FieldWise: A Mobile Knowledge Management Architecture

Henrik Fagrell[a,d], Kerstin Forsberg[a,b,c], Johan Sanneblad[a,d]
The Viktoria Institute[a], Adera[b], ICTech[c], Newmad Technologies[d]
Viktoria Institute, Viktoriagatan 13, S-405 30 Göteborg, SWEDEN
+46 (0) 31 774 11 00
[Henrik.Fagrell, Johan.Sanneblad]@newmad.com and kerstin@instantcontext.com

ABSTRACT
The paper presents results of a research project that has aimed at developing a knowledge management architecture for mobile work domains. The architecture developed, called FieldWise, was based on fieldwork in two organisations and feedback from users of prototype systems. This paper describes the empirically grounded requirements of FieldWise, how these have been realised in the architecture, and how the architecture has been implemented in the news journalism domain. FieldWise adds to the field of CSCW by offering an empirically grounded architecture with a set of novel features that have not been previously reported in the literature.

Keywords
Mobile CSCW, Knowledge management, Organisational memory, Hand-held devices

INTRODUCTION
Recently, the issues of "knowledge management" [19, 1] and "mobility" [31, 6, 7] have received much attention in the CSCW literature. The interest in these issues is often motivated by the fact that work in many organisations is "knowledge intensive" and "mobile." These issues have been explored separately. However, so far few researchers have explored the topic of knowledge management in mobile work domains.

This paper reports the final step of a research project with the objective to design novel and commercially interesting IT support for knowledge management in mobile work. The design has been informed by:

- an empirical study of mobile and distributed service electricians, i.e., observation of work and workshops [19].
- a field study of mobile news journalists at a radio station, i.e., observation of work [18].
- experiences with a prototype system for mobile news journalists, i.e., observation of work under real working conditions with the prototype and design workshops [17].

Based on the results a novel mobile knowledge management architecture, called FieldWise, was constructed.

The kind of typical work organisation that can benefit from a FieldWise implementation is one where:

- people's tasks are time critical and driven by deadlines
- the result relies on the creativity of autonomous, but interrelated people
- there is a culture of co-operation and sharing of knowledge amongst people
- people are mobile and distributed

Typical work of this sort is, for example, news journalism, sales and real estate brokering. The first implementation of FieldWise is adapted for news journalism. Each new work domain and installation of FieldWise requires a definition of the informational context [cf., 21] of the organisation, e.g., integration with existing personnel- and record-keeping systems.

FieldWise provides mobile access for five client platforms, i.e., Pocket PC, Windows CE 2.11 (Palm-sized and Handheld), EPOC, and PalmOS. But just offering access to stationary systems cannot solve the knowledge management problems, in mobile work. Several novel requirements must be met.

REQUIREMENTS
This section describes the requirements derived from the empirical work and experiences with prototypes in mobile settings.

Evolving and interdependent tasks
When people produce material with the objective to accomplish a coherent result there is a need for co-ordination. On the other hand, the autonomous nature of mobile work makes central control unsuitable. The local and unique circumstances that people confront give them knowledge that is situated and local. Also, the actions of people give rise to new local knowledge and situated actions, thus the tasks evolve [cf., 38, 18]. In situations when the tasks of co-workers are overlapping, it is useful to

CSCW'00, December 2-6, 2000, Philadelphia, PA.

interrelate the knowledge that each person possesses [38]. In a mobile and distributed organisation it is never possible for anyone to have complete overview at any point. On the other hand, planning of potentially interesting tasks can be done by a common resource, e.g., an editor at a newspaper.

Empirical research implies that people take notes to reduce the complexity of their tasks [cf., 26]. Our empirical results suggest that task-related notes are useful in mobile situations to facilitate remembering [19, 17]. In this respect, the notes are a representation of a "prospective memory" [cf., 9].

This suggests that a mobile knowledge management architecture should support the user's tasks, as they evolve. It should also notify the users of interdependencies between tasks, and provide access to tasks that are potentially interesting.

Overview of records

Having accurate records is not always desirable in an organisational sense. It is often more expensive to store and retrieve than to re-discover [8]. Finding out what to do in a situation is a matter of recognising similarities with previous situations, i.e., to remember matching cases and creatively find a category of what needs to be done [34]. Our empirical observations suggest that in mobile situations an overview of what has been done previously can facilitate remembering and ease the rediscovering of relevant details [18]. Also, feedback annotations that are made in the context of a persistent record can point out ways of taking a task further [cf., 19, 18]. For example, it is common that the editorial staff evaluate news reports and discusses improvements at daily meetings. Such annotations can contain suggestions on how to follow-up on a task or how to improve quality in a more general sense.

This suggests that a mobile knowledge management architecture should give an overview of records rather than an exhaustive collection. Annotations to records should also be displayed.

Location of available expertise

Mobile users often confront situations that they are not totally familiar with. The most efficient way to get help is often to talk to co-workers who have relevant expertise. Knowing who knows has been highlighted as important [34, 32]. It is often a question of getting a "second opinion" on how to solve the task [cf., 13].

It is, however, especially problematic for mobile workers to know whom to contact, since they are often away from their co-workers. Accordingly, knowing whom to contact is not enough, they need to be available for interaction as well [18].

