# SIDE Surfer: a Spontaneous Information Discovery and Exchange System

David Touzet
IRISA/INRIA
Campus de Beaulieu
35042 Rennes
France
dtouzet@irisa.fr

Jean-Marc Menaud
Ecole des Mines
4, rue Alfred Kastler
44307 Nantes
France
jmenaud@emn.fr

Frédéric Weis
IRISA/Univ. Rennes 1
Campus de Beaulieu
35042 Rennes
France
fweis@irisa.fr

Paul Couderc
IRISA/INRIA
Campus de Beaulieu
35042 Rennes
France
pcouderc@irisa.fr

Michel Banâtre
IRISA/INRIA
Campus de Beaulieu
35042 Rennes
France
banatre@irisa.fr

## Abstract

*Development of wireless communications enables the rise of networking applications in embedded systems. Web interactions, which are the most spread, are nowadays available on wireless PDAs* [1]*. Moreover, we can observe a development of ubiquitous computing. Based on this concept, many works aim to consider user's context as part of the parameters of the applications. The context notion can include the user's location, his social activity . . . Taking part from emerging technologies enabling short range and direct wireless communications (which allow to define a proximity context), the aim of our study is to design a new kind of application, extending the Web paradigm:* spontaneous *and* proximate *Web interactions.*

## 1. Introduction

Development of new hardware architectures enabling proximity wireless communications, such as Bluetooth [7], allows to envision direct information exchanges between neighbouring nodes such as PDAs. Indeed, PDAs can nowadays be used to store personal information. Assuming users declare some of this information as freely accessible, we could then envisage direct exchanges of information when two mobile PDAs meet (*i.e.* are able to communicate together).

[1] Personal Digital Assistant

One possible application of this new kind of networking could be an extension of the Web interactions: a *proximity Web* in which people may *spontaneously* collect proximate information, *i.e.* information stored on neighbouring PDAs. Due to users' mobility and to the volatile aspect of the system, such an application raises some problems that are not handled by the classical Web architecture. In this paper, we propose an architecture adapted to the constraints of a proximity Web.

This paper is organised as follows. Section 2 describes our objectives and the constraints we have to take into account. In the third section, we highlight the mechanisms we propose to achieve our goals. The forth section presents some of the unsolved problems and remaining works. Finally, we briefly remind the studies related to our works in section 5 before concluding.

## 2. Motivations

As given above, we aim to enable spontaneous exchanges of information between mobile PDAs. For simplicity sake in the following, we only consider the encounter of two PDAs. However, the proposed mechanisms can easily be extended to manage meetings of several PDAs. We focus our work on the adaptation of the Web interactions to this mobile and spontaneous environment.

As an example, we could consider the use of a proximity Web during a conference where attendees are carrying PDAs equipped with short-range wireless communication facilities. Let us assume that each attendee stores information

of interest on his PDA. In the conference case, this could be a set of Web pages extracted from his project Web site. In such an environment, a proximity Web system could collect interesting information (such as common research interests) from PDAs carried by people met during the conference. The spontaneous aspect of this system means that users are not disturbed neither to define their topics of interest nor when proximate requests are performed.

To support such Web interactions, three main problems have to be addressed:

- *Users' profiling.* In order to perform automatically relevant requests, the user's topics of interest have to be discovered. For this purpose, the system has to build a user's profile without resorting to external instructions. To make use of the locally stored documents is a good way to achieve this goal.

- *Indexation of public information.* PDAs have to automatically organise their public documents in order to enable efficient retrieval during encounters. We believe that providing a thematic indexation of the stored documents is a good mean to speed up the exchanges.

- *Spontaneous discovery of relevant information.* The way relevant remote information is retrieved has to be defined. During the encounter of two persons, such a mechanism has to perform the selection of documents to exchange according to the profile of each protagonist.

