RFID-based user profiling of fashion preferences: blueprint for a smart wardrobe

Sea Ling* and Maria Indrawan

Faculty of Information Technology,
Monash University, Caulfield East 3145, Melbourne, Australia
E-mail: chris.ling@infotech.monash.edu.au
E-mail: maria.indrawan@infotech.monash.edu.au
*Corresponding author

Seng W. Loke

Department of Computer Science and Computer Engineering,
La Trobe University, Bundoora 3086, Melbourne, Australia
E-mail: s.loke@latrobe.edu.au

Abstract: In this paper, we explore the possibility of generating user profiles of fashion preferences from information captured by RFID technology. Proposing a design of a smart wardrobe, we investigate the appropriateness of the technology as an identification tool of real objects and to aid in detecting and tracking real objects movements. We then present a model of user fashion profile, which is generated through queries and data mining techniques. In order to illustrate the usefulness and real world feasibility of our proposed model, we build a working prototype as proof of concept. For evaluation purposes, we have created a random generator, which is able to generate random clothing items and dressing events that serve as input to our model for creating user profiles. Our experimental results clearly indicate that RFID technology is suitable to aid in creating smart systems.

Keywords: RFID technology; user profiling; smart wardrobe.

Reference to this paper should be made as follows: Ling, S., Indrawan, M. and Loke, S.W. (xxxx) 'RFID-based user profiling of fashion preferences: blueprint for a smart wardrobe', *Int. J. Internet Protocol Technology*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: S. Ling received his PhD in Computer Science from Monash University in 2000. He is currently a Senior Lecturer in the Faculty of Information Technology, Monash University, Australia. His current research interests include software engineering, pervasive computing and context-aware systems.

M. Indrawan is a Senior Lecturer of Computer Science in the Faculty of Information Technology, Monash University. She received her PhD in Computer Science from Monash University. Her current research focuses on pervasive computing, information retrieval and context-aware systems.

S. Loke is a Senior Lecturer of Computer Science in the Department of Computer Science, La Trobe University. He received his PhD in Computer Science from University of Melbourne. His research interests are pervasive computing, smart containers, social devices and context-aware pervasive systems.

1 Introduction

Nowadays, there has been a rapid proliferation of high tech devices. New technologies are no longer reserved for PDAs and mobile phones, but also for home devices, e.g., fridge. A fridge manufactured by LG Electronics (Harvey, 2002) is built with on-board monitor and also equipped with small processor unit. With the increase in popularity of online shopping, this fridge is also built with the capability to connect to the internet, which allows the user to shop online. Evidently, information technology has merged with human's daily surroundings. However, the usage of such devices can be taken further. For example, in the case of a fridge, it could be very valuable if the fridge can also automatically detect 'the milk is running out', and it would 'smartly' place an order online. Furthermore, it may be able to order milk of a particular brand that is preferred by the owner. This can only be done if a user profile is kept within the memory of this fridge. The proposed concept is one example of a smart application in a fridge. A smart application is distinguished from a normal application if it can provide more useful functionalities to its users by personalising its services to a specific user. As stated in Amato and Straccia (1999), a requirement to develop personalised services is to rely on a user profile that represents user preferences.

There are many ways to realise the idea of a personalised system. Current techniques used to create user profiles include content based filtering and collaborative filtering. These techniques are, however, mostly used in creating user profiles based on users' web access data. Content based filtering involves keyword search based on what previously have been selected by the user. Collaborative filtering uses data mining techniques based on multiple users' data. It gives recommendations based on similar patterns found between a specific user and other users.

These techniques are nonetheless not directly applicable to building a user profile for personalised smart system. Collaborative filtering requires information from other users' data, which is often unavailable during a person's day to day living. Content based filtering is based on a user's past history and to our knowledge, this technique has only been used to build user's web access profile.

In this paper, we propose a model that enables us to create a user profile automatically by observing user's behaviour. The aspect of a user's day-to-day living we chose to facilitate user profiling is the user's daily clothing (fashion) usage. In order to generate a user's fashion profile from his or her clothing selection, the system must first be able to identify what specific clothing items are worn by the user. The mechanism to identify these items is Radio Frequency Identification (RFID) technology.

Today, RFID technology is often used to integrate a virtual world with the physical environment (Want et al., 1999). As the cost of RFID components decreases tremendously, the technology is gradually becoming more popular.

Our exploratory research proposes a model of smart wardrobe that utilises RFID technology to identify and capture a user's dressing events. We then apply statistical methods and appropriate data mining techniques on the captured events to extract relevant information. Based on this information, the system will automatically generate a user's fashion profile. Using this profile, the system is then able to give the user recommendations regarding what items to wear for different situations.

2 Related work on RFID applications

Radio Frequency Identification (RFID) has been around since World War II. It was used to help soldiers identify fighter planes as friend or foe (Evans, 2004). Its usage in modern era is also for identification purpose. As stated in Vogt (2002a, 2002b), "Identification is a central concept in user-oriented and ubiquitous computing". RFID is used to identify real objects such as humans, books, as well as movement of these objects. There has been research on ubiquitous ("being or seeming to be everywhere at the same time", WordNet, 2003) computing, which utilises RFID system architecture to pursue the seamless integration of technology and human environment (Want et al., 1999). The same research objective was also undertaken by Floerkemeier and Lampe (2004), which presented the idea of having services provided by virtual worlds such as computers to assist users in their daily tasks.

