecs.umich.edu/cogarch0/ [Accessed 13th June 2004].

Zhou, X. and Conati, C. (2003) Inferring user goals from personality and behaviour in a causal model of user affect. 8th International Conference on Intelligent User Interfaces (IUI'03), Miami Florida USA. ACM Press, 211-218.

Appendix A: Study one – observation

A.1: Tagged behaviours

TIME	EVENT	KEY	MODIFIER	BEHAVIOUR	CODE	DESCRIPTION	APLICATION
02:44.3	Marker			open: file (ValidTriangle1.java)	F	File/Application Navigation	
02:44.3	Marker			thinking: sample code (ValidTriangle1.java)	U	Read Code/Errors/Thinking	
02:57.8	Marker			open: new file	F	File/Application Navigation	
03:19.6	Keystrokes	/		type: comment			SciTE.exe
03:32.4	Marker			thinking: sample code (ValidTriangle1.java)	U	Read Code/Errors/Thinking	
03:34.4	Keystrokes	p		type: class declaration (complete)			SciTE.exe
03:34.6	Keystrokes	u		type: modifier public			SciTE.exe
03:36.0	Keystrokes	c		type: class			SciTE.exe
03:46.2	Marker			read: tasks	R	Read Questions	
03:53.4	Keystrokes	c	Ctrl	copy: class name (tasks)			firefox.exe
03:54.6	Marker			paste: class name (AverageMark)	V	Copy Paste Code	
03:56.1	Marker			thinking: sample code (ValidTriangle1.java)	U	Read Code/Errors/Thinking	
04:00.4	Keystrokes	{	Shift	type: brace pair (class)			SciTE.exe
04:07.2	Keystrokes	/		type: comment			SciTE.exe
04:30.5	Marker			use: previous code (ValidTriangle1.java)	C	Uses Previous Code	
04:31.8	Marker			paste: main method declaration (complete)	V	Copy Paste Code	
04:34.5	Marker			edit: main method declaration (incorrect - previous)	T	Delete Code	
04:35.0	Keystrokes	BS		delete: parameter declaration (incorrect)			SciTE.exe
04:40.6	Keystrokes	}	Shift	type: right brace (main)			SciTE.exe

A.2: Self-assessed Java questionnaire

Phase 1, Post-observation questionnaire

Please enter your participant code here:

Questionnaire purpose

This question is designed to gather the dynamic aspects of your personality and will be used to populate the personality profile. This is not a test and will be kept anonymous . Please answer as accurately as you can. Please enter your code before starting.

A. Demographic questions

Q1. What is your gender?

Male Female

Q2. How old are you?

Q3. What country are you from?

Q4. What is your highest attained education level?

GCSE or O' level	Bachelor degree
AS level	Masters degree
A' level or A2 level	PhD
HNC/HND	Other

Q5. What is the title of your course?

Q6. If you are studying at University, what year of study are you currently in?

Not at University	Year 2	Year 4
Year 1	Year 3	Other

B. Programming Experience

Q7. How long have you been programming computers?

Years Months

Q8. How long have you been programming in the Java Language?

Years: Months

Q9. Which of the following programming languages do you Know of, use Some times, use Frequently:

C	Lisp	Prolog
C#	Oberon	Python
C++	Pascal	Ruby
Java	Perl	Visual Basic
JavaScript	PHP	

Q10. How would you rate your general programming ability? (where 1=low and 5=high).

Very Poor Average Very Good

Q11. How would you rate your Java programming ability? (where 1=low and 5=high).

Very Poor Average Very Good

Q12. Before the observation study, how long has it been, since you last spent more than 2 hours coding?

Years Months

Q13. Did you use Java in your last coding activity, before the observation study?

Yes No … if No go to Q14.

Q14. What language did you use?

C. Learning Programming

Q15. During term time, how many hours per week do you spend practicing Java programming, in your spare time?

under 5	11-20	31-40
6-10	21-30	Over 41

Q16. If you were having difficulty with a programming task, and need help, which of the following is the first place, (choose 1 only), you would try, and what other places, (choose 1 to many), would you try if you needed more help

a. The lecturer
b. Ask the lab assistant
c. Search on the Internet,
d. Read the course text book
e. Read other text books
f. Ask other students
g. Review lecture notes
h. Review the API's
i. Ask friends (outside university)
j. Never ask for help

Q17. When you are learning a new concept in programming which of the following methods do you use most often (choose 1 only) and which others, (choose 1 to many), do you also use?

a. Think about it
b. Read other text books
c. Read the coarse text book
d. Review your lecture notes
e. Practice the concept with code
f. Discuses it with other students
g. Research it on the Internet
h. Do nothing

Q18. How would you rate your understanding of the following programming concepts? (where 1=low and 5=high)

Very Poor Average Very Good

a. The use of Pseudo-code
b. Variable declaration and assignment
c. How to create an instance of a class
d. How to call a method on an object
e. How to call a static method from another class
f. How to write a method that returns a value
g. What a primitive type is
h. The function of a class constructor
i. How to pass parameters into methods
j. What the "main" method is for
k. The difference between global and local variables
l. When static should be applied to variables and methods
m. The purpose of public, protected and private

Q19. How important is too you, to be a good programmer? (where 1=low and 5=high)

Not Important Average Very Important

Q20. How important is feedback on your lab work, too you? (where 1=low and 5=high).

Not Important Average Very Important

D. Programming Preferences

Q21. Given a set of tasks to complete in a lab, which of the following are you most likely to do?

a. Pick the easiest one to do first
b. Pick the one that interests you the most
c. Randomly pick one
d. Do the first one first
e. Do the hardest one first

Q22. Which of the following best describes your use of commenting in code?

a. I frequently comment my code
b. I always write header comments, but nothing more
c. I only write comments if the work is assessed
d. I very rarely comment my code

Q23. How important is code formatting, too you? (where 1=low and 5=high).

Not Important Average Very Important

Q24. Do you have a coding convention regarding variable/class/method names?

Yes … if Yes go to Q25 No

Q25. Briefly describe or show examples of your coding convention?

A.3: Java questionnaire results

The following presents the raw data gathered from the self-assessed java questionnaire. Values indicate the number of participants that responded to the question with a particular response.

Part A. Participant demographics (Q1 to Q6)

Q1. Gender	
Male	Female
8	2

Q2. Age	
25 & under	over 25
7	3

Q3. Home Country	
UK	Other
9	1

Q4. Current Education Level	
GCSE	2
AS/A2	4
HND	1
Bachelor	2
Masters	1

Q5. Studying for	
Bachelor	7
Masters	1
PhD	2

Q6. Study Year	
1	0
2	7
3	1
4	1
other	1

Part B. Programming experience (Q7-Q14)

Q7-Q8 Total programming experience

Time	Q7. General Programming	Q8. Java Programming
less than 6 months	0	0
6-24 (inclusive) months	1	5
2-5 (inclusive) years	3	4
5-10 (inclusive) years	2	1
10-20 (inclusive) years	2	0
greater than 20 years	2	0

Q9 Understanding and using programming languages

Language	Know	Sometimes	Frequently	Language	Know	Sometimes	Frequently
C	7	0	2	C#	6	0	0
C++	8	0	1	Java	2	5	5
JavaScript	4	0	6	Lisp	4	0	0
Oberon	5	0	0	Pascal	7	0	3
Perl	8	0	2	PHP	3	5	2
Prolog	4	0	0	Python	6	0	2
Ruby	6	1	1	Visual Basic	7	0	2

Q10-Q11 Self-assessed programming ability

	Very Poor (1)	(2)	Average (3)	(4)	Very Good (5)
Q10. General programming	0	2	6	2	0
Q11. Java programming	0	3	4	3	0

Q12-Q14 Last programming experience before the observation study

Q12. Last 2 hour session		Q13. Session used Java		Q14. Other Language used	
1 month or less	6	**Yes**	6	PHP	3
1-3 (inclusive) months	1	**No**	3		
3-6 (inclusive) months	1	**null**	1		
6-12 (inclusive) months	0				
a Year or more	2				

Part C. Learning programming (Q15-Q20)

Q15 How often they practice programming

Hours per week					
under 5	6-10	11-20	21-30	31-40	0v34 40
5	4	1	0	0	0

Q16 Where to go for help

	First place	Other places
Ask the lecturer	0	7
Search on the Internet	4	5
Read other text books	2	4
Review lecture notes	2	7
Ask friends (outside university)	0	6
Ask the lab assistant	0	8
Read the course text book	0	4
Ask other students	0	9
Review the API's	2	7
Never ask for help	0	

Q17 Learning a new programming concept

	First place	Other places
Think about it	1	6
Read the course text book	1	3
Practice the concept with code	2	7
Research it on the Internet	4	5
Read other text books	1	5
Review your lecture notes	1	5
Discusses it with other students	0	6
Do nothing	0	

Q18 Self-assessed understanding of concepts

	Very poor (1)	(2)	Average (3)	(4)	Very good (5)
The use of Pseudo-code	0	0	4	4	2
Variable declaration and assignment	0	0	0	4	6
How to create an instance of a class	1	3	0	2	5
How to call a method on an object	0	3	1	2	4
How to call a static method from another class	1	3	1	2	3
How to write a method that returns a value	0	0	3	2	5
What a primitive type is	0	3	0	2	5
The function of a class constructor	1	3	0	2	4
How to pass parameters into methods	0	0	3	3	4
What the "main" method is for	0	0	2	2	6
The difference between global and local variables	0	1	1	4	4
When static should be applied to variables and methods	1	2	2	3	2
The purpose of public, protected and private.	0	1	1	4	4

Q19-Q20 How important

	Not important (1)	(2)	Average (3)	(4)	Very important (5)
Q19. To be a good programmer	1	1	3	3	2
Q20. Getting feedback	0	0	1	3	6

Part D. Programming preferences (Q21-Q25)

Q21-Q24 Programming preferences

Q21. Task order		Q22. Use of commenting		Q23. Importance of code formatting		Q24. Coding convention	
Pick the easiest one to do first	0	I frequently comment my code	0	Not Important (1)	5	Yes	3
Pick the one that interests you the most	2	I always write header comments, but nothing more	3	(2)	0	No	6
Randomly pick one	0	I only write comments if the work is assessed.	6	Average (3)	5		
Do the first one first	8	I very rarely comment my code.	0	(4)	0		
Do the hardest one first	0	null	1	Very Important (5)	0		

Q25 Descriptions of coding conventions

Participant	Description
PA01	public convertDate(paramInputDate){WesternDate returnDate
PB03	Class TestForLoop { public static void main (String[] arguments) { for (int i
PC03	Variables: 'exampleVariablePassing 'example Variable' to a method the receiving methods version of the variable would be called 'inExampleVariable'

A.4: Conversation table and methods

Dynamic Aspect	Questions and behaviour description	Original data type	Method
Values	Q15 - Practicing code	Selection 1 from 6	Method 2
	Q22 - Using comments	Selection 1 from 4	Method 2
	Q23 - Formatting code	Scale 1-5	Method 1
Abilities	Q10 - General programming	Scale 1-5	Method 1
	Q11 – Java programming	Scale 1-5	Method 1
	Q9a - Know of languages	Select many from14	Method 3
	Q9b - Understanding languages	Select many from 14	Method 3
	Q9c - Using languages	Select many from 14	Method 3
	Q18a – Use pseudo code	Scale 1-5	Method 1
	Q18b – Variable declaration	Scale 1-5	Method 1
	Q18c – Instance of class	Scale 1-5	Method 1
	Q18d – Method call	Scale 1-5	Method 1
	Q18e – Static method call	Scale 1-5	Method 1
	Q18f - Write returning method	Scale 1-5	Method 1
	Q18g – Primitive types	Scale 1-5	Method 1
	Q18h - Class constructor	Scale 1-5	Method 1
	Q18i - Parameter passing	Scale 1-5	Method 1
	Q18j- Understand main method	Scale 1-5	Method 1
	Q18k – Understand variable scope	Scale 1-5	Method 1
	Q18l – Static methods	Scale 1-5	Method 1
	Q18m – Protection modifiers	Scale 1-5	Method 1
Attitudes	Q20 - Getting feedback	Scale 1-5	Method 4
	Q16b - Help form lecturer	Select or not select	Method 5
	Q16b - Help from lab assistant	Select or not select	Method 5
	Q16b - Help from the Internet	Select or not select	Method 5
	Q16b - Help from course text	Select or not select	Method 5
	Q16b - Help from none course text	Select or not select	Method 5
	Q16b - Help from peers	Select or not select	Method 5
	Q16b - Help from notes	Select or not select	Method 5
	Q16b - Help from API	Select or not select	Method 5
	Q16b - Help from none peer friends	Select or not select	Method 5
	Q17b - Learning via thinking	Select or not select	Method 5
	Q17b - Learning via practice	Select or not select	Method 5
Goals	Q19 - Good programmer	Scale 1-5	Method 1
Preferences	Q16a - Needing help	Select 1 from 9	Method 6
	Q17a - Learning new concept	Select 1 from 7	Method 6

Method 1 – Where scale data from one to five was translated into a range from one to a hundred. The scale was rearranged to align negative or undesirable ratings with zero. Then each rating score was multiplied by twenty. For example Q10 on general programming ability a self-score of two became an ability score of forty.

Method 2 – Similar to method one in that a scale is transferred to a value between one and hundred, but this time the original scale numbers are not discreet. One hundred was divided by the number of items in the scale, and rounded up to the nearest five or ten. For example with Q15, practice code, the selection was from six ranges of hours spent practicing. It is assumed that the more time a participant spent practicing code the more they valued the activity. Each of the six levels were incremented in values of fifteens, so if they had selected the third option 11-20 hours, then their value score would be forty five.

Method 3 – Generic abilities such as knowledge of programming languages can be assessed by asking the participants to select from a range of languages those that they are aware of or have used. This was done with Q9a-c. To translate these selections into values the PR can use the number of selections made were simple counted and used to calculate the percentage of selections from the whole. The result was rounded to the nearest whole number and set as the ability level value.

Method 4 – As discussed above attitudes can only contain one of three values which represent whether the participant is positive, negative or indifferently disposed towards the focus behaviour. Those attitudes that were derived from scale data required the five step scale to become three. So the scale values one and two become equal to a negative attitude represented by one. Scale values four and five as positive and represented by three. The middle value of the scale, three, equates to indifference and becomes the attitude value two.

Method 5 – The remaining attitudes were extracted from Q16b and Q17b. If the subject was selected it was deemed that the participant had a positive, value three, towards that subject. If not then the attitude towards that subject was set to one, negative. Not all the subjects listed in

Q17 were entered into the profile as some proved to be irrelevant in the context of the SOs, or difficult to identify by the system when monitoring the users interactions.

Method 6 – It was intended that preferences would be derived from an ordered list provided by the participant. The questionnaire would present a grid of radio buttons labelled one to ten, and ten items to be ordered. This method was piloted in an earlier questionnaire and proved to be unreliable to capture this data, particularly on large groups of items to be ordered. For the dynamic aspect questionnaire it was decided to ask the participants to only select the most important subject from a list, followed by all other subjects they that felt had secondary relevance. This effectively created a preference list that contained a single top preference, followed by an unordered group of secondary preferences, and ending with a final group of unordered least preferred subjects.

A.5: Task set one

Introduction:

The following tasks have been designed to simulate a Java Programming Lab at the Centre for Internet Computing. They are not intended to test your programming ability in any way, but to allow you to show us your behaviours and preferences while performing programming tasks.