The expertise does not have to fit exactly, i.e., an expert does not need to be a specialist in the specific topic, but rather knowledgeable on the general genre of the task.

This suggests that a mobile knowledge management architecture should suggest experts and present their accessibility.

Filtering based on task and long-term interests

Mobile workers are often interested in the latest information that is related to the current task [18, 17]. For example, a person who is travelling to visit a customer may want to be notified about the customer's latest press releases. After the visit, however, the person is less interested in getting this kind of information. We call this *task-based* interests.

Task-based interests are different from both short-term and long-term interests. A short-term interest is, for example, to retrieve a phone number based on a query, and long-term interests is to filter email from a specific sender based on a profile [5]. Task-based interests are similar to long-term interests, but are only active for the duration of a task.

This suggests that a mobile knowledge management architecture should have filters based on task-based interests. A task-based interest should easily convert into a long-term interest.

Dynamic configuration of mobile services

The capabilities of mobile devices change rapidly. What once were simple calendar replacements are now expected to have both Internet and Wireless LAN features. Due to limitations in hardware design most of these devices are not upgradeable, which means that an organisation must invest in new devices to get all the latest services. This has led to a wide range of handheld devices from different manufacturers with different revisions of the same operating system. Therefore, implementing a mobile system in an organisation requires the support for several mobile devices and their specific capabilities.

Handheld devices are also expensive, which in some organisations has led to device sharing among the employees. A shared device is not usable as standard personal calendar or phone register since the applications can only store data from one user at a time. Custom built applications may work, however, if they are designed with this in mind.

Organisations may want to provide unique content and services specifically tailored for each co-worker based on their personal preferences and work environment, for instance restricting access for freelancers.

This suggests that a mobile knowledge management architecture should allow dynamic configuration of the mobile knowledge support according to current user preferences and mobile device capabilities.

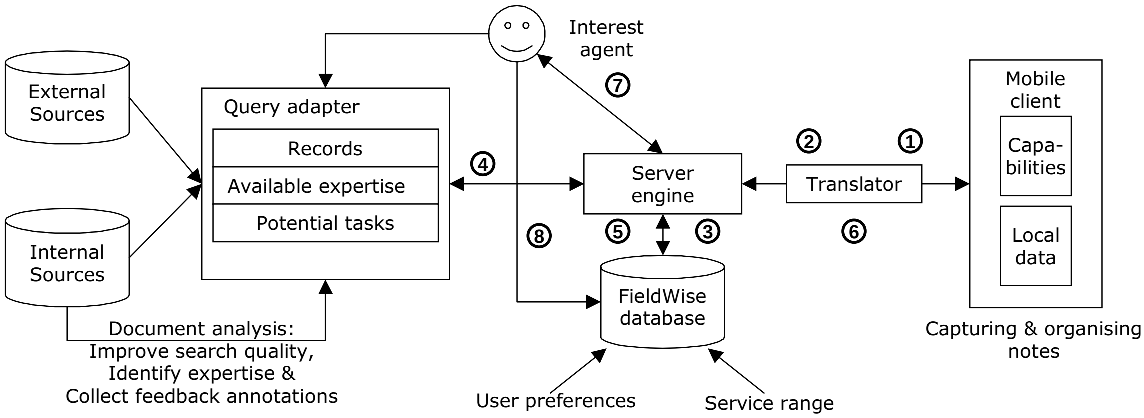


Figure 1: The FieldWise architecture.

Summing up the requirements

Based on our empirical work and experiences with prototypes, we propose the following main requirements for a mobile knowledge management architecture. It should:

1. Support evolving tasks and notify users of interdependencies.
2. Offer an overview of records, including annotations.
3. Suggest available expertise.
4. Filter information based on task and long-term interest.
5. Enable adaptation to user preferences and mobile device capabilities.

There are of course more general requirements for a CSCW architecture, but our focus is on the specifics of mobile knowledge management.

THE FIELDWISE ARCHITECTURE

Based on the requirements above, this section describes the general architecture of FieldWise. The following listing describes a use case that illustrates how large parts of the architecture works (figure 1).

1. The mobile client connects and the user's notes are sent to the server engine.
2. The server engine analyses the capabilities of the client and selects the appropriate translator to adapt its response.
3. The server engine creates an interest profile based on the notes and stores it in the FieldWise database.
4. The interest profile is sent to the query adapter, which performs a search in internal and external sources. The server engine receives the results.
5. The server engine looks in the FieldWise database for overlaps with the interest profiles of other co-workers. If there is an overlap, the server engine notifies the co-workers. The response to the user is also adapted according to the user's preferences stored in the FieldWise database.
6. The results are translated and returned to the mobile client, which presents and stores the results locally.
7. The server engine configures the interest agent to continuously monitor for updates that match the user's interest profile.
8. If the interest agent identifies an update, the FieldWise database is modified and the user is notified.

There are, of course, other use cases that are important, e.g., accessing the common resource of potential tasks and feedback annotations, but they are described in of the next section.

How the architecture meets the requirements

Evolving and interdependent tasks

The mobile client captures and organises the users' task related notes. Based on the notes, as well as the forthcoming query results, a profile is created, which makes it possible to compare users' tasks to identify interdependencies. As the task evolves, due to situated and local circumstances, the user updates the notes and the profile is updated. If two tasks are significantly similar the concerned users are notified, letting them decide if it is necessary to take action.

