User-centric Interests Manager (UcIM)

Yijun Ying, Dominic Hugh Jones, Declan O’Sullivan

Knowledge and Data Engineering Group
School of Computer Science & Statistics, Trinity College, Dublin.
{yingy | jonesdh | Declan.O'Sullivan}@cs.tcd.ie

Abstract

The Publish/Subscribe paradigm exposes a user to a wide wealth of information, to which they may be possibly interested. This poses the question: how can we best track the real interests of the user and provide them with the information that is related to what they are interested in? In beginning to answer this question this paper introduces a new model for personalised interest management, the User-centric Interests Manager (UcIM), and presents a prototype implementation that can be used to express users interests in a technically simple form and thus obtain the most relevant information according to the routing of the underlying Publish/Subscribe network.

Keywords: Knowledge-based Networks, Ontologies, Interest Representation, Semantic web, Usability.

1 Introduction

Whilst searching for information online in which they may be interested, users often find the process involves their full attention as well as a lot of effort and time to reliably figure out good information sources related to their interests from the vast number of information sources available. Moreover, people receive the query results passively, without the ability to decide what they are really interested in, how the information should be shown, and the timeframe of their interest (whether once off where search results presented once or a persistent interest indicating a long lived search such that information should be gathered from new sources as they become available). Current internet users search, digest and discard each and every search session. Additionally with regard to the conventional search engine, users have to pick a set of suitable keywords, refine this set of search terms, browse through a vast number of websites, seek out only those sites relevant to their interests, and only then hope that the information they find is actually interest them. Web favourites allow users to represent their interests successfully, however to be able to deduct automatically users’ full set of interests from a large body of websites is a hard task to achieve.

Given the limitations of searching for information, as previously outlined, search engines cannot track their users’ interest information, hoping to deliver more relevant information later in the search period. In other words, the query for a particular interest will only return information relevant at that point in time, based on websites which the search engine has crawled. In comparison to the traditional search engine, the evolving Publish/Subscribe paradigm [1] provides a more efficient manner in which to deliver information from a range of publishers to a number of interested subscribers. RSS feeds [2], which point to rapidly changing sources of information, provide a good set of publication data for use within a publish/subscribe network. Currently after users have subscribed to the RSS feeds they are interested in, the RSS Reader automatically “Polls” the RSS feed website to see if they are any changes in the content. Even so services such as RSS require a user to search and subscribe to the feeds, which match their specific set of interests. A major drawback to these services is that it is not that easy for users to find accurate information reflecting their full set of interests, without conducting
broad and detailed searches, which still does not guarantee that relevant information will be found. Users do receive some relevant information related to the RSS feeds to which they have subscribed, but they still have to search for the information to obtain the real content represented by the feeds, in the case that maybe only one item from a feed represents a real interest of the user.

In this paper, we describe and analyse the prototype UcIM system which is used to capture users’ interests, and relies upon the Knowledge-based Network (KBN) [3] for timely delivery of RSS content. UcIM provides a multi-option interface in which a user can interact with the system. In expressing their interests, users use their own natural language to define their interests and select their particular interest concepts, which are passed to the KBN as subscriptions. For example, using “MU” to indicate they are interested in “Manchester United Football Team” or using “Footie” to indicate they have an interest in “Football”. Based on the routing mechanism of the KBN, the system automatically delivers only the relevant published information through a keyword and semantic matching mechanism. These results are only delivered to the user after they are ranked based on the users’ preference. Given that routing of information within the system is managed by the KBN, the UcIM system allows users to customise their preferred way in which they wish to interact with the system. For example users can change the design of the interface, when they would like to receive notifications of matching RSS feeds and how they would like to be notified. The prototype UcIM system works with the users defined interests, represented within a semantic domain and explores the full set of possible interaction models between the users and the system.