Instructions:

Please treat this session as you would any other lab exercise. You are free to use any of the resources you would usually use, including the lab officer. If you find some resource you usually use is missing please state this vocally. As you start each task please state aloud which task you are starting.

You may now begin.

Task 1:

Lisa, Bart, Ralf, Nelson and Milhouse sit an exam and achieve the following marks respectively (expressed as percentages) 56, 75, 40, 86, 32. Write a program to calculate the average mark achieved by this group of students. Include an appropriate output message.

Please save this file (and all attempts) in the directory provided for you and call it "AverageMark.java".

Task 2:

Write a program to prompt the user to enter a number and then print out the times table for that number. For example:

Please enter a number:- 4

Here is the 4 times table:- 1 * 4 = 4

2 * 4 = 8

3 * 4 = 12

Note: Your output should look exactly like this.

Before attempting any coding you should produce a pseudo code algorithm, hand written, on paper, to document your design. This should be submitted when the tasks are complete.

Please save this file in the directory provided for you and call it "TimesTables.java"

Task 3:

Write an int function (method that returns an int result) that takes 2 integer parameters and returns an integer that is the first number to the power of the second. Then write a program that prompts the user for 2 integers, calculates the value of the 1st raised to the power of the 2nd, using your function, and outputs the result. e.g.

Please enter the first number: 2

Please enter the second number: 3

The value of 2 to the power of 3 is: 8

Please save this file in the directory provided for you and call it "PowerOf.java".

Task 4:

Develop a class called "Car" (in a file called "Car.java") that provides a new type to represent motor vehicles. Your new Car type should be able to characterise all types, and represent them as an instance of this class.

Before attempting any code you should model the Car class fully, and produce a class diagram to illustrate the main attributes and operations. Don't be tempted to start any coding until you have a clear understanding of the requirements of your class.

Example attributes you may consider include: engine capacity, registration date, registration number, colour, etc... In addition to get and set methods for each attribute you should also include an equals method, a toString method, and a method that allows a target Car to be compared with another Car (provided as an argument) to establish which Car is newer (based on the registration date). Make use of a class variable to keep track of the number of instances that are created. Each time a constructor is called this variable should be incremented. Provide a class method called "numberOfInstances" which returns the number of Car objects that have been created so far in this run of the program.

You should provide a number of constructors to allow new Car objects to be created and should at least include constructors that allow Car objects to be created from the following:

A set of primitive values

A String

Another Car object (i.e. clone)

Your class should employ good software engineering practice (including providing information hiding through encapsulation), and should be well commented to make clear the operation provided by each method and the purpose of any return value and parameters associated with it. You should also ensure that good field names are chosen, and appropriate naming conventions are adopted.

Create a client programme called "CarTest" to test your Car class, and store this in the directory provided in a field called "CarTest.java". The CarTest programme should be used to thoroughly test all constructors and methods provided by your Car class to ensure that the class works correctly.

NB: In deciding what attributes and operations would be useful to provide in your Car class, you may find it helpful to think of applications that could make use of such a class. A motor insurance company, Garage, The Police, DVLA, Vehicle recovery organisations like the AA, etc.. all need to store information about Cars.

A.6: Allocation of task marks

	PA01	PA02	PB01	PB02	PB03	PC01	PC02	PC03	PC04	PC05	Total
Task1 – Total	3	1	3	1	2	1	2	3	2	2	
Answer 57.8 or 58	1	0	1	0	0	0	0	1	0	0	**3**
Output message suitable	1	1	1	1	1	0	1	1	1	1	**9**
File and class name as requested	1	0	1	0	1	1	1	1	1	1	**8**
Task 2 - Total	3	0.5	4.5	2	2	4	3	2.5	3	3.5	
Pseudo-code written prior	0	0	0.5	0	1	1	0.5	0.5	1	0.5	**5**
User prompted for input	1	0.5	1	1	0	1	0.5	0.5	0.5	1	**7**
Output in requested format	0	0	1	0	0	0	0	0	0	0	**1**
Maths correct	1	0	1	1	0	1	1	1	1	1	**8**
File and class name as requested	1	0	1	0	1	1	1	0.5	0.5	1	**7**
Task 3 - Total	4	x	3	5	x	1	0	3	4	3	
User prompted for input * 2	1	0	1	1	0	0	0	0	1	1	**5**
int function with 2 * int parameters	0	0	0	1	0	0	0	1	0	0	**2**
Output message suitable	1	0	1	1	0	0	0	1	1	1	**6**
Maths correct	1	0	0	1	0	0	0	0	1	0	**3**
File and class name as requested	1	0	1	1	0	1	0	1	1	1	**7**

A.7: Compilation and runtime errors

Theme	Num Users	Num Errors
Exception Handling	4	9
Import statements	4	8
Global variables	4	5
Main Method	4	8
Correct use of types (parameters/with operators)	2	6
Undeclared variables	4	9
Capitalisation errors	6	12
Incorrect method/class names	2	8
Code placement	2	3
Correct use of operators	4	8
Missing () ; brace pair	4	15
Others	5	25

Appendix B: Study two – evaluation

B.1: Introductory text

Introduction to phase two: prototype evaluation

Overview

This system has been developed to support students through the early stages of leaning to programme. It uses a reflective approach to the support process, rather than directly correcting mistakes, or misconceptions. While you are programming the system will out put a series of statements that have been design to motivate you to think more carefully about what you are doing. Some will be directly related to what you are doing at the time, while others are more about developing ‘best practice’ habits.

The statements that the system presents to you are determined by the personality profile developed form the data gathered in study one. This is in two parts. The first links your personality traits, (defined by the online personality survey presented prior to the observation study), to the programming behaviours you exhibited, during the observation study. These have been statistically correlated across all participants and are used to predict, which behaviours you are most likely to exhibit while using the prototype. The second part has been developed from the data gathered from the java questionnaire, (presented after the observation study). This data, containing self-rated abilities, values and preferences, was used to populate your personal experience profile.

This evaluation study has two goals:

Testing the predictive ability of the personality profile – which will be done behind the scenes by recording your interactions, and analysed later

Evaluating the statements the system produces and how appropriate they are to supporting the learning process.

With this in mind we ask you, to interact with the prototype in the same why you would while programming any other text editor. While the functionalities may be different, or unavailable, the basic requirement is the process you undertake while writing code.

The process

This part of the study has two parts.

The first requires you to interact with the prototype, as you would any text editor, while solving some programming tasks. During this process the software will output a variety of statements. As you notice, and/or respond to each of these statements, we would like you to make a quick evaluation, on the sheets provided, before you continue with your code. This will act as a reminder for later of what you thought of each, as well as giving us an insight into your first impressions. If you notice statements that you have missed, don’t worry there is opportunity later to comment on these.

Example

Statement No: 4 **Negative** **Indifferent** **Positive**

	Strongly Disagree 1	2	3	4	Strongly Agree 5
1 – Understandable					X
2 - Relevant				X	
3 – Helpful				X	
4 - Timeliness			X		

Are there any comments you wish to add regarding this statement?

I didn't think of it like this before!

The coding period will last for about 30 minutes. When you have finished, or time is called please DO NOT exit the prototype. The recording will be stopped at this point.

Next we would like you to complete an in-depth online questionnaire, taking between 20 and 30 minutes. If you have participated in study 1, the observation phase, there is no need to complete sections A and B, go directly to section C. Remember to fill in your participant number before you start.

Please feel free to ask questions at any time. As previously I will be acting as your lab assistant, and you are quite welcome to use any code, books, internet etc, at anytime.

If you have a large USB drive please over 256Mb, please disconnect it form the laptop between views, or copy the relevant folders onto the desk top.

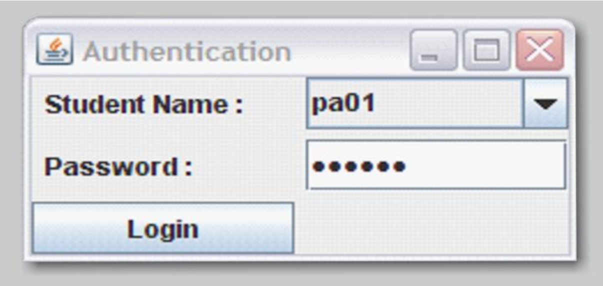
A dos prompt will be running in the bottom half of the screen, please keep this visible at all times.

The following pages give more detailed information about the prototypes functionality and limitations.

The Prototype

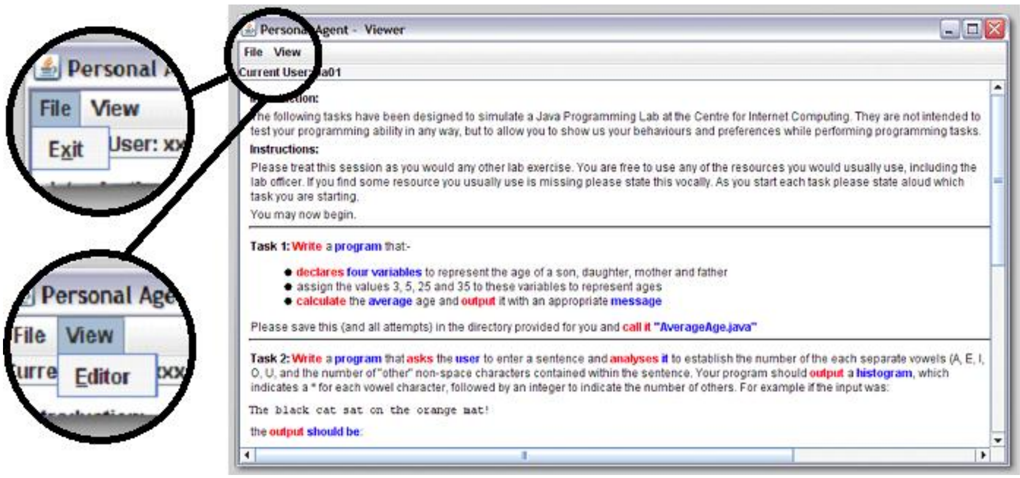
This consists of three parts, a login box, a HTML viewer and a Text Editor.

Login



Login using your participant number, by selecting from the drop down list. The password is fixed at the moment, so does not require changing. This will bring up the HTML viewer, and populate your personal predictions list.

HTML Viewer



The viewer has two menus:

File: with exit selection - exits the application

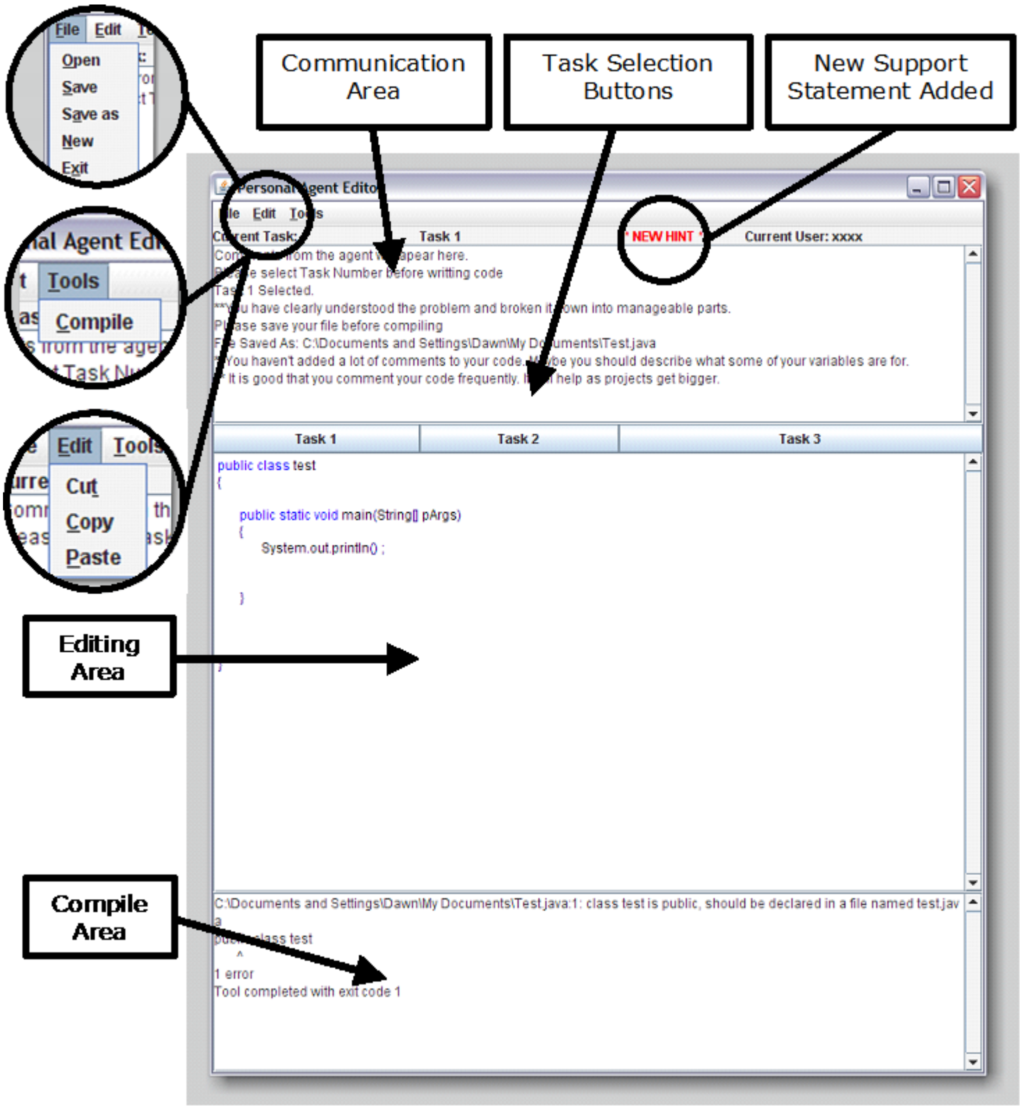
View: with editor selection – opens the text editor

Other Functions:

Right Mouse Click: brings up the copy command only.

Once you are ready to begin coding select the Editor from the View menu.

Text Editor



The text editor is split into three sections:

The Communication area – this is where the system will respond to your actions, with confirmation messages, and display the support statements. This is a none editable area.

The Editing area – this is where you type your code. Cut copy and paste are available from a right mouse click. This is none editable if no task has been selected.

The Compile area - results from compilation will be displayed here.

The Editor has three menus:

File: provides the standard, Save, Save as, New & Exit.

Edit: provides cut, copy and paste.

Tools: provides compile function.

Please not quick keys are currently none operational.

Process

Before you start, you need to let the system know which task you are going to do, by selecting one of the buttons between the communication and edit area. Once this has been done the edit area will be available for typing code.

As you type the system will monitor your interactions and use the predicted behaviours to output support messages. When a new support statement is added to the communication area, the message “NEW HINT” will appear at the top of the screen. Once you start typing again this will disappear until the next statement is generated.

You can use the compile and save functions any time, although the system will not let you compile without saving. There is currently no auto save function attached to compile.

Once you have compiled your code, you will need to go to an external dos prompt to run it. This function is currently unavailable. The command for running a java application in dos is: java CLASS NAME.