A commonly shared list of potentially interesting tasks is accessible to all mobile clients. The user can copy a task from the list and develop it further, i.e., adapt it to local circumstances. This also reduces the extra work that is required for note taking.

Overview of records

The query adapter uses the interest profile to search sources that are within the informational context for the organisation, e.g., internal and external archives. The search result is then processed by the server engine. The purpose is to provide the mobile clients with extracts of records. The extracts are displayed as summarised lists to provide the user with an overview of what has previously been done on the topic. The complete record can also be accessed, but the idea is that the overview, in itself, should remind the user of the context for the task.

Furthermore, if feedback annotations are available they are presented as a part of the result.

Location of available expertise

Based on the profile, people with related expertise, are displayed on the mobile client. The availability of the people is also presented.

Expertise is automatically identified through analysis of records, e.g., authorship and mentioning of internal documents, using the query adapter. This approach is more suitable than to ask people to explicitly state their area of expertise and knowledge level [cf., 25].

To find out if someone is available is an adaptation for each installation, but typically organisations have back-end security systems for entering the office. Another alternative is to use work schedules or shared calendars. The server engine keeps track of this information by using the query adapter.

Filtering based on task and long-term interests

As tasks evolve throughout the day, new information becomes available. Also, new or evolved tasks may overlap. The interest agent filters out this information for each task, based on the interest profile stored in the FieldWise database.

As long as a task is active the interest agent notifies its user of relevant updates. When a task is deleted or de-activated the interest agent stops the monitoring. Another alternative is that a task-based interest grows into a long-term interest that does not necessarily represent a current work task, but rather an area of expertise. In this case the interest agents remain active.

Dynamic configuration of mobile services

Each mobile client has unique capabilities. The server engine investigates the capabilities of the mobile client, as well as the current network capacity, and translates its response. For example, less information is sent if the mobile client has a slow network connection.

The organisation can adapt the range of services for specific users by updating the user's preferences, which are stored in the FieldWise database. This also affects the responses of the server engine. The services are distributed as components, making it easy for the organisation to add new services.

Maintaining data (user preferences and tasks) on a mobile device is always a problem. The mobile devices may crash, shared between people, or data may be corrupted over time. For this reason, all client data is replicated in the FieldWise database. A user can roam between different FieldWise installations and thus, the user authentication must be handled by another system. The user can rewind to the latest state of the mobile client. The mobile client can also be the host for transporting the user's preferences to another FieldWise installation.

A FIELDWISE IMPLEMENTATION

In this section we describe an implementation of FieldWise adapted for the news journalism domain. In this case, FieldWise is integrated with an editorial system, called IMpress, which is running at a Swedish newspaper. The overall implementation is illustrated in figure 2. We use the latest mobile devices and (wireless) network technologies that are available. Let us go into the technical details and then see how the implementation meets the requirements.

Technical details

Network access

The communication is done with TCP/IP sockets through a server on an internal structure. The clients use GSM (Global System for Mobile communication) and Point-to-Point Protocol (PPP) to communicate in wireless mode. GSM is the largest digital wireless communications standard in the world with its 284 million subscribers [15]. The clients are connected to a GSM phone with cable or infrared (IrDA 1.0). Today, the bandwidth for GSM is 9600 bits per second and the time to get online is about 20 seconds. Users that are not online are notified using the Short Message Service (SMS). SMS is a part of GSM and makes it possible to send and receive text messages with the mobile phone.

The clients with PCMCIA slot can use wireless LAN (IEEE 802.11b). The client is then connected instantly with a bandwidth of about 11 megabits per second.

The clients can be connected in a stationary mode with a network card or via a serial cable to a networked PC.

Mark-up languages

The internal structure is encoded in XML (eXstensible Mark-up Language) and we use XSLT (eXxtensible Stylesheet Language Transformations) for transformations [11]. In the implementation the query adapter uses XMLNews. In the news industry there are two emerging XML-compliant mark-up languages: XMLNews [33] and News Information Text Format (NITF). The reason we choose XMLNews, is that it is more extensible as the data is separated from the meta-data, allowing multiple meta-data schemas. To describe user preferences and device capabilities we use an XML application called Composite Capability/Preference Profiles (CC/PP) [35].

Operating systems and development platforms

The server is implemented in Java on the Microsoft Windows 2000 platform using an Oracle database manager, which is also the database manager for the editorial system IMpress. To find out the availability of people we use the shared electronic calendar in Microsoft Exchange.