To address these issues, we have to deal with the constraints inherent to the mobile computing environment. Indeed, compared to workstations, PDAs suffer from a lack of computational, storage and energy resources. Besides, we have to manage a totally distributed architecture since no server is available to store all accessible information. This fact implies that each PDA will have to act as both server (for its own information) and client (with respect to remote information). Finally, the communication capabilities are direct and based on temporary links (because of the unconstrained mobility of the users) instead of a fixed infrastructure as in the traditional Web.

In order to meet our objectives in spite of these constraints, we have designed two mechanisms which are detailed in the next section.

## 3. Design of a Proximity Web

To manage the process of spontaneous information discovery, we choose to resort to the Web model in which documents are labelled with keywords. In order to alleviate the spelling divergences, keywords matching methods usually relies on close keywords matching mechanisms [6]. However, these mechanisms seems difficult to adapt to our embedded environment. Indeed, algorithms used to provide such a service are quite greedy: they are not adapted neither to the poor resources of appliances nor to the limited communication times. In order to enable a search process running without this service, we consider that keywords chosen by users to label their documents are got from a common *ontology*. We could assume, for example, that users could use a common keyword-based classification of topics (such an assumption is realistic in the case of a conference).

Concerning the architecture, our system dissociates itself from the Web: it relies on peer-to-peer *symmetrical interactions* instead of the Web *client/server* model. Indeed, during an encounter between two mobile PDAs, each node has to find the remote information matching with its profile. In fact, as profiles are deduced from public databases, the nodes have to discover common interests on behalf of their user. Such a task is typically performed more efficiently by a cooperative scheme than by two independent client/server processes. We therefore choose to focus our study on such interactions.

The design of this symmetrical mechanism is based on two main components. The first one is the *database manager*: it deals with PDAs' public database, which contains the set of public Web pages. The manager has to extract a user's profile from this database in order to perform some automatically relevant requests. It also has to build a thematic indexation of the stored Web pages in order to quickly satisfy requests the PDA receives.
The second component is a *distributed search engine*: it organises the information discovery between two PDAs using information provided by their database manager.

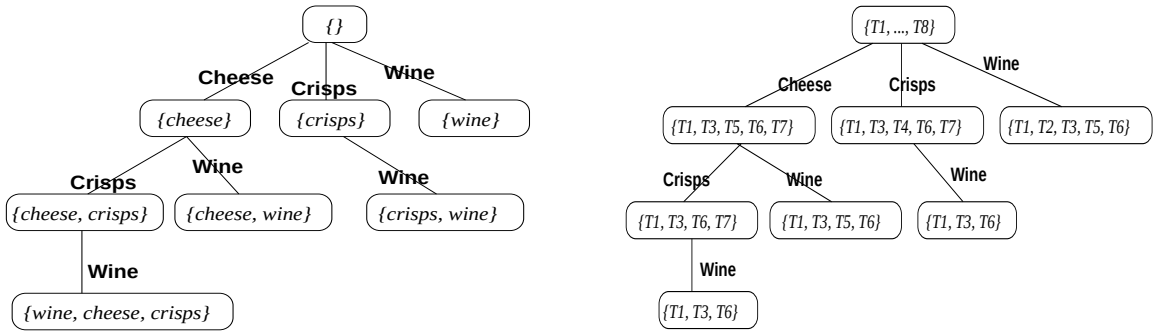
The following subsections are devoted to the description of these two components.

### 3.1. The database manager

The public database of a PDA is the set of documents its user declares as freely accessible. Assuming these documents are representative of his interests, this database can be used to automatically profile the user. The other issue to address is the thematic indexation of documents stored in such a database.

To reach these goals, we adapt some data mining techniques. Indeed, applying such techniques on a user's public database is a way to discover his main interests in order to build a set of relevant requests that could be executed during future encounters. We design a profiling scheme, which can also be used as an indexation structure for the public stored documents. By this mean, we gather in a single structure the user's requests and an index for his public database.

Before further describing this algorithm, we present the




(a) Products

(b) Transactions

**Figure 1. Data mining indexation structures**

generic data mining method.