B.2: Task set two

Task 1:

Write a program that:-

declares four variables to represent the age of a son, daughter, mother and father

assign the values 3, 5, 25 and 35 to these variables to represent ages

calculate the average age and output it with an appropriate message

Please save this file (and all attempts) in the directory provided for you and call it "AverageAge.java"

Task 2:

Write a program that asks the user to enter a sentence and analyses it to establish the number of the each separate vowels (A, E, I, O, U, and the number of "other" non-space characters contained within the sentence. Your program should output a histogram, which indicates a * for each vowel character, followed by an integer to indicate the number of others. For example if the input was:

The black cat sat on the orange mat!

the output should be:

A: *****

E: ***

I:

O: **

U: *

Other (non-space) Characters : 20Before attempting any coding you should produce a pseudo code algorithm, hand written, on paper, to document your design. This should be submitted when the tasks are complete.

Please save this file in the directory provided and call it "VowelCounter.java"

B.3: Quick-sheet results

All statements made by participants are presented as written by the participants and may include spelling and grammatical errors.

Attitude scores for evaluated hints

Key: 1 = negative, 2 = indifferent, 3 = positive

Task	Strict	Hint	PA02	PC01	PC02	PB01	PB02	PD01	PA01	PC04	MODE	SO MODE
1	N	CA1					2				-	
1	N	CA2									-	
1	N	CA3		3		3				2	3	3
1	Y	MO1		3		3				3	3	
1	Y	MO2					1				-	
1	Y	MO3					1				-	3
1	N	PT2		2		2	1				2	
1	N	PT3		3		2	3				3	
1	N	PT6		2							-	
1	N	PT7D		3		3					3	
1	N	PT9D									-	3
1	N	WC1					3				-	-
1	N	WC2D				1	2				-	
1	N	WC3D								3	-	3
2	N	OV1								3	-	-
2	N	WC1									-	
2	N	WC3D								3	-	-
2	N	WP1									-	
2	N	WP2				2					-	
2	N	WP3								3	-	
2	N	WP5D				1				3	-	
2	N	WP6D									-	3
		Static									3	
		Dynamic									3	
		Overall									3	

Evaluation questions

1. I understood the statement.

Task	Strict	Hint	PA02	PC01	PC02	PB01	PB02	PD01	PA01	PC04	Mean	SO MEAN
1	N	CA1			2		4				3.00	
1	N	CA2			4						4.00	
1	N	CA3		5		5			5	5	5.00	4.00
1	Y	MO1	5	5		5		5	5	5	5.00	
1	Y	MO2					5				5.00	
1	Y	MO3					5				5.00	5.00
1	N	PT2	5	3		1	3	5			3.40	
1	N	PT3	5	4		5	5	5			4.80	
1	N	PT6		3	4						3.50	
1	N	PT7D		5	5	5					5.00	
1	N	PT9D			2						2.00	3.74
1	N	WC1					5	5	5		5.00	
1	N	WC2D			5	5	5				5.00	
1	N	WC3D		5					5	5	5.00	5.00
2	N	OV1							1	4	2.50	2.50
2	N	WC1							5		5.00	
2	N	WC3D							5	4	4.50	4.75
2	N	WP1							5		5.00	
2	N	WP2				5					5.00	
2	N	WP3								5	5.00	
2	N	WP5D				5				5	5.00	
2	N	WP6D							5		5.00	5.00
		Static									4.41	
		Dynamic									4.50	
		Overall									4.44	

2. The statement was relevant to what I was doing at the time.

Task	Strict	Hint	PA02	PC01	PC02	PB01	PB02	PD01	PA01	PC04	Mean	SO MEAN
1	N	CA1			1		2				1.50	
1	N	CA2			4						4.00	
1	N	CA3		5		5			4	4	4.50	3.33
1	Y	MO1	5	5		4		5	2	5	4.33	
1	Y	MO2					1				1.00	
1	Y	MO3					2				2.00	2.44
1	N	PT2	5	4		2	1	4			3.20	
1	N	PT3	5	4		2	5	4			4.00	
1	N	PT6		2	3						2.50	
1	N	PT7D		5	2	3					3.33	
1	N	PT9D			2						2.00	3.01
1	N	WC1					2	5	5		4.00	
1	N	WC2D			4	3	2				3.00	
1	N	WC3D		3					4	5	4.00	3.67
2	N	OV1							1	5	3.00	3.00
2	N	WC1							5		5.00	
2	N	WC3D							4	3	3.50	4.25
2	N	WP1							4		4.00	
2	N	WP2				1					1.00	
2	N	WP3								5	5.00	
2	N	WP5D				1				5	3.00	
2	N	WP6D							4		4.00	3.40
		Static									3.27	
		Dynamic									3.26	
		Overall									3.27	

3. The statement helped me to think about what I was doing at the time.

Task	Strict	Hint	PA02	PC01	PC02	PB01	PB02	PD01	PA01	PC04	MEAN	SO MEAN
1	N	CA1			2		1				1.50	
1	N	CA2			3						3.00	
1	N	CA3		4		2			5	3	3.50	2.67
1	Y	MO1	5	5		2		5	2	5	4.00	
1	Y	MO2					1				1.00	
1	Y	MO3					2				2.00	2.33
1	N	PT2	5	3		1	1	4			2.80	
1	N	PT3	4	4		2	5	4			3.80	
1	N	PT6		2	3						2.50	
1	N	PT7D		4	1	2					2.33	
1	N	PT9D			1						1.00	2.49
1	N	WC1					3	5	5		4.33	
1	N	WC2D			5	3	3				3.67	
1	N	WC3D		4					4	5	4.33	4.11
2	N	OV1							1	5	3.00	3.00
2	N	WC1							5		5.00	
2	N	WC3D							4	4	4.00	4.50
2	N	WP1							5		5.00	
2	N	WP2				1					1.00	
2	N	WP3								5	5.00	
2	N	WP5D				1				5	3.00	
2	N	WP6D							4		4.00	3.60
		Static									3.16	
		Dynamic									3.19	
		Overall									3.17	

4. The statement appeared at an appropriate time.

Task	Strict	Hint	PA02	PC01	PC02	PB01	PB02	PD01	PA01	PC04	Mean	SO MEAN
1	N	CA1			1		5				3.00	
1	N	CA2			2						2.00	
1	N	CA3		4		1			4	4	3.25	2.75
1	Y	MO1	5	5		2		4	4	5	4.17	
1	Y	MO2					1				1.00	
1	Y	MO3					5				5.00	3.39
1	N	PT2	3	3		1	1	5			2.60	
1	N	PT3	4	4		3	5	5			4.20	
1	N	PT6		3	2						2.50	
1	N	PT7D		3	2	3					2.67	
1	N	PT9D			2						2.00	2.79
1	N	WC1					5	4	5		4.67	
1	N	WC2D			1	2	5				2.67	
1	N	WC3D		4					4	5	4.33	3.89
2	N	OV1							1	5	3.00	3.00
2	N	WC1							5		5.00	
2	N	WC3D							4	3	3.50	4.25
2	N	WP1							4		4.00	
2	N	WP2				1					1.00	
2	N	WP3								5	5.00	
2	N	WP5D				1				5	3.00	
2	N	WP6D							5		5.00	3.60
		Static									3.36	
		Dynamic									3.31	
		Overall									3.34	

Comments made by participants

Hint Code	Participant	Comment
CA1	PC02	?
CA1	PB02	I am not too sure what has it got to do with anything. Have I donr it right or wrong?
CA2	PC02	would of become better earlyer
CA3	PB01	Not sure why it apeared at end of saving

CA3	**PC04**	I did not find it useful
MO1	**PA01**	Doesn't correspond to what I did with the output.
MO1	**PD01**	Not certain of timeliness as concentrating on editor.
MO2	**PB02**	It occurred only when I entered a ":" in the output statement. As well as bad timing it was Irelavant too.
MO3	**PB02**	There is only one result. It might be useful to keep students brian occupied with statements like that.
PT2	**PC01**	Wasn't sure if this cams up because I typed a / or because of task.
PT2	**PB01**	Not understanding why it should ask me this
PT2	**PB02**	Makes you wonder what is hidden behind this hint. Is it trying to tell me that I shouldn't forget to out put the result or is there something I don't know.
PT2	**PD01**	makes me think
PT3	**PC01**	Useful, as I hadn't ever considered having a decimal as an answer.
PT3	**PB01**	Making think whether to use double or int
PT3	**PB02**	Motivates you to find out other ways for calculate the average.
PT3	**PD01**	question my ability
PT6	**PC01**	I've not done a divison statement. Ive done an addition? (Writing -Typo)
PT6	**PC02**	Wasn't sure it was right but I made a mistake on this one
PT7D	**PC01**	Came up a bit later than I'd have expected.
PT7D	**PC02**	Moral boosting to a new learner
PT9D	**PC02**	look at what
WC1 T1	**PB02**	The only comment that I could have added was the description of the program.
WC1 T1	**PD01**	As for 3 (didn't think comments would ???? Here
WC2D T1	**PC02**	Comment code FtW
WC2D T1	**PB01**	Stating its good practice to comment
WC2D T1	**PB02**	I think I;ve done quite a lot of commenting for such a small program.
WC3D T1	**PC01**	I had already commented at the beginning
WC3D T1	**PC04**	Promt user to write more commnets
WC1 T2	**PA01**	fair point :-)
WC3D T2	**PC04**	Promts user to write more
OV1	**PA01**	I got distracted with the comment. I was distracted out of 'The Zone'. Didn't notice untill later. No Idea!
OV1	**PC04**	I forgot to declare my variable it reminded me of this.
WP1	**PA01**	This is similar to statement 1

WP2	**PB01**	Presumming I havent read full question, is users fault if question has not been read.
Not Hint related	**PB02**	The hints could be separated into different types. For example if the hint is gree – just remember it for future programming. If the hint is red – you better change your code.

B.4: Evaluation questionnaire

Phase 2, Prototype Evaluation

Please enter your participant code here:

Questionnaire purpose

This questionnaire is focused on evaluate the prototype, which was developed to utilise your personality profile. If you did not participate in study 1 of this research please complete all the sections A-F otherwise, complete sections C-F

A. Demographic questions

Q1. What is your gender?

Male Female

Q2. How old are you?

Q3. What country are you from?

Q4. What is your highest attained education level?

GCSE or O' level	HNC/HND	PhD
AS level	Bachelor degree	Other
A' level or A2 level	Masters degree	

Q5. What is the title of your course?

Q6. If you are studying at University, what year of study are you currently in?

Not at University	Year 2	Year 4
Year 1	Year 3	Other

B. Programming Experience

Q7. How long have you been programming computers?

Years Months

Q8. How long have you been programming in the Java Language?

Years Months

Q9. Which of the following programming languages do you Know of, use Some times, use Frequently

C	Lisp	Prolog
C#	Oberon	Python
C++	Pascal	Ruby
Java	Perl	Visual Basic
JavaScript	PHP	

Q10. How would you rate your general programming ability? (where 1=low and 5=high).

Very Poor Average Very Good

Q11. How would you rate your Java programming ability? (where 1=low and 5=high).

Very Poor Average Very Good

C. Your Last Programming Experience

Q12. Before this prototype evaluation study, how long has it been, since you last spent more than 2 hours coding?

Years Months

Q13. Before this prototype evaluation study, did you use Java in your last coding activity?

Yes No … if No go to Q14

Q14. What language did you use?

D. Evaluating the prototype

If you have scored any of the statements during the study, please answer questions 15 to 18. Otherwise answer questions 17 to 19.

Q15. Look back at the statements that you have already scored, choose one that you had the strongest positive attitude towards, and answer the following questions.

Statement number chosen:

Q15a. How did the statement support you in thinking about what you were trying to do?

Q15b. Would you have preferred the statement to have appeared:

Earlier … go to question 15c

Later … go to question 15c

or it was About Right … go to question 16

Q15c Why did you feel the timing for this statement was not suitable?

Q16. Again look back at the statements that you have already scored, and choose one that you had the strongest negative attitude towards, and answer the following questions.

Statement number chosen:

Q16a How did this statement fail to support you in thinking about what you were trying to do?

Q16b. Would you have preferred the statement to have appeared:

Earlier … go to question 16c

Later … go to question 16c

or Not At All … go to question 17

Q16c. Why did you feel the timing for this statement was not suitable?

Q17. Now scroll back through the comments section of the prototype, and look at the statements you may have not noticed. Choose one and answer the following questions.

Statement number chosen:

Q17a. Why do you think you did not notice this particular support statement?

Q17b. If you had noticed this statement do you think it would have been useful to you:

No Maybe Yes

Q18. If there are any other statements you particularly wish to comment on please do so here. Please remember to add their ID number at the begriming of each comment you make.

Q19. Which of the following statements best describes, why you did not score any statements during the study

a. I just forgot to look.

b. I looked but was more interested in finishing the code.

c. I looked but did not think the statements were of much help.

d. I was intending to fill them out at the end

e. I didn't realise what I was supposed to do.

f. None of the above.

Q20. Overall how would you rate the prototype in the following areas. (where 1=low and 5=high).

Very Poor Average Very Good

a. The Editor's Layout

b. The Editor's Functionality

c. The Level of Support Provided

d. The Promptness of the Support Statements

e. The Relevance of the Support Statements

f. The Helpfulness of the Support Statements

Q21.Are there any comments that you wish to add regarding the prototype?

E. Evaluating the support concept

Please read the following scenarios and rate them according to how beneficial, you think, a remote artificial support system would be. (where 1=low and 5=high).

Q22. To Students practicing their programming outside of lab time.

Not Beneficial Beneficial

Q23. If you had a programming task to do, when you had not programmed for a period of time?

Not Beneficial Beneficial

Q24. When learning a new programming language, within a work environment?

Not Beneficial Beneficial

Q25. If you had, had access to a remote system such as this, would you have?

Poor Average Good

a. Found it easier to practice java programming.

b. Been motivated to practice outside of university more

c. Found it useful in labs, when a lab assistance was unavailable.

Q26. Are there are any other comments you wish to make regarding the use of artificial support systems.

F. Evaluating the personality profile concept

Q27. Was the personality survey,(before the observation study). (1 is easy and 5 is hard)

Easy Average Hard

Q28. How long did the personality survey, (before the observation study) take. (1 is short 5 is long)

Short Time Average Time Long Time

Q29. Do you think the length of time the personality survey, (before the observation study) took, was. (1 is acceptable and 5 is unacceptable)

Acceptable Unacceptable

Q30. Was the java questionnaire, (after the observation study). (1 is easy and 5 is hard)

Easy Average Hard

Q31. Did you think the java questionnaire, (after the observation study) was. (1 is short 5 is long)

Short Time Average Time Long Time

Q32. Do you think the length of time the java questionnaire, (after the observation study) took, was. (1 is acceptable and 5 is unacceptable)

Acceptable Unacceptable

Q33. Are there any other comments you wish to make regarding the gathering of personality related information by survey.

Thank you very much for taking part in these studies and completing the questionnaire. Please feel free to ask any questions about the research or the study process

B.5: Evaluation questionnaire results

The following presents the raw data gathered from the self-assessed java questionnaire. Values indicate the number of participants that responded to the question with a particular response. All statements made by participants are presented as written by the participants and may include spelling and grammatical errors.