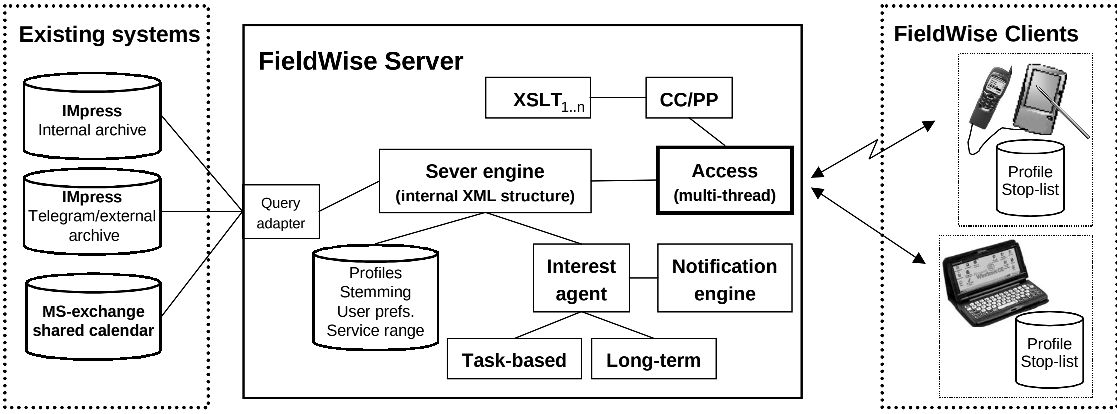


Figure 2: The detail architecture of a FieldWise implementation in the news journalism domain.

Algorithm of the interest profiles

The interest profile is represented with keywords and a blueprint of its search results.

The keywords are filtered out of the task notes through a stop-list, which takes away common words, e.g., he, she and it. For each work domain the stop-list is adapted to filter out high frequency words with low content value. In the news domain the word "interview" is filtered out because it is to common to be a part of interest profile. The stop-list is about 1,500 words (10 kilobytes) and stored on the client. The stop-list is also refined for the informational context for the organisation.

A stemming dictionary is also used to improve the interest profile. It is used to put the keywords on their basic form and manage mis-spellings. For example, the word "Ericsson" will match on different spelling, e.g., "Ericson," "Erikson," "Eriksson," and plural version. The stemming dictionary is generated from two years of text from the internal archive. The dictionary is about 300,000 words (4 megabytes) and is placed on the server because of its size.

Each keyword is weighted with TF-IDF [36]. Also, a blueprint of the search result is added to the interest profile, giving the algorithm a collaborative filtering aspect, i.e., overlaps between interest profiles can be detected, even if they do not contain the same keywords.

The interest profile is used in conjunction with the database manager's free text functionality. This enables a suitable quality of the search results.

How the requirements were implemented

Our focus in this section is on the user interface and the specifics of the adaptation for the news journalism domain.

Evolving and interdependent tasks

The application lets the user add, edit and remove tasks and organise the tasks in folders. All of the folders are personal to the user except the "list of the day," which contains potentially interesting tasks that are shared (figure 3a).

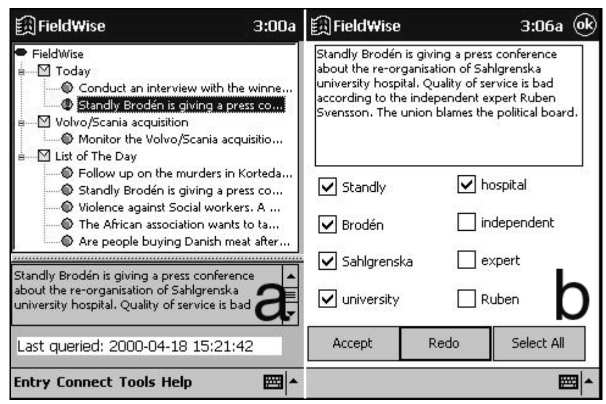


Figure 3: The tree view of tasks and folder (a) and (b) generate keywords. The illustration is on Pocket PC.

The "list of the day" is timely tasks stored in the editorial system, which are suggested by the news editor or by other colleagues. A task can be copied from the "list of the day" folder into another folder of the tree view. The sharing also reduces the amount of input required to construct a new task.

Keywords are automatically generated from the text of the task (figure 3b). The user can choose the keywords that represent what the task is about. Pressing the Redo button generates new keywords. The user does not have to write complete sentences; adding keywords at the end of the text is enough.

If the user is satisfied the Accept button is pressed. After this, the user chooses if the task should be activated. The reason for this is that the users may be interrupted while editing and wants to continue the note taking later on. When the user is online the tasks can be registered by choosing Send task in the Connect menu.

If the task of another co-worker overlaps, the co-workers name and task text is displayed on the Match tab (figure 4d). The co-worker is also be notified (via e.g., SMS). As the task evolves during the day the user updates the notes, re-sends it and receives new information.