In Section 2, we introduce the background Knowledge and Content-based network used in our system and in Section 3, we briefly describe some related work in the interest management area. Section 4 presents our research aims and goals. Sections 5 outlines the technical design and implementation of the prototype system. The initial evaluation shown in section 6 illustrates the way in which the UcIM system will be evaluated and how we will evaluate the usability and usefulness of UcIM. Section 7 draws together initial conclusions drawn from the prototype to date.

2 Defining Knowledge and Content-based Networks

Within publish/subscribe networks, subscribing clients send subscriptions to distributed message brokers, and publishing clients send publications to the same brokers, whilst the brokers match subscription filters to publications and route publications back to subscribing clients. This de-coupled relationship between subscribers and publishers allows for message matches to occur between source and destination without an explicit relationship being created between the publisher and subscriber, and without the two knowing of each other’s existence. Content-based pub/sub systems, such as Siena [4] allow for arbitrary subscriptions filters to be applied to any part of the contents of the publication, rather than its type or channel etc. The filters which match these subscriptions are constructed using a set of operators. The range of these operators determines the type of pub/sub network in which the message is being sent. Most used yet least expressive are Topic-based networks such as Scribe [5]. More expressive content-based networks use content-based filtering, e.g. the core message matching algorithm within the Siena CBN [6]. Mühl et al. describes this content-based matching as a set of “Filters [which] are evaluated against the whole contents of notifications” [7]. Within this paper, notifications can be thought of as publications. Publications are only forwarded to a user when the contents of the filter, which is made up from a set of constraints, matches a subset of the messages contents. This allows for a more flexible message format and increases, in comparison to topic-based networks, the desired network separation between the publishers and subscribers. Thus making more likely the possibility that a producer and consumer, using separate formats, may match based on a subset of their publication and subscription respectively.

The work described within this paper involves the use of Knowledge-based Networks (KBN). The KBNs allow for the traditional content-based types which include Strings, Byte Arrays, Integers, Doubles, Floats and Booleans and their operators Equals, !Equal, Less Than, More Than, Prefix/Suffix to be extended with additional semantically enhanced ontological operators. The KBNs extended Type set includes Ontological Classes, Properties, Individuals and Bags of Typed Attributes. The operators
which apply to these new types include Equivalent, 'Equivalent, More Specific (Sub-Class), Less Specific (Super Class), Is_a and 'Is_a with regard to instances belonging to a class and Equal, Sub and Super bags. The KBNs have been discussed in detail in [3]. The KBN semantics further improves the expressiveness of the subscription mechanism to allow subscribers to describe their interests more accurately and concisely, while further decoupling the publisher/subscriber. One key difference between Knowledge-based and Content-based networks, other than the increased expressiveness presented by the networks operators [8], is that an ontology must be placed on each router/broker, and then reasoned over. Within a KBN, messages may be formed around any ontological relationship between classes, individuals and properties, these ontologically enhanced messages are only successfully routed as long as each broker within the network have an understanding of the ontological knowledge on which a message may be formed. For this reason each broker must reason over an ontology, representing the message scope, and re-reason when and if the ontology changes.

3 Related Work

Aiming at providing tailored services for individual users, much research is ongoing related to managing users’ interests, especially in semantic domains. In terms of representing people’s interests, APML (Attention Profiling Mark-up Language) [9] is a good example. It collects and aggregates all of the data that people are interested in and puts them together into a single “attention profile” in the form of a consolidated, structured XML file. The APML attention profile not only contains the contents that people are interested in when surfing the web, such as the URLs visited, the blogs or websites tagged and the multimedia shared, but also ranks the information to indicate how interested people are in the various contents. Nowadays, a growing number of sites and services have already supported APML, including “Engagd” tool which creates APML file based on RSS feeds’ selection, “Particles” service which makes use of APML file to alert the ranked news that users are interested in, “Clutzr” social network which shares users’ attention data to their friends, and “Dandelife” service which creates a socially powered biography for users [10]. The project conducted by Alexandra Roshchina et al [11] recently, also makes use of APML format, with the combination of the Galaxy (IBM LanguageWare Miner for Multidimensional Socio-Semantic Networks) [12] API to construct dynamic ontological data on the social web, to represent users’ interests domain as expressively and accurately as possible.