Part A. Participant demographics (Q1 to Q6)

Q1. Gender		Q4. Current Education Level		Q5. Studying for		Q6. Study Year	
Male	Female	GCSE	1	Bachelor	5	1	1
6	2	AS/A2	3	Masters	1	2	5
Q2. Age		HND	1	PhD	2	3	1
25 & under	over 25	Bachelor	2			4	1
6	2	Masters	1			other	0
Q3. Home Country							
UK	Other						

Part B. Programming experience (Q7-Q14)

Q7-Q8. Total programming experience

Time	Q7. General Programming	Q8. Java Programming
less than 6 months	0	1
6-24 (inclusive) months	2	4
2-5 (inclusive) years	3	2
5-10 (inclusive) years	2	1
10-20 (inclusive) years	1	0
greater than 20 years	0	0

Q9. Understanding and using programming languages

Language	Know	Sometimes	Frequently	Language	Know	Sometimes	Frequently
C	7	1	0	C#	6	0	0
C++	7	0	0	Java	3	3	5
JavaScript	4	4	1	Lisp	3	0	0
Oberon	4	0	0	Pascal	7	2	1
Perl	7	2	0	PHP	4	3	3
Prolog	3	0	0	Python	6	1	0
Ruby	6	0	1	Visual Basic	7	1	0

Q10-Q11. Self-assessed programming ability

	Very Poor (1)	(2)	Average (3)	(4)	Very Good (5)
Q10. General programming	0	3	4	1	0
Q11. Java programming	1	3	2	2	0

C. Your Last Programming Experience

Q12-Q14. Last programming experience before the observation study

Q12. Last 2 hour session		Q13. Session used Java		Q14. Other Language used	
1 month or less	2	**Yes**	5	PHP	1
1-3 (inclusive) months	1	**No**	2	Pascal	1
3-6 (inclusive) months	0	**null**	1		
6-12 (inclusive) months	1				
a Year or more	1				
null	3				

D. Evaluating the prototype

Q15a-c. Hints selected as most positive attitude towards

Hint Code	Participant	15a. Comment on how hint supported	15b. Preferred timeliness	15c. Comment on timeliness
MO1	PC01	The statement confirmed that using variables in my output statements was better than putting the calculation in it instead. This was a nice back up but I always do that process and it seemed a little trivial but a nice thumbs up or encouragement. Would have helped me a lot when I started learning Java because I wouldn't think of things like that.	Earlier	I'm not sure if this statement would have popped up if I had put the calculation in the output string. It might have been better popping up earlier to advise people not to make the mistake but at the same time we learn by mistakes so it might not have been. But if I was a novice I'd have preferred spending less time making mistakes like that.
MO1	PA02	well it suggested that = have you selected the correct type of variable for the average counter	About right	
PT3	PB02	It motivates to find out other ways to calculate the avegare.	About right	
PT7D	PB01	Didn't support me just indicated that i understand how to declare variables. It gives the user a clear indication that this is a good practice	About right	
WC1 T1	PA01	This statement reminded me to comment my code variables and functions. Something which is useful for maintainability.	About right	
WC1 T1	PD01	As a reminder for commenting under any circumstances and conditions	About right	
WC2D T1	PC02	Al though I Did not Add Comments i know that they are an important Part of coding in any language to help understand and correct 1rrors.	Earlier	it was a bit Late into the work it would of been better near the start to remind people to do it.
OV1	PC04	This statement had picked up on the fact that I had forgoten to decare a variable. it is very useful because it would have caused great problems for me	About right	

Q16a-c. Hints selected as most negative attitude towards

Hint Code	Participant	16a. Comment on how hint supported	16b. Preferred timeliness	16c. Comment on timeliness
CA1	PC02	i failed to see the Relivence to what i was doing since i had already 5 variables in one step...	About Right	
CA3	PC04	This statement had told me that I had understood the problem and broken it down into manageable parts I found that I was still working on the problem in my head so did not find it a use ful hint	Later	may be when I was commplining my code so that or when the code is my complete so the user could have a better break down if the code on in manageable part. Or if a proggramer is programmin on big section it would be advisiable to advise them to break the code up.
PT2	PC01	I wasn't too sure if the statement appeared because of a typo of a / or because of the task objective. At the time it didn't make sense but when backed up by statement 3 (are you using the best TYPE for finding an average) it made more sense and I realised I'd put integers instead of doubles without really thinking.	Earlier	Earlier and probably with statement 3. I think they might have appeared at the same time come to think of it but statement 2 alone had me a little confused. Just wasn't sure what it applied to.
PT2	PB02	Did not see the point of this statement. Makes you wonder wether it is trying to tell me not fo forget to output the result or is there something I should be doing differently.	About Right	
PT2	PD01	Because I already know what result I expect when using it.	Later	As a reminder of other possibilities should things not be as expected
PT3	PA0 2	delay time message	Earlie r	alert message would be nice!!
OV1	PA01	I didn't notice it at the time. When I did I couln't understand what I meant in my context and lost my train of thought with the code.	About right	It may have been better to highlight the line of code. A popup may have been useful but then again I get extremely annoyed with all the popups with a modern operating system so this would have made me lose my concentration even more.
WP5D	PB01	Was presuming that i did not read the task given. Would be difficult to tell if the user has done any pseudocode thus meaning this statement may be pointless for the user.	About Right	

Q17. Hints missed during observation study

Hint Code	Participant	17a. Comment on why you missed this hint	17b. would it have been useful to you
OV1	PA01	I wasn't aware of doing anything 'a bit dodgy' at that point.	Maybe
PT7D	PC02	It Stood out to me because i have coded alot in java i think this was a bit patirnising if i was a new programmer i can imagin been Quite happy with that comment but a more adept programmer is likely to be a bit... why?	No
-	PC01	I noticed all 7 statements. Only problem I had with statements was working out if they all appeared at once or one by one.	
WC3D	PC04	I already put a lot of comments in my code already then when I have completed section I add more once I have tryed and tested solutions to the problems	Maybe

Q18. Other comments made by participants

Hint Code	Participant	18 Other comments
	PA02	Alert boxes
CA1	PB02	[3]Again I wasn't too sure how should i respond to this hint. You might want to separate the hints into different types for example hints that need some action taking (in red) or just general hints that should be remembered (in green). Because at this point i was confused wether im doing it right or wrong.

Q19. Why statements were not scored.

No responses given for this question

Q20 a-f. Cross participant weighted mean for each question

Rating	a. Layout	b. Functionality	c. Level of support	d. Promptness of support	e. Relevance of support	f. Helpfulness of support
null	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	1	2	2	2	2
3	0	2	0	1	1	2
4	6	4	4	5	4	2
5	2	1	2	0	1	2
Weighted Total:	34	29	30	27	28	28
Mean:	**4.25**	**3.63**	**3.75**	**3.38**	**3.50**	**3.50**
Overall:						**3.67**

Q20 a-f. Cross question mode for each participant

	a. Layout	b. Functionality	c. Level of support	d. Promptness of support	e. Relevance of support	f. Helpfulness of support	Mode
PA02	4	4	4	4	4	3	4
PC01	5	4	5	4	4	5	5
PC02	4	3	4	2	2	2	2
PB01	4	3	2	2	3	3	3
PB02	4	5	4	4	2	2	4
PA01	4	2	2	4	4	4	4
PC04	4	4	5	3	4	4	4
PD01	5	4	4	4	5	5	5
Overall:							**4**

Q21. Other comments about the prototype

Participant	Comment
PA02	aleet message
PC01	I liked the colour coding that it used for different aspects of the code. Though I think there was one point (pity I can't remember exactly when) that I was expecting a colour change and none came though I generally code in SciTE.
PC02	it is a Very good tool when finished i would be Tempted to use it.
PB01	Comments could be more clear just by making them stand out from the text area at the top.
PB02	You might want to add the line numbers in the editors layout (but im sure you have thought about that already.
PA01	I am used to the way that the Eclipse IDE works and perhaps have become accustomed to it's prompts.My complaint is that I often missed the support statements as I was looking at the main editor.
PC04	when the comments appear may a noise or a little flashing icon just 2 make you aware of it but not annoy you.
PD01	Maybe some indication within the editor itself when a hint occurs i.e flash the background once etc. Found myself not noticing hints until late on.

E. Evaluating the support concept

Q22-25. Cross participant weighted mean for each question

Rating	Q22. Programming outside of lab time	Q23. Programming after absence period	Q24. Learning a new language during work	Q25a. Easier to practice	Q25b. Motivated to practice	Q25c. Useful in labs when help unavailable
null	0	0	0	0	0	0
1	0	1	0	0	1	1
2	1	0	0	1	0	0
3	0	0	1	1	1	0
4	3	1	2	3	2	3
5	4	6	5	3	4	4
Weighted Total:	34	35	36	32	32	33
Mean	**4.25**	**4.38**	**4.5**	**4**	**4**	**4.13**
Scenario mean:			**4.38**			
Hindsight mean:						**4.04**
Overall:						**4.21**

Q26 Other comments about artificial support systems

Participant	Comment
PA02	i love AI :P
PC01	I'd still have preferred to consult a lab assistant or code examples to learn than wait for statements to appear. For task 2 I was expecting some to appear but none were (the program had stopped working in the background we found). Presumably some would have normally. Artifical support systems are fine as long as you can turn them off and they aren't anywhere near as annoying as that paperclip in word!
PC02	making this system Support another language as well as Java would be handy
PA01	It does encourage good practice.
PC04	It is a good piece of software that helps you when the lecture can not. It is also good that can be used inacordace with lecturer tasks
PD01	I think they work when they don't give answers but provide food for thought

F. Evaluating the personality profile concept

Q27- 32 Cross participant weighted mean

	Q27. Difficulty of Personality Survey (1 Hard - 5 Easy)	Q28. Length of time Personality Survey (1 Long – 5 Short)	Q29. Acceptability of time Personality Survey (1 Unacceptable – 5 acceptable)	Q30. Difficulty of Java Survey (1hard - 5 easy)	Q31. Length of time Java Survey (1 Long – 5 Short)	Q32. Acceptability of time Java Survey (1 Unacceptable – 5 acceptable)
null	0	0	0	0	0	0
5	3	0	5	3	0	6
4	3	2	3	3	1	1
3	1	6	0	1	7	1
2	1	0	0	1	0	0
1	0	0	0	0	0	0
Total	32	26	37	32	25	37
Mean	**4.00**	**3.25**	**4.63**	**4.00**	**3.13**	**4.63**
Personality Survey			**3.96**			
Java Survey						**3.92**
Overall						**3.94**

Q33. Other comments about data gathering by survey

Participant	Comment
PA01	I can't really remember the details but I don't remember it being too much of a difficult task.
PC04	Please type your comments here. Thank you.
PD01	If you have to return to a page it is annoying to have to complete the next page again rather than those results being stored.

Appendix C: Support objects

C.1: Static hint rules

A hint rule consists of an ID, two behaviours, a keyword and the hint text in the following format:

```
ID/Behaviour A/KEYWORD/Behaviour B/hint text
```

These rules are applied in an IF…THEN conditional process, to the edited behaviour pattern. The condition type is provided by the keyword. Both behaviours, or in some cases just behaviour A, can be used in the condition statement. Behaviour B is always the behaviour that is edited in the edited behaviour pattern. The Keyword also provides information as to whether one or all repeating behaviours within a behaviour pattern have hint text added.

The order that the rules are applied also forms part of the logical process. Some keywords included instructions to `OVERWRITE` *hints* already attached to behaviours, if certain conditions are met. The default is to add multiple *hints* to a single behaviour. Both behaviour A and B can be composed of a single behaviour or a behaviour pattern. In the case of a behaviour pattern the last behaviour in the pattern is edited by default. If the rule has `FIRST` attached to it, this indicates that it is the first behaviour within a behaviour pattern that should be edited. The Keywords and their extensions are as follows:

`AFTER` and `AFTER_OVERRIGHT` - If behaviour A comes after behaviour B add the hint to behaviour B.

`BEFORE` and `BEFORE_OVERRIGHT` - If behaviour A comes before behaviour B add the hint to behaviour B.

`MISSING` and `MISSING_OVERRIGHT` and `MISSING_FIRST` - If the behaviour A is not present then add the hint to behaviour B

PRESENT and PRESENT_OVERRIGHT - If both behaviour A and B are present, regardless of their order, add hint to B.

EQUAL and EQUAL_OVERRIGHT_FIRST - If the predicted behaviour pattern and the ideal behaviour pattern are equal then add the hint to behaviour B.

For example the use:primitive_types SO contains the following hint rule:

```
PT2/type:double_dec,type:double_dec/MISSING_OVERRIGHT/type:double_dec/What
type of result are you expecting when you use this variable in division?
```

Given the following predicted behaviour pattern:

```
type:int_dec,type:int_dec,type:int_dec,type:int_dec,type:int_dec,type:int_dec,
type:int_dec,compile:code
```

The PT2 rule would determine that type:double_dec, type:double_dec pattern is missing from the predicted pattern. This would result in the *hint* being added to the last behaviour in the behaviour pattern listed after the keyword. As this particular SO is of the non-strict variety, it is to be the ideal behaviour pattern that is edited, not the predicted behaviour pattern shown here.

C.2: Dynamic hint rules

Similar to the static rules an IF…THEN condition is used to determine if the rule should be applied. Rather than comparing behaviours, the dynamic aspects (values, attitudes, ability and goals) are tested via a Boolean condition. Each so the behaviours can be associated with a value for each of these dynamic aspect types. This is not true with the dynamic aspect preference as it is a list of preferred behaviours given some context. This aspect is treated in a comparison process similar to that of the static hints. The dynamic rule has the following format:

```
ID/Behaviour A/ASPECT_KEY/Boolean Condition/Behaviour B/ Hint text
```

The `ASPECT_KEY` tells the system which of the dynamic aspects the behaviour in this rule refers to. Given the aspect the system then checks whether there is a value for the given behaviour in the current PR profile. If there is then that value is inserted into the left hand position of the Boolean condition. If the condition equates to true the *hint* text is added to all the behaviours within the behaviour pattern that are equal to behaviour B. For example, the use `use:primitive types` has the following dynamic rule:

```
PT8D/type:double dec/ABILITY/>= 80/type:double dec/You quite clearly understand how to different primitive types work.
```

If the PR profile has an `ABILITY` value for the behaviour `type:double dec` that is ninety, the condition statement becomes `90 >= 80`. This equates to true in Boolean logic, and the *hint* text would be added to all occurrences of `type:double dec` in the behaviour pattern that is currently being edited, either the predicted behaviour or the ideal behaviour deepening on the `<STRICT>` tag as describe in Appendix C.1.

C.3: Support object input file (partial)

<INSTRUCTION>

use:primitive types

<IDEAL>

type:int dec,type:int dec,type:int dec,type:int dec,type:int dec,type:double dec,type:double dec,compile:code

<STRICT>

false

<TRIGGERS>

write:maths expression

<HINTS>

PT1/type:double dec/MISSING_FIRST/type:double dec/If you use int's in a calculation that has a double result, do you have loss of precision?

PT2/type:double dec,type:double dec/MISSING_OVERRIGHT/type:double dec/What type of result are you expecting when you use this variable in division?

<DYNAMIC>

PT7D/type:int dec/ABILITY/>= 80/type:int dec/You quite clearly understand how to declare variables and assign values to them.