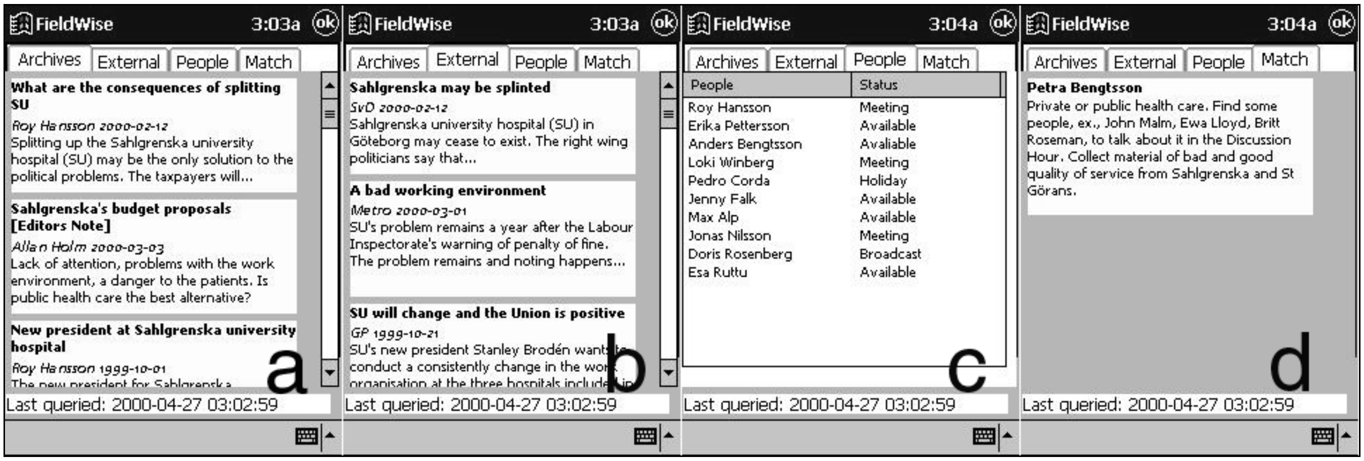


Figure 4: The Archive (a), External (b), People (c), and Match (d) tabs. The illustration is on Pocket PC.

Overview of records

All the information on the tabs in figure 4 depends on the active and sent task. The tabs are displayed when a task is registered or Result is chosen in the Entry menu.

The Archive tab displays a list of stories from the internal archives (figure 4a) where each item contains: title, author, publication date and ingress. The External tab displays a list of stories from predefined external sources, e.g., newspaper articles and telegrams (figure 4b). The items on the external list contain title, source, publication date and ingress. Approximately ten items are displayed on each tab.

The intended use of the listed stories is to give the user an overview, i.e., a reminder of what has been done previously. The user can click on a title and get the full text of the story. This is, however, accomplished through a separate system, e.g., a web browser, when the mobile client is online.

Editors evaluate, comment and annotate published stories. This coaching is captured by the editorial system IMpress, and we display this with the article entry on the Archive tab (figure 4a [Editors Note]). By clicking on the Editors Note the user gets the comment (like a Post-it note). Usually these are suggestions of how to follow-up on a story or, on a more general level, how the genre of reporting can be improved.

Location of available expertise

People tab displays a list of co-workers and their availability who have previously been working on a topic related to the current task (figure 4c).

The expertise is identified through the authorship of published articles from the internal archive. We also include documents of the category "issues to watch," that are common in the news domain.

The availability of people is found in the shared electronic calendars system (Microsoft Exchange).

Filtering based on task and long-term interests

As long as a task is active the user is notified about the latest information, e.g., press releases and overlapping tasks of other users.

Today, the notifications are done via SMS (figure 5b), but we also support email. The implementation sends the SMS through a gateway that converts emails into SMS.

A task can be converted into a long-term interest by selecting it and choosing Item options in the Tools menu (figure 5a).

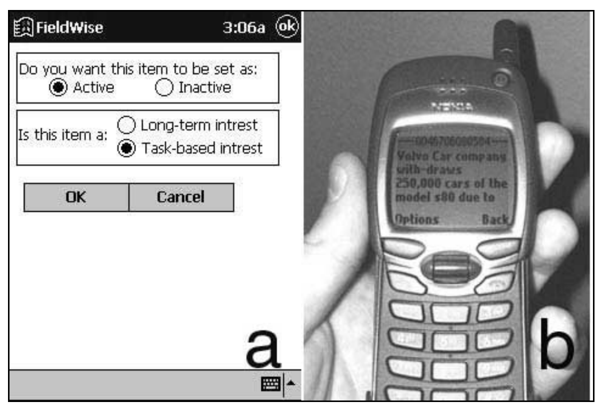
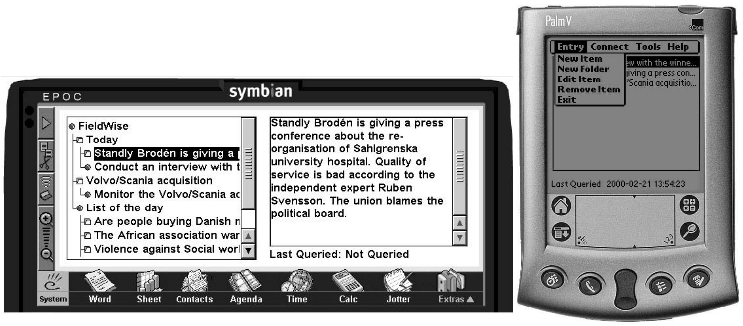


Figure 5: The item options (a) and a GSM phone showing an SMS message (b).

In the case of an overlap between two task-based profiles, both parties are notified.

The functionality of the filter for task-based and long-term interests differs a bit in this implementation. A long-term interest only notifies new information. It does not notify overlaps with a task-based interest. A user with a long-term interest should not take the initiative to approach a colleague that has not asked for help. Instead, if the colleague wants help, the list of expertise can be consulted.

Eventually the colleague's task results in an article that becomes available.




```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.w3.org/TR/WD-profile-vocabulary#”
xmlns:fw="http://www.instantcontext.com/FieldWise-vocabulary#”>
 <rdf:Description about="SoftwarePlatform">
  <prf:Defaults
   OS=”WinCE2.11/HPC"/>
 </rdf:Description>
 <fw:Description about=”Network">
 <fw:Defaults
     Connection="GSM9600" />
 <fw:Description about=”NewsServices">
 <fw:Defaults
     <rdf:li
     resource="http://www.newspaper.com/profiles/FieldWise"/>
   <fw:Modifications
   Match=”No" />
</rdf:RDF>
```


Figure 6: The implementation on EPOC ER5 and Palm OS. The textbox is an extract of a CC/PP identification of a client.