Want et al. (1999) states that the RFID technology has been used for sometime. However, only simple prototypes were developed in many cases. This is due to the relatively high cost of implementation in the past. Recent development allows the reduction of the implementation cost of RFID systems which results in the emergence of many smart and useful applications. These applications include:

- *Tagged business card*. The application involves automatic detection of a person through the embedded RFID in the business card. When a reader senses the business card, it will trigger the user's computer to open the home page of the person represented in the business card (Want et al., 1999).
- *Tagged printed documents*. In this system, printed documents are tagged such that a document can be linked to its electronic copy for an up to date version so that readers are always aware of the version of the document (Want et al., 1999).
- *Tagged dictionary.* Once reader senses the tag, the system will perform a language translation of the document displayed (Want et al., 1999).
- *Smart toolbox.* There are laws in aircraft maintenance business that requires continuous checking and monitoring of tools used in the maintenance process to ensure quality and safety of aeroplanes. To date, these routines are done manually and thus time consuming and prone to human errors. Smart toolbox is proposed to automatically detect missing tools (Lampe and Strassner, 2003).
- *Smart medicine cabinet* (Floerkemeier et al., 2003). This system is designed to provide visual representation of RFID tracked medicine in the cabinet. The visual representation can be adjusted for visually impaired users. Example of the visual display is the packaging information of the medicine. Moreover, it also gives extra functionalities such as listing out expiry dates of all medicine in the cabinet. Thus user does not have to check it one by one manually.
- *Smart surgical kit* (Floerkemeier et al., 2003). Often doctors forget to remove some swabs and bandages that have been used in surgical operations. This prototype is proposed to continuously track and monitor how many RFID-tagged swabs and bandages have been used and whether all have been returned or disposed. Otherwise it will send out an alarm signal to doctors.

- *RFID chef* (Langheinrich et al., 2000). This prototype illustrates how an RFID mechanism can be incorporated into smart home. Apart from providing the users with easy access to information, it also provides the idea of context awareness. To support the information access and context awareness, grocery items are equipped with RFID tags. When a user places these groceries near a reader, the computer screen will display dynamically the current list of grocery items and a recipe list for dishes that can be prepared with the available ingredients. This prototype is a good example where RFID can be used to support daily human activities. An advancement of the RFID chef's capability has been mentioned in Römer et al. (2003). In this system, a user preference is added to the selection of recipes, e.g., vegetarian or Asian dishes. The system identifies the user from the tag that is attached to him or her. A similar idea is also proposed by Schneider (2003) where an application called Smart Shopping Assistant is designed to help the shopper by displaying various information regarding some products in a display unit. The shelves and the products are equipped with RFID reader and tags respectively so that the application can identify which product information to display.
- *Flytrap* (Crossen et al., 2002) is a smart system that will play appropriate songs based on the music preferences of occupants in a room. It has the knowledge of the occupants' musical tastes and will construct a play list of songs that will please everyone in the room. The system will construct each of the user's songs preferences by monitoring what music each individual person plays on his or her own computer. Each user has a RFID badge which enables the system to identify which user(s) are currently in the given room. Using the preference information of each user, the system will construct the play list. This system is a good example of how a system can be built for a truly ubiquitous environment. The presence of the system is seamless in the room but it is useful for everyone. In this system, RFID technology is also successfully used as an identification mechanism.

From these examples of prototypes and applications of RFID, it is evident that RFID technology is a simple, unobtrusive identification tool that can be used to track and monitor real objects. One area that has not been addressed by the sample prototypes is the utilisation of RFID to build personal profile. Personalisation of services is crucial for pursuing ubiquitous computing. In many cases, the collection of data to build user profile is done through WWW interactions. Example of such system is amazon.com. In this type of profiling system, the data collected is owned by the organisation that owns the server. It is not possible for individual to get the data for their personal use. RFID provides an alternative to the way data can be collected. It shifts the data collection process from the organisation to the individual, hence will shift the ownership of the data to the individual and enhanced privacy. In this research, we illustrate how and RFID system can be used to create individual-owned profile. We use the scenario of fashion profile which is created based on observing the users daily clothing choices.

3 User profiling

3.1 What is a user profile?

Today's world is congested with information especially from the Internet. With the vast amount of information available, it is very time consuming for people to search for their particular interests and to finally find what they are looking for. A mechanism or a tool could be useful to act on behalf of the users to filter the information available. The system must be able to distinguish what will be useful and informative to the users, so that a personalised service can be delivered to each user.

An observation by Hirsh et al. (2000) is that a personalised system can be built by recognising patterns of behaviour so that the system will be able to predict the user's next move. The simplest way is to learn from the user's past behaviour because usually a user exhibits regularity in his or her behaviour. This recognised pattern of a specific user will become the user's profile. "A profile is a description of user interests" (Wasfi, 1999). Adomavicius and Tuzhilin (2001) illustrate that a user profile consists of two components: the factual profile and the behaviour profile. The factual profile contains information such as name, gender, and date of birth, while the behaviour data consists of information on how the user will behave, e.g., "When purchasing an item X, a user will also buy item Y". Liu et al. (2003) concur that a profile consists of both user defined information and system detected user behaviour patterns.

According to Amato and Stracia (1999), a user profile is defined as "a representation of the preferences of any individual user". Therefore, an application should be able to rely on a user profile for information with regards to what the user likes and what the user needs.

In the following section, we will present some current techniques and technologies used in creating a user profile.

3.2 How to create user profiles

There are two common ways to create user profiles:

- *Traditional manual approach.* In a very simple system, the application designer can directly interview the users and then create the profiles based on the information collected. Each user will continually update the information about his or her preference information. This technique is known as the direct learning technique (Wasfi, 1999). The system is forced to directly learn by asking the users. However this method can be cumbersome as the number of users of can be very large and it can be too time consuming to collect the data. Furthermore, this technique is not appropriate when one attempts to create a smart application whose main goal is to recommend what will be useful for the user with minimal directions from the user.

- *Data mining.* An intelligent way to create user profiles is to use data mining techniques. Once data is collected, the system will use algorithms to discover patterns, i.e., rules from observing the data. These generated rules will be validated based on the validation rules from the domain expert. The validated rules are the properties of the user profile (Adomavicius and Tuzhilin, 2001).

In data mining, data can be information on one specific user or information on a group of users i.e., multiple users. When data of multiple users is used, the technique is called collaborative filtering; the system will search for other users that have similar tastes to a given user and then use their preferences as search criteria for the given user. The system assumes that the interests of other users will also be of interest of the current user since they had similar tastes in the past (Balabanović and Shoham, 1997). In other words, the collaborative filtering approach "computes similarities between the user profiles" (Chen and Chen, 2001). In this approach, users with similar profiles will be grouped together and thus each member will be given recommendations from other users' profiles in the same group.

