ESPRIT Project N. 25 338

Work package I

Pilot Application 1

Evaluation Report (DI5)

ID:	WP_I_EvaluationReport	Date:	20.02.1999
Author(s):	Hans-Guenter Stein, FAST e.V.	Status:	draft
	Ariel Sindermann, FAST e.V.		
	Lioba Gebauer, Bavaria Online		
Reviewer(s):		Distribution:	internal reviewers

Change History

Document Code	Change Description	Author	Date
WP_I_EvaluationReport	Version 1.0. No changes.	Stein, Gebauer, Sindermann	20.02.99
WP_I_EvaluationReport	Version 1.1. Changes after internal review	Stein	30.03.99

Overview

This document reports on the evaluation processes and results of the pilot applications developed and implemented in FollowMe WP I. We summarize architectural and technological concepts used to implement the pilots, provide a description of the pilot application domains and a user model of the applications. Furthermore we report on application deployment in the test field and describe the methodology used for the evaluation process. We deliver the results of the evaluation in form of statistics derived from usage monitoring and user feedback. Additionally we report on lab tests with components of the FollowMe framework that have not yet been integrated into the pilot applications deployed in the field test. We compare the pre-defined pilot success criteria against what has been achieved in the project both from a technological and a user centered point of view. Finally we outline future plans in maintaining the existing pilot applications and developing new applications beyond the scope of the project.

1 Technological Background

The concepts used to develop the pilot applications in WP I were derived from the users' need for customizable information retrieval and filtering services. The approach taken was to design a generalized framework for these kind of applications and to use FollowMe architectural concepts and code libraries to implement it.

In the following sections we outline the concepts of the WP I application framework and describe its FollowMe-based components.

1.1 Pilot 1 Application Framework

Today the World Wide Web is a loosely coupled and fairly unstructured environment composed of millions of information consumers and providers. Anyone familiar with the Web knows of its major contradiction: *Whatever information you look for is available somewhere on the net, but locating the appropriate information sources is extremely time consuming and therefore very costly.*

Any solution to this contradiction should aim to leverage the existing infrastructure of the Web to the worlds largest, truly distributed database. In approaching a solution towards this aim, we identified the following core issues:

- The usefulness of a database depends not only on the sheer amount of collected data, but on the applications or services that operate on the contents of the database to provide information in form of customized results to the users of such systems.
- In order to enable the development of useful database applications, any database needs to provide a meta-model describing the structure of the offered data.
- To gain the most from largely distributed databases, the development of database applications should be de-coupled from the maintenance of the databases themselves. That way, the contents of data sources can be re-used and re-combined when developing new applications according to the needs of information consumers.

A first analysis of above issues leads to the most important design decision for the WP I pilot application framework: *to de-couple the roles of service providers and content providers*. *Service providers* implement applications that make use of raw data offered by *content providers*. They define meta-models describing the data structures their applications are capable of dealing with. In order to enable the *service providers*' applications to make use the data offered by a *content provider*, the data needs to be structured according to meta-models that form supersets of the meta- models of the *service providers*. This decision leads straightforward to the following core axioms:

- Users of our applications will no longer (as is with the Web) address *content providers* to obtain information in raw or proprietary data format. Instead they will address services that provide them with already refined information according to their individual needs. The services therefore need to be customizable by the individual user.
- Services do not operate on a predefined or hard-coded set of data sources, but on specific data structures. They may use any data source available at runtime that offers relevant data as long as it offers an interface the service knows how to use.

These axioms impose the introduction of components *that glue things together*. On the one hand, users need to have an effective way of locating services that fit their needs. On the other hand, there need to exist mechanisms that enable services to locate relevant information sources at runtime. The appropriate concepts to fulfil these requirements are the ideas of brokers, matchmakers or – more simply – directory services.

Thus the core components of the application framework have been identified as (see figure 1):

- *content providers* (offering access to data-objects),
- *service providers* (providing services operating on data available from the *content providers*),
- *information consumers* (users of available services) and
- *directory services* (mediating between the other components).

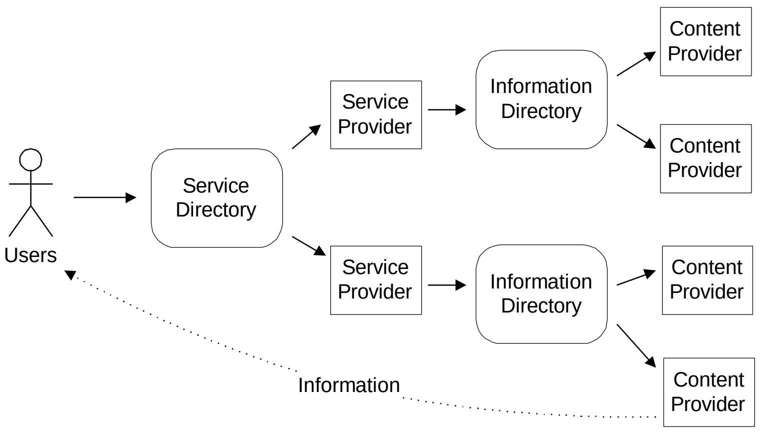


Figure 1: Core components of the application framework

The FollowMe architectural framework provides agent, scripting and profiling concepts to build up an agent interaction framework that enables the deployment of agent driven applications meeting the above stated needs.