Dynamic configuration of mobile services

The mobile client platforms supported are Pocket PC, Windows CE 2.11 (Palm-sized and Handheld), EPOC ER5, and PalmOS 3.5 (see figure 6).

The mobile devices have different capabilities, e.g., software platform versions, size of display and storage capacity. Also, each device can be connected to the network in a number of ways, e.g., GSM, Wireless LAN and serial cable. Accordingly, the server engine adapts its responses to the capabilities of the device and its network connection. This is done through the selection of a device specific translator (encoded in XSLT), that adapt the responses.

An organisation can define its range of mobile services. For a specific role, or for a unique user, a set of services can be provided. For example, a freelance journalist can be offered a different set of services then the editor. We choose to see each tab on the mobile client as a service, which is dynamically configured.

In order for the server engine to adapt its responses and dynamically configure its services, the clients must be identified. Figure 6 shows an extract of the information needed by the server engine to adapt its responses and dynamically configure its services for the mobile client. The identification is encoded using CC/PP, extended with a FieldWise-specific namespace. It also exemplifies how a default profile, for the range of services, is modified.

The appropriate translator and service configuration is selected depending on:

- **Platform.** Different platforms have different capabilities. For example, the display capacity of the device affects the length of the title and ingress for each article.
- **Connection speed.** If there is a slow connection, less information is transferred to the client. Figure 6 illustrate an identification of a connection with 9,600 bit per second over GSM. For a client with instant access, the notification method is affected as well, i.e., there is no need for SMS.
- **Software version**. Different versions of the client need information in different formats.
- **Services**. The server needs to know the services of the client. Each tab (figure 3) is considered to be a service. Figure 6 illustrate an identification where the Match tab is cancelled.

DISCUSSION

The FieldWise architecture as it is described here is especially useful for work organisation where:

- people’s tasks are time critical and driven by deadlines
- the result relies on the creativity of autonomous, but interrelated people
- there is a culture of co-operation and sharing of knowledge amongst people
- people are mobile and distributed

The domain of work may be, for example, news journalism, sales and real estate brokering. However, every installation requires integration and adaptation to the local work practice of the organisation. For example, the implementation for the news journalism described here has a special filtering policies for task and long-term interests. The informational context for organisation must be considered, e.g., what are the relevant external sources and how is the information structured?

Since we have not been able to identify any similar architectures for mobile knowledge management the related research is discussed on a feature level.

The features include (1) support for evolving tasks with notifications of interdependencies; (2) overview of records and annotations; (3) suggestions of available expertise; (4) filtering based on task and long-term interest; (5) adaptation to user preferences and mobile devices.

Evolving and interdependent tasks

A foundation for the architecture is that the users evolving task is captured. The representation of the task makes it possible to continuously provide notifications of interdependencies.

Shared task lists has been used by, e.g., Kreifeldt et al [27], but they display the lists in a common information space, whereas our approach is notify if there is an overlap. Similar to FieldWise, the Yenta system notifies its user of others' with similar interests [20]. The difference is however that FieldWise match the users based on their current tasks. Also, Cadiz et al [10] show the importance of matching the awareness tool with the task support in a recent study.

Overview of records

FieldWise use text analysis algorithms to improve search quality. The principle for the visualisation mechanism is to provide overview of records rather than an exhaustive collection.

The quality of the result is based on how well the index works. It needs to be tuned in for the local circumstances. In some cases there is a need to incorporate a separate index to get good quality of the search result. The index does not need to be very advanced to solve the problem (see, e.g., the NewsMate prototype [17]). Another approach is to use a commercially available product for information filtering, e.g., Autonomy (www.autonomy.com).

In knowledge management systems, e.g., Answer Garden 1-2 [2, 3], Bubble-up, [14], and Project memory [39] there are search features. The ways in which users can survey search results are very limited. Clearly, these systems are much more oriented towards finding "the matching record" rather than providing the user with an associative overview. There are some exceptions, however, as represented by gIBIS [12], Designer assistant [37] and RepTool [23]. These systems offer features to get overview of complex design problems using graphics and hypertext. However, the stationary nature of these systems makes them less useful in the mobile work supported by FieldWise.

Location of available expertise

FieldWise determines if someone is knowledgeable on a topic through the analysis of internal documents. If a person is the author of a document we assume that they have expertise on the topic. The availability of the expertise is an adaptation for each specific FieldWise installation. Emerging technologies that can identify the position of a mobile device can be used. For example, the Ericsson Mobile Positioning System can locate a GSM phone with the accuracy of about 300 meters.

There are PC based applications like ICQ (www.icq.com) that help people to find out if colleagues are available (or busy, etc.). The system is not very sophisticated when it comes to expert location. The Answer Garden 2 [3] and TeamBuilder [24] supports the location of predefined experts. Referral Web helps people to find experts based their relationship in a social network, assuming that topical exercise among co-authors, i.e., an expert can be identified by their participation in co-author relationships or Web page listing [25]. The Expert Finder [29] is tailored for Java programmers and supports the location of expertise by automatically analysing the programmers source code. None of the systems considers if the expertise is available for interaction.

