A Flexible Personalization Architecture for Wireless Internet Based on Mobile Agents

George Samaras and Christoforos Panayiotou

Department of Computer Science, University of Cyprus
CY-1678 Nicosia, Cyprus
{cssamara,cs95gp1}@cs.ucy.ac.cy

Abstract. The explosive growth of the Internet has fuelled the creation of new and exciting information services. Most of the current technology has been designed for desktop and larger computers with medium to high bandwidth and generally reliable data networks. On the other hand, hand-held wireless devices provide a much more constrained and poor computing environment compared to desktop computers. That is why wireless users rarely (if ever) benefit from Internet information services. Yet the trend and interest for wireless services is growing with fast pace. Personalization comes into aid by directly toning down factors that break up the functionally of the Internet services when viewed through wireless devices; factors like the "click count", user response time and the size of the wireless network traffic. In this paper we present a flexible personalization system tuned for the wireless Internet. The system utilizes the various characteristics of mobile agents to support flexibility, scalability, modularity and user mobility.

1 Introduction

One of the problems of the Internet today is the tremendous quantity and unstructured information a user needs to search and navigate through to locate the desired one. To alleviate this problem the solution of personalization and user profiling (representing in some form the user's interests) is being lately proposed. The design, however, and implementation of such systems poses many challenges. These challenges become even bigger within the context of the wireless Internet. Some of the added issues introduced are the low bandwidth, the unreliable connectivity, the lack of processing power, the limiting interface of wireless devices and user's mobility. Adding to all these, is the huge variety and diversity of wireless devices with different capabilities and limitations. Thus, in order to build a personalization system that is tuned for the wireless Internet we must extend the user profile to include the characteristic of his device. In doing so we effectively introduce the notion of a device profile.

In a nut shell, the proposed personalization system provides innovative solutions to cellular network subscribers, it provides them with personalized content according the end-user preferences taking into account not only his/her profile but the profile of his/her handset device as well. The innovation relies in the way that the end-user receives the information to his/her mobile handset, i.e. with minimized clicks, and exactly the information required, thus reducing access time, browsing time and cost. As

Y. Manolopoulos and P. Návrat (Eds.): ADBIS 2002, LNCS 2435, pp. 120-134, 2002.

a matter of fact, for each end-user the system builds a unique wireless (e.g. WAP) portal that is created according to the user profile. To this end and in simple terms the system devises a flexible middleware and protocols to handle the diversity of content structures and their semantics. Performance and evaluation is also performed demonstrating the effectiveness and viability of the system.

The system utilizes the technology of mobile agents taking advantage their various characteristics such as asynchronicity, and mobility. Via mobile agents the system becomes modular, scalable, flexible but above all truly mobile. Consider, for example, the obvious requirement of having the profiles able to move around following the clients; user profiles implemented as mobile agents can do the trick!

The remaining sections of this paper are organized as follows. Section 2 presents a short introduction to the mobile agents technology. Section 3 presents the problem of personalization in general and section 4 presents our proposed architecture for this problem. Section 5 demonstrates the extensibility of proposed architecture. In section 6 we have a summary of the advantages of the utilization of mobile agents in our approach. Section 7 presents the implemented prototype, our experimentation and performance analysis. Finally section 9 concludes this report.

2 Mobile Agents

Mobile agents [3-5] are self-contained processes that can navigate autonomously. On each node they visit they can perform a variety of tasks. The underlining computational model resembles the multithreading environment in the sense that like individual threads each agent consists of program code and a state. The communication is usually done through the exchange of messages or RMI. The fundamental extension of this model is the mobility: each agent can autonomously relocate itself or its clones.

Each mobile agent needs to interact with the visited host environment to perform useful work. Thus, a daemon like interface (i.e., an agent execution environment) is provided that receives the visiting agent, interprets their state and send them to other daemons as necessary. This provides the visiting agent access to services and data on the current host. Issues of security are also handled by the agent interface. Mobile agents have been proved effective in a variety of Internet applications [6-9].

3 The Personalization Problem

The problem of personalization is a complex one with many aspects and issues that need to be resolved. Such issues include, but are not limited to, the following:

- ***What content to present to the user.*** This alone is a major problem as there is quite a large number of "sub issues" to deal with; How to decide what to show, using user profiles, using the user history to predict future needs etc.
  - When using user profiles we must address the need to store the interests of the user in a format that is easy to be used and be updated [15,17,23]. The main

problem here is the unpredictability of the user. *On the other hand within the wireless Internet we must also address the notion of profile mobility*.

- When using user profiles and thematic interests there is the problem of what the thematic interest really means. There is the need to be able to relate interests and items based on a semantics level. For example, lets consider the theme interest of "flowers"; the system must return everything that is related, such as florists or even fertile producers.
- Another aspect of content selection is the exploitation of user histories [19,20, 22,24,25]. The problem here is to find efficient and accurate mechanisms to read and comprehend the user history in order to make a good prediction of what the user will want. Machine-learning techniques to discover patterns, or data mining techniques to find rules are usually employed.