**3.1.1. Data mining presentation.** Data mining is usually used to discover hidden dependencies between objects of huge databases. The basket market example is the most frequently used. In this example, we consider a supermarket database whose records are products bought by customers. Data mining can here be used to discover which products are often bought together and which ones never appear in the same market basket. Such information can then be exploited to realise promotional offers.

Let us introduce a few definitions. A set of $k$ products is called a *k-itemset*. The *support* of a k-itemset is the number of database records containing it. Finally, a *minimum support* (called $MinSup$) from which itemsets are considered as *frequent* is defined by the user.

The traditional data mining algorithm is the following. A first pass on a whole database enables to discover all frequent 1-itemsets (*i.e.* the frequent products). Finding the frequent 2-itemsets (the frequent sets of two products) can be achieved by only scanning the set of frequent 1-itemsets, as shown in [1], and so on ...

| | $wine$ | $crisps$ | $cheese$ | $ham$ |
|---|---|---|---|---|
| $T_1$ | X | X | X | |
| $T_2$ | X | | | |
| $T_3$ | X | X | X | X |
| $T_4$ | | X | | |
| $T_5$ | X | | X | |
| $T_6$ | X | X | X | |
| $T_7$ | | X | X | |
| $T_8$ | | | | X |

**Table 1. Database example**

**Example.** Table 1 shows an example of purchases made in a supermarket (each line corresponds to a purchase). Considering $MinSup$ equal to 3, we begin to search for frequent 1-itemsets. The count gives us the following results: $\{wine\}$ 5, $\{crisps\}$ 5, $\{cheese\}$ 5 and $\{ham\}$ 2. So, the frequent 1-itemsets are $\{wine\}$, $\{crisps\}$ and $\{cheese\}$. To discover the frequent 2-itemsets, we now only consider the set of frequent 1-itemsets: among the combinations of elements in frequent 1-itemsets, which ones are frequent 2-itemsets ? We have the following counts: $\{wine, crisps\}$ 3, $\{wine, cheese\}$ 4 and $\{crisps, cheese\}$ 4. All these sets are therefore frequent. We proceed in the same way to discover the frequent 3-itemsets. The only one contained in this database is $\{wine, crisps, cheese\}$ with a support of 3.

**3.1.2. Toward a profile-based indexation scheme.** The main application of data mining is the discovery of frequent itemsets among databases. This technique is here applied to extract frequent sets of keywords characterising the stored documents. To achieve this goal, we consider the set of public Web pages of a PDAs as a database. Stored documents then correspond to the database records and the keywords describing them are viewed as the items of these records.
To profile the user's interests, we adapt the data mining algorithm in order to supply a tree structure on frequent keywords (see Figure 1(a)). As only frequent sets of keywords are indexed, the structure built by the algorithm provides a good representation of the user's main interests. This structure can also be used to classify the documents characterised by these frequent keywords (see Figure 1(b)): a document is indexed from a node if the set of keywords describing this node is included in the set of keywords characterising the document. Thus, the structure provided by this method can be exploited as both a user's profile and a thematic index for public documents.

Recalling the example above, we now describe how the

**Algorithm 1** Discovery algorithm of the initiator

$commonList \leftarrow$ subtrees($Root$)
send($commonList$)
**repeat**
  $subList \leftarrow$ subtrees($commonList$)
  receive($remoteList$)
  $toSendList \leftarrow$ select($remoteList, commonList$) $\cup$ ($remoteList \cap subList$)
  transfer($objects\ indexed\ from\ toSendList$)
  $commonList \leftarrow subList \cap remoteList$
  **if** $remoteList \neq \emptyset$ **then**
    send(subtrees($commonList$))
  **end if**
**until** $commonList = \emptyset$

