Mining Context–Aware User Preferences for m-Services Applications

[1]Jembere E, [2]Adigun M.O, [3]Xulu S.S, [4]Emuoyibofarhe O J
Centre for Mobile e-Services for Development
University of Zululand.
P. Bag X1001, KwaDlangezwa, 3886, RSA
1. Tel: +27-35-902-6012, e-mail: ejembere@yahoo.co.uk, 2.Tel: +27-35-902-6189, e-mail: madigun@pan.uzulu.ac.za, 3. Tel: +27-35-6188 e-mail: ssxulu@pan.uzulu.ac.za 4. eojustice@justice.com

Abstract- Challenges raised in mobile computing in relation to Human Computer Interaction (HCI) can be solved by tailoring the access and use of the mobile services (m-Services) to user preferences. Our investigation of state of the art work in personalisation and context-aware computing found that user preferences are assumed to be static across different context descriptions. Further more the assumed preference models do not give an intuitive interpretation of a preference and lack user expressiveness. This paper presents a hybrid user preference model, based on strict partial order preferences and feature preferences to address these issues. We also present an approach for mining context-aware user preferences in an m-services environment. A use case example is also presented to illustrate how the results of the user preferences mining model developed in this work will work in an m-Services environment.

Key Words: User preferences, Context awareness, m-Services, Data mining

I. INTRODUCTION

The influx of web based mobile applications and web content that can be accessed through mobile devices presents new challenges to the area of Human Computer Interaction (HCI). A lot of services and content are made available to the user and the user is left alone to find preferred services or products in an environment characterised by constraints such as small keyboard, flaky network connections, low processing power, etc. Tailoring the access and use of mobile services to the user's preferences is one approach that can be used to solve the HCI challenges in mobile computing. Traditionally user preferences are captured in to the system explicitly from the users. However a closer look at the nature of user preferences in an m-services environment reveals that user preferences are heavily dependent on the dynamic user, environmental and application context. This implies that user preferences change as the context changes. In this case it will be a very cumbersome exercise for users to explicitly give their preferences for each and every context. Recent work [1] in the development of personalised web applications is moving towards the use of data mining for extracting user preferences from user session data.

Automatically analysing user preference models can be classified into the following three classes depending on how they represent preferences: *vector similarity*, *association*, and *probability* [2]. When preferences are represented as vector similarities, it is hard to get the intuitive interpretation about which items are preferred more than others. The concept of an association is different from that of a preference on the part that association deals with relations between two item sets rather than the intuitive interpretation of a preference. When preferences are represented as probability that a user selects an item, a preferred item with low frequency or not so preferred items with high frequency, is not measured correctly. In search of an appropriate representation of a preference this work develops a hybrid user preference model based on two frameworks, the strict partial order preferences framework by Kiessling [3] and feature preference model proposed by Jung et al [2]. Strict partial order preferences provide a very semantically rich way of representing preferences, but the framework use frequency of selection (which does not give an intuitive measure of a preference) as a preference measure. Frequency of selection exhibits the weakness similar to that when preferences are represented as selection probabilities. The feature preference model gives an intuitive measure of a preference but it lacks user expressiveness. This paper presents a preference model based on the strength of these two models to complement each other's weaknesses. To make the resultant model useful in a mobile services environment characterised by dynamic user, environmental and application context (and hence the user preferences), it is necessary to enable it to detect context-aware user preferences.

In an m-services environment user preferences can be mined and used in two areas, which are (i) preferences for automatic service discovery, selection and composition, which can be handled at middleware level and (ii) preferences for personalisation within m-Services applications, which can be handled at application level. Our preference mining approach is envisaged to work in both these areas (see Figure 3).

This paper is organised as follows: Section II covers the related work which forms the basis of this work. Section III discusses the user preference model developed in this work and the user preference mining algorithm for categorical preferences. Section IV presents the architecture for mining context-aware user preferences in an m-services environment. Section V gives a use case example for the preferences mined using the model presented in this paper. Section VI gives a discussion of issues raised in this paper. Section VII concludes this paper and also gives a discussion of some open problems.