In content-based filtering, the system will only retrieve information based on what has been selected previously by the specific user, i.e., what has been of interest to the user (Balabanović and Shoham, 1997). As explained by Chen and Chen (2001), information or data items that have been accessed by users in the past will be stored as user profiles and new data items or information will be compared and similarities found between this new information and the user profile will determine whether new information will be recommended.

It is quite often that collaborative and content-based filtering techniques are used in conjunction; this combination is called the hybrid technique. Like in the Fab system, which maintains user profiles based on content-based techniques, it also identifies similar users to make collaborative recommendations (Wasfi, 1999).

Content-based technique is mainly used in personalised web browsing because it uses keywords as matching criteria to make selection. On the other hand, collaborative filtering uses other users' information as the basis to make recommendations. In our work to create a smart wardrobe, since the focus is on identifying patterns in our collected data on clothing preferences, we choose to select the data mining technique. Moreover, information from other users is not needed and therefore collaborative filtering technique is not appropriate in this context.

Before we can apply any data mining technique to build user profiles, we must collect data about user's daily routines or events. There are many ways to capture and log this information. To this end, we propose the use of RFID technology which is used in many smart systems as an identification mechanism. However, in the context of our project, we have to discover what information can be captured and logged, taking into consideration also the information needed to build a user profile. From a user evaluation on project Gate Reminder (Kim et al., 2004), it is stated that "RFID received the best score in providing natural and transparent interaction" which enables a system to identify objects.

4 Smart wardrobe architecture

4.1 Architecture overview

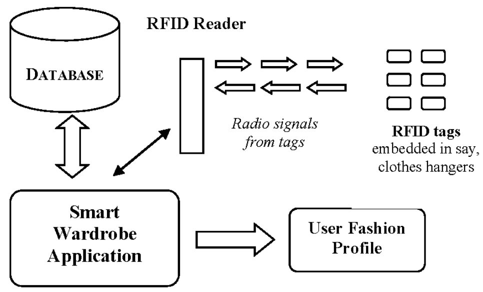
Our model, smart wardrobe is related to the Digital Wardrobe project (Mora, 2005). In her thesis, Mora suggested a model, making use of RFID technology to identify clothing items that are worn by the user. However, no experimentation with this technology is performed. Moreover, its functionalities are only limited to the display of clothing items that were used in the past.

Our model consists of four components:

- RFID technologies; reader installed in wardrobe and clothing items tagged with RFID tags
- smart wardrobe database
- smart wardrobe application
- generated user fashion profile.

As shown in Figure 1, the RFID reader is used to detect the presence or absence of clothing items inside the wardrobe. Therefore, each clothing item that the user has must be tagged with RFID tags. Each tag has a unique id, thus enabling the application to differentiate one item from another. The RFID reader is designed to periodically read all tags attached to clothing items in the wardrobe. Signals are sent to the smart wardrobe application if items are present (available) in the wardrobe.

Figure 1 Smart wardrobe architecture



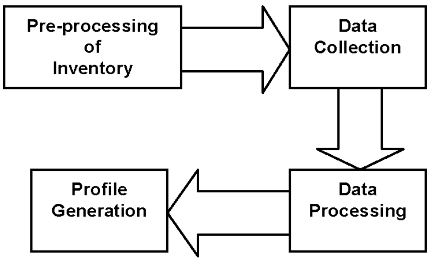
This application also utilises a database consisting of several tables; one table stores information on clothing attributes retrievable using the id; another table records the data generated by the user's wardrobe dressing activities.

The smart wardrobe application is the core processing component. It handles and processes the captured RFID events, which are used to identify the presence and absence of clothing items. The application will conclude an item is worn if it is absent from the wardrobe for a certain time period. Tags of items read by the reader will indicate to the application that these items are in the wardrobe and are not worn by the user. The application will then generate a user fashion profile that reflects the user's dressing preferences.

4.2 Architecture design

The system comprises four main phases: Pre-processing of inventory phase, data collection phase, data processing phase and profile generation phase (see Figure 2). Each phase will be explained below.

Figure 2 Four main phases of smart wardrobe architecture



4.2.1 Phase 1: Pre-processing of inventory

Each clothing item in the wardrobe is to be embedded with an RFID tag with a unique id from the manufacturer and it is essential to map this id information to attribute information of the clothing item. This information will be stored in a database. In brief, the process has two stages: entering information into database about the id as well as other attributes of the item; and assigning the tag to the item. RFID tags are available in many forms: key chains, buttons, etc. (www.phidgetsusa.com).

Initially, a suggested solution is to embed the tag into the clothes hanger. However, this approach has its limitation as to make the system work; the user must use the same hanger for the same clothing item in the wardrobe. A more convenient approach is to use washable RFID tags KSW-Microtec (http://www.ksw-microtec.de) according to McCullagh (2003). These tags can be embedded in the clothing label.

Mora (2005) has suggested that in USA, there has been an increase in demand of RFID chips as non-profit businesses and government organisations are considering implementing the technology. A breakthrough is the announcement made by WalMart, the leading department store in the USA, according to Foster and Scheepers,

> "WalMart announced in June 2003 that it would require its top 100 suppliers to equip incoming crates and palettes with RFID chips by January 2005 to link into WalMart's RFID infrastructure." (Foster and Scheepers, 2005)

A common use of the technology is in the area of supply chain management. It is however inevitable that RFID technology will soon be the new technology in our everyday lives (Evans, 2004). It is foreseeable that the RFID tag of a clothing item could be 'pre-recorded' by the clothing departmental store with id and information regarding the item. Therefore, our database for smart wardrobe home can simply utilise this 'pre-recorded' information, requiring minimal data input from the user.

Examples of clothing attributes to be stored are colour, pattern, material, type e.g., jacket, shirt, pants, etc; brand, purchase price, season e.g., summer or winter clothing. The system will generate user preferences based on this information. For example, during weekdays in the morning, user's preference is to wear shirt with pants. Table 1 shows a snapshot of clothing table in the database.