<LEXICON>

// all whole number declarations to simple int declaration

type:int dec/partial:short dec assign literal/partial:long dec assign literal/partial:int dec assign literal/type:short dec assign literal/type:long dec assign literal/type:int dec assign literal/partial:short dec/partial:long dec/partial:int dec/type:short dec/type:long dec/partial:short dec assign maths expression/partial:long dec assign maths expression/partial:int dec assign maths expression/unchecked:short dec assign maths expression/unchecked:long dec assign maths expression/unchecked:int dec assign maths expression/type:short dec assign maths expression/type:long dec assign maths expression/type:int dec assign maths expression

<OPTIONAL>

type:array dec/type:variable dec,type:int dec,type:int dec,type:int dec,type:int dec

<END>

C.4: Support object hint text

Hint Code	Hint Description	Tasks	Presented	Dynamic	Strict
USE GLOBAL VARIABLES					YES
UGV1	You have just created a global variable, should it have any special modifiers applied to it?	1,2	0,0	NO	
UGV2	Do you think that the main method declaration should ever need to be changed?	1,2	0,0	NO	
UGV3	It is good practice to give global variables an access modifier.	1,2	0,0	NO	
UGV4	You don't necessarily have to move the global variable to correct this problem. The main method is a static method and thus global variables that it use also are required to be static	1,2	0,0	NO	
UGV5	To have access to global variables from static methods, the variables need to be static to. It is good that you understand this.	1,2	0,0	NO	
UGV6D	Have you ever needed to edit the main method before? Why don't you look at some of your previous code and see if it often changes.	1,2	0,0	YES	
UGV7D	If you are not so sure about how to use a global variable, are really sure that you need to use one?	1,2	0,0	YES	
UGV8D	Is the main method a static method? How do you think this affects access to global variables?	1,2	0,0	YES	
WRITING REQUESTED PSEUDO-CODE					NO
WP1	The task asks for pseudo-code are you going to write it first?	2	1	NO	
WP2	Have you noticed the task asks for pseudo-code?	2	1	NO	
WP3	It's good to write pseudo-code, it will help you understand the problem.	2	1	NO	
WP4	Maybe you should write some pseudo-code before continuing? It will help you understand the problem.	2	0	NO	
WP5D	Pseudo code is used to understand the processes and attributes of a program. Try writing single sentences for each thing you want the program to do.	2	2	YES	
WP6D	Pseudo code is a useful tool during problem solving, it is good that you understand it.	2	1	YES	
USE PRIMITIVE TYPES CORRECTLY					NO
PT1	If you use int's in a calculation that has a double result, do you have loss of precision?	1	0	NO	

Hint Code	Hint Description	Tasks	Presented	Dynamic	Strict
PT2	What type of result are you expecting when you use this variable in division?	1	5	NO	
PT3	Are you using the best TYPE for finding an average?	1	5	NO	
PT4	If you using integers in a calculation that has a double result, do you have loss of precision?	1	0	NO	
PT5	Is this the best type to use if you are going to use it within a division expression?	1	0	NO	
PT6	You clearly understand the best types to use when writing a division expression.	1	2	NO	
PT7D	You quite clearly understand how to declare variables and assign values to them.	1	4	YES	
PT8D	You quite clearly understand how to different primitive types work.	1	0	YES	
PT9D	If you are not sure which is the best primitive type to use why don't you have a look in the course text book.	1	1	YES	
SEPERATING STEPS WHEN FINDING AN AVERAGE					NO
CA1	It is good practice to use a variable for only one step in an algorithm.	1	2	NO	
CA2	It is good practice to break problems down and keep the total separate from the average. This will also make error checking much easier.	1	1	NO	
CA3	You have clearly understood the problem and broken it down into manageable parts.	1	4	NO	
USING MATHS EXPRESSIONS WITHIN OUTPUT STATEMENTS					YES
MO1	Using variables in your output statements is better than putting the calculation in the output statement.	1	6	NO	
MO2	It is not good to put maths expressions in your output statements.	1	1	NO	
MO3	You should divided problems up and use more variables to assign results too.	1	1	NO	
MO4	It is not good to put any maths expressions in your output statements.	1	1	NO	
USING IMPORT STATEMENTS					YES
UI1	Good programming skills are about developing good habits. Writing import statements before using the reference type is a good habit to develop.	2	0	NO	

Hint Code	Hint Description	Tasks	Presented	Dynamic	Strict
UI2	Good programming skills are about developing good habits. You should try to think ahead with what you are going to need, and write your import statements first.	2	0	NO	
USING COMMENTS					NO
WC1	You haven't added a lot of comments to your code. Maybe you should describe what some of your variables are for.	1,2	3,0	NO	
WC2D	You don't rate commenting code very highly, but you should practice doing it, even on small tasks.	1,2	4,3	YES	
WC3D	It is good that you comment your code frequently. It will help as projects get bigger.	1,2	3,0	YES	
OUTPUT MESSAGE AS REQUESTED					YES
OM1	Getting small details like output content as requested is good programming practice.	2	0	NO	
OM2	Have you looked at the output content carefully. It is important to get small details like this correct.	2	0	NO	
USING VARIABLES IN OUTPUT					NO
OV1	Are sure you don't need to add any variables to your output.	2	1	NO	

C.5: Support object behaviour patterns

Calculate:average

`write:average_in_full`

R2: 0.851

Constant: -0.411

Facet weightings: E3:-0.009, O6:0.024

Behaviour pattern:

```
type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:maths_expression

type:maths_expression
```

The values used on this behaviour have been calculated with a previous set of groups. The behaviour pattern was identical. The correct values for this behaviour with this group set should have been as follows:

R2: .667

Constant: -.205

Facet weightings: E3: -.007, O6: .021

`write:average_some_variables_one_expresion`

R2: 0.711

Constant: -0.081

Facet weightings: E6:0.010, C6:-0.009

Behaviour pattern:

```
type:variable_dec
```

```
type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:maths_expression
```

write:average_one_variable_one_expression

R2: 0.382

Constant: -0.314

Facet weightings: C2:0.010

Behaviour patterns:

```
type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:variable_dec

type:maths_expression
```

Maths_expresion:in_output

part_maths:in_output

R2: 0.536

Constant: 0.430

Facet weightings: C6:-0.002, N5:-0.002, O1:-0.006, O4:0.001, E4:0.003

Behaviour pattern:

```
type:variable_dec_&_assign

type:maths_expression

type:output_statement

type:output_text
```

```
type:maths_expression
```

`all_maths:in_output`

R2: 0.382

Constant: -0.314

Facet weightings: C2:0.010

Behaviour pattern:

```
type:output_statement
type:output_text
type:maths_expression
```

`no_maths:in_output`

R2: 0.833

Constant: 0.013

Facet weightings: A5:0.010, C2:-0.013, C6:0.017, O1:0.008, O4:-0.008

Behaviour pattern:

```
type:variable_dec_&_assign
type:maths_expression
type:output_statement
type:output_text
type:variable_name
```

Write:comments

`write:few_comments`

R2: 0.324

Constant: 0.176

Facet weightings: O2:0.010

Behaviour pattern:

```
type:comment
```

`write:many_comments`

R2: 0.324

Constant: 0.824

Facet weightings: O2:-0.010

Behaviour pattern:

```
type:comment
type:comment
type:comment
type:comment
type:comment
```

Use:import_statement

`type:import_correct`

R2: 0.550

Constant: 2.153

Facet weightings: O6:-0.028

Behaviour pattern:

```
type:import_statement
type:reference_dec_assign
compile:code
```

`type:import_incorrectly`

R2: 0.550

Constant: -1.153

Facet weightings: O6:0.028

Behaviour pattern:

```
type:reference_dec_assign

compile:code

type:import_statement

compile:code
```

Write:pseudo-code

`write:pseudo-code_after`

R2: 0.624

Constant: 0.236

Facet weightings: N3:-0.006, O2:0.007

Behaviour pattern:

```
read:tasks

edit:code

read:tasks

write:pseudo-code
```

`write:pseudo-code_before`

R2: 1.000

Constant: 1.218

Facet weightings: E1:0.026, E5:-0.022, A2:-0.013, A3:-0.007, A4:-0.007, A6:-0.002, C1:0.004, N1:-0.014, N6:0.017

Behaviour pattern:

```
read:tasks

write:pseudo-code

edit:code

write:no_pseudo-code
```

`write:pseudo-code_during`

R2: 0.815

Constant: -0.658

Facet weightings: A1:0.003, C3:0.001, N1:-0.004, O6:0.014

Behaviour pattern:

```
read:tasks

edit:code

read:tasks

write:pseudo-code

edit:code
```

use:global_variables

`edit:main_make_local`

R2: 0.724

Constant: -0.487

Facet weightings: E2:-0.004, C4:-0.002, C5:0.015, O3:0.010

Behaviour pattern:

```
partial:global_variable_dec_&_assign

missing:modifier_static

compile:code

partial:main_method_declaration

compile:code

type:main_method_declaration

compile:code

type:variable_dec_&_assign

compile:code
```

`edit:main_add_static`

R2: 0.382

Constant: -0.314

Facet weightings: C2:0.010

Behaviour pattern:

```
partial:global_variable_dec_&_assign
missing:modifier_static
compile:code
partial:main_method_declaration
compile:code
type:main_method_declaration
compile:code
type:global_variable_dec_&_assign
type:modifier_static
compile:code
```

`add:modifiers_public_static`

R2: 0.394

Constant: 0.584

Facet weightings: E2:-0.008

Behaviour pattern:

```
partial:global_variable_dec_&_assign
missing:modifier_static
compile:code
partial:global_variable_dec_&_assign
type:modifier_public
compile:code
type:global_variable_dec_&_assign
type:modifier_static
compile:code
```

`dont_use:global_variables`

R2: 0.978

Constant: 0.576

Facet weightings: A6:0.008, N3:0.005, O3:-0.006, O5:-0.014

Behaviour pattern:

```
type:local_variabel_dec_&_assign

compile:code
```

use:primitive_types

`declare:correctly`

R2: 0.507

Constant: 0.829

Facet weightings: E4:-0.009, C3:-0.005

Behaviour pattern:

```
type:int_dec

type:int_dec

type:int_dec

type:int_dec

type:int_dec

type:double_dec

type:double_dec

compile:code
```

`declare:incorrectly`

R2: 0.552

Constant: 2.055

Facet weightings: A5:-0.008, O6:-0.018

Behaviour pattern:

```
type:int_dec
type:int_dec
type:int_dec
type:int_dec
type:int_dec
type:int_dec
type:int_dec
compile:code
```

declare:correctly_after_error

R^2: 0.815

Constant: -0.658

Facet weightings: A1:0.003, C3:0.001, N1:-0.004, O6:0.014

Behaviour pattern:

```
type:int_dec
type:int_dec
type:int_dec
type:int_dec
type:int_dec
type:int_dec
type:int_dec
compile:code
edit:double_dec
edit:double_dec
compile:code
```

Output:message

static_content:correct

R^2: 0.797

Constant: -0.263

Facet weightings: A6:0.012, C4:0.002, N5:-0.007, O6:0.009

Behaviour patterns:

```
type:output_statement

type:output_text

compile:code
```

static_content:incorrect

R2: 0.797

Constant: 1.263

Facet weightings: A6:-0.012, C4:-0.002, N5:0.007, O6:-0.009

Behaviour pattern:

```
type:output_statement

type:output_text_incorrect

compile:code
```

Output:variables

output:variables_correct

R2: 0.403

Constant: 0.194

Facet weightings: E3:0.009

Behaviour pattern:

```
type:output_statement

type:output_text

type:variable_name

compile:code
```

output:variables_correct

R2: 0.403

Constant: 0.906

Facet weightings: E3:-0.009

Behaviour pattern:

```
type:output_statement

type:output_text

compile:code
```

Appendix D: Statistics

D.1: Example correlation table

See Appendix F.1 for personality code values.

Use Primitive Types		Incorrect	Correct No Error	Correct After Error
Incorrect	Pearson Correlation	1	-0.802**	-0.408
	Sig. (1-tailed)		0.003	0.121
	N	10	10	10
Correct No Error	Pearson Correlation	-0.802**	1	-0.218
	Sig. (1-tailed)	0.003		0.272
	N	10	10	10
Correct After Error	Pearson Correlation	-0.408	-0.218	1
	Sig. (1-tailed)	0.121	0.272	
	N	10	10	10
E	Pearson Correlation	0.089	-0.384	0.441
	Sig. (1-tailed)	0.403	0.137	0.101
	N	10	10	10
A	Pearson Correlation	-0.193	-0.107	0.479
	Sig. (1-tailed)	0.296	0.384	0.081
	N	10	10	10
C	Pearson Correlation	0.292	-0.530	0.333
	Sig. (1-tailed)	0.207	0.058	0.173
	N	10	10	10
N	Pearson Correlation	-0.070	0.412	-0.515
	Sig. (1-tailed)	0.424	0.118	0.064
	N	10	10	10
O	Pearson Correlation	0.130	-0.020	-0.182
	Sig. (1-tailed)	0.360	0.478	0.308
	N	10	10	10
E1	Pearson Correlation	-0.069	-0.234	0.471

Use Primitive Types		Incorrect	Correct No Error	Correct After Error
	Sig. (1-tailed)	0.425	0.257	0.085
	N	10	10	10
E2	Pearson Correlation	0.056	-0.371	0.475
	Sig. (1-tailed)	0.439	0.146	0.083
	N	10	10	10
E3	Pearson Correlation	0.111	-0.295	0.270
	Sig. (1-tailed)	0.380	0.204	0.226
	N	10	10	10
E4	Pearson Correlation	0.516	-0.696*	0.219
	Sig. (1-tailed)	0.063	0.013	0.272
	N	10	10	10
E5	Pearson Correlation	-0.111	-0.168	0.437
	Sig. (1-tailed)	0.381	0.322	0.103
	N	10	10	10
E6	Pearson Correlation	-0.105	-0.118	0.351
	Sig. (1-tailed)	0.387	0.373	0.160
	N	10	10	10
A1	Pearson Correlation	-0.061	-0.386	0.690*
	Sig. (1-tailed)	0.433	0.135	0.014
	N	10	10	10
A2	Pearson Correlation	-0.070	-0.127	0.309
	Sig. (1-tailed)	0.423	0.363	0.192
	N	10	10	10
A3	Pearson Correlation	-0.042	-0.156	0.307
	Sig. (1-tailed)	0.454	0.333	0.194
	N	10	10	10

Use Primitive Types		Incorrect	Correct No Error	Correct After Error
A4	Pearson Correlation	-0.309	-0.012	0.523
	Sig. (1-tailed)	0.192	0.487	0.061
	N	10	10	10
A5	Pearson Correlation	-0.607*	0.438	0.322
	Sig. (1-tailed)	0.031	0.103	0.182
	N	10	10	10
A6	Pearson Correlation	0.328	-0.310	-0.061
	Sig. (1-tailed)	0.178	0.191	0.433
	N	10	10	10
C1	Pearson Correlation	0.313	-0.510	0.269
	Sig. (1-tailed)	0.190	0.066	0.227
	N	10	10	10
C2	Pearson Correlation	0.357	-0.512	0.199
	Sig. (1-tailed)	0.155	0.065	0.291
	N	10	10	10
C3	Pearson Correlation	0.174	-0.622*	0.667*
	Sig. (1-tailed)	0.316	0.027	0.018
	N	10	10	10
C4	Pearson Correlation	0.256	-0.295	0.032
	Sig. (1-tailed)	0.238	0.204	0.465
	N	10	10	10
C5	Pearson Correlation	0.254	-0.525	0.387
	Sig. (1-tailed)	0.240	0.060	0.135
	N	10	10	10
C6	Pearson Correlation	0.368	-0.525	0.200

Use Primitive Types		Incorrect	Correct No Error	Correct After Error
	Sig. (1-tailed)	0.148	0.060	0.290
	N	10	10	10
N1	Pearson Correlation	-0.052	0.428	-0.569*
	Sig. (1-tailed)	0.444	0.109	0.043
	N	10	10	10
N2	Pearson Correlation	0.075	0.273	-0.539
	Sig. (1-tailed)	0.419	0.223	0.054
	N	10	10	10
N3	Pearson Correlation	-0.197	0.343	-0.201
	Sig. (1-tailed)	0.292	0.166	0.289
	N	10	10	10
N4	Pearson Correlation	-0.127	0.414	-0.425
	Sig. (1-tailed)	0.363	0.117	0.110
	N	10	10	10
N5	Pearson Correlation	0.200	0.145	-0.549
	Sig. (1-tailed)	0.289	0.344	0.050
	N	10	10	10
N6	Pearson Correlation	-0.118	0.443	-0.485
	Sig. (1-tailed)	0.373	0.100	0.078
	N	10	10	10
O1	Pearson Correlation	-0.170	0.431	-0.381
	Sig. (1-tailed)	0.319	0.107	0.139
	N	10	10	10
O2	Pearson Correlation	0.224	-0.072	-0.255
	Sig. (1-tailed)	0.267	0.421	0.238
	N	10	10	10

Use Primitive Types		Incorrect	Correct No Error	Correct After Error
O3	Pearson Correlation	0.176	-0.075	-0.173
	Sig. (1-tailed)	0.314	0.418	0.317
	N	10	10	10
O4	Pearson Correlation	-0.016	-0.303	0.488
	Sig. (1-tailed)	0.483	0.197	0.076
	N	10	10	10
O5	Pearson Correlation	0.383	-0.158	-0.384
	Sig. (1-tailed)	0.137	0.331	0.136
	N	10	10	10
O6	Pearson Correlation	-0.584*	0.220	0.618*
	Sig. (1-tailed)	0.038	0.271	0.028
	N	10	10	10

**. Correlation is significant at the 0.01 level (1-tailed).

*. Correlation is significant at the 0.05 level (1-tailed).