Concerning how to gather and mine users’ personal data to deduce interests, the concept of the Semantic Desktop has been introduced by Sauermann et al.[13] bringing together the ideas and technologies of the Semantic Web into the personal computer. The Semantic desktop stores all digital information such as documents, multimedia files and messages transferring among web services as Semantic Web resources and integrates the Semantics within applications used for collecting information about the users. Based on the Semantic Desktop, different projects such as Growsis [14], MindRaider [15], Haystack [16], MOSE [17], Chandler [18] use different approaches to model the personal data of desktop users and manage the concepts related to their interests.

With regard to matching information to a user’s real interests, the Cobra (Content-based RSS Aggregator) [19] system is designed to deliver personalised feed to an individual user based on his set of interests from a large numbers of RSS feeds. Consisting of a three-tiered network of “crawlers”, “filters” and “reflectors”, the Cobra system uses “crawlers” to periodically pull data from web feeds, uses “filters” to match the keywords of subscriptions against the feeds from “crawlers”, and uses “reflectors” to deliver the matched feeds to users. Another project which is described by Michael G. Noll in [20] provides a new approach to personalised web search by using what is called the “Tagmarking” technique. Combining the popular tools of bookmarking and tagging, “Tagmarking” provides an efficient and convenient approach to gather users’ interests by storing the bookmarks with tagged metadata. By matching the similarity of user’s profile containing personal “tagmarking” against the search results returned by ordinary search engines, the system re-ranks and personalises these results to satisfy an individual user’s interests.
4 Research Aims

Through the User-centric Interest Management UcIM system, represented within this paper, we aim at addressing the following issues: Firstly establish what kinds of interactions are necessary in making interest management efficient for the user. Specifically with regard to the non-technical users, how can the system involve them in searching their interests actively, without interfering with their ordinary daily routine? Secondly investigate whether UcIM is a possible approach to be used in the tracking of users’ interests and extracting only the relevant semantic information from the vast numbers of online resources in satisfying users’ requirements. Thirdly explore to what extent Web 2.0, Semantic Web and KBN technologies facilitate this research. Through the implementation of the prototype system, we will examine how evolving technologies such as APML [9], Jena [21], JDOM [22], ROME [23], AJAX [24], and the KBN [3] can combine and integrate to achieve the primary research aims.

5 Design and Implementation

5.1 System Overview

UcIM is designed to track users’ interests, subscribe the users to these interests and, using the KBN [3], route published information towards users and notify them automatically of its availability. Users use their own, non-technical, language to create their own personal concepts of interest, and then express these interests by means of KBN. A user’s interests are recorded in his Interest Profile File (IPML) which is then passed to the KBN as subscriptions. Publications within the KBN are formed from RSS [25] feeds which are filtered against subscriptions (according to the KBN routing ontology and ontology mapping mechanism) before the system ranks results based on the expressed user interests. After reviewing these feeds, users can give feedback to the system as to whether or not the filtered results fit their interests. This feedback mechanism allows the system to develop a better understanding of the users’ interests and subsequently provide more appropriate information. Apart from the feedback mechanism, users also have full options in selecting their preferred way to interact with the system by changing the application settings. The architecture of UcIM is shown in Figure 1 below:

![Figure 1 Architecture of User-centric Management System](image-url)
Briefly, Figure 1 can be described as follows. In the graphical user interface, users establish and modify their personal profiles and their application settings. Based on the user ontology imported by the system, users express their interests by selecting single or multiple concepts and/or the use of keywords. The user-centric interest information is recorded in an IPML file, explained in detail in Section 5.4. Each time a change occurs related to users' interests, the system generates a corresponding KBN subscription. Based on the routing ontology and the ontology mapping within the user ontology, the KBN manages the delivery of the published RSS feeds, which match the users' subscriptions. After filtered, the matched feed is delivered to the user as a notification, through the notification browser. After viewing the notifications, users give feedback in terms of the accuracy of these notifications delivered, which are also recorded in the users' IPML file, through this mechanism, it is hoped that the system will become more accurate in delivering future publications, without any explicit action from the user.