Table 1 Snapshot of clothing attributes stored in a table

id	*type*	*colour*	*brand*	*material*	*pattern*
000001	shirt	white	A	cotton	stripes
000002	skirt	black	B	polyester	plain
000003	pants	dark blue	C	polyester	plain
000004	jacket	brown	A	wool	plain
000005	jumper	red	A	cotton	flower

Depending on the occasion, people will wear different outfit. For example, working professionals will wear professional suit when they go to work; university student most of the time will wear casual wear such as t-shirt and jeans when they go on campuses. In this context, one can decide that an item can be categorised as formal wear, casual wear, eveningwear or uniform etc. However, it is possible for an item to be classified as formal as well as casual, depending on individual tastes. The user can decide these classifications during the initial data entry stage.

Once this information has been stored in a database and each of the clothing items has been assigned a RFID tag, we can now progress to the next phase.

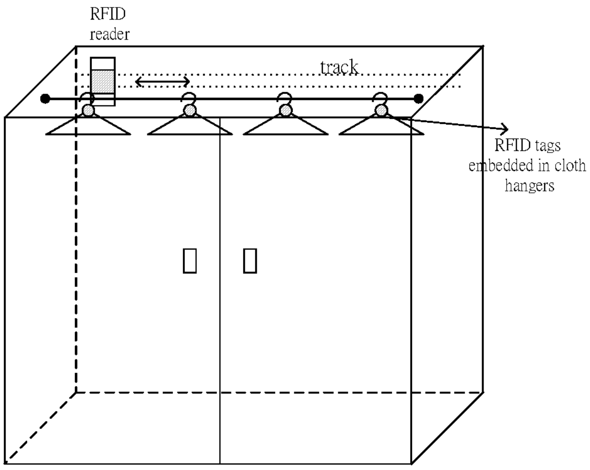
4.2.2 Phase 2: Data collection

Smart wardrobe utilises an RFID reader, which is to be installed near the wardrobe door to detect the movement of items being taken out or put back into the wardrobe. This phase is concerned with the collection of RFID signals that detected by the RFID reader. Specific processes need to be applied to the collected data (i.e., event data) in order to distinguish an event of removing a piece of clothing item from the wardrobe from an event of putting the item back into the wardrobe.

RFID readers are available in many forms. There are certain types which are only able to read one tag at a time, but there are also others which can read up to 30 tags within 100×100 mm area simultaneously. The reader in the RFID Chef project (Langheinrich et al., 2000) reads at once all tagged ingredients inside the shopping bag and then the application lists out recipes of possible dishes given the available ingredients on the display monitor. However, this type of reader is not appropriate for our system since the tag detection area to be covered in the wardrobe will be far greater than the area covered by this type of reader.

Intuitively, for our system, the general rule is that the reader must have a reading range, which covers the entire area of the wardrobe, when multiple tags RFID reader is used; or when a single tag reader is used, the selected RFID reader can have narrower reading range. In addition, the positioning of the reader in the wardrobe will also affect the selection of the reader, especially in the case where a single tag RFID reader is used. In such a case, the RFID reader will have to be put on a top of a motor, which is able to move on a track installed along the length of the wardrobe. The illustration of the model is shown in Figure 3. The moving reader will pass over the RFID tags and detect clothing items.

Figure 3 RFID technology installation in wardrobe



It should be noted that a design with multiple RFID readers will certainly take less time to read the tags compared to a wardrobe with a single RFID reader. For a design with single reader, the application has to wait for a longer duration to obtain the information since the reader has to move over all the clothings and notify the application each time it reads a tag.

Information (tag id) retrieved by the reader, although remotely identifies the presence of an object (Foster and Scheepers, 2005), is essentially meaningless to an application. There is a need to process it further in order to make sense of this information for the application. The presence of a tag should be transformed into some higher-level inferences that can be utilised to achieve the goal of the application. In the smart wardrobe, tag presence infers that the corresponding item is in the wardrobe, i.e., not being worn by the user. Moreover, when a tag is present at a certain time, but not present at a later time will indicate that the corresponding item has been taken out of the wardrobe. If this tag is not present for a longer period, it is reasonable to conclude that the item is being worn. These conclusions can be inferred from the captured RFID events for the application.

In our proposed model, there will be an RFID reader installed near the wardrobe door or in a place where the reader will be able to detect the tags inside the wardrobe. For a single reader, a possible implementation is that the wardrobe must be equipped with a motor and reel installed along the length of the wardrobe and the reader positioned on top of the motor such that the reader is able to move along the reel to read the tags inside the wardrobe one by one, as shown in Figure 3.

Once the reader is installed, it needs to be started so that the reader can start moving along the reel to read tags inside the wardrobe, i.e., polling for data, one tag at a time. This polling can be scheduled to happen periodically. For example, every two minutes, the reader will be activated to start polling i.e., reading radio frequency signal that is reflected back from the tag within the range. The reader signals and notifies the application with unique ID information of which tags have been successfully read. From this unique ID, the application is able to cross match the read id with the corresponding clothing item, enabling it to identify which items are currently in the wardrobe. The process of reading tags will be done periodically within certain time periods set by the user.

In between each reading, the application compares the data received in the previous reading and data received in current reading. If there are one or more tags that are present in previous reading, but absent in the current reading, the application will decide that those item(s) have been taken out by the user. Conversely, if the reader reads some tags present in the current reading but not previously, the application will conclude that the item(s) has been returned to wardrobe. Table 2 illustrates this scenario.

Table 2 Illustration of RFID readings

Reading	*Tags read*	*In wardrobe*	*Out of wardrobe*
1	001, 002, 003	001, 002, 003	–
2	001, 002	001, 002	003
3	001, 002	001, 002	003
4	002	002	001, 003
5	002	002	001, 003

The scenario starts with every item in the wardrobe. However, in between first and second reading, the user removes item 003. This causes the reader only to read two tags in the second reading. Therefore, the application can conclude that the item 003 is taken out. Further, in the third reading, the tag 003 is being read, but not item 001. After this round, the application decides that item 001 has been removed from the wardrobe, and after the fourth reading, all items are back in the wardrobe.

The application uses the following algorithm (Figure 4) below to determine whether the user takes out clothing items from the wardrobe.