- ***How to show the content to the user***. Many users want to see the same things but their needs differ as to what form they want the data presented to them. The main issues here are, (a) the recording and storage of widely varying users needs (with user profiles) and (b) the set up of mechanisms that take the wanted content in an intermediate form and transform it to the appropriate form. This could be done through the use of XML and RDF [26]. *In the wireless environment this also relates to the used mobile device and its specific characteristics.*
- ***How to ensure the user's privacy***. Every personalizing system needs (and records) information about the habits of each user. This leads to privacy concerns as well as legal issues [16]. It also leads to lack of trust from the side of the user and could result in the failure of the system due to the avoidance of its use.
- ***How to create a global personalization scheme***. The user doesn't care if a set of sites can be personalized but at each one of them he has to repeat the personalization process. Efforts in this area take the form of personalized navigational spaces [18,21]. This includes, dynamic link updating, reduction of old links, "Meta searching" of multiple search engines and relocating information.

These are the major issues of personalization. It could be summarized in the following phrase: "***What, how and for everything***." The solution to these problems is often elusive and incomplete. There are many approaches to personalization and each one of them usually focuses on a specific area, whether that is profile creation, machine learning and pattern matching, data and web mining or personalized navigation. So far, to our knowledge, there hasn't been an approach that combines all the above techniques, or that gives a complete solution to the whole problem.

4 The Architecture: Personalization Systems Based on Mobile Agents

One of the issues of the personalization problem is the lack of an architecture that combines the existing techniques in a component-based fashion. Our aim is to propose such architecture, which in addition is flexible and scalable. Furthermore, we focus our efforts on the wireless Internet in general. We do that by avoiding tying up our proposal to specific wireless protocols (today that would be the WAP [10]) and taking into consideration mobility, device characteristics and any other limitations

imposed by mobility and the wireless medium. This is achieved by using, as much as possible, autonomous and independent, components. Thus, we can replace any component as needed without making the system inoperable. To achieve such a high degree of independence and autonomy we based our approach on mobile agents.

In a nutshell, our architecture suggests a system that resembles the notion of proxy that it sort of stands between the client and the content (and thus the content provider) and effectively separating the client's platform from the provider's platform. Specifically at one end we have the entry point of the client into our architecture (fig. 4.1:8) and on the other the servers that link the content providers with it (fig. 4.1:7). In between these ends is the heart of our approach, implemented by mobile agents. So we have agents that select, reform and deliver the desired content (fig. 4.1: 2 & 3).

However, for this to be possible we need a way to first describe the content and it's structure, for each participating provider, and secondly to manage the user profile. Thus two more components were added, one for each task (fig. 4.1:1 and 4.1:4 respectively). Having these two components we employ the mobile agents of our approach to take the profile of the user and "apply it" on the content's structure in order to restructure and reform it into the users personalized portal. In summary the components of our architecture are:

- Content Description component (Fig. 4.1: 1 & 5).
- Content Selection component (Fig. 4.1: 1 & 6).
- Content Reform component (Fig. 4.1: 2 & 3).
- User Profile Management component (Fig. 4.1: 4).

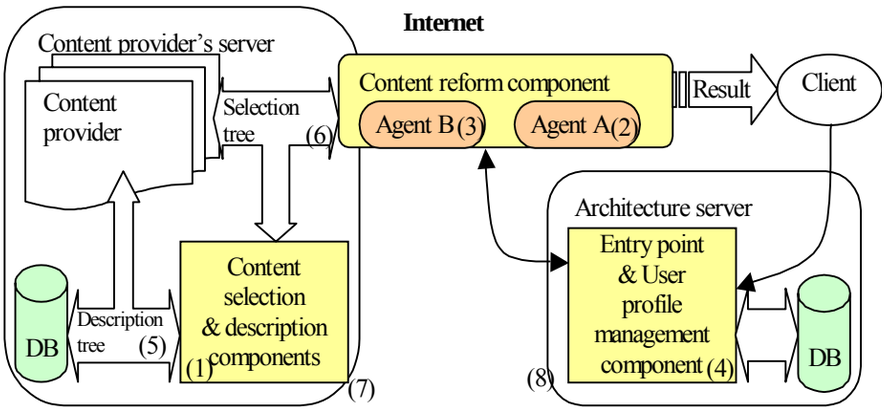


Figure 4.1. General View of the Architecture

4.1 Content Description Component

To be able to build this component we first need to decide how to represent the structure of the content. The common structure of a content provider's site in the wireless environment is a hierarchy of information types: from the most generic to the most

specific (fig. 4.2 shows such an example). This hierarchy can be easily represented by a tree structure.