II. RELATED WORK

State of the art personalisation techniques for personalised web services suffer from preference models with limited expressiveness [1]. The work by Holland et al in [1] noted that most approaches use either score to describe preferences or just distinguish liked and disliked values. Thus, complex "I like A more that B" relationships as well as numeric attributes can not be expressed in a natural way. A very expressive and mathematically well founded framework for preferences was introduced in [3]. In this framework, customer preferences are modelled as strict partial orders with "I like A more than B" semantics where negative, numeric and complex preferences form special cases. This framework allows for representation of negative preferences and gives user expressiveness. A preference P is defined as a strict partial order $P = (A, <p)$, where $A = (A_1, A_2, ..., A_k)$ denotes a set of attributes with corresponding domains (A_i). The domain of A is defined as the Cartesian product of the $dom(A_i)$, $<p \subseteq dom(A) \times dom(A)$ and $x <p\ y$ is interpreted "*y* is better than *x*".

A set of intuitive preference constructors for base preferences is defined. The constructors for base preferences on categorical domains are *POS(A, POS-set)*, *NEG(A, NEG-set)*, *POS/NEG(A, POS-set; NEG-set)*, *POS/POS(A, POS1-set; POS2-set)* and *EXP(A, E-graph)*. The *POS-set* $\subseteq$ *dom(A)* of a POS preference defines a set of items that are better than all other items of *dom(A).* Analogously, the *NEG-set* of a NEG preference defines a set of items that are worse than all the other items in *dom(A),* i.e. any other items which are not in the NEG-set are preferred to any of the items in the *NEG-set*. The *POS/NEG* preference is a combination of the previous preferences, where items in the *POS-set* are preferred to all the other items, items in the *NEG-set* are the least preferred and any other items neither in the *POS-set* nor the *NEG-set* are preferred to the items in the *NEG-set* and less preferred to the items in the *POS-set*. In *POS/POS* preference an optimal set of items (*POS1-set*) and an alternative set (*POS2-set*) can be specified. The items in *POS1-set* are the most preferred, followed by the items in the *POS2-set*. All the items not in the *POS1-set* and *POS2-set* are less preferred. In an Explicit-graph (*E-graph*) of an EXPLICIT preference a user can specify any better-than relationships. An E-graph is a directed "better than" acyclic graph. All items in an E-graph are better than all other items in *dom(A).*

The preference constructors for numerical domains include *AROUND(A, z)*, *BETWEEN(A, [low, up])*, *LOWEST(A)* and *HIGHEST(A)*. In an AROUND preference the desired value is *z*, but if it is not available values with nearest distance apart from z are best alternatives. For a BETWEEN preference the values within [*low, up*] are optimal. For LOWEST (HIGHEST) preferences lower (higher) values are better.

The actual preferences will be predicted from the implicit preferences hidden in the user session data through use of some data mining techniques. To get these Holland et al [1] introduced the concept of data driven preferences denoted by $P_D = (A, <_{PD})$

- For categorical preferences in a given domain, *dom(A),* a data-driven preference $P_D = (A, <_{PD})$ is defined as: $x <_{PD} y$ iff $freq_A(x) < freq_A(y)$ and
- For numerical preferences in a given domain, *dom(A), a* data driven preference $P_D = (A, <_{PD})$ is defined as $x <_{PD} y$ iff $\exists\, \varepsilon > 0$: freq([$x - \varepsilon, x + \varepsilon$])< freq([$y - \varepsilon, y + \varepsilon$])

where $freq_A(x)$ is the frequency of selection of item x in domain *A*. In order to detect negative preferences the Closed World Assumption (CWA) is defined. It states that a user is aware of all possible choices that he/she can make. In this sense detection of negative preferences is enabled through the fact that only if the user is aware of all the possible choices, we can assume dislike for items he/she never selected.