5.2 The Publication Source

The KBN [8] source of publication comes from RSS [25] feeds. In utilising development tools such as JDOM [26] and ROME [23], which will be described in detail in Section 5.6, the UcIM application parses RSS feeds from multiple sources, including BBC, RTE, Yahoo, iTunes, etc., and publishes them to the KBN brokers. Each publication contains three attributes and two bags. According to [27], a bag (also called multiset) is a set-like object in which order is ignored, but multiplicity is explicitly significant. As described in [28], a bag can contain various types, including string, long, integer, double, Boolean and Ontological class. For example, {'Manchester United', 'Football Team', 'Sport Domain', 0.98} is a bag contains both string and double types. Two bags can be compared according to the three well-known binary bag relations: equal, subbag, and superbag, which indicate the relations of “=”, “⊆”, and “⊇”. For example, the bag {'Football Team', 'Manchester United'} is a subbag of the bag {'Manchester United', 'Football Team', 'Sport Domain', 0.98} and the latter is the superbag of the former.

In each publication, the three attributes include the Title, Description and URL of the items in RSS feeds, which indicate the content of each particular item of a feed. The two bags within each publication are named as Category Bag and Keyword Bag, representing a set of values that contain the category concepts and keywords attached with each feed item. Based on the KBN's routing ontology, the corresponding categories in the routers ontology will be tagged with each item, which forms the Category Bag. Concurrently, the content of item's Title and Description will form the Keyword Bag after all of the texts stop words have been removed. Stop words are extremely common words such as “in”, “and” and “the.” This assures that the Keyword Bag only contains relevant keywords and not commonly used words. Using these two bags, the publication can be filtered against the brokers' subscriptions to check whether they match the users' interests or not using the KBN's built-in semantic operators. The publication is then routed over the KBN and when a match is found the UcIM system notifies the user of relevant content to which they may be interested.

5.3 The User Interface

5.3.1 Profile Establishment

When a user initially loads UcIM, the system requires them to create an account by providing some basic personal information, such as their user name, age, gender, occupation, email address, etc., this information can be edited at any later point, the user ID is regarded, within the system, as the unique identification of a user. The profile management function can be extended to share the user information, assuring the user privacy, with other applications to allow for the analysis of groups of similar users, based on age, gender, occupation etc., in aiming to provide more specific semantic information which may be suitable to the users' interests.
5.3.2 Application Settings

Based on a user’s personalised preferences, the system provides different modes of interaction. There are two main methods of interaction, the first being through the basic settings panel, and the second being through the notification settings panel. The basic settings panel allows the user to personalise their application interface by selecting their preferred GUI format such as colour, font or layout. The notification settings panel offers to the user the option of setting the way in which they will be notified of matches to their subscriptions. By defining “what”, “when” and “how” the notification should be delivered, the user has the full freedom to decide their preferred way of notification, as described below:

The what setting refers to the kind of information which will be provided to the users, for example this allows a user to set variables such as only receiving notifications about their latest subscriptions, represented by time. Or to tell the system to rank the notifications according to when they created the subscription based on a specifically prescribed level of interest in a notification. The when setting allows users to stipulate the time range in which they would like to receive notifications. For example every time they open the system or do a manual update, every half hour, or only within office hours. Being able to control the time in which a user has messages delivered, in a push based system such as UcIM, is of the utmost importance to the final usability of the system. The how settings allow users to set how notifications are delivered, which includes methods such as message pop-ups in the corner of the users screen, email notification, or manual collection from the application itself. The introduced application settings allow users to customise the interaction, delivery and format of the delivery of messages matched by the KBN to their interests. With the final assessment of usability being of such importance, the ability to change and adapt the user experience is seen as an important aspect of UcIM.