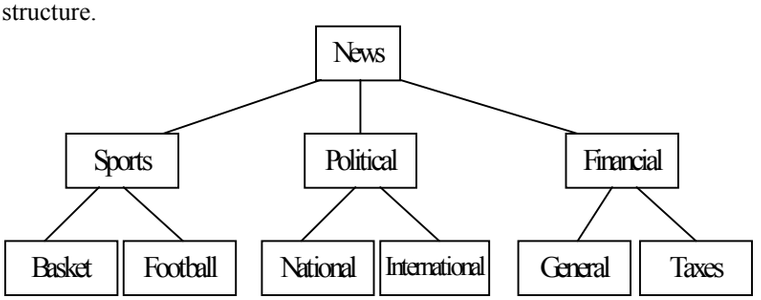


Figure 4.2. An Information Hierarchy of a site

Knowledge, however, of the content structure alone is not enough; we also need the context description of each content node. Thus, the sole purpose of this component is to construct a tree where each node of the tree contains this metadata. Fig. 4.3 shows one such tree (we call it the Description or Metadata Tree). In this way we are able to fully describe both the navigational structure as well as the context structure of the site of a content provider.

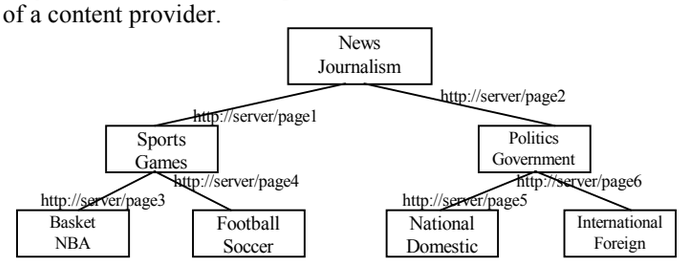


Figure 4.3. Description Tree example

At this point we introduce the concept of the "thematic interest" of each content node. The thematic interest is a context description of each node and represents the metadata of that node. In essence this metadata associates each node with one or more topics ("thematic interests" as we call them) that match the type of information under that node. For example, when we have a node that contains information on flower shops we associate with it the thematic interests of "florist", "flowers" etc. Once the Metadata Tree is produced existing techniques (such as pattern matching) can be used to select the wanted content. Note that, what selection technique is used is purely an implementation detail that has no impact on the design of the architecture. If for some reason an implementation that uses a specific technique proves inadequate it can be simply replaced within the appropriate component (i.e., the Content Selection component, see below). It is obvious that the Content Description component should be as close as possible to the real content (both for performance and security reasons).

4.2 Content Selection Component

This component uses the Description Tree and the user profile to produce a new trimmed tree (called the "Selection Tree") based on the user's needs. This tree represents the user's personalized portal. The needed steps are: (1) Finding all the useful nodes from the Description Tree, (2) Discarding the unneeded nodes and (3) Restructuring and further reducing the Description tree.

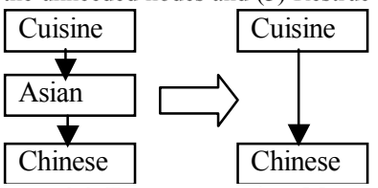


Figure 4.4: Tree restructuring. *We are interested only for Chinese Cuisine*

Matching and comparing the theme interests taken from the user profile and the Description Tree achieves the first step. The matching is done via any of the existing techniques or a combination of them (e.g. pattern matching and other AI techniques). In our experiments we used the "key words" approach. The second step just removes the unwanted nodes/branches from the Description Tree. In the final step we further reduce the tree by removing the inner unneeded nodes. Fig. 4.4 shows an example of step 3. Note the difference between steps 2 and 3: Step 2 drops all the nodes that aren't part of any navigational path to the interesting nodes while step 3 shortens these paths by eliminating unneeded inner nodes.

The final product is a reduced tree that contains only the desired nodes. The resulting tree does not need to contain the context metadata of the Description Tree. It only needs to contain pointers to the actual content pages/nodes. Fig. 4.5 shows this process.

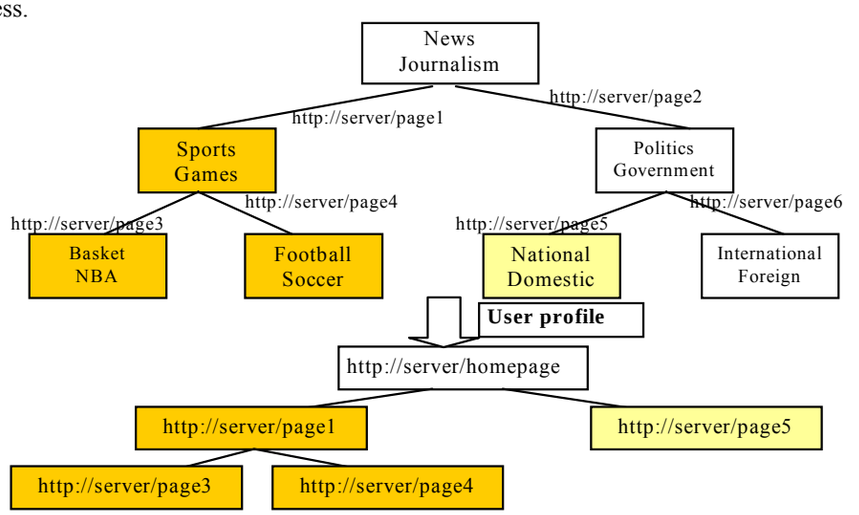


Figure 4.5. Transforming a Description Tree to a Selection Tree

This component should also recite at the provider's site and it is implement as a static agent. It must be an agent, as this will make it easier to dynamically install it,