Figure 4 Event data collection algorithm

```
Algorithm 1:
Deciding whether item is INSIDE/not INSIDE
the wardrobe

Let T = {t_1,t_2, ... , t_n} represents the set
of items which are tagged with n = number of
tags

Let P = {p_1,p_2, ... , p_∞} represent
collections of RFID polling during the life
time of the application.

Let d be the time duration (in seconds/mins)
between one polling and the next such that d
= time of p_j - time of p_j-1, for
j∈{1,2,..,∞}.

Let t[n,∞] be the collections of tags
states such that t[i,j]=0 when t_i is OUT
during polling p_j or t[i,j]=1 when tag t_i is
INSIDE during polling p_j for t_i∈T and p_j∈P

Let S_r^2 be the collection of tags that are
read during polling p_j

Let S_r^1 be the collection of tags that are
read during polling p_j-1

Initialization:
 1                  S_r^1 := {}
 2                  S_r^2 := {}
 3                  j := 1
 4     BEGIN
 5           WHILE (true)
             {
 6              start polling p_j
 7              during polling
 8              {
 9                 IF tag t_i is read, t_i∈T
10                 S_r^2 := S_r^2 ∪ {t_i}
11              }
12              FOR each t_i∈T
13              {
14                 IF t_i∈S_r^2
15       // item t_i is inside the wardrobe
16                    t[i,j] := 1
17                 IF t_i∉S_r^2 AND t_i∈S_r^1
18       // item t_i is taken out of wardrobe
19                    t[i,j] := 0
20                 }
21              S_r^1 := S_r^2
22              S_r^2 := {}
23              j := j+1
24           }
```

At the start of the application, the application initialises the data storage that is used to store the tag ids read by the reader (lines 1 and 2); variable *j* represents the reading number. During polling, every tag id that is read by the reader will be added to the list S_r^2 (line 10). Line 12–20 compares the list of tags ids that are read in the current polling S_r^2 with the list from the previous polling S_r^1. If the tag is present, it will be concluded as being inside the wardrobe (line 16). The status of the tag, whether it is inside or not inside the wardrobe, is stored in a two dimensional array $t[i, j]$ with *i* representing the tag id and *j* representing the reading number during which the status of tag *i* occurs. The value of this array is set to 1 to denote the item is present.

However, when a tag is not present during the current reading, but present in a previous polling, the application will conclude that the corresponding item is taken out of the wardrobe. Line 17 shows this comparison. Therefore, in line 19, the value of the $t[i, j]$ is set to 0 to denote the absence of the tag during polling *j*.

The application will continuously execute these steps for the duration of the application run. However, when an item is concluded as being taken out, it does not mean the user is wearing it. Consider the scenario where the user takes an item out of wardrobe but decides not to wear it and puts it back into the wardrobe within a certain time frame; the application should be able to differentiate this event from other events, which the user decides to wear the item he or she takes out. This is accomplished in the next phase: Data Processing.

4.2.3 Phase 3: Data processing

In the previous phase, the application has identified whether each item is currently inside the wardrobe or has been taken out of the wardrobe. The result of the previous phase is a two dimensional array $t[i, j]$ that records the status of each tag during each polling period.

The Data Processing phase consists of two main procedures: identifying whether an item is actually worn and from this collection of worn items, deciding which items are worn together.

4.2.3.1 Identification of whether an item is worn

We need to distinguish between the event in which an item is just removed for a short time and then returned back to the wardrobe and the event in which an item is taken out for a much longer duration i.e., being worn by the user. The application uses a time out mechanism. For example, if the item is taken out, i.e., not being read during polling for certain time period, e.g., 60 minutes; then application concludes that this particular item is worn by the user. However, if during this time limit, the item is put back i.e., is present to be read during the polling, the application will conclude that this particular item is not worn.

Table 3 illustrates the value of the array $t[i, j]$ after six polling periods from the previous scenario described in Table 2.

Table 3 Snapshot of $t[i, j]$ after readings illustrated in Table 2

Polling no.	*Tag 001*	*Tag 002*	*Tag 003*
1	1	1	1
2	1	1	0
3	1	1	0
4	0	1	0
5	0	1	0
6	0	1	0

After the first polling, all tags are present, i.e., $t[001, 1] = 1$. At the time of polling 2, $t[003, 2] = 0$ denotes that the corresponding item of tag 003 is not present in the wardrobe, i.e., has been taken out by the user. Subsequently, at time polling 3, $t[001, 4] = 0$ denotes that during this polling period, the user has taken out the item with tag 001. Tag 003 is subsequently not present from pollings 2 to 6; and tag 001 is subsequently not present from pollings 4 to 6.

The steps taken by the application to make these findings is described in the algorithm below (Figure 5).

Figure 5 Event data processing algorithm: Part 1

```
Algorithm 2:
Deciding whether item is WORN / NOT WORN

Let W be the set of collections of tag and
polling time (t_i,j) such that tag t_i is
concluded as being worn at time of polling
p_j.

Let m be the number of consecutive pollings
after which every tag that is not present
during this number of consecutive pollings
has its associated item declared as having
been worn.

Initialisation: W = {}

BEGIN
     WHILE (true)
     {
        FOR each t_i ∈ T
        {
  // refer to Figure 3 for explanation of P
            IF  t[i,j] = 0 where
                       j ∈ {1,2,..,size of P}
            {
               FOR  k:= 1 to m
                  status_i^k := t[i,j+k]

               IF all status_i^k ,status_i^(k+1), ..
                                ,status_i^m = 0
     // item t_i is declared worn at time of
                                 polling p_j
                  W := W ∪ {(t_i,j)}

            }
        }
     }
```

4.2.3.2 Deciding which items are worn together

One example is in the morning the user wears a shirt with a pair of pants for work. To identify this kind of pairing, the application again utilises a time boundary mechanism. For example, an item is taken out at time 10.00 and a second item is taken out at time 10.20. Therefore, the time difference between the taking out of these two items is 20 minutes. If the time difference allowed between the taking out of the two items such that these two items will be concluded as being worn together is 30 minutes, then the application will conclude that these two items have been worn together.

Expanding on Table 3, Table 4 illustrates a scenario where corresponding items tagged 001 and 003 are worn together.