One weakness of the strict partial order preference mining framework as presented in [1] is that it uses the frequency of selection of an item as the measure of a preference for that item. The probability that a user selects an item is determined mainly by two factors: the preference and the accessibility of the item [4]. Although a user likes an item, the selection frequency can be low if the item is rarely distributed in that domain. On the other hand, although a user does not like an item, the selection frequency can be high if the item appears very frequently in that domain. Measuring the accessibility probability of an item in an m-service environment poses another challenge. To counteract this problem this research adapts the idea proposed by Jung et al [2], [4]. The work in [4] defines a preference as the concept to make a relationship between a person and a target item which contains several kinds of attributes/features. From this point of view a preference of an item is indirectly related to the preferences of the attributes/features contained by that item. This implies that the preference of an item can be represented by a combination of the preferences of the features contained in it. A preference was then mathematically represented as a function of item *x* and the user profile *G.* The user profile will be approximated by the User History *U*, i.e

$$\Pr ef_A(X) = f(x, G) \approx f(x, U) \tag{1}$$

The user history is represented by a set of selected items $U = (x_1, x_2, x_3, x_4, \cdots, x_n)$. Each item has a set of several attributes/features denoted by *w*. Item *x* is then represented as a set of features, $x = (w_1, w_2, w_3, \cdots, w_m)$. The User Profile *G* is defined by the preference of each feature w, i.e. $G = \{pref_A(w_1), pref_A(w_2), pref_A(w_3), \cdots\}$. The feature preference $\Pr ef_A(w)$ is computed from the user history. Since the user profile *G* and the item *x* can be represented by common attributes, w_i's, they can be compared through these attributes. As a consequence, a preference of an item *x* is then represented as follows:

$$\Pr ef_A(x) = \frac{1}{M(x)} \sum_{w \in x} \Pr ef_A(w_i) \tag{2}$$

where *M(x)* is a normalisation term which is defined as the number of features in item x.

Mutual information is used as a measure of a feature preference, $\Pr ef_A(w) = I_A(X(w), U)$, where X(w) = $\{x \mid w \in x\}$, $I_A(X(w), U) = \ln\{P_A(X(w); U / P_A(X(w))\}$. $P_A(X(w); U)$ is the feature selection probability given the user history and $P_A(X(w))$ is the unconditional feature probability, which gives a measure of the feature's accessibility.

The strict partial order preference model has been successfully implemented in the COSIMA B2B sales automation e-procurement system [5]. The preferences used in COSIMA are either explicitly obtained from the user through query interfaces and user feed back or are mined from the user session data. The work done in [5] assumes that preference mining can only give long term (static) preferences which are not dependent on context. This paper presents an approach where the user preference mining algorithms are able to detect context-based user preferences.

III. THE USER PREFERENCE MINING MODEL

We define a *user preference as the concept where a user makes a binary choice relation on a given set of items*. Taking the strength of the preference models presented in Section I and II, the strict partial order preference and the feature preference model, we introduce a model for mining situated user preferences in an m-Services environment. Instead of using frequencies as a measure of a preference in the strict partial order preference model as proposed in the work done by Holland et al in [1], this work uses the value of $Pref_A(x)$ defined in Table 1.

TABLE I
NOTATION AND DEFINITIONS

Notation	Definition
$N_A(X(w))$	Number of items with feature w in the whole item set in domain A.
$N_A(X)$	Number of all items in the whole item set in domain A.
$freq_A(X(w);U)$	Frequency of selection of items with feature w in domain A given the user history
$freq_A(S;U)$	Total number of all item selections in domain A given the user history
$P_A(X(w))$	$N_A(X(w))/N_A(X)$, measures accessibility probability of feature w domain A.
$P_A(X(w;U)$	$freq_A(X(w);U)/freq_A(S;U)$, measures the probability that a given user selects an item with feature w in domain A based only on the selection frequencies.
$\Pr ef_A(w)$	$\ln\left(\frac{P_A(X(w);\ U)}{P_A(X(w))}\right)$, measures the preference of feature w.
$\Pr ef_A(x)$	$\frac{1}{M(x)}\sum_{w\in X} pref_A(w)$, measures the preference of item x.