move it as needed and support direct communication with the other components (i.e., the Content reform) that are made up by mobile agents.

4.3 Content Reform Component

This component is split in two parts. The first part takes as input a Selection Tree and delivers a modified version of the currently requested content node. The modification made is the removal or addition of links from the node according to the Selection Tree. Having done that it passes the (navigationally) modified node to the second part of this component. The second part reforms the received content node according to the user's device profile and then delivers it back to the client.

For this to be possible the Content Reform component must be able to understand the syntax of the content node. To satisfy this it is advisable for the content provider to describe its content in a standardized language. The easiest (and best) way to define such a language is the use of XML [11]. Note that the use of XML doesn't tie up our approach with a specific protocol. With XML we can easily transcode between various dialects.

This component is made up by two mobile agents and is based on the client/agent/agent/server (client/intercept) [1] mobile computing model (fig. 4.6). On the server side we have the agent (agent B) that reforms the navigation between the nodes and on the client side the one (agent A) that reformats the final results. Agent B moves in order to follow the requests of the user while agent A follows the user.

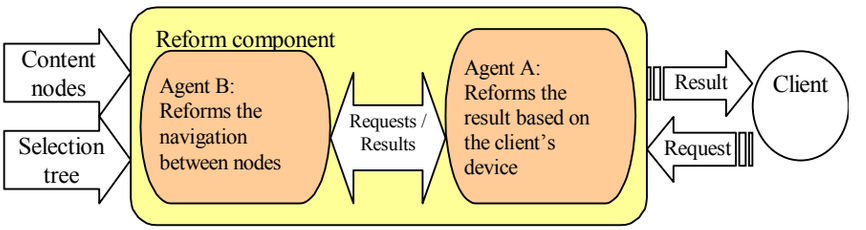


Figure 4.6. Content Reform Component

Agent B also carries part of the user profile and delivers it to the Content Selection component whenever it visits a new content provider. Thus agent B provides the collection of theme interests needed by the Selection component and gets in return the appropriate Selection Tree. Agent A on the other hand applies the user's device specifications on the currently requested node.

4.4 User Profile Management Component

This component provides a way to store the user's needs. By user's needs we mean both, his thematic preferences (i.e., the traditional notion of profile) as well as the characteristics of his personal device on which any requested info will be displayed. By focusing on wireless Internet the capabilities of the user's device become crucial,

as these wireless devices are quite restrictive on the form and length of the received content. It is therefore necessary to store and manage both the user's interest and the device's specifications. Yet these two kinds of data are quite different and indicated the need to split the user's profile in two parts.

The first part is used to hold the interests of the user. This part is actually a collection of theme interests (the same type used in the Description Tree). Thus we call this part the "*theme profile*". The second part of the user profile holds information about the user's device and is called respectively "*device profile*". Notice that these two parts are completely independent from each other.

The creation and management of these profiles can follow either a very simplistic or very complicated approach and is implementation specific. This is especially true for the "theme profile" where we can have (as a profile) just a collection of keywords that represent theme interests, or a complicated scheme that exploits the user's history. Similarly, for the device profile we may opt to just have a simple collection of device characteristics or to use a standardized format such as CC/PP [12]. The modularity and component independence that we implanted in our design allows us this flexibility. Indeed, if we modified the profile representation the only affected components are the Selection component (which uses the theme profile) and agent A of the Reform component (which utilizes the device profile).

Finally this component serves as the entry point of the client in our system. Our architecture supports a network of entry points (called "homes") resulting in a true distributed system. This "home" also provides the necessary interface to the user in case he needs to manipulate its profile. It also employees a mobile agent that can move in order to deliver securely the user profile to a specific "home".

4.5 Putting Everything Together: Integrating the Various Components

The two end points of the architecture represent services/servers for the moving client and servers for the content providers. These end-point servers are located somewhere in the Internet. On the client side we have the "home" server of our architecture, which is the entry point of the client in our system. Except of the Profile Management component it also holds the necessary services to initialize all the needed mobile agents (and thus components) used by the client. Upon the registration of a new client the initialization process begin. During initialization, agents A and B of the Reform Component upload the device and theme profile associated with the current user respectively. In this way we are able to move around the user profile without actually allowing access to it by the various content providers. Only components of our architecture have access to the two parts of the profile. Thus we secure the user's privacy, as well as improving the security provided by the mobile agent platform.