5.3.3 Interests Expression

Different users have different ways in which they wish to express their interest in ontological concepts. For this reason UcIM contains the individual user’s ontology which helps users express their interests in their own personalised manner. Compared to content-based matching, the UcIM system has the ability to handle the semantic matching between the users’ self-defined interest concepts and the relevant information defined in other ways, in the form of KBN subscriptions and publications.

The prototype implementation, discussed within this paper, provides two main ways for users to express their interests: categories and keywords. The category values are extracted from the hierarchical concepts in the users’ ontology, and will be shown in listed menu for user selection. By using the KBN “More Specific”, “Less Specific” or “Equivalent” operators, which indicates the Sub-class, Super-class and semantic equivalence [8] relationships between ontological concepts, users can begin to outline the scope of their selected categories of interest. Presented in the work of Kenney et al. [8], the Wine ontology [29] offers an example used to illustrate the main KBN operators. The ontological type “wine” is less specific than the type “white wine”, “white wine” is more specific than “wine” since “wine” is a superclass of “white wine”. The keyword values within UcIM are typed by users manually, which express the particular set of words they are interested in. For example, if a user types “Manchester United” and “Football” as keywords, selects the bag operator as “Superbag” and the sub-operator as “Equal”, which is passed into the KBN as a subscription by the UcIM system, then any publication containing the bag of these two keywords, such as bag {“Sport”, “Manchester United”, “Football”} will be matched and a notification will be sent to the user.

Apart from expressing the content of interests, users can also select the degree they are interested in, by selecting “Less Important”, “More Important”, “Very Important” and “Most Important”. The importance degree generates an “Interest Value” which ranges from 0 to 1, attached with each of the users’ interests in IPML file. This “Interest Value” will be used to rank the sequence of matched notifications that may be sent back to the users by the KBN.
5.3.4 Interest Feedback

After browsing the notifications matched with their original interests, users provide feedback to the system by tagging their level of interest in each of the notifications. The system acts upon the tags assigned by the user to the notifications by adjusting the “Interest Value” of each concept, in order to provide users with more accurate information in future notifications. An example of this can be seen as the system reducing a user’s interest value in a subscription to the Olympic Games automatically, once a user begins to stop selecting the feeds associated with the Olympic Games itself. The user never explicitly “un-subscribes” from a subscription, they may however reduce their interest value to 0 and thus be unsubscribed by the UcIM system. The Interest Feedback loop is designed, as is central to UcIM system, to remove any complexity in creating subscriptions, their subsequent refinement and possible future un-subscription from the users’ perspective.

5.3.5 Notification Browser

Based on users’ notification settings, as previously described in Section 5.3.2, users browse the notifications that match with their interests in their own personalised manner. One way in which notifications may be delivered includes the notification browser. The browser lists all the notifications which match users’ subscriptions and displays in the format of: Title, Description and URL. The Title and Description are parsed from this feed to give users a brief summary of the notification, and the URL links to the website of the original article.

5.4 IPML File

Inspired by APML (Attention Profiling Mark-up Language) [9], IPML (Interest Profile Mark-up Language) is a way to track users interests in the form of an XML file. As described in Section 3, APML has become an evolving standard that is gradually being adopted by more and more websites and services as a way of gathering the attention data to represent people's interests and dislikes [10]. The UcIM system has not been developed to only use APML, therefore the APML schema cannot be directly imported into the system. However to track users interests and make these interests more easily manipulated by the UcIM system and web services alike, we establish the IPML Schema by adapting the APML format. The resulting IPML file is much more suitable for use within the UcIM system. To assure the continued convenience brought to this research by the adoption of the standard of APML, we keep the changes made in the IPML file to the minimum, and make it possible for IPML to be easily compatible with additional APML services. An IPML file is created from three parts, the user profile, personalised application settings and interest profiles, as created within the system. The user profile contains the account information created and modified by users, which are shown in the head of IPML file. The additional two components are both stored in the body of IPML file. The personalised application section includes all the users application settings, while the interest profile records all the interests tracked by the system, with regard to the individual user. Every time a user changes their profile, resets the application, or re-expresses their interests the IPML file will be updated accordingly. If the changes happen in the users’ interest profile, representing a change in their interests, the system will send this updated profile as a new subscription to the KBN.