The resultant hybrid model is then developed in such a way that it can detect context-aware user preferences. In order to be able to detect such preferences the item/attribute domains will be defined in terms of the context. We also define the concept of a strong negative preference on an item. A user preference on an item is regarded as Strong Negative preference if the user avoids the item even though he/she had a significant number of chances of selecting it. The CWA is adopted to detect negative preferences.

Assuming the Closed World Assumption (CWA), the following criterion is used to determine Negative and Positive categorical preferences:

- An item x is a Strong Negative (STRONG_NEG) preference in domain A if $freq_A(x)$[1] $>$ $freq_A(threshold)$ and $freq_A(x;U)$[2] $=0$
- An item x is a Soft-Negative (SOFT_NEG) preference in a given domain if $\Pr ef_A(x) < 0$.
- An item x is a Neutral preference in a given domain if $\Pr ef_A(x) = 0$.
- Item x is a Positive (POS) preference only if $\Pr ef_A(x) > 0$

The following data driven categorical preferences are defined from the above mentioned criterion:

- There is a data-driven POS preference, iff $\forall x \in POS-set, \forall y \notin POS-set: y <_{PD} x$.
- There is a data-driven STONG_NEG preference, iff $\forall x \in STONG\_NEG-set,\ \forall y \notin STRONG\_NEG-set$: $x <_{PD} y$.
- There is a data-driven SOFT_NEG preference, iff $\forall x \in SOFT\_NEG-set,\ \forall y \notin (SOFT\_NEG \cup STRONG\_NEG-set)$: $x <_{PD} y$.
- There is a data-driven POS/POS preference, iff $\forall x \in POS1-set, \forall y \in POS2-set, \forall z \notin (POS1-set \cup POS2-set), : y <_{PD} x$ *and* $z <_{PD} y$.
- There is a data-driven POS/STRONG_NEG preference, iff $\forall x \in POS-set, \forall y \in STRONG\_NEG-set, \forall z \notin (POS-set \cup STRONG\_NEG-set): z <_{PD} x$ and $y <_{PD} z$.
- There is a data-driven POS/SOFT_NEG preference, iff $\forall x \in POS-set, \forall y \in SOFT\_NEG-set, \forall z \notin (POS-set \cup SOFT\_NEG \cup STRONG\_NEG-set)$, $z <_{PD} x$ and $y <_{PD} z$.
- There is a data-driven SOFT_NEG/STRONG_NEG preference, iff $\forall x \in SOFT\_NEG-set$, $\forall y \in STONG\_NEG-set, \forall z \notin (SOFT\_NEG \cup STRONG\_NEG-set)$, $y <_{PD} x$ and $x <_{PD} z$
- Let $<_{EP}$ be a strict partial order on positive preferences on E. A data-driven EXPLICIT Positive preference holds, iff
  - $\forall x, y \in E$ with $x <_E y : x <_{PD} y$,
  - $\forall u \in E, \forall v \notin E : v <_{PD} u$
- Let $<_{ESN}$ be a strict partial order on soft negative preferences on E. A data-driven EXPLICIT Soft Negative preference holds, iff
  - $\forall x, y \in E$ with $x <_E y : x <_{PD} y$
  - $\forall u \in E, \forall v \notin E : v <_{PD} u$

[1] *$freq_A(x)$* is the frequency item x was made available to a given user for selection in domain *A*.
[2] *$freq_A(x;U)$* is the frequency item x was selected by the user in domain *A*.

Using the above criterion a data mining algorithm (shown below) was developed to detect the categorical preference.

A. Algorithm for Categorical Preferences

Input: Context tagged Log Relations.