On the other side we have the participating provider's servers. These servers host the Content Description and Selection components. These components are initiated asynchronously, before any client activity. After all initiation tasks are completed we can serve user requests. This is done with the following procedure (fig. 4.7):

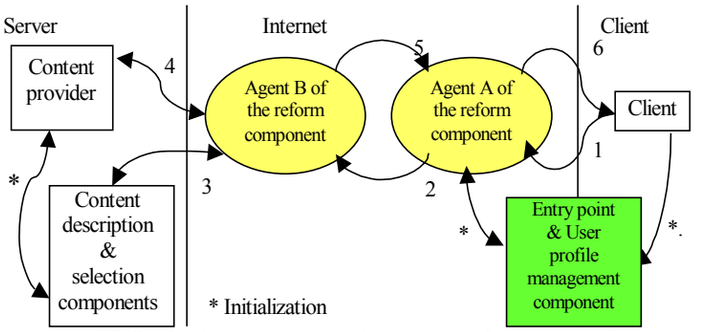


Figure 4.7. Graphical representation of the architecture's components in action

1. The client requests a node from agent A (either using HTTP or some other network protocol such as TCP/IP depending on the implementation). If this is the first request, it can only be the homepage of a particular content provider.
2. The request is passed on to agent B.
3. If not already done, agent B delivers the theme profile to the Content Selection component and receives the Selection Tree (which was produced dynamically based on the profile). This is done once, on the first request of the client.
4. Agent B pulls the relevant content node and modifies it according to the Selection Tree. It actually transforms the node to only contain the selected links discarding the rest.
5. Agent B returns the navigationally modified node to agent A.
6. Agent A reforms the received content node based on the device profile and returns it to the client.

5 Extensibility of the Architecture

The component-based structure of our architecture allows us to change or replace various components without further propagating any other changes. This level of "componization" is by and large due to the fact that our basic building blocks are mobile agents. Indeed the very nature of mobile agents and their object orientation offers a great degree of independence and flexibility.

We can add new components in our architecture just as easily. An obvious extension would be the incorporation of a mobile agent (component) that utilizes the offline time of the client; asynchronicity is one of the main virtues of mobile agents technology. Its functionality could include (without been limited to) prefetching, caching or the materialization of the users private wireless portal. In essence extend the capabilities of step 4 of fig. 4.7. Furthermore, we can add components that visit the participating content providers in order to gather and process user's historical data to aid prediction on future demands.

This architecture via the utilization of mobile agents and its design has great potential as far as extensibility is concerned. To make this extensibility potential more understandable let's consider that if we removed agent B from our system it would

still continue to work without any problem. Of course we cannot do that for any component.

6 Advantages of the Use of Mobile Agents

Mobile agents technology has been proven quite suitable for wireless systems. Thus a system based on them, such as ours, is much better prepared to face the challenges of the new era of wireless Internet. This is due to the fact that mobile agent are excellent for asynchronous communication and execution (e.g., the creation of the Description Tree, the creation of the personalized portal), for mobility (e.g., having agents A and B to follow the user and the user's request respectively, in fact agent B can transfer the user profile to the various destinations asynchronously as needed).

A less obvious advantage is the increased protection of the user's privacy. Keep in mind that the user-profile data remain within the mobile agents, blocking the provider's access to them. Furthermore by utilizing the built-in security mechanisms of the mobile agent's platform we strengthen the privacy of the user by securing the transmission of sensitive information.

Another important advantage that flows from the use of mobile agents is the ability to dynamically incorporate various mobile computing models [1]. In fact we have extensively utilize/materialized the client/intercept model, which has been proven quite effective in the wireless environment [1,2,8].

One other important advantage of the use of mobile agents is the flexibility and object orientation that enhances the autonomy needed in a wireless environment.

7 Prototype Implementation and Experimentation

Our prototype implementation is a personalization system for WAP services. We selected WAP as our target as today this is the standard of the wireless Internet. Note that our system is not tide to any particular protocol and can be used long after WAP is no more the wireless Internet standard

As discussed previously we have the Content Description component, which builds the Description Tree that fully describes the content of a provider. In order to construct this tree we have implemented a crawler that automates as much as possible this procedure. This crawler resides at the provider's site and remains within the provider's boundaries.

One final note on the implementation is that we added the factor of "locality". It is common in WAP application to provide location-based information. Thus by knowing the location of the user beforehand, we can further trim the Selection Tree based on the locality of the content. As an example consider a service that gives information on restaurants. If I wanted to find a restaurant to eat I'd like that to be near me.