**Algorithm 2** Discovery algorithm of the passive entity

$commonList \leftarrow Root$
$subList \leftarrow$ subtrees($commonList$)
**repeat**
  receive($remoteList$)
  $toSendList \leftarrow$ select($remoteList, commonList$) $\cup$ ($remoteList \cap subList$)
  transfer($objects\ indexed\ from\ toSendList$)
  $commonList \leftarrow$ subtrees($subList \cap remoteList$)
  $subList \leftarrow$ subtrees($commonList$)
  **if** $remoteList \neq \emptyset$ **then**
    send($commonList$)
  **end if**
**until** $commonList = \emptyset$

tree structure is obtained. A product labels each edge (and its following node) of the structure. Those labelling the first level of nodes are obtained by considering the frequent 1-itemsets. Second level nodes represent the frequent 2-itemsets. They are labelled by the keyword completing their parent itemset.
Frequent 2-itemsets are obtained by combining two frequent 1-itemsets: second level nodes have thus two potential parents. Among these, the one that is elected is the one with the associated higher support (in the event of support equality, the alphabetical order is used). Thus, each node has only one parent, is reached by only one edge and, consequently, is labelled by a unique keyword.
In the transactions indexation structure 1(b), transactions are indexed from each node representing a k-itemset they contain (so, a transaction can be indexed from several nodes).

**Memory management.** The value of $MinSup$ can be used to control the size of indexation structures. Indeed, the smaller this value, the greater number of frequent itemsets. Thus, by increasing this threshold we limit the number of frequent itemsets and, in this way, the size of the profile decreases. By this mean, users can adapt the size of their profile to their PDA capabilities.
However, this parameter has to be carefully managed due to its impact on the information discovery performances: by reducing the accuracy of its profile, a user also reduces the set of interesting information which could be automatically found during encounters.

### 3.2. A distributed search engine

The spontaneous information discovery process relies on the profile-based indexation structure described above. This discovery is gradually performed by computing successive intersections between two remote profiles. We first present a basic protocol for the mutual information discovery. In a second part, we propose an improved scheme, which makes use of the relevance concept.

**3.2.1. A basic information discovery protocol.** During the encounter of two mobile PDAs, each entity has to discover among the remote indexation structure which documents are relevant to its user's profile. Those considered as relevant have then to be spontaneously downloaded. To achieve this goal, our protocol looks progressively for the intersections between the two users' profiles.

When an encounter between two mobile PDAs, $A$ and $B$, occurs, the mutual information discovery protocol is initiated. In order to perform this discovery in an efficient way, the entities begin by electing an initiator for further exchanges. Let us assume that $A$ is the initiator in the following. Once this first step is completed, $A$ sends to $B$ the set of keywords describing the first level of its profile. As soon as $B$ receives such a message, it can perform the intersection between its own first level profile and the remote one. $B$ then answers $A$ with the descending nodes of its calculated intersection. In its turn, $A$ computes a new intersection based on $B$'s answer and the process is repeated until the computed intersection is empty. Transmitted documents are those indexed from the successive computed intersections (taking care to transfer only once those that are multi-indexed).

Whole mechanisms are given by Algorithm 1 for the initiator and by Algorithm 2 for the passive entity.

**Example.** Figure 2 presents the communications generated by the process of mutual interesting information discovery between two entities $A$ and $B$ (assuming that $A$ initiates the exchange).
As a first step, $A$ sends to $B$ the keywords of the first level of its profile *(i)*. $B$ then computes the intersection between the received information and its local profile. It returns to $A$ the subtree of each node of the computed intersection *(ii)* and