1. For each x_i in given context domain, A, compute $freq_A(x;U)$ and $freq_A(x)$
2. Remove all X_i: $freq_A(x;U) = 0$ and $freq_A(x) \geq freq_A(threshold)$ from the item set. This is a set of Strong Negative preferences.
3. Extract the set of attributes/features of all items remaining in the item set.
4. For each feature, compute, $P_A(X(w))$, $P_A(X(w);U)$, $\Pr ef_A(w)$ as shown in Table 1
5. For each item, X, in the context domain, A, compute $\Pr ef_A(X)$ as shown in Table 1
6. Remove all x : $\Pr ef_A(x) = 0$. This is a set of neutral items
7. Compute a clustering of the x_i's with $\Pr ef_A(x) > 0$, using a clustering technique
   Depending on the clustering results we have the following possibilities:
   a. One cluster, C_1: we have a SOFT_NEG(A, C_1; $\{x \in dom(A) : \Pr ef_A(x) < 0)\}$
   b. Two clusters, C_1 and C_2, where $\forall c_1 \in C_1, \forall c_2 \in C_2, \Pr ef_A(c_2) < \Pr ef_A(c_1)$. We have POS/SOFT_NEG(A,C_1:$\{x \in dom(A) : \Pr ef_A(x) > 0\}$; C_2:$\{x \in dom(A) : \Pr ef_A(x) < 0\}$)
   c. Three clusters, C_1, C_2 and C_3 where $\forall c_3 \in C_3, \forall c_2 \in C_2, \forall c_1 \in C_1; \Pr ef_A(c_3) < \Pr ef_A(c_2) < \Pr ef_A(c_1)$. We have POS/POS(A, C_1: $\{x \in dom(A) : \Pr ef_A(x) > 0\}$ C_2: $\{x \in dom(A) : \Pr ef_A(x) > 0$ & $\forall x \in C_1, \forall y \in C_2, y <_{PD} x\}$)
   d. More than three clusters $C_1, C_2, \ldots. C_n$, where $\forall c_1 \in C_1, \forall c_2 \in C_2, \cdots, \forall c_n \in C_n$: $\Pr ef_A(c_n) < \cdots < \Pr ef_A(c_2) < \Pr ef_A(c_1)$. We have an EXPLICIT positive preference $E(A, <_{EP})$ with $c_n <_E c_{n-1} <_E \ldots\ldots\ldots <_E c_1$, $\forall c_1 \in C_1, \forall c_2 \in C_2, \ldots \forall c_n \in C_n$
8. Compute a clustering of the x_i's with $\Pr ef_A(x) < 0$, using a clustering technique
   Depending on the clustering results we have:
   a. One cluster, C_1: we have a POS(A, C_1; $\{x \in dom(A) : \Pr ef_A(x) > 0\}$)
   b. Two clusters, C_1 and C_2, where $\forall c_1 \in C_1, \forall c_2 \in C_2$, $\Pr ef_A(c_2) < \Pr ef_A(c_1)$ .We have POS/SOFT_NEG(A,C_1: $\{x \in dom(A) : \Pr ef_A(x) > 0\}$; C_2:$\{x \in dom(A) : \Pr ef_A(X) < 0\}$)
   c. Three clusters, C_1, C_2 and C_3: $\forall c_3 \in C_3, \forall c_2 \in C_2, \forall c_1 \in C_1$; $\Pr ef_A(c_3) < \Pr ef_A(c_2) < \Pr ef_A(c_1)$. We have SOFT_NEG/SOFT_NEG(A, C_3: $\{x \in dom(A) : \Pr ef_A(x) < 0\}$; C_2: $\{x \in dom(A) : \Pr ef_A(x) < 0$ & $\forall x \in C_3, \forall y \in C_2, x <_{PD} y\}$)
   d. More than three clusters $C_1, C_2, \ldots. C_n$, where $\forall c_1 \in C_1, \forall c_2 \in C_2, \ldots \forall c_n \in C_n$: $\Pr ef_A(c_n) < \ldots < \Pr ef_A(c_2) < \Pr ef_A(c_1)$. We have an EXPLIXIT Soft Negative preference $E(A, <_{ESN})$ with $c_n <_E c_{n-1} <_E \ldots <_E c_1$, $\forall c_1 \in C_1, \forall c_2 \in C_2, \ldots \forall c_n \in C_n$.
9. In all the other case there are no data-driven preferences