Table 4 Expanded version of Table 3, illustrating items that are worn together

Polling no.	*Tag 001*	*Tag 002*	*Tag 003*
1	1	1	1
2	1	1	0
3	1	1	0
4	0	1	0
5	0	1	0
6	0	1	0
7	0	1	0
8	0	1	0
9	0	1	0
10	0	1	0
11	0	1	0
12	0	1	0

The following algorithm (Figure 6) describes the steps taken to discover pairs of items that are worn together.

Figure 6 Events data processing algorithm: Part 2

```
Algorithm 3:
Selecting which items are worn together

Let h be the allowed time duration between
two polling times such that two items that
are concluded as worn can be declared as
being worn together.

Let WT be the set of pairs of items that are
worn together.

Initialization: WT = {}

BEGIN

// W is the set of tags and polling time of
// worn items (refer to Figure 5)

      WHILE size of W ≥ 2
      {
            FOR each (t_i,j) ∈ W
            {
            Find (t_x,y) ∈ W where y ≤ j+h

            IF found
      // item t_i and item t_x are worn
       // together
               WT := WT ∪ {(t_i,t_x)}

            Remove t_i from W
            Remove t_x from W
            }
      }
```

Variable h is calculated by dividing a time boundary with the time duration between two consecutive polling. For example if the time limit is set at 30 minutes and the time duration between pollings is set at 10 minutes, therefore h is 3; which means the two items must be taken out either during the same polling or during the next three consecutive pollings in order to conclude that these two items are being worn together.

Using the result from the previous phase, this algorithm checks if there are at least two items concluded as being worn (line 1). For the first item that is concluded as worn, the application will try to find the corresponding match for this item by checking the polling period in which the other item is taken out (line 5). In this context, we assume that at any one time, the user wears only two items, top and bottom items. If the application can find another item that meets the criteria, i.e., also taken out during the time allowed, then this pair will be added to list *WT* as a pair of clothing items worn together (line 8). These items are then removed from the list of worn items (lines 9–10).

The application will also make database entries into a table with this information as well as the time of usage, i.e., the day and the time. Generally, it is assumed that there are two categories of time periods: morning and evening. Every dressing event that occurs between 6 a.m. to 5 p.m. will be recorded as a morning event, whereas if it occurs between 5 p.m. and 6 a.m., it is categorised as an evening event. We use this categorisation to differentiate between morning and evening outfit.

Table 5 shows a snapshot of database entries of pair of items being worn.

Table 5 Usage data

Event ID	*Worn items ID*	*Day*	*Time*
0	001, 003	1	MORNING
1	005, 002	1	EVENING
2	004, 008	2	MORNING
3	006, 009	2	EVENING
4	002, 007	3	MORNING
5	011, 015	3	EVENING

Once we have the usage data of user dress events, the application is now ready to use the result to generate user fashion profiles. In fact, each of the records in the database that represent the usage of clothing items in the inventory can be considered as a single transaction and the whole database as a data set that can be mined during the data mining process. The data processing conducted in this phase can be considered as data preparation in the data mining process.

4.2.4 Phase 4: User profile generation

There are two main sources of data to be used in generating user fashion profiles: the inventory data (Table 1) and the usage data (Table 5).

From the information, the application is able to analyse and generate this part of the user's fashion profile (Figure 7).

In this example, the left column shows that the user has 100 clothing items, broken down into various attributes: ten pair of pants and ten shirts; based on colour, the user has 20 blue items and 15 black items; based on pattern, user has ten plain pattern items and five striped items; and so on. On the right column, it shows the favourites from each category, i.e., favourite brand is A, favourite colour is blue, favourite pattern is plain and favourite material is cotton. To generate the above user profile, simple SQL queries are used to query the inventory.

Figure 7 A snapshot of a user fashion profile: Part 1

Inventory:
Total: 100
Number of items (Type based):
Pants: 10
Shirts: 10 …
Number of items (Brand based):
A: 10
B: 5 …
Number of items (Colour based):
Blue: 20
Black: 15 …
Number of items (Category based):
Formal: 20
Casual: 30 …
Number of items (Pattern based):
Plain: 10
Stripes: 5…
Number of items (Material based):
Cotton: 20
Nylon: 15…

Favourites:
Brand: A
Colour: Blue
Pattern: Plain
Material: Cotton

The second part of the user fashion profile (Table 6) is generated based on the usage data generated by the events detected by the RFID reader, as discussed in the previous phases. It is necessary to analyse the usage data because there are times when what the user has in the inventory does not necessarily imply what he or she likes. Therefore, the application is designed to generate the profile from two different perspectives.

Table 6 shows that the most frequently worn brand is brand B, most frequently worn colour is red and so on. The bottom half of the profile generated based on usage statistics of individual item; for example, it shows that item 1 is worn most often. In addition, it also shows which items are most often worn together at certain time period. For example, item 1 and item 20 are the most often used combination during daytime and so on. As before, the application uses SQL queries to query the usage data table (Table 5) to generate the profile.

Table 6 A snapshot of user fashion profile: Part 2

Usage statistics
Most frequently worn:
Brand: B
Colour: Red
Pattern: Plain
Material: Cotton
Most frequently worn: item 1
Most in-frequently worn: item 10
Most frequent combination:
Daytime: item 1 and item 20
Weekend: item 2 and item 10
Evening: item 11 and item 15

Another user profile generated by the application is the collection of usage combination pattern. Combination patterns can be found from the way the user dresses such as the colour combination, pattern combination, and type combination. In our scenario, these combinations suggest the user's personal preference on how the user matches his or her top item with his or her bottom item.

An example is shown below.

Table 7 Usage combination patterns

Colour
1. White and black
2. Blue and black
3. Pink and blue
Pattern
1. Plain and plain
2. Stripes and plain
3. Stripes and stripes
Types
1. Shirt and pants
2. T-shirt and jeans
3. Shirt and skirt

Table 7 shows the three most often combinations from each category. The most common colour combination is white top and black bottom; the most common pattern combination is plain top and bottom; and the most common type combination is to wear shirt and pants.