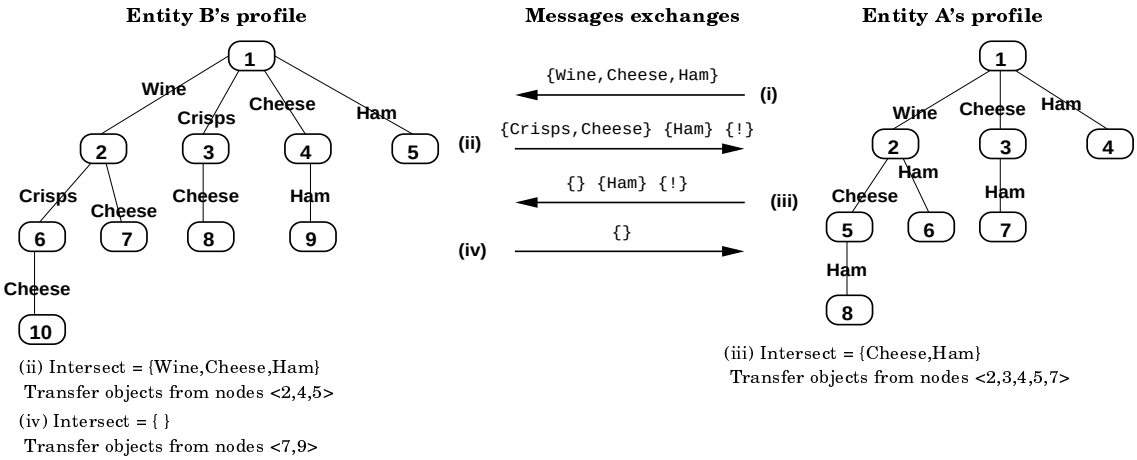



**Figure 2. Common interests discovery mechanism**

starts to transfer the Web pages indexed from these nodes. In our example, $B$ returns to $A$ the set $\{Crisps, Cheese\}$ for the keyword $Wine$, the set $\{Ham\}$ for the keyword $Cheese$ and an leaf set (symbolised by *!* on the figure) for the keyword $Ham$ (which means that it belongs to the intersection but has no child). Based on the $B$'s answer, $A$ is now able to compute its first level intersection (this task is performed by the $select$ function in Algorithms 1and 2). A keyword belongs to this intersection if the received message contains a corresponding and non empty set (leaves sets are considered as non empty). It can also computes a new intersection with its second level nodes *(iii)*. It has no $Crisps$ keyword following $Wine$, so it notifies $B$ by sending it an empty set. Concerning the keyword $Cheese$, it has a corresponding node with a child, $Ham$, that is returned to $B$. $A$ has equally a corresponding node for the received $Ham$ keyword. As this last node locally has no child, $A$ sends to $B$ a leaf set. Finally, the received leaf set only indicates that $B$ has no more keyword corresponding to the $A$'s profile in this part of its indexation tree.
The process then repeats in the same way until an empty intersection is calculated and all relevant documents are transferred *(iv)*.

**Protocol improvements.** This first protocol presents some limitations. Indeed, the only exchanged documents are those that belong to strict profiles intersection. That means that Web pages can be transmitted as soon as one common keyword is discovered, even if it is the only one that matches between two profiles.
Another limitation appears when a part of a profile matches completely with a remote profile. If the corresponding remote profile is deeper than the local one, it is highly likely that documents indexed by corresponding deeper remaining nodes are also relevant for the local profile. However, using our basic protocol, such documents are not exchanged.
Addressing these limitations can be achieved by introducing a relevance calculation between profiles (as it is used by Web search engines [6] between requests and indexed documents). A way to compute this relevance is based on the ratio of common keywords between documents indexed by the profiles. We adapt this technique to the constraints of mobile PDAs by applying it directly between the profile structures (instead of indexed documents). A *relevance threshold* (specified by the users) is used to distinguish interesting documents from others. Principles of this new protocol are exposed in the next subsection.

**3.2.2. A relevance-based information discovery protocol.** This protocol roughly follows the same scheme than the basic one. The first difference is the use of the relevance concept. The discovery protocol associates with each node of the indexation structure a relevance value that measures its relevance with respect to the remote profile. We then consider a node as relevant if its relevance is greater or equal to the remote user's relevance threshold. The relevance of a node $x$ is obtained by computing the quotient of the two following parameters:

- the number of matched keywords in order to reach $x$ (lower or equal to $depth(x)$).
- the length of the shortest path (from the root to a leaf) passing through $x$.

We assume the relevance value of the root is 0.