Output: The detected preferences or no preference was found

Within the preference categories, preferences will be mined and represented as strict partial order relations as proposed in [3]. For detection of context-aware user preferences, the

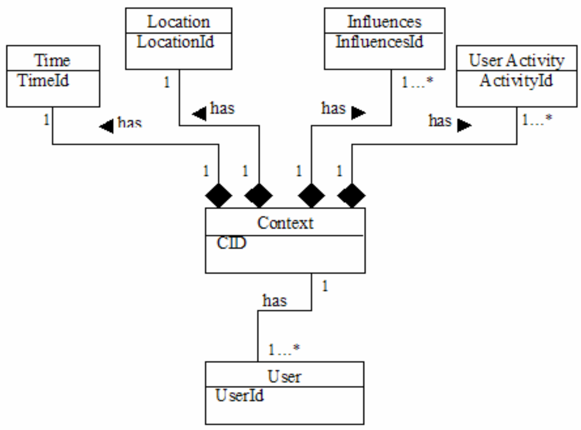

Figure 1: m-Services Context Meta-Model

item/attribute domains are defined in terms of the abstracted high level context. Holland and Kiessling [6] proposed an e-commerce context meta-model with three high level context components, location, time and influences. Due to the fact that in a mobile computing scenario the user might be involved in some other activities other than computing (e.g. driving, running, etc), the user's activity has to be considered to avoid recommendations that are irrelevant to the user's activity. Based on this consideration we came up with the context meta-model shown in Figure 2, with four high level context abstractions *location*, *activity*, *influences* and *time*.

IV. USER PREFERENCE MINING ARCHITECTURE

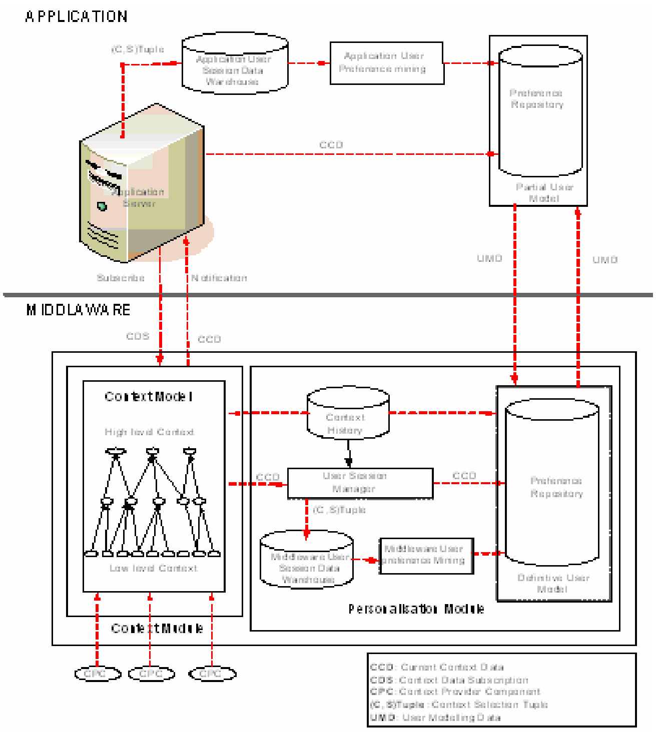

Figure 2: An architecture for user preference mining in an m-services environment