Filtering based on task and long-term interests

FieldWise use agents to monitor the users task and long-term interests. Notifications are issued about the latest information and task overlaps as long as the interest is active.

Support for long-term interests can be found in several systems, e.g., Bubble-up [14], Fab [4], IntraNews [16], Knowledge Pump [22] and Soap [40] and Yenta [20], but neither of them supports task-based interests in combination with notifications.

Dynamic configuration of mobile services

The FieldWise architecture adapt to user preferences and several mobile devices. The users preferences can also be transported between different installations of FieldWise. The architecture is developed to easily deploy future mobile devices and wireless communication technologies.

The only system that we have found that supports knowledge management on mobile devices is Darwin [28]. Darwin supports the distribution and exchange of lessons learned within a dispersed IT-support group.

CONCLUDING REMARKS

In this paper we describe an architecture for mobile knowledge management that can easily be adapted for new mobile work domains, mobile devices and wireless communication technologies. As opposed to similar design efforts the requirements are derived from empirical studies of mobile work.

The research adds to the field of CSCW by offering a set of novel features that have not been reported in the literature previously.

The architecture has been applied in a commercial implementation, which may assure its relevance and usefulness (www.instantcontext.com).

The next implementation of FieldWise focuses on the work domain of mobile sales personnel.

ACKNOWLEDGEMENT

This research is a part of the Mobile Informatics programme that is funded by the Swedish Information Technology research Institute (SITI).

The authors like to thank the colleagues in the Mobile Informatics group at the Viktoria Institute, especially Fredrik Ljungberg and Jens Bergqvist. Also, thanks to Adera Business Solutions and ICTech.

A special thanks to Kerry Northrup at IFRA Center for Advanced News Operations.

REFERENCES

1. Ackerman M.S. and C. Halverson (1998) "Considering an organization's memory," In *Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work,* pp. 39-48. Seattle, WA: ACM Press.
2. Ackerman, M.S. (1994) "Augmenting the organizational memory: a field study of answer garden," In

Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, pp. 243-252. Chapel Hill, NC: ACM Press.

3. Ackerman, M.S. and D.W. McDonald (1996) "Answer Garden 2: Merging organizational memory with collaborative help," In *Proceedings of ACM 1996 Conference on Computer Supported Cooperative Work,* pp. 97-105, Cambridge, MA. ACM Press.
4. Balabanovic, M and Y. Shoham (1997) "Fab: content-based, collaborative recommendation," *Communications of the ACM*, 40 (3), pp. 66-72.
5. Belkin, N.J. and W.B. Croft (1992) "Information Filtering an Information Retrieval: Two sides of the same coin?" *Communication of the ACM*, 37 (7), pp. 29-38.
6. Bellotti, V. and S. Bly (1996) "Walking away from the desktop computer distributed collaboration and mobility in a product design team," In *Proceedings of ACM 1996 Conference on Computer Supported Cooperative Work*, pp. 209-218, Cambridge, MA: ACM Press.
7. Bellotti, V. and Y. Rogers (1997) "From Web Press to Web Pressure: Multimedia Representations and Multimedia Publishing," In *Proceedings of ACM 1997 Conference on Human Factors in Computing Systems*, pp. 279-286. Atlanta, GA: ACM Press.
8. Bowker, G.C. (1997) "Lest we remember; organizational forgetting and the production of knowledge," *Accounting, Management and Information Technologies,* 7 (3), pp. 113-138.
9. Brown, B.A., A. Sellen and K. O'Hara (2000) "A Dairy Study of Information Capture in Working Life," In *Proceedings of the ACM 2000 Conference on Human Factors in Computing Systems,* pp. 438-445, The Hauge, The Netherlands: ACM Press.
10. Cadiz, J., A. Espinosa, L. Rico-Gutierrez, R. Kraut, W. Scherlis and G. Lautenbacher (2000) "Coming to the Wrong Decision Quickly: Why Awareness Tools Must be Matched with Appropriate Tasks," In *Proceedings of the ACM 2000 Conference on Human Factors in Computing Systems,* pp. 392-399, The Hauge, The Netherlands: ACM Press.
11. Clark, J. (1999) XSL Transformations (XSLT) Version 1.0. W3C Recommendation 16 November 1999. http://www.w3.org/TR/xslt
12. Conklin, J. and M.L. Begeman (1988) "gIBIS: a hypertext tool for exploratory policy discussion," *ACM Transactions on Office Information Systems,* 6 (4), pp. 303-331.
13. Ehrlich, K. and D. Cash (1999) "The invisible world of intermediaries: a cautionary tale," *Computer Supported Cooperative Work: The Journal of Collaborative Computing,* 8 (1-2), pp. 147-167.
14. El Sawy, O.A. and G.E. Bowles (1997) "Redesigning the customer support process for the electronic economy: Insights from storage dimensions," *MIS Quarterly*, 21 (4), pp. 457-484.
15. EMC World Cellular Database. http://www.gsmworld.com/
16. Fagrell, H. (1999) "IntraNews: A News Recommending Service for Corporate Intranets," In *Proceedings of Computer Supported Cooperative Work in Design*, pp. 323-328, Compiègne, France.
17. Fagrell, H. (2000) "NewsMate: Providing Timely Knowledge to Mobile and Distributed News Journalists," Revised version accepted for publication in *Beyond Knowledge Management: Managing Expertise, M. Ackerman et al. (eds.)*.
18. Fagrell, H. and F. Ljungberg (2000) "A Field Study of News Journalism: Implications for Knowledge Management Systems," Accepted *for publication in Proceedings of the Sixth Biennial Participatory Design Conference*.
19. Fagrell, H., S. Kristoffersen and F. Ljungberg (1999) "Exploring Support for Knowledge Management in Mobile Work," In *Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work,* pp. 259-275, Copenhagen, Denmark: Kluwer Academic Publishers.
20. Foner, L. (1997) "Yenta: A Multi-Agent, Referral Based Matchmaking System," In *Proceedings of the first international conference on Autonomous agents,* pp 301-307, Marina del Rey, CA: ACM Press.
21. Forsberg, K. and L. Dannstedt (2000) "Extensible use of RDF in a Business context," In *Proceedings of the Ninth International World Wide Web Conference,* Amsterdam, The Netherlands: Elsevier Science.
22. Glance, N., A. Damián and M. Dardenne (1998) "Knowledge Pump: Supporting the Flow and Use of Knowledge," In *Information Technology for Knowledge management,* U.M. Borghoff and R. Pareschi (eds.). Berlin: Springer-Verlag.
23. Jordan, B., R. Goldman and A. Eichler (1998) "A Technology for Supporting Knowledge Work: The RepTool," In *Information Technology for Knowledge management*, U.M. Borghoff and R. Pareschi (eds.), pp. 79-96. Berlin: Springer-Verlag.
24. Karduck, A. (1994) "TeamBuilder: a CSCW tool for identifying experts and team formation," *Computer Communications*, 17 (11), pp. 777-887.
25. Kautz, H., B. Selman and M. Shah (1997) "Referral Web: combining social networks and collaborative