D.2: Example regression table

See Appendix F.1 for personality code values.

Variables Entered/Removed(b)

Model	Variables Entered	Variables Removed	Method
1	O6, A5(a)	.	Enter

a. All requested variables entered. b. Dependent Variable: Incorrect

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.743	0.552	0.424	0.392

a. Predictors: (Constant), O6, A5

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	**Regression**	1.324	2	0.662	4.309	0.060
	Residual	1.076	7	0.154		
	Total	2.400	9			

a. Predictors: (Constant), O6, A5

b. Dependent Variable: Incorrect

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	2.055	0.586		3.509	0.010
	A5	-0.008	0.004	-0.479	-1.812	0.113
	O6	-0.018	0.011	-0.447	-1.692	0.134

a. Dependent Variable: Incorrect

D.3: Distribution across support object groups

Calculate:average

	5 Variables & 2 Expressions	5 Variables & One Expression	5 Variables & No expressions	No Variables & Two Expressions
PA01	1	0	0	0
PA02	0	1	0	0
PB01	1	0	0	0
PB02	0	0	1	0
PB03	0	1	0	0
PC01	0	0	0	1
PC02	0	1	0	0
PC03	1	0	0	0
PC04	1	0	0	0
PC05	1	0	0	0
Total	**5**	**3**	**1**	**1**

Maths_expresion:in_output

	Not in Output	All in Output	Part in Output
PA01	1	0	0
PA02	1	0	0
PB01	1	0	0
PB02	0	1	0
PB03	0	0	1
PC01	1	0	0
PC02	1	0	0
PC03	1	0	0
PC04	1	0	0
PC05	1	0	0
Total	**8**	**1**	**1**

Use:import_statement

	Incorrect	Correct
PA01	.	.
PA02	0	1
PB01	1	0
PB02	0	1
PB03	0	1
PC01	1	0
PC02	0	1
PC03	.	.
PC04	0	1
PC05	1	0
Total	**3**	**5**

Write:pseudo-code

	After	Before	None	During
PA01	1	0	0	0
PA02	0	0	1	0
PB01	0	0	0	1
PB02	1	0	0	0
PB03	0	1	0	0
PC01	0	1	0	0
PC02	0	1	0	0
PC03	0	1	0	0
PC04	0	1	0	0
PC05	0	1	0	0
Total	**2**	**6**	**1**	**1**

Use:global_variables

	Edit Main Local	Edit Main Static	Add Public Static	Don't Use	Use correct
PA01	0	0	0	0	1
PA02	0	1	0	0	0
PB01	0	0	0	1	0
PB02	1	0	0	0	0
PB03	0	1	0	0	0
PC01	0	0	0	1	0
PC02	0	0	0	1	0
PC03	0	0	0	1	0
PC04	0	0	0	1	0
PC05	0	0	1	0	0
Total	**1**	**2**	**1**	**5**	**1**

output:message

	Correct	Incorrect
PA01	1	0
PA02	.	.
PB01	1	0
PB02	0	1
PB03	.	.
PC01	1	0
PC02	0	1
PC03	0	1
PC04	0	1
PC05	1	0
Total	**4**	**4**

output:variables

	Correct	Incorrect
PA01	0	1
PA02	.	.
PB01	1	0
PB02	1	0
PB03	.	.
PC01	1	0
PC02	1	0
PC03	0	1
PC04	0	1
PC05	1	0
Total	**5**	**3**

Appendix E: Prototype design

E.1: Example of a tagged HTML page

Top portion of image indicates the page text that is associated with the action (red) focus (blue) attribute pair used to denote tasks and sub tasks.

Task 1: Write a program that:-

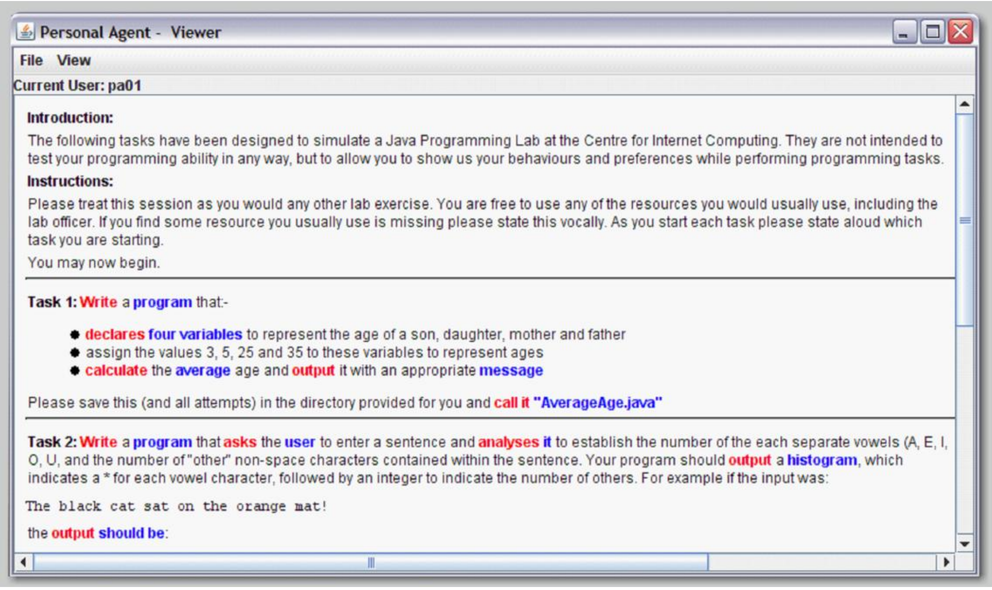
- declares four variables to represent the age of a son, daughter, mother and father
- assign the values 3, 5, 25 and 35 to these variables to represent ages
- calculate the average age and output it with an appropriate message

Please save this file (and all attempts) in the directory provided for you and call it "AverageAge.java"

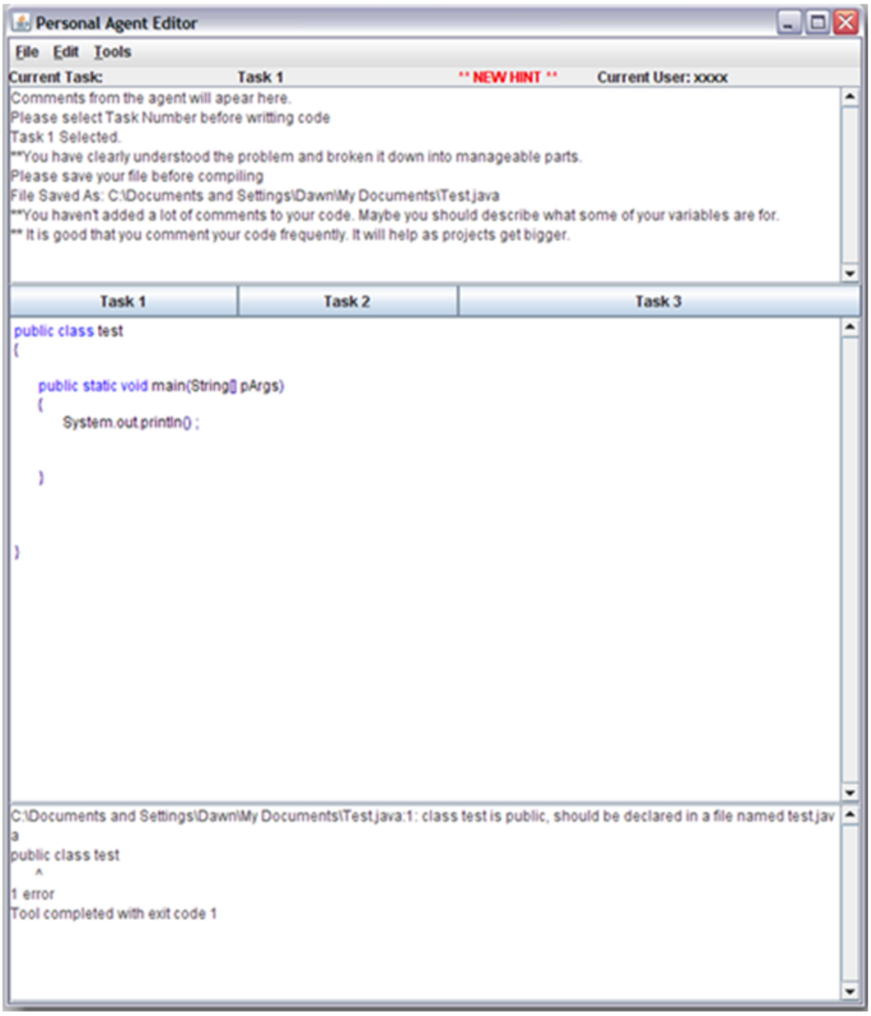
```
<tr><td><p><b><span class="taskBegin">Task 1</span>:</b> <span class="action">Write</span> a <span class="focus">program</span> that:-
</p>

<ul><li><span class="action">declares</span> <span class="focus">four variables</span> to represent the age of a son, daughter, mother and father </li>

<li><span class="action">assign</span> the <span class="focus">values</span> 3, 5, 25 and 35 to these variables to represent ages </li>

<li><span class="action">calculate</span> the <span class="focus">average</span> age and <span class="action">output</span> it with an appropriate <span class="focus">message</span></li>
```

E.2: Interface layouts

Web browser window



Editor window



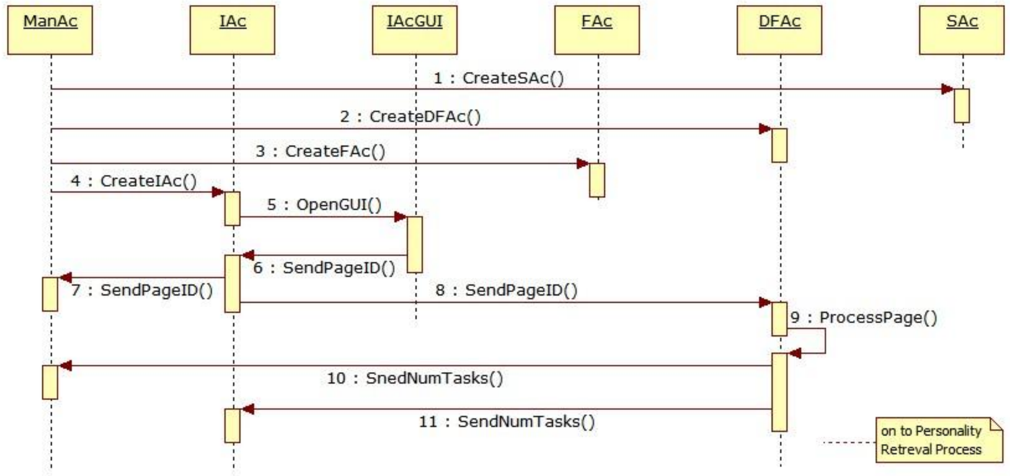
E.3: Agent communication protocol

FIPA Performative	Keyword	Other Content	Response Types	Used Between (sender – receiver)
QUERRY_IF	CREATE_AGENT	Agent Name	YES or NO	FFMAc – ManAc DFAc - ManAc
INFORM	DELETE_AGENT	Agent Name	none	FFMAc – ManAc DAAc - ManAc
INFORM	TASK_NUM	File Name ~ Number of Tasks	none	DFAc – ManAc DFAc - FAc
INFORM	OPEN	User Name	none	IAc - FAc
INFORM	LOST_FOCUS	none	none	IAc – FAc
INFORM	FOCUS	none	none	IAc – FAc
INFORM	None	Page Package	none	ManAc - SAc
INFORM	None	Page Package	none	SAc - FAc
INFORM	None	Task Package	none	FFMAc – ManAc DFAc – FFMAc
INFORM	None	Page Identifier Package	none	IAc – DFAc IAc - ManAc
INFORM	None	Instruction Package	none	DAAc – FFMac FFMac - DAAc

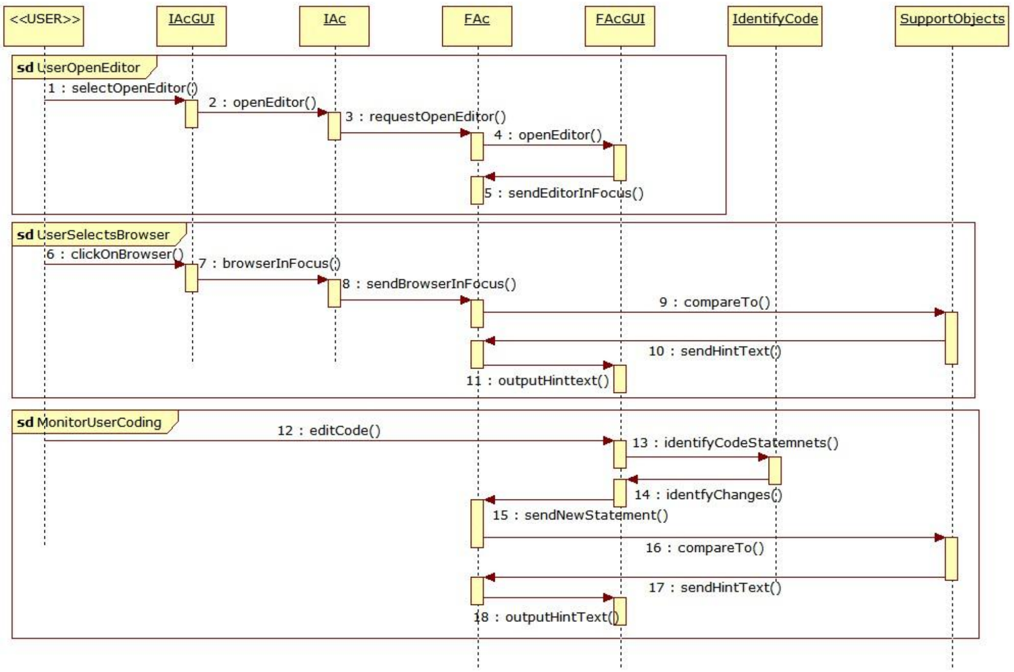
E.4: Agent interaction

The following diagrams show the main communications, agent creation and Java object passing within the prototype.