Our preference mining model is implemented in an architecture shown in Figure 3. The architecture shows two layers where user preference can be mined and used in an m-service environment, the middleware and the application layer. The middleware layer is divided into two modules the context module and the personalisation module. The modules are separated in order to utilise the benefits (discussed later in Section VI) which come with the separation of context and personalisation in mobile computing. The context module obtains low level context data from the Context Provider Components (CPC) and synthesis high level context data. The personalisation module handles all the functionalities for user preference mining and use at middleware level. The definitive user model holds, among other personalisation functionalities at middleware level, a repository of user preference for automatic service discovery, selection and composition. It also facilitates sharing of user modelling data among similar application. In the application layer the application server is responsible for handling user session management. Each application has its own partial user model responsible for personalisation within that application. The user preferences are mined from the user session data warehouses. These data warehouses hold the user session data in the form user history tuples, consisting of the service/item selected and the context under which it was selected. We call this tuple the Context-Selection Tuple ((C,S)Tuple).

The user preference mining algorithm is implemented as an autonomous agent that can be called by different applications, to mine for preferences from the user session data warehouses held by each application. The same algorithm is also responsible for mining user preferences at middleware level.

V. A USE CASE EXAMPLE

In this section we demonstrate how the preferences mined using the criteria presented in this paper works in an m-services environment.

TABLE II:
A CONTEXT DESCRIPTION EXAMPLE

	Context description
Time	13:00 pm, Sunday
location	Durban, home
Activity	Lying on the bed, Browsing the internet
Influences	Cold and rain, alone, PDA

A. Personalisation at middleware level

Considering the context given in Table II and a closed world with the m-services shown in Figure 5, the following applications may be relevant to the current context: News feeds, M-shopping, Weather reports, Racing Reports, Restaurant Reservations, and Movie Theatre Reservations. This list forms the items in the current context domain. The system discovers, composes and avails all the m-services that are relevant to the current context description according to the user's preferences. The user then selects the preferred m-service. After logging out of the service, the system automatically updates the user session data warehouse in the middleware, and initiates user preference mining algorithms to update the definitive user model.

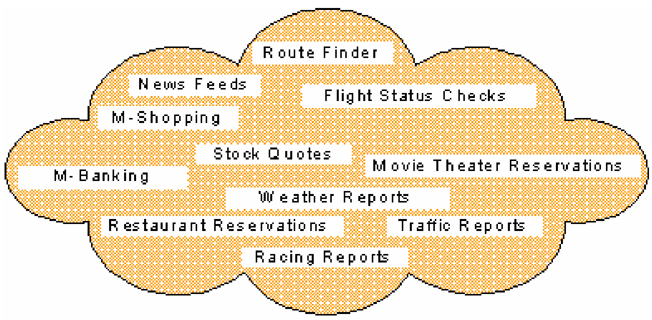

Figure 3: Different web-based m-services applications

B. Personalisation within an application

As soon as the user selects an application/service, the attention shifts from the middleware to the application. The application automatically subscribe for relevant context data to the middleware. Suppose the user selects a movie theatre reservation service. The application gets the context it had subscribed from the uniform service infrastructure and uses it to query for user preferences from the partial user model held within the application. The application then uses these preferences to recommend movie theatres, which will be shown on the user interface according to the user's preferences. Features that can be used to get the preferred theatre include the actor, genre, and production year of the movie being shown, services available in the theatre, admission fees, etc.

VI. DISCUSSION

Most of the work in context-aware computing [7], [8], [9] take user preferences as part of context data while on the other hand personalisation is loosely defined as adaptation to user preferences. So we are left with the question, "Are user preferences part of context or personalisation data?" In context-aware computing personalisation is implicitly treated as a subset of context-aware computing. Our work takes context and personalisation as two separate dimensions of mobile computing. A clear separation of these concerns seems to be difficult to realise but has important benefits and these include: independent modification of the context network; independent usage of established techniques from both the fields of personalisation and context-awareness; initial development of non-personalised context aware application, empowerment and simplification of context history management among others [10].

The efficiency of our user preferences mining approach to ensure advanced personalisation, for automatic service discovery and composition is hinged on the enriched semantic service description. Recent semantic efforts around UDDI, WSDL and SOAP will enable the description of the service features/attributes.