This above is generated by applying data mining algorithms to usage data. In this instance, we use association rules to capture patterns. The result is useful for the application to make recommendations to the user. For example, when a user has selected a shirt, it will recommend pants with it since the combination pattern derived from the usage data suggests the user often wears shirt with pants (rather than jeans).

4.3 Data mining

In the generated profile, the combination patterns reveal how the user likes to match his or her outfits; for example: t-shirt with jeans; white top with blue pants; stripes top with plain bottom; etc. The information is derived using data mining techniques based on association rules mining.

In our prototype, we choose to use Weka, a system written in Java that provides algorithms to perform data mining techniques on a set of data. It stands for the Waikato Environment for Knowledge Analysis, developed at the University of Waikato in New Zealand (Witten and Frank, 1999). These algorithms are available for free on the WWW (www.cs.waikato.ac.nz/ml/weka).

In performing the association rules mining, the generated data collected during the data processing is used as the data set. In our simulated system, the data processing of the system creates two entries for each day, being the morning and evening. We generate usage records or transactions over a two week period. In total we have 56 transaction records in our data set. The number of transactions can be easily increased by the simulator, however for the purpose of our experiments we consider four weeks worth of usage data. A snapshot of the data set is depicted in Figure 8. We call each of the generated set of transaction as event.

Figure 8 Statistics of usage: working professional

Events Table:

Day	Time	TopID	TopType	BottomID	BottomType
1	MORNING	23	shirt	20	skirt
2	MORNING	22	jacket	2	pants
3	MORNING	3	sweater	6	skirt
4	MORNING	9	cardigan	19	skirt
5	MORNING	18	jacket	7	skirt
1	MORNING	16	sweater	2	pants
2	MORNING	11	cardigan	7	skirt
3	MORNING	15	jacket	4	pants
4	MORNING	18	jacket	4	pants
5	MORNING	10	cardigan	20	skirt
1	MORNING	18	jacket	2	pants
2	MORNING	9	cardigan	21	skirt
3	MORNING	16	sweater	19	skirt
4	MORNING	8	cardigan	7	skirt
5	MORNING	5	cardigan	4	pants
1	MORNING	11	cardigan	4	pants
2	MORNING	18	jacket	2	pants
3	MORNING	10	cardigan	25	skirt
4	MORNING	22	jacket	7	skirt
5	MORNING	22	jacket	25	skirt

The Apriori algorithm in WEKA was chosen to generate the combination patterns. Since we are only interested in discovering the combination patterns of the 'top' and 'bottom' items, we limit the number of attributes to two in the event record. The two attributes are TopID and BottomID. The support is set to be 80%.

5 Implementation of prototype

A prototype of the smart wardrobe application has been built. Random generated data were populated into the event data storage. We created simulation data from two different types of user, a working professional and a student. These two categories are chosen because these types of users may be considered to be different in the way they dress. The idea is to test whether the application can appropriately generate the profile.

When creating this simulation data, some assumptions are made:

- The inventory generator creates only clothing items for females. We chose to generate female data because there is broader range of female clothing items compared to mens wear.
- All clothing items inside a single wardrobe belong to a single user.
- All clothing items inside the wardrobe are tagged with RFID tags.
- When one decides not to wear the item, one will always put the item back into the wardrobe. Therefore, application can be certain that the user wears clothing items that have been taken out of the wardrobe for a long time period.

Based on the simulated data, the system generates profile of the inventory statistics, usage statistics and usage combination patterns. Figure 8 shows the statistic of usage of a working professional which is summarised in Figure 9 as a pie chart for his or her preference on weekday mornings.

Figure 9 Usage statistics of user on weekday mornings

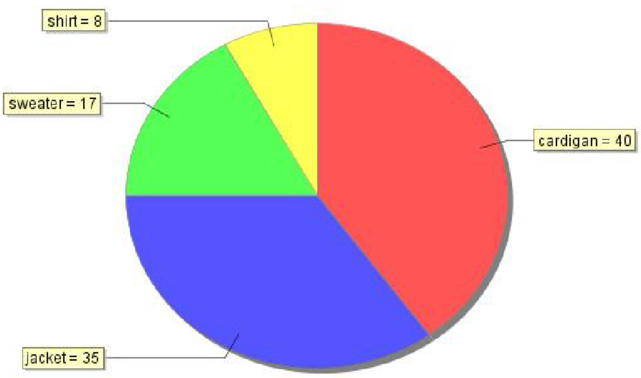
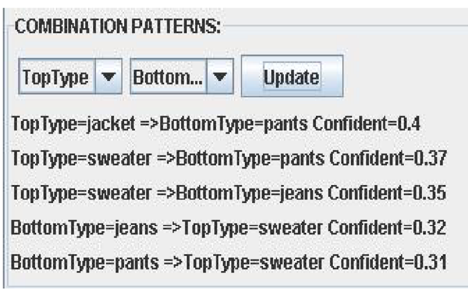


Figure 10 shows an example of the result of mining frequent pattern of clothing item usage. According to this data set, the person is likely to wear ‘jacket’ with ‘pants’ rather then together with ‘jeans’. The WEKA system reports confidence factor of 0.4 for this rule.

Figure 10 Association of clothing items generated



6 Conclusion and future work

In this paper, we have presented a model of a smart wardrobe, which attempts to generate information regarding the user’s preferences on the selection of appropriate clothing. The recommendations given to the user will be based on the user past behaviour which will be captured in his or her user profile. Our model integrates the RFID technology in order to achieve automatic item identification with minimal user intervention. RFID technology is used to capture the events on the user dressing activities. The captured data becomes sets up the foundation to generate the user fashion profile.

In order to illustrate the feasibility and to evaluate the usefulness of our proposed model, we have built a prototype as proof of concept. Due to time and resource constraints, we are unable to incorporate our model into a physical wardrobe. However, using a number of test scenarios applied to our prototype, we are able to illustrate that RFID technology can be successfully utilised in our model as an identification tool. Using this technology, we are able to capture real world events, i.e., the events of taking out a piece of clothing and putting it back into the wardrobe. Statistical methods such as data mining technique, in particular, association rules have been successfully adapted in our model as the mechanism to create user profile.