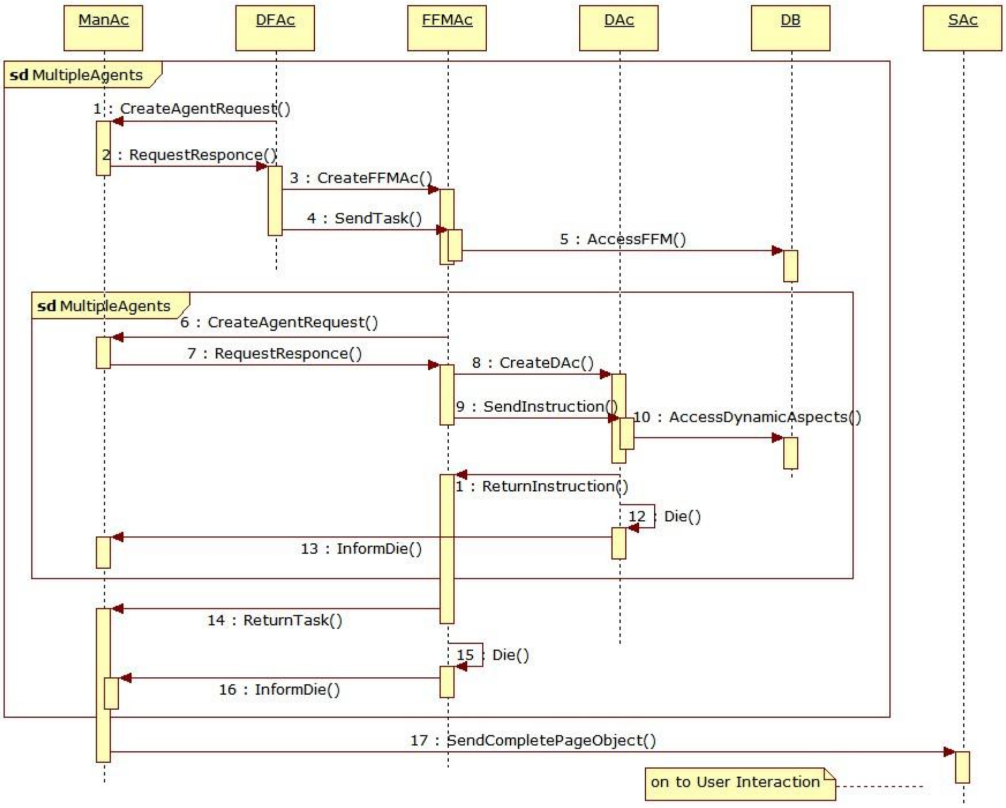
Agent initialisation



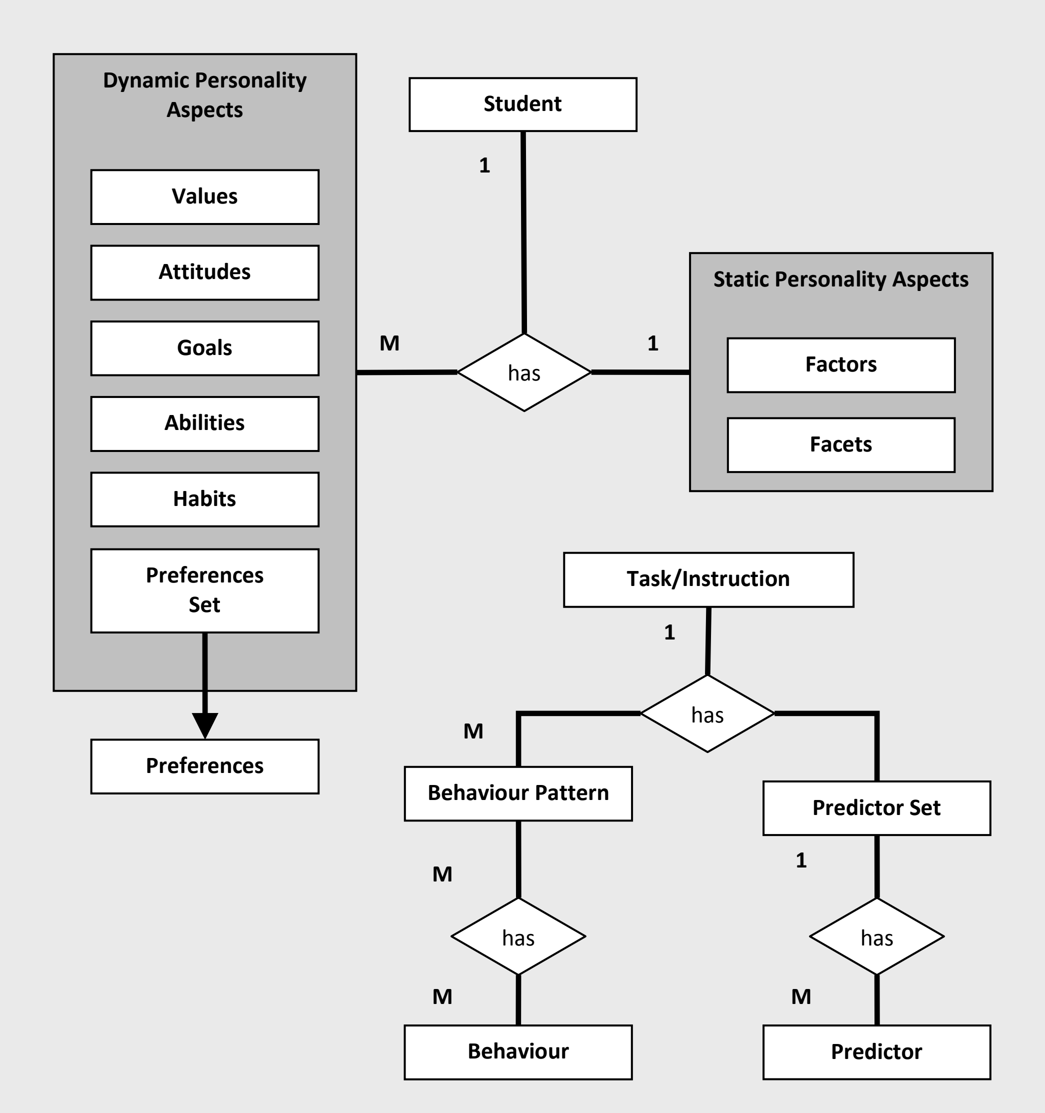
User Interaction



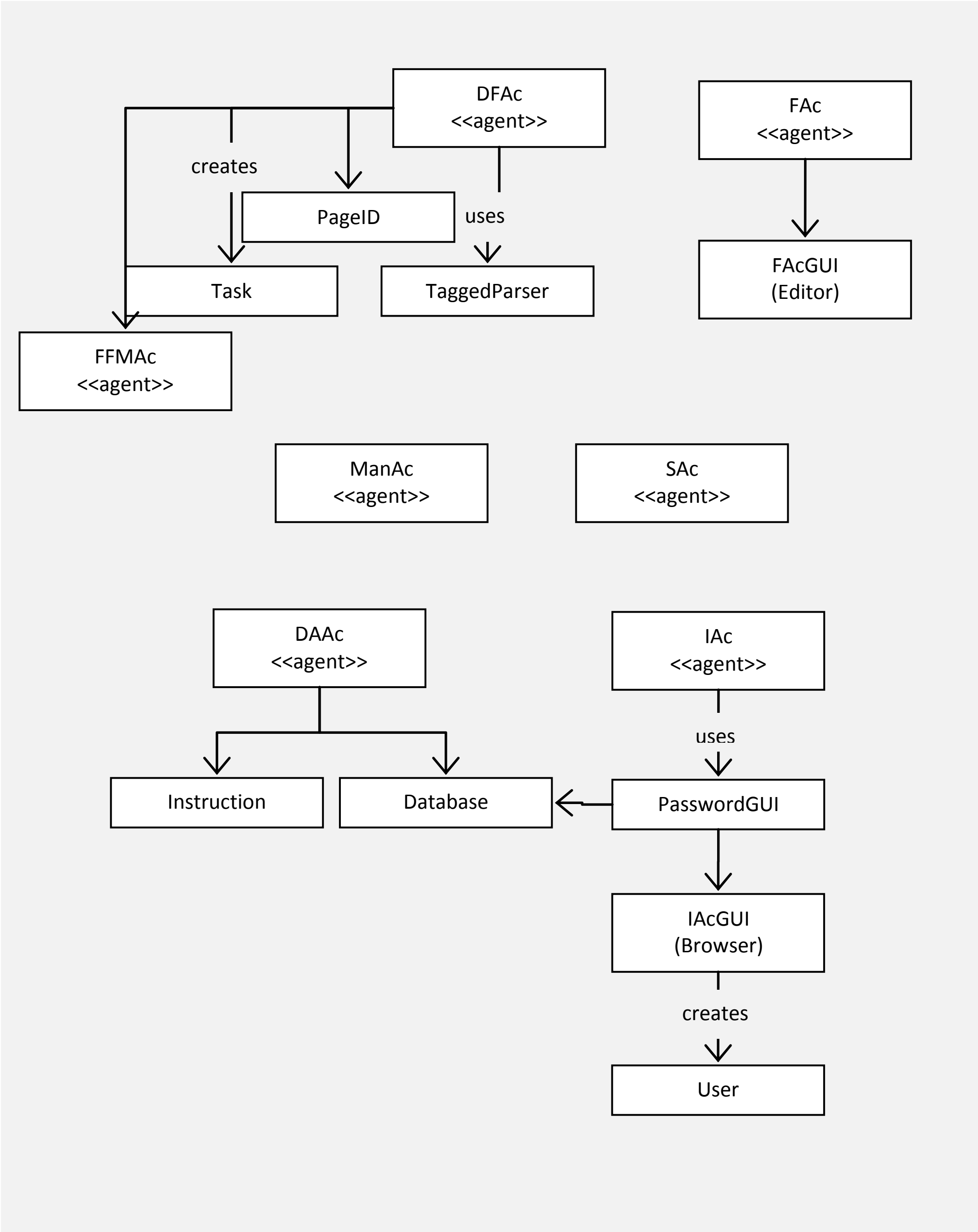
Personality retrieval process



E.5: Entity relationship diagram

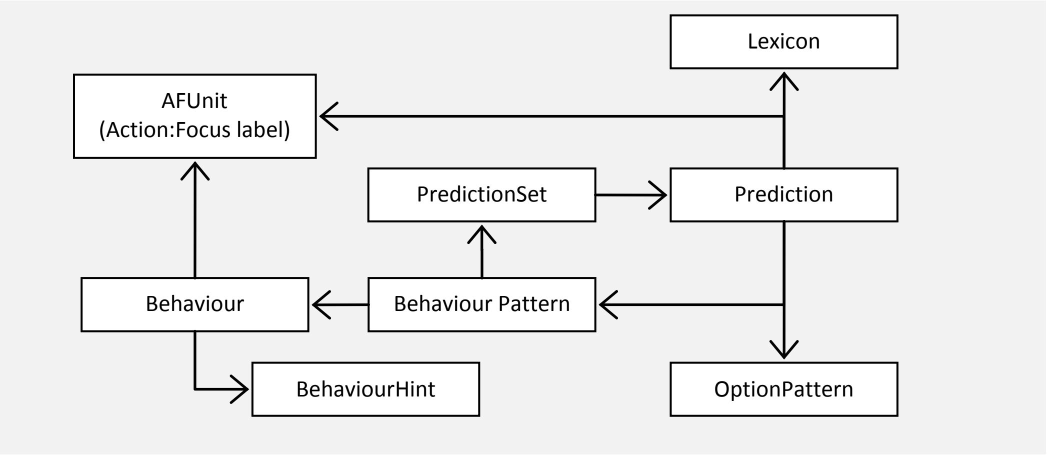


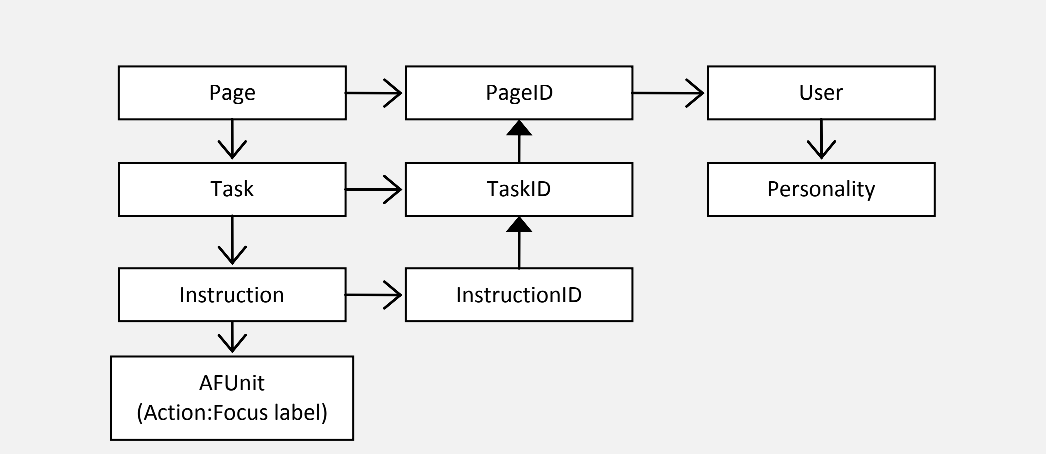
E.6: Agent class diagrams

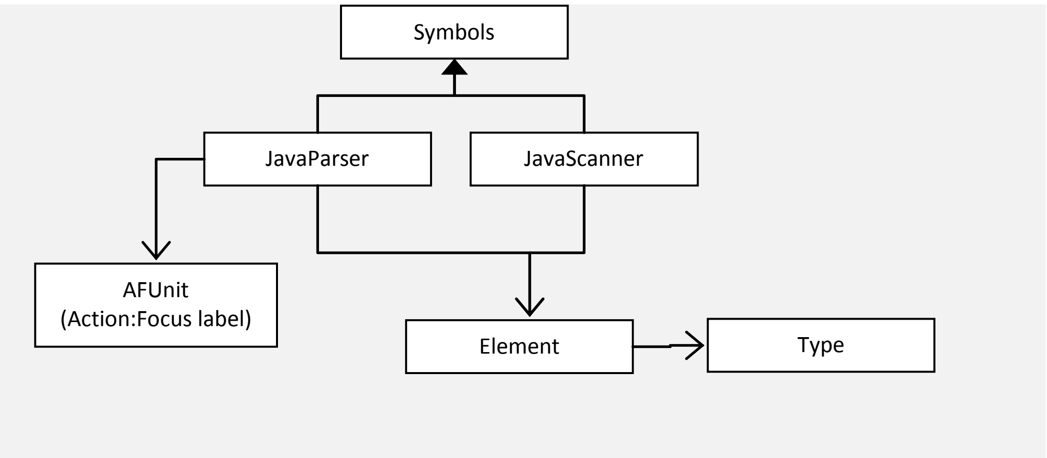


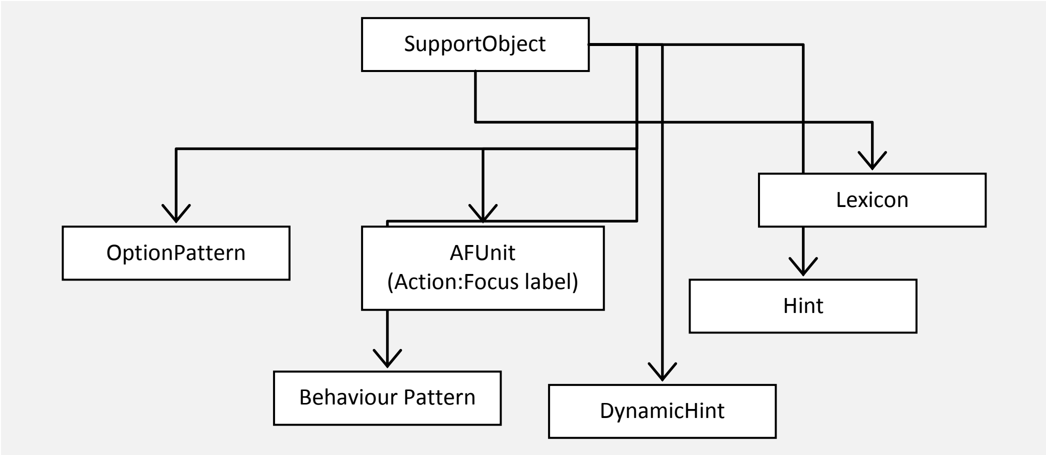
E.7: Object class diagrams

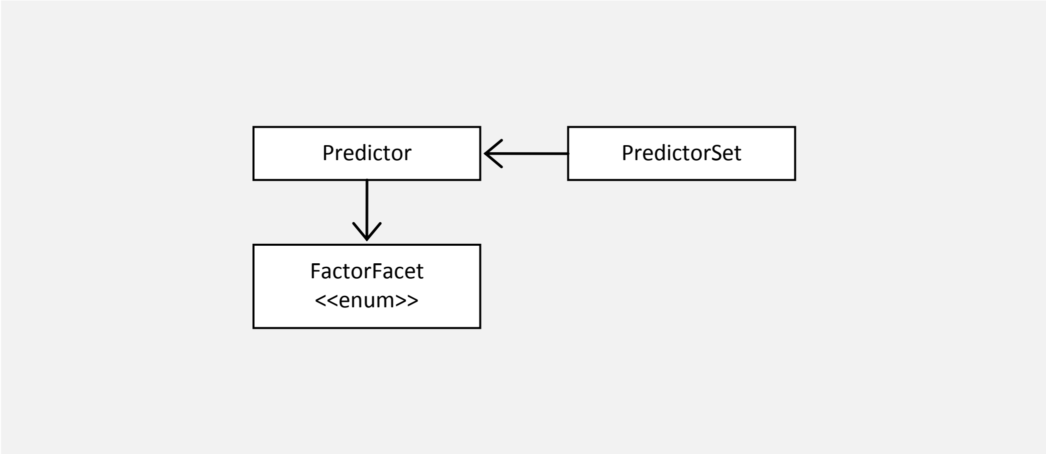
High-level diagram showing the object classes and their relationships, as implemented within the PReSS prototype system. The `Page` object is the primary data carrier around the system, backup at various points in a flat file format which can be captured at the end of a session. The page (representing the web page of tasks) is updated by the various agents responsible for gathering and processing particular aspects of data.





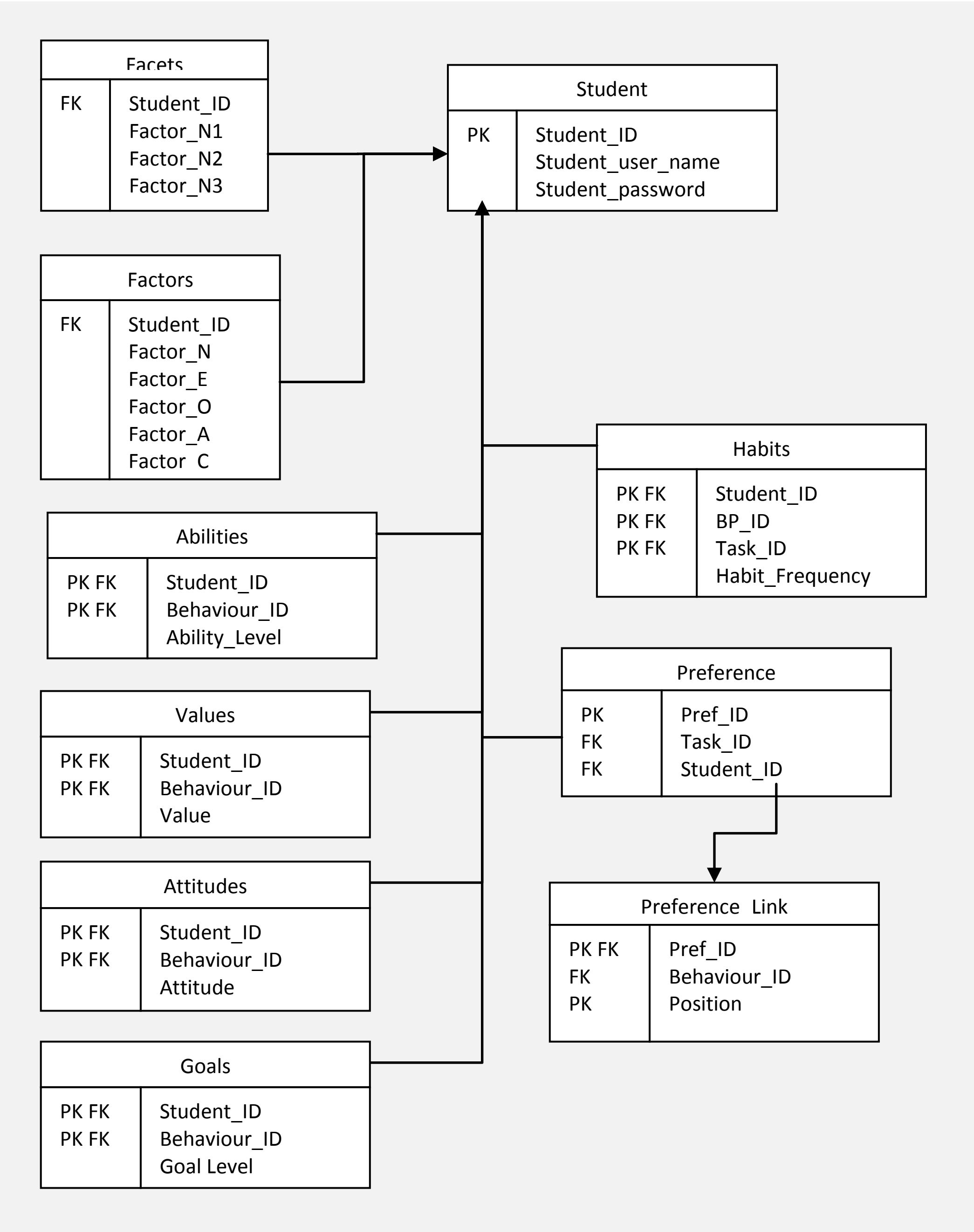



SupportObject
OptionPattern
AFUnit
(Action:Focus label)
Lexicon
Hint
Behaviour Pattern
DynamicHint

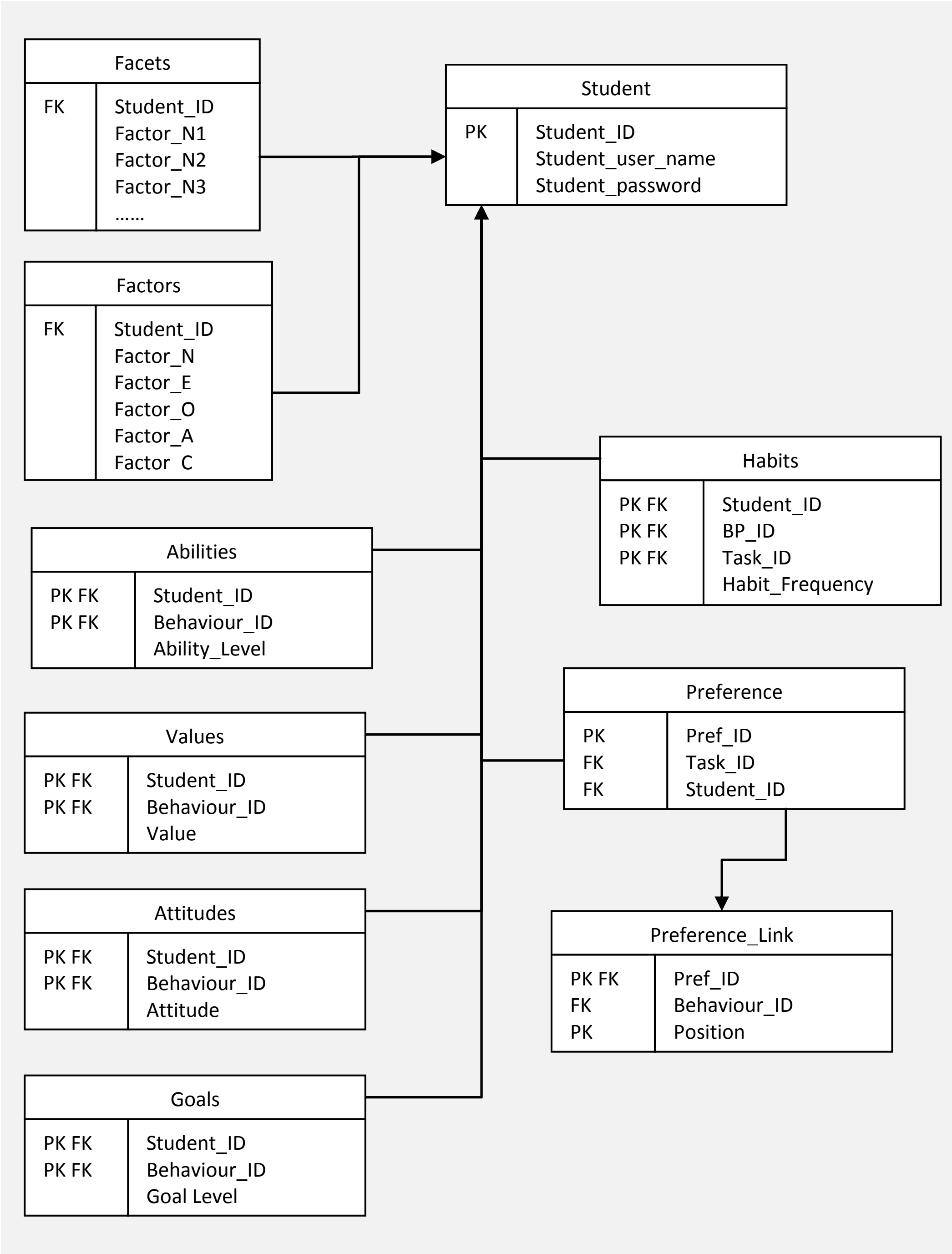

Predictor
PredictorSet
FactorFacet
<<enum>>

E.8: Database structure

Profile structure



Behaviour – prediction structure



Appendix F: Five Factor Model of personality

F.1: Factor and facet codes

Alphanumeric codes used to represent the factors and facets of the Five Factor model.

N NEUROTICISM

N1 Anxiety

N2 Anger

N3 Depression

N4 Self-Consciousness

N5 Immoderation

N6 Vulnerability

A AGREEABLENESS

A1 Trust

A2 Morality

A3 Altruism

A4 Cooperation

A5 Modesty

A6 Sympathy

E EXTRAVERSION

E1 Friendliness

E2 Gregariousness

E3 Assertiveness

E4 Activity Level

E5 Excitement-Seeking

E6 Cheerfulness

C CONSCIENTIOUSNESS

C1 Self-Efficacy