Having the prototype implementation we needed to find a way to evaluate it. Thus we needed to find a set of metrics that could give as a representation of the added benefits of the system. We have selected the following metrics:

- *The "click count".* The number of links that the user must follow in order to reach the desired content.
- *The size of the network traffic*. The size of the content delivered over the wireless link. This is especially important as the wireless link is very slow and often a bottleneck. Note that this is linked with the click count: fewer clicks means (most of the times) less traffic.

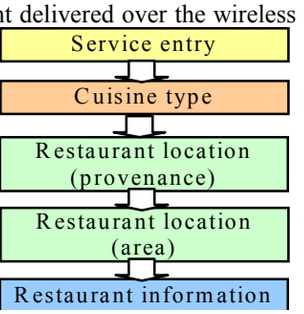
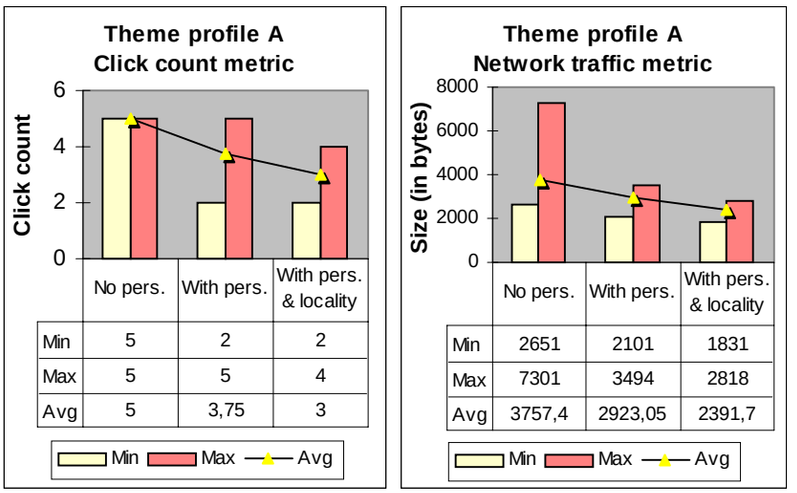


Figure 7.1: Provider's Content Hierarchy

Furthermore something that can't be objectively measured is the time that the user needs to navigate from one node to another. This is depended on the number of options the user has as well as the confusion produced by these options. By eliminating undesired options we keep the confusion to a minimum and speeding the process of finding the desired link to follow. This was obvious during our tests.

For our experimentation we came up with two testing scenarios (effectively two different user profiles), one that would produce a "wide" Selection Tree (many different branches - scenario A) and one that would produced a "deep" Selection Tree (few long branches - scenario B). These scenarios were tested both with and without our system. In addition within our system we have tested the effect of exploiting the locality factor.

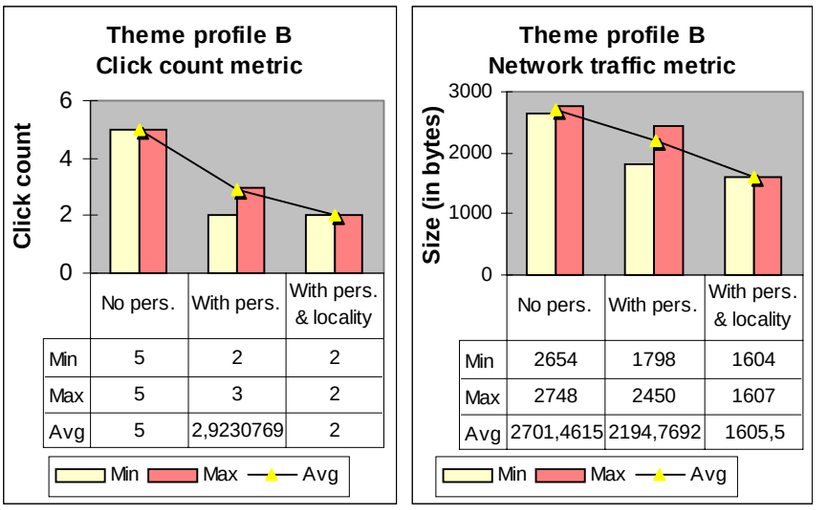


Graph 7.1: Scenario A

Our measurements were made using a real content provider's data[1]. The tested service was a restaurant information service with almost 2200 leaf nodes and a hierarchy of up to 5 levels (fig. 7.1 shows the hierarchy). The performed test was very rigorous. We have evaluated both metrics for EVERY possible selection produced from the user's profile. In essence from the profile we collected all the qualified restaurants

[1] WINMOB Technologies is a wireless service provider residing in Cyprus.

and we access each one of them with and without our system. Finally, note that the tests were concluded using the Nokia Mobile Internet Toolkit, which accurately emulates a real WAP client.



Graph 7.2: Scenario B

As shown in the Graphs we present the best, the worst and the average case of the "personalization effect" on the service (min, max and avg. respectively). Graph 7.1 (i.e., scenario A) shows that with no personalization every selection requires 5 clicks while with personalization the average case is reduced to 3,75 clicks and further reduced to 3 clicks when we exploit the locality factor. Similarly the graph of the network traffic metric shows significant reduction with personalization. Notice that due to the varying size of the content nodes the best and worst case of wireless network traffic without personalization is different.