The following extensions to our model have been identified as possible future work:

- Our system is equipped with relevant knowledge to make reliable and reasonable recommendations. However, this capability is not implemented in our prototype.
- The current system uses a single RFID tag reader. Further extension to include the use of multiple RFID tag readers is envisaged.
- Currently, the generated profile is the only information available for the system to make recommendations. However, it is also possible to include external information, such as the weather and the user’s diary as context for the system to determine what further recommendations can be offered.
- One characteristic of a smart system is its ability to evolve over time. It is possible to design a system able to learn and be aware of changes in the user’s preferences and be able to adapt to these changes appropriately.
- A possible usage of the generated profile is to create a personalised e-catalogue.
- This model is not only applicable to wardrobe, i.e., to build user fashion profile, but also to other home appliances such as fridge, bookshelf or other appliances where user profile can be generated by observing user activities. A collection of these profiles generated by different appliances can formulate a single user profile based on the user’s daily activities around the house.

Acknowledgement

This work is partially funded by the Australian Research Council. We are extremely grateful to Frida Samara for her contribution to the smart wardrobe project.

References

Adomavicius, G. and Tuzhilin, A. (2001) 'Using data mining methods to build customer profiles', *Computer*, Vol. 34, No. 2, pp.74–82.

Amato, G. and Straccia, U. (1999) 'User profile modeling and applications to digital libraries', *Proceedings 3rd European Conf. Research and Advanced Technology for Digital Libraries*, Springer-Verlag, pp.184–197.

Balabanović, M. and Shoham, Y. (1997) 'Fab: content-based, collaborative recommendation', *Communications of the ACM*, Vol. 40, No. 3, pp.66–72.

Chen, H.C. and Chen, A. (2001) 'A music recommendation system based on music data grouping and user interests', *Proceedings of the 10th International Conference on Information and Knowledge Management*, ACM Press, Atlanta, pp.231–238.

Crossen, A., Budzik, J. and Hammond, K.J. (2002) 'Flytrap: intelligent group music recommendation', *Proceedings of the 7th International Conference on Intelligent User Interfaces*, January 13–16, ACM Press, New York, pp.184, 185.

Evans, N. (2004) 'Why Wal-Mart had to deploy RFID', *RFID Journal* [Online], Available: http://www.rfidjournal.com/article/articleprint/1257/-1/82/ (Accessed 8 September 2005).

Floerkemeier, C. and Lampe, M. (2004) 'Issues with RFID usage in ubiquitous computing applications', in Ferscha, A. and Mattern, F. (Eds.): *Pervasive Computing: Second International Conference*, April 18–23, Springer-Verlag, Vienna, pp.188–193.

Floerkemeier, C., Lampe, M. and Schoch, T. (2003) 'The smart box concept for ubiquitous computing environments', *Smart Objects Conference*, Grenoble.

Foster, S. and Scheepers, H. (2005) 'RFIDs: from invention to innovation', *IMMA Conference: Information Technology in the 21st Century: Meeting the Challenges of a Changing World*, September, 2005, Dublin, Ireland, pp.1–13.

Harvey, F. (2002) 'Home smart home', *Engineer*, Vol. 291, No. 7605, pp.16, 17.

Hirsh, H., Basu, C. and Davison, B.D. (2000) 'Learning to personalize', *Communications of the ACM*, Vol. 43, No. 8, pp.102–106.

Kim, S.W., Kim, M.C., Park, S.H., Jin, Y.K. and Choi, W.S. (2004) 'Gate reminder: a design case of a smart reminder', *Proceedings of the 2004 Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, August 1–4, Cambridge, Massachusetts, USA, pp.81–90.

Lampe, M. and Strassner, M. (2003) 'The potential of RFID for moveable asset management', *Workshop on Ubiquitous Commerce, Ubicomp'03*, October, Seattle, USA.

Langheinrich, M., Mattern, F., Romer, K. and Vogt, H. (2000) 'First steps towards an event-based infrastructure for smart things', *Ubiquitous Computing Workshop, PACT 2000*, October, Philadelphia.

Liu, J., Wong, C.K. and Hui, K.K. (2003) 'An adaptive user interface based on personalized learning', *IEEE Intelligent Systems*, Vol. 18, No. 2, pp.52–57.

McCullagh, D. (2003) 'RFID tags: big brother in small packages', [Online], Available: http://news.com.com/2010-1069-980325.html (Accessed: 8 September 2005).

Mora, C. (2005) *Eternal Style Capturing Sartorial Expressions*, MFADT Dissertation, Parsons University, NY, USA.

Römer, K., Schoch, T., Mattern, F. and Dübendorfer, T. (2003) 'Smart identification frameworks for ubiquitous computing applications', *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, March, IEEE Press, Washington DC, pp.253–273.

Schneider, M. (2003) 'A smart shopping assistant utilizing adaptive plan recognition', *ABIS Workshop on Adaptivity and User Modelling in Interactive Software Systems*, pp.331–334.

Vogt, H. (2002a) 'Efficient object identification with passive RFID tags', in Mattern, F. and Naghshineh, M. (Eds.): *Proceedings of the International Conference on Pervasive Computing*, *Pervasive2002*, Springer-Verlag, Zurich, Switzerland, Vol. 2414, pp.98–113.

Vogt, H. (2002b) 'Multiple object identification with passive RFID tags', *IEEE International Conference on Systems, Man and Cybernetics*, October.

Want, R., Fishkin, K.P., Gujar, A. and Harrison, B.L. (1999) 'Bridging physical and virtual worlds with electronic tags', *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI'99),* May 15–20, ACM Press, New York, pp.370–377.

Wasfi, A.M.A. (1999) 'Collecting user access patterns for building user profiles and collaborative filtering', *Proceedings of the 1999 International Conference on Intelligent User Interfaces (IUI'99)*, May 12–15, ACM Press, California, pp.57–64.

Witten, I. and Frank, E. (1999) 'Chapter 8: nuts and bolts: machine learning algorithms in Java', in *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation*, Morgan Kaufmann Publishers, pp.265–320.

WordNet (2003) *Free On-line Dictionary of Computing*, (Dictionary.com), Available: http://dictionary.reference.com/search?q=ubiquitous (Accessed: 25 April 2005).