The second difference is the way the intersection sets are computed. Indeed, calculated intersections (called $PseudoIntersect$) now contain, for a given level of a profile, the set of relevant nodes rather than the strict intersec-

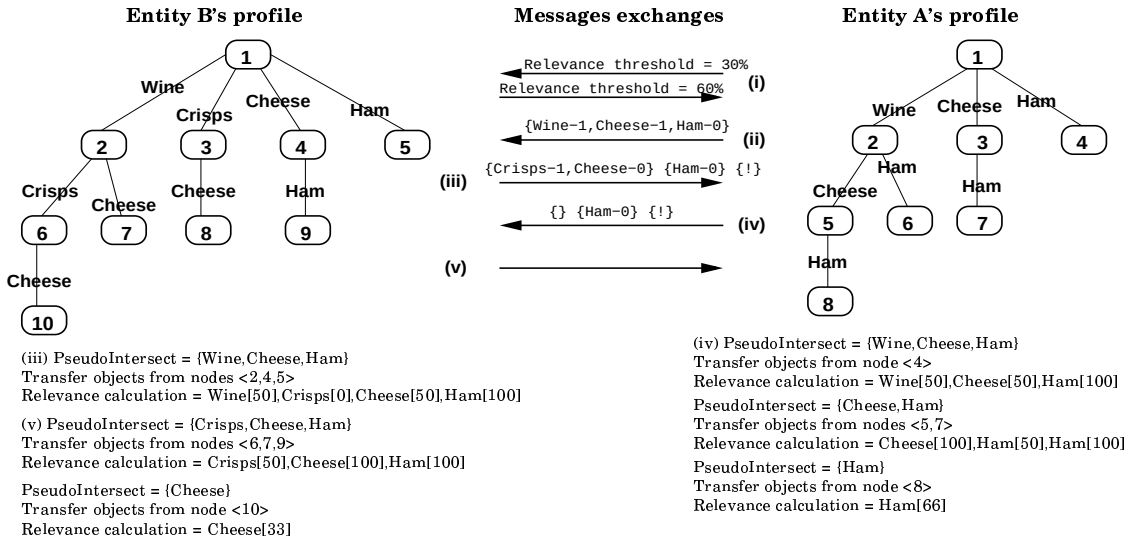



**Figure 3. Relevance-based common interests discovery mechanism**

tion of two remote profiles.
The global scheme of this protocol is the following. Once an initiating entity has been elected, communicating entities exchange the relevance threshold of their user. Then, as in the previous scheme, the initiating entity sends to the other its profile first level of keywords. In order to enable remote PDAs to calculate relevance values, each entity associates with each keyword sent the remaining number of keywords in the shortest branch of this part of its profile. Using this information, entities can compute $PseudoIntersect$ sets by applying the following rules:

1. A node belongs to the remote profile and to the local one. We have to explore the subtree of this node. In order to achieve this goal, the corresponding subtree is transmitted.

2. A node does not belong to the intersection but its relevance is greater than the threshold. Its subtree is *locally* explored as long as the relevance is greater than the threshold (assuming no more keywords are matched).

3. A node does not belong to the intersection and its relevance is smaller than the threshold. The subtree is no longer explored.

4. Finally, if a ramification of the remote profile has totally matched with the local one, all documents indexed deeper have to be transmitted.

Transmitted Web pages are those indexed from nodes whose relevance is greater than the user's threshold.