The improvement is quite significant; the percentage improvement of "with personalization" vs. "without personalization" for the average clicks case is 33,3%. Including localization is even better. Graph 7.2 presents a similar analysis for scenario B.

It is clearly seen that we have a significant benefit from the use of our system. We also see that the locality factor plays a crucial role on location-based services. One important observation is that the second scenario presents us with much better performance improvement. This is attributed to the type of the Selection Tree. In the first scenario we have a "wide" Selection Tree with many different sub-branches and this in general reduces the number of unneeded inner nodes, which in turn hampers performance. In fact it minimizes the effect of step 3 of the Selection Tree algorithm.

8 Related Work

The problem of personalization is quite complex. It is a general problem relating to information retrieval and it can be found even outside the Internet domain. The major encounters of this problem, though, are usually within the Internet. To our knowledge, however, no other work has so far focus on the wireless Internet and its specificities. Most of the known work is Internet based.

We encounter approaches that are based on agents (not mobile) that act as proxies in offering the personalization services. WBI [15,17] is such a system. The difference with our approach is at first, the flexibility provided by the mobile agents, and secondly the operation mode. WBI works by using several plugins, which must be at the client's machine in order to automate tasks that the user would perform on his own. BASAR [21] is a similar system (also uses static agents) that manages and updates the "personal webspace" which is created by the user's bookmarks collection. Siteseer [18] is a system that is based on the user's bookmarks analysis in order to predict and suggest relevant, possibly interesting Internet sites.

Another approach to the problem is the analysis and modeling of the user in order to predict the user's future moves. [22] and [25] describe two such systems that incorporated machine learning and artificial intelligence techniques respectively. The first describes a system that recognizes user behavioral patterns and predicts the user's future moves. The later focuses in harvesting and managing the knowledge that comprises the user profile. Modeling the user through rule discovery and validation is another approach. The 1:1Pro [19] system is a representative of this category, and uses data mining techniques to achieve its goal.

Yet another approach is the exploitation of histories in order to reduce the results of information retrieval. Haystack [20] is a system that gathers the transactional history of the user in order to discover knowledge that will use to limit the results of information retrieval to the interesting information.

One other AI based solution is the one presented in Proteus [13,14]. This system creates models of each user for the purpose of adjusting the nodes of some Internet site to the needs of the user. This adjustment is made by rearranging the order in which pieces of information are presented in the resulting page. Furthermore it has the ability to reorder or remove links to other pages. This last point is very similar with our approach except that we don't reorder the information of the page. We do, however, perform a multiple level tree restructuring.

Another interesting, and with great potential, approach is the use of "theme profiles". Such a profile holds the theme interests of the user. The biggest problem here is the management of this profile. [23] is a subsystem of the CiteSeer services that follows this approach. The difference with our approach is that we just kept the main idea of the "thematic profiles" leaving the specifics as implementation details. [26] presents an approach that is based on the use of XML and RDF (Resource Description Framework).

Finally we have the approach followed by the eRACE [27] system, which is a prefetching and precaching system that further personalizes the results. It utilizes user profiles (written in XML) to search the Internet for possibly interesting nodes. Afterwards it downloads all the potentially interesting nodes presenting them in a uniform fashion. eRACE searches to HTTP, NTTP, SMTP, and POP3 sources.

9 Conclusions

In this paper we have presented a flexible personalization architecture for Wireless Internet based on Mobile Agents. The system utilizes mobile agents as the fundamental building block and in doing so capitalizes on a number of specific to mobile agents technology advantages. Some of them being the following:

- Flexibility and independence between the various components.
- Asynchronous communication and execution (e.g., the creation of the Description Tree, the creation of the personalized portal),
- Mobility (e.g., by moving the user profile asynchronously as needed)
- Increased protection of the user privacy and increased security.

One other important strength of our approach is flexibility. Being component based we can easily extent our architecture by adding new components as necessary. Furthermore, this flexibility allows us to split the user profile in two: the theme profile and the device profile. Wireless Internet makes the existence of a device profile a necessity. There are many different devices (mostly handheld) with quite different capabilities and, most importantly, different limitations. Supporting device profile eliminates many unneeded problems. The incorporation of the device profile in the design of our architecture relieves the content provider from the necessity to handle different devices as this is now done by the personalization system.

Finally, the prototype shows proof of concept with its viability as well as its promising results during our tests. Our performance evaluation indicates significant improvement over the traditional with no personalization approach. Initial results show improvement that ranges from 35% to 140% dependent on the profile. We do expect this figures to fare even better once we incorporate better profile management algorithms in our architecture.