filtering," *Communications of the ACM*, 40 (3), pp. 63-65.

26. Kidd, A. (1994) "The Marks are on the Knowledge Worker," In *Proceedings of the ACM 1993 Conference on Human Factors in Computing Systems,* pp. 186-191. Boston, MA: ACM Press.
27. Kreifeldt, T., E. Hinrichs and G. Woetzel (1993) "Sharing To-Do Lists with a Distributed Task Manager," In *Proceedings of the third European Conference on Computer-Supported Cooperative Work,* pp. 31-45. Milan, Italy: Kluwer Academic Publishers.
28. Kristoffersen, S. and F. Ljungberg (1998) "MobiCom: Networking dispersed groups," *Interacting with Computers,* 10 (1), pp. 45-65.
29. Lieberman, H. and A. Vivacqua (2000) "Agents to Assist in Finding Help," In *Proceedings of the ACM 2000 Conference on Human Factors in Computing Systems,* pp. 65-72, The Hauge, The Netherlands: ACM Press.
30. Ljungberg, F., B. Dahlbom, H. Fagrell, M. Bergquist and P. Ljungstrand (1998) "Innovation of new IT use: Combining approaches and perspectives in R&D projects," In *Proceedings of the Fifth Biennial Participatory Design Conference*, pp. 203-209, Seattle, GA: CPSR and ACM.
31. Luff, P. and C. Heath (1998) "Mobility in Collaboration," In *Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work*, pp. 305-314, Seattle, WA: ACM Press.
32. McDonald, D.W. and M.S. Ackerman (1998) "Just talk to me: a field study of expertise location," In *Proceedings of ACM 1998 Conference on Computer Supported Cooperative Work,* pp. 39-48. Seattle, WA: ACM Press.
33. Megginson, D. (1999) XMLNews-Story (1999-04-05) Technical Specification. http://www.xmlnews.org/
34. Randall, D., J. O'Brien, M. Rouncefield and J.A. Hughes (1996) "Organisational Memory and CSCW: Supporting the 'Mavis Phenomenon'," In *Proceedings of the sixth Australian Conference on Computer-Human Interaction*. Hamilton, New Zealand: IEEE Computer Society.
35. Reynolds, F., J. Hjelm, S. Dawkins and S. Singhal (1999) Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation. W3C Note 27 July 1999. http://www.w3.org/TR/NOTE-CCPP/
36. Salton, G. (1971) *The SMART Retrieval System*, New Jersey: Englewood Cliffs, Prentice-Hall, Inc.
37. Terveen, L.G., P.G. Selfridge and M.D. Long (1995) "Living Design Memory: Framework, Implementation, Lessons Learned," *Human-Computer Interaction*, (10) 1, pp. 1-37.
38. Tsoukas, H. (1996) "The firm as a distributed knowledge system: a contsructionist approach," *Strategic Management Journal*, 17, Winter Special Issue, pp. 11-25.
39. Weiser, M. and J., Morrison (1998) "Project Memory: Information Management for project teams," *Journal of Management Information Systems*, 14 (4), pp. 149-167.
40. Voss, A. and T. Kreifelts (1997) "SOAP: Social Agents Providing People With Useful Information In *Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration challenge*, pp. 291-298, Phoenix, AZ: ACM Press.