C2 Orderliness

C3 Dutifulness

C4 Achievement-Striving

C5 Self-Discipline

C6 Cautiousness

O OPENNESS TO EXPERIENCE

O1 Imagination

O2 Artistic Interests

O3 Emotionality

O4 Adventurousness

O5 Intellect

O6 Liberalism

F.2: Personality inventory results

	PA01	PA02	PB01	PB02	PB03	PC01	PC02	PC03	PC04	PC05
Cautiousness	33	0	48	55	76	63	1	0	21	26
Self-Discipline	32	61	65	27	73	60	6	14	23	7
Achievement-Striving	74	87	48	43	83	64	19	3	16	15
Dutifulness	33	27	80	49	45	53	3	0	33	16
Orderliness	40	49	54	78	59	57	20	23	29	17
Self-Efficacy	33	58	51	60	51	50	1	1	2	3
CONSCIENTIOUSNESS	38	42	61	53	72	62	0	0	11	4
Sympathy	86	95	54	59	54	48	38	19	48	79
Modesty	76	94	88	17	23	20	76	85	79	29
Cooperation	91	80	96	33	41	14	31	14	57	2
Altruism	78	93	82	82	63	29	63	0	35	17
Morality	80	65	65	30	65	12	10	27	46	35
Trust	51	78	99	33	56	57	42	29	66	25
AGREEABLENESS	89	97	96	38	50	20	39	14	60	20
Liberalism	68	37	79	38	50	58	50	63	58	61
Intellect	80	55	0	46	55	17	3	0	14	80
Adventurousness	45	20	71	57	76	34	25	6	9	20
Emotionality	47	86	32	64	68	3	63	18	12	64
Artistic Interests	68	8	11	81	43	1	15	8	24	77
Imagination	84	91	23	41	0	57	70	54	47	57
OPENNESS TO EXPERIENCE	75	49	26	57	41	13	28	9	16	66
Cheerfulness	67	96	81	67	47	51	72	19	26	42
Excitement-Seeking	41	99	88	75	37	60	64	41	25	21
Activity Level	23	40	58	39	84	83	2	14	43	22
Assertiveness	28	65	74	94	65	80	56	24	5	5
Gregariousness	53	99	99	66	87	68	57	39	52	17
Friendliness	82	91	99	67	78	53	58	13	45	21
EXTRAVERSION	51	98	97	77	74	72	52	17	25	12
Vulnerability	56	51	12	13	47	34	99	92	97	99
Immoderation	44	81	10	89	5	54	94	68	96	76
Self-Consciousness	73	32	16	20	46	18	77	69	93	97
Depression	17	17	33	14	47	48	89	91	85	64
Anger	38	57	7	12	61	38	94	72	77	99
Anxiety	84	18	4	32	56	37	84	56	87	93
NEUROTICISM	51	39	6	20	41	34	98	83	96	99