**Example.** Figure 3 presents an example of information discovery using the relevance-based mechanism (assuming the exchange is initiated by $A$). The first step is the exchange of the users' interest threshold *(i)*. Using this value, each entity can locally distinguish interesting documents from others. As a second step, $A$ sends its first level of keywords to $B$. Each keyword (labelling a node $x$) of this message is associated with the length of the path from $x$ to the closest leaf *(ii)* in order to enable $B$ to compute relevance values.
When $B$ receives this message, it builds the set of $A$'s relevant nodes by calculating the relevance value of each of its first level nodes. Subtrees that will be further explored are those that belong to the strict trees intersection and those obtained from nodes whose relevance is greater than the user's threshold. These nodes are stored in the $PseudoIntersect$ set *(iii)*. Only documents indexed from relevant nodes are transferred. $B$ then replies $A$ with the results of its computation: the set $\{Crisps, Cheese\}$ obtained for the keyword $Wine$, the set $\{Ham\}$ for $Cheese$ and a leaf set for $Ham$ *(iii)*.
Based on $B$'s answer, $A$ can compute, in a single step, $PseudoIntersect$ sets for its profile two first levels *(iv)*. For the first level, three keywords are put in the $PseudoIntersect$ set but the only relevant one is $Ham$ (the others nodes are belonging to the strict intersection). For the second level, the $PseudoIntersect$ set contains only two keywords, which are both relevant: $Cheese$ and $Ham$. Moreover, $A$ can deduce from the received message that $B$ has no more descending node for $Cheese$: as this keyword is relevant, its subtree is also locally explored and the keyword $Ham$ on the third level is also considered as relevant.

Once these calculations are completed, documents from nodes $\langle 4 \rangle$, $\langle 5 \rangle$, $\langle 7 \rangle$ and $\langle 8 \rangle$ are sent to $B$. Finally, $A$ replies $B$ with subtrees obtained from the last computed intersection: an empty set for $Crisps$, the keyword $Ham$ for $Cheese$ and a leaf set for $Ham$.
When $B$ receives this new message, it computes new $PseudoIntersect$ sets for its second and third profile level basing its calculation on the last computed $PseudoIntersect$ set. Three interesting keywords are found on the second level: $Crisps$, $Cheese$ and $Ham$. Although $Crisps$ does not belong to the profiles intersection, its relevance is greater than $A$'s threshold: its subtree is therefore locally further explored. This search enables the discovery of another relevant node on the third level. Thus, documents to transfer are those indexed from nodes $\langle 6 \rangle$, $\langle 7 \rangle$, $\langle 9 \rangle$ and $\langle 10 \rangle$ *(v)*.

### 3.3. Overview of the architecture

We are currently developing a prototype for the architecture described above. We chose, for our implementation, to use PDAs with infrared communication facilities. The prototype is composed of four main modules.

**Storage.** This module is divided in two parts. The first one manages the storage of the public documents present on a PDA, whenever indexed or not (recalling that public documents are not all indexed). The second one stores the spontaneously downloaded documents. As resources of PDAs are limited, it is designed as a cache: only the most recently downloaded documents are kept inside.

**Database Manager.** This module implements the data mining mechanism (presented in section 3.1.2). It provides an API that permits to insert, to remove and to find indexed documents in a data mining based tree and to discover common parts between two tree structures.

**Information Discovery.** This module implements the algorithm described in section 3.2.2. Its objective is the discovery of common parts between remote profiles and the local one. For this purpose, it interacts with the communication module to send and receive discovery messages and relevant documents. It also questions the information indexation module in order to perform the intersection sets.

**Communication.** This last module ensures the interface with the infrared communication card.

## 4. Future works

In this section, we briefly present some aspects of our system which will be the object of further researches.

Firstly, we plan to enable users to specify different constraints about the documents to be spontaneously downloaded. For example, because of a small cache size, users may choose to settle a per document maximum size. Another possible constraint is the type of the downloaded documents. Indeed, various kinds of documents (sounds, videos) are often attached to Web pages and users are probably not interested in downloading those they can not read.

The spontaneous aspect of the exchanges raises the problem of repetitive encounters. For example, two persons will probably meet several times during a same conference. Assuming they do not update their set of public documents, these different encounters will lead to several exchanges of the same data. Such an example is a typical case of resources waste. That is why a mechanism able to distinguish, in small intervals, already downloaded documents from unknown ones is required.

Finally, we notice that our prototype suffers from a platform-based problem. Indeed, the use of infrared communications requires communicating nodes to be each one in the line of sight of the other. With such a restriction, communications can not happen spontaneously. As this aspect of our study is particularly important, we plan to test our work with Bluetooth communication facilities. This could simply be achieved by rewriting a part of the communication module.