The structured format of an interest profile is shown below, in figure 2. Each user’s interest profile contains Explicit Data and Implicit Data. The Explicit Data contains the interests directly expressed by users, as discussed in sector 5.3.3. The Implicit Data contains the users implicit interests which may in future work be able to be extracted from additional desktop applications or web services, which are detected by the UcIM system automatically. These two unique ways to represent interest expressions provide a method to realise the semantics of the interest management. However in the current prototype, the main focus is on the use of Explicit Data, and the introduction of Implicit Data is seen as future work.
5.5 Ontology Mapping

As described in Section 5.3.3, users use their own words to express interests by establishing the user ontology, whose classes form the user subscriptions passed to the KBN. This is beneficial as it allows each user to form their own subscriptions around their own collection of concepts. However, the published RSS feeds are based around the classes of the KBN routing ontology, which is a well-defined ontology used to tag each RSS feed within specific domains. Due to the difference between the user ontology and the router ontology, it is necessary to use ontology mapping to establish the relationship between the user and routing ontology. The fully automatic generation of mappings between different ontologies is generally considered impractical [30]. For this reason the mapping between the router and user ontology is conducted using a manual approach within the project. By defining “equivalence”, “subclass” and “superclass” relationships between the classes in the router and user ontologies, we express the mappings between ontological concepts. The work of Song et al. [31] has developed a KBN in which mapping strategies are supported with the KBN routers themselves. When mappings are included within the KBN routers ontology, the KBN is able to match the publication and the subscription based on the routing ontology and user ontology respectively. This allows users to express their interests individually, whilst preserving the KBN routing mechanism and ontological classification, a harmonisation of the two systems and a unique approach to interest representation.

5.6 Implementation

As well as the KBN, the UcIM system makes use of Java and other Web 2.0 technologies such as ontological data representation, Jena, JDOM, ROME, and AJAX. This section describes the basic concepts of these technologies and how they are used in the UcIM system.

**Ontologies** in both computer and information science describe the concepts of a domain and also the relationships held between those concepts. Ontologies are used as a form of knowledge representation regarding the world or some part of it [32]. In the UcIM prototype, we establish the user ontology to represent the knowledge regarding the users’ interests and use ontological mapping to represent the mutual relationship between the user ontology and the routing ontology. Both the user and routing ontology are expressed using the Web Ontology Language (OWL) [33] designed with the use of Protégé-OWL editor [32]. The **HP Jena Toolkit** is a Java Application Programming Interface (API)
and software toolkit used for manipulating RDF, RDFS SPARQL and OWL with an included rule-based inference engine [21]. The Jena OWL API provides the tools for use within the Java programmatic environment for dealing with the concepts and relationships represented within ontology. In the UcIM prototype, Jena is used to create a model that contains all the concepts of user ontology and then to read and operate these concepts as java objects.

Java Document Object Model **JDOM** is a unique Java toolkit for working with XML, engineered to enable the rapid development of XML applications [26]. As an open source library for manipulating XML data, JDOM is described by Jason Hunter [22] as “being created to be Java-specific and thereby take advantage of Java’s features, including method overloading, collections, reflection, and familiar programming idioms.” The UcIM system uses JDOM to create and subsequently modify the IPML file which contains meta-data about the users profile, application settings, and interests. JDOM is again used to read information from the interests profile of the IPML file to generate the subscriptions for use within the KBN. **ROME** comprises of a set of open source Java tools for the parsing, generating and publishing of RSS and Atom feeds using the JDOM XML parser [23]. ROME can parse, generate, and convert all of the popular syndication feeds formats by using Java objects in either specific types or generic normalised class. The set of tools, represented by ROME, makes it possible to parse RSS feeds from various sources, extract the information from each feed that the UcIM system is interested in, including the feed’s title description and URL and pass this information towards the KBN as publications.