References

[1] Evaggelia Pitoura and George Samaras. "*Data Management for Mobile Computing"*. Kluwer Academic Publishers, 1997

[2] B.C. Housel, G. Samaras, and D.B. Lindquist. WebExpress: "Client/Intercept Based System for Optimizing Web Browsing in a Wireless Environment". *ACM/Baltzer Mobile Networking and Applications*, 1997.

[3] C. Harrison, D. Chess, and A. Kershenbaum, "Mobile Agents: Are they a good idea?", IBM Research Division, T.J. Watson Research Center, 1995.

[4] D.B. Lange and M. Oshima. "Seven Good Reasons for Mobile Agents". *Communications of the ACM*, 42(3):88-91, 1999.

[5] Robert Gray, David Kotz, George Cybenko, and Daniela Rus. "*Agent Tcl"*. In William Cockayne and Michael Zyda, editors, "*Mobile Agents: Explanations and Example"s, Manning Publishing*, 1997.

[6] D. Barelos, E. Pitoura, and G. Samaras, "Mobile Agent Procedures: Metacomputing in Java", *Proceedings ICDCS Workshop on Distributed Middleware*, (in conduction with 19th IEEE International Conference on Distributed Computing Systems (ICDCS'99)), pp. 90-93, Austin, TX, 1999.

[7] P.E.Clements, Todd Papaioannou, and John Edwards. "Aglets: Enabling the Virtual Enterprise". *Proceedings MESELA'97 Conference,* Loughborough University, UK, 1997.

[8] G. Samaras, E. Pitoura, and P. Evripidou, "Software Models for Wireless and Mobile Computing: Survey and Case Study". Technical Report TR-99-5, University of Cyprus, 1999.

[9] P.K. Chrysanthis, T. Znati, S. Banerjee, and S.K. Chang. "Establishing Virtual Enterprises by means of Mobile Agents". *Proc*eedings *10th IEEE RIDE Workshop,* pp. 116-125, Sydney, Australia, 1999.

[10] WAP Forum, Technical specifications, http://www.wapforum.org

[11] T. Bray, J. Paoli and, C.M. Sperberg-McQueen. "Extensible Markup Language (XML) 1.0 Specifications". World Wide Web Consortium, http://ww.w3.org/TR/Rec-xml

[12] "Composite Capabilities/Preference Profiles", World Wide Web Consortium, http://www.w3c.org/Mobile/CCPP/

[13] Corin R. Anderson, Pedro Domingos, and Daniel S. Weld. "Adaptive Web Navigation for Wireless Devices", *Proceedings 17th IJCAI Conference*, 2001.

[14] Corin R. Anderson, Pedro Domingos, and Daniel S. Weld. "Personalizing Web Sites for Mobile Users*"*. *Proceedings 10th WWW Conference*, 2001.

[15] Paul Maglio and Rob Barrett, "Intermediaries Personalize Information Streams", *Communications of the ACM*, 43(8):96-101, 2000.

[16] Eugene Volokh, "Personalization and Privacy", *Communications of the ACM*, 43(8): 84-88, 2000.

[17] Rob Barrett, P. Maglio, and D. Kellem. "How to Personalize the Web". *Proceedings CHI Conference*, 1997.

[18] J. Rucker, J.P. Marcos, "Siteseer: Personalized Navigation for the Web", *Communications of the ACM*, 40(3):73-75, 1997.

[19] G. Adomavicius and A. Tuzhilin. "User profiling in personalization applications through rule discovery and validation", *Proceedings KDD Conference*, 1999.

[20] E. Adar, D. Karger, and L. Stein. Haystack: "Per-user information environments". *Proceedings CIKM Conference*, 1999.

[21] C. Thomas and G. Fischer. "Using agents to personalize the web". *Proceedings ACM IUI Conference*, pp.53-60, Orlando, FL, 1997.

[22] Haym Hirsh, Chumki Basu, and Brian D. Davison. "Learning to personalize". *Communications of the ACM*, 43(8), 2000.

[23] Kurt Bollacker, Steve Lawrence, and C. Lee Giles. "A system for automatic personalized tracking of scientific literature on the web". *Proceedings 4th ACM Conference on Digital Libraries*, pp.105-113, New York, 1999.

[24] M.D. Mulvenna, S.S. Anand, and A.G. Buchner. "Personalization on the Net Using Web Mining". *Communications of the ACM*, 43(8): 123-125, 2000.

[25] Sung Myaeng and Robert Korfhage. "Towards an intelligent and personalized information retrieval system". Technical Report 86-CSE-10, Dept. of Computer Science and Engineering, Southern Methodist University, Dallas, TX, 1986.

[26] I. Cingil, A. Dogac, and A. Azgin. "A Broader Approach to Personalization". *Communications of the ACM*, 43(8):136-141, 2000

[27] M. Dikaiakos, D. Zeinalipour-Yazti, "A Distributed Middleware Infrastructure for Personalized Services," Technical Report TR-2001-4, Department of Computer Science, University of Cyprus, December 2001.