VII. CONCLUSION AND OPEN PROBLEMS

This paper presented an approach for modelling and mining situated user preferences in an m-services environment, as one way of solving HCI challenges in an m-services environment. However there are a number of open problems arising from this work that require further research.

Our approach to mining situated user preferences assumes that there is a sound context model on ground. Context is in its essence complex, subjective and application specific. Different applications consider different context descriptions to be context to them. For example location is context to a weather report service but it is definitely not context to an online banking services. This requires sound context semantic labelling, which can be attained through definition of context ontology.

The other problem which arises from this work is on the definition of context domains. If we consider the nit grits of the context atoms, almost all context descriptions are unique and hence it is not possible to formulate context domains. This can be solved by clustering the context descriptions using some vector similarity measures to form some clusters which will be used as context domains. This implies that when a new context description is discovered, it is first compared for similarity with existing context domains (clusters) in the user history before it is treated as a significant new context description. However this still require further research.

ACKNOWLEDGEMENTS

This work was done under the Centre for Mobile e-Services for Development at the University of Zululand. The centre is funded by Telkom, THRIP, NRF and Huawei, and Alcatel.

REFERENCES

[1] Holland S, Ester M, Kiessling W; A Novel Approach on Mining User Preferences for Personalised Applications. *In: Knowledge Discovery in Databases (PKDD 2003), Dubrovnik, Croatia, pp 204-216*, 22-26 September (2003). http://www.cs.uoi.gr/~kstef/PreferenceMining.pdf (last accessed on 24 April 2006)

[2] Jung S.Y, Hong J.H, Kim T.S; A Statistical Model for User Preferences, *IEEE Transaction on Knowledge and Data Engineering, Vol. 17, No 6*, June 2005.

[3] Kiessling W. Foundations of Preference in Database Systems, *In proceedings 28th International Conference on Very Large Databases ,Hong Kong, China*, 2002 http://www.vldb.org/conf/2002/S09P04.pdf (last accessed on 24 April 2006)

[4] Jung S.Y, Hong J.H, Kim T.S, A Formal Model for Preference, In proceedings of the 2002 IEEE international Conference on Data Mining (ICDM'02), Maebashi TERRSA, Maebashi City, Japan, pp 235, 9-12 December; 2002

[5] Kiessling W, Fischer S, Doring S (2004); COSIMA2B2-Sales for E-Procurement; In proceedings of the IEEE International Conference on E-Commerce Technology,

[6] Holland S, Kiessiling W; Situated Preferences and Preference Repositories for Personalised Database Applications*; 23rd International Conference on Conceptual Modelling,* (Nov 2004). http://www.cs.uoi.gr/~kstef/2004_hol_kie_er2004.pdf (last accessed on 24 April 2006)

[7] Biegel G, Cahill V (2004); A Framework for Developing Mobile, Context-aware Applications, *2nd IEEE Conference on Pervasive Computing and Communications, Orlando,FL, March 14-17*

[8] Dey A. D, Abowd G. D, and Salber D (2001); A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware applications; http://www-static.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf (last accessed 24 on April 2006)

[9] Riva O (2004); A Conceptual Model for Structuring Context-Aware Applications; *Forth Berkeley-Helsinki student workshop on telecommunication Software architectures*; *university of Berkeley USA*; http://www.cs.helsinki.fi/u/kraatika/Courses/Berkeley04/RivaPaper.pdf (last accessed on 24 April 2006)

[10] Coppola P et al; MoBe: A Framework for Context-aware Mobile Applications; *Workshop on Context-Aware Proactive Systems, Helsinki, Finland*; (16-17 June 2005) www.cs.helsinki.fi/hiit_bru/conferences/caps2005/presentations/dellamea_et_al_CAPS2005.pdf (last accessed on 24 April 2006)

Edgar Jembere is a holder of a BSc Honours degree in Statistics. He is currently a final year MSc student in Computer Science in the Centre for Mobile e-Service for Development at the University of Zululand. His area of research is Data Mining and Knowledge Engineering for Mobile e-Services.