**AJAX** (Asynchronous JavaScript and XML) consists of HTML, JavaScript, DHTML, and DOM and offers an approach that helps to transform Web interfaces into interactive Web applications [24]. Using AJAX, web applications retrieve data using the “XMLHttpRequest” Java Script object from the server asynchronously [34]. In the UcIM prototype, the techniques central to AJAX, such as JavaScript, HTML, XML, and DOM are used to develop the web application client-side and Java is used as the server-side language for client-server interaction.

### 6 Evaluation

The prototype UcIM system is designed to track users’ real interests and deliver only relevant information to them by providing a number of personalised interactions with system. Hence, it is planned that two sets of experiments will be conducted during August 2008 to evaluate the system in terms of the usability and usefulness respectively. The usability evaluation is designed to explore whether the interfaces provided are easy to use for non-expert users, how well the multiple interaction models provided by the system satisfy a broad range of users. Additionally the evaluation of “usefulness” mainly focuses on the basic “publish/subscribe” functionality provided by the system and aims to establish whether the system can deliver the set of notifications, corresponding to a users real interests, appropriately.

In the usability experiment, the “Think Aloud Method” [35] will be used to establish what users are thinking whilst using the system. In using this method the user is observed whilst performing various tasks and asked to describe what he/she is trying to do and why, what he/she thinks is happening etc. This approach documents how the system is actually used and provides rapid, high-quality, qualitative user feedback. To implement the think aloud method of evaluation a number of steps need to be followed: Firstly a number of tasks covering the use of each interface in the system is outlined, from profile establishment to the notification browser. Secondly, it is intended that 8 to 10 participants will be invited to test the system, each tester with a slightly different background based around gender, age, interests and importantly that each tester has a different level of the knowledge regarding the technologies used in the system. Thirdly, participants will perform tasks according to task scripts and describe their actions, intentions and thoughts to the observer. During this period the observer will guide the participant by asking short questions and take notes to record the evaluation process as well as an audio/video recording. Lastly, based around a questionnaire filled in by participants and the recording of think aloud experiments, it will be possible to analyse the current usability of the system to make future usability improvements a possibility.
In regards to the evaluation of the systems usefulness, it is planned that users will be asked to explore the system by themselves, paying particular attention to the contents of the delivered notifications. In aiming to ensure all the participants have a similar knowledge of how the system operates, the usefulness testing will be conducted with those who have participated in the previously conducted usability study. The testers will be asked to use the system within their day to day work cycle for one week and asked to interact with the system in multiple ways. By taking notes on what interests they expressed, and what notifications were sent to them, they will be able to judge whether the system can provide relevant information based upon their previous representation of interests. By making records on the specific functions they are satisfied or dissatisfied with, the users will be able to give a wide range of feedback on the overall performance of the system. Upon conclusion of the one-week experiment, participants will be interviewed to convey their point of views concerning the overall usefulness of the system.

Based on the two sets of experiments, we believe the user testing will provide a wide range of information to evaluate the overall usability and usefulness of the UcIM system. The results of this evaluation can then be used to make further improvements and outline future changes to the system with regard to these two aspects.

7 Conclusion and Future Work

In this paper is presented a new model for personalised interest management based on the publish/subscribe paradigm has been presented. The design and implementation of a prototype system has been documented in which users can express their real interests in the form of KBN subscriptions. The KBN is then used to deliver RSS notifications that semantically match the users’ subscriptions. The UcIM system has been designed to provide a number of personalised models of interaction. As mentioned, in Section 5.4 regarding the IPML File, an extension to this project is seen as the ability to detect and track users’ implicit interests, which could be sourced from various desktop applications such as documents and calendars, web history, email and bookmarks. How to extract meta-data reflecting users’ interests from these various sources is an open question for reflection in the projects future work. Only if the system is able to learn and crawl the implicit data sources will the semantic meaning of the user-centric interests manager be fully realised.

References