## 5. Related works

Many researches are led in the Ad Hoc networks area [5]. This domain considers sets of independent mobile nodes equipped with wireless communication facilities. However, these works differ from ours by their objectives. Indeed, they aim to allow nodes to communicate by providing routing schemes whereas we only consider direct communications between neighbouring nodes. Another difference is that researches in the Ad Hoc networks area only focus on routing protocols whereas we are also dealing with applications.

Ubiquitous computing [4] aims to enable applications to access to contextual information (user's location or activity) in order to adapt the delivered services to the current situation. In the ParcTAB project [9], people can communicate via a cell-based infrared network. Their position is tracked and stored in a centralised localisation server. Using this information, proximate devices (such as printers) can be automatically selected. As in the ParcTAB project, most of the ubiquitous works are based on fixed infrastructures (like fixed network access points), unlike our study which proposes a totally mobile system.

Another ubiquitous study, the CoolTown project [3], proposes to associate Web presence to people and places. They use the Web architecture to spontaneously provide people

with access to Web pages related to the place they are located. Although this work seems quite close of ours, it is distinguishable from it by at least two points. On the one hand, whereas our prototype is completely distributed, the CoolTown architecture relies on servers (associated to places) to provide people with an ubiquitous access to Web pages. On the other hand, we address the issue of the spontaneous relevant information discovery (using users' profiles) what is not treated by the CoolTown study.

Finally, the Proem system [8] aims to enable, during physical encounters, direct exchanges of user's profile. Their profile notion includes a set of personal informations such as phone numbers, e-mail addresses or a list of interests. Although it deals with a distributed architecture and spontaneous interactions, Proem does not manage some automatic user's profiling.

## 6. Conclusion

In this paper, we propose and design a new kind of application extending the Web paradigm to the domain of direct and proximate communications. Assuming that a part of the information stored on PDAs is freely accessible, we aim to enable *spontaneous* exchanges of relevant Web pages between mobile users.

In order to achieve this goal, we design an algorithm automatically providing users' profile (using their public database) which can be used to index the public documents. We also propose two protocols enabling spontaneous interesting information discovery.

Although we believe our approach is promising, we are aware of some current limitations of our work. Indeed, it is actually based on the use of a common ontology between all the users. Such an assumption restricts the diffusion of our system to confined communities.

This work is part of a larger study, the SIS (*Spontaneous Information System*) project [2], within which other aspects of proximate interactions (such as communication issues [10]) are considered.

## References


[1] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th Int'l Conference on Very Large Databases*, 1994.

[2] M. Banâtre and F. Weis. SIS: a new paradigm for mobile computer systems. In *Proceedings of the Information Society Technologies Conference (IST'99)*, 1999.

[3] D. Caswell and P. Debaty. Creating Web Representations for Places. In *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC'2000)*, pages 114–125, 2000.

[4] G. Chen and D. Kotz. A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Darmouth College, 2000.

[5] L. M. Feeney. A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks. Technical Report T99-07, Swedish Institute of Computer Science, 1999.

[6] Grossman and Frieder. *Information Retrieval: Algorithms and Heuristics*. Kluwer Publishers, 1998.

[7] J. Haartsen, M. Naghshined, J. Inouye, O. J. Joeressen, and W. Allen. Bluetooth: Vision, Goals and Architecture. *Mobile Computing and Communications Review*, 2(4), 1998.

[8] G. Kortuem, Z. Segall, and T. Thompson. Close Encounters: Supporting Mobile Collaboration through Interchange of User Profiles. In *Proceedings of the International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, pages 171–185, 1999.

[9] B. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The ParcTAB Mobile Computing System. Technical Report TR-CSL-93-20, Xerox Palo Alto Research Center, 1993.

[10] A. Troël, M. Banâtre, P. Couderc, and F. Weis. Predictive Scheme for Proximate Interactions. In *Proceedings of the International Workshop on Smart Appliances and Wearable Computing (IWSAWC'2001)*, pages 235–239, 2001.