In terms of the FollowMe *Agent Framework* (see WP D, E and F) a service is composed of a component related to the information consumer (referred to as *task agent*) and a component implementing an interface to content providers (referred to as *service interaction interface*). A special user related agent (referred to as *personal assistant*) assists the user in organizing the usage of services and handling personalized information.

In most client/server or network centric applications currently available on the Internet users connect to remote applications and specify their interests in form of application specific parameters. All computation is done on the server side and results are delivered to the clients. This enables application providers to gather and data-mine lots of personalized information about users of their system. In contrast, within the agent based approach of the FollowMe system, the user's privacy is respected by hosting all user related information and computation in a location close to the user and in an environment trusted by the user (see concepts of *profiles* and *information spaces* in the FollowMe architecture documentation). All agents acting on behalf of a user in FollowMe are instantiated on request by downloading the respective Java classes from so called *agent factories* to a user trusted environment (referred to as *FollowMe places*). This environment is located on a host with permanent online connection (i.e. a local ISP).

1.2 Scenario Description

To illustrate the above outlined basic concepts of the WP I application framework we describe a fairly generic scenario. We assume that the user in the described scenario already owns a *personal assistant* and is now on his way to select one of the services offered by *service providers*. As an example we introduce a service that delivers information on regional events (i.e. concerts, cinemas, markets, exhibitions). This specific service is one of the two applications that have been implemented to validate the concepts of the application framework (see chapter 2 for details on this specific application domain).

Step 1:

A user connects to the system by contacting his *personal assistant* (*PA*). The user may then make changes to his personal *diary*, i.e. by stating, that he will be reachable by e-mail during working hours and by fax otherwise (see figure 2).

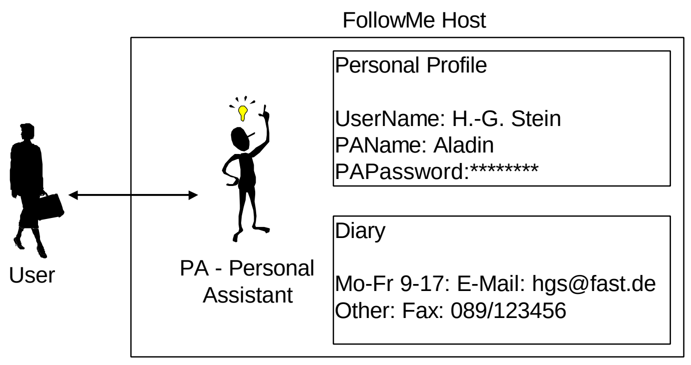


Figure 2: Contacting the *personal assistant*

Step 2:

The user wants to use one (or more) of the offered services.

The *PA* connects the user to a *service directory* that allows the user to select a specific service the user is interested in. Note that the *PA* does not require any knowledge about specific services. This de-couples the user specific components (such as the *PA*) from the rest of the system (available information sources and services). After selecting a service to subscribe to, the *directory service* links to the appropriate *service provider* (*agent factory*) and an instance of a *task agent* representing the respective service is created on the FollowMe host to service the individual request of its owner (see figure 3).

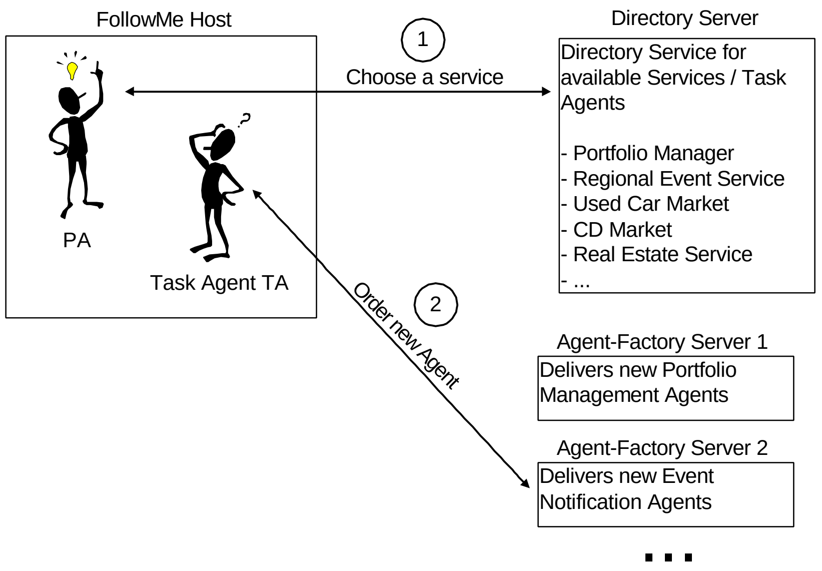


Figure 3: Instantiation of a new *task agent*

Step 3:

The user may now provide the new *task agent* with personalized parameters. In case of an event notification service, parameters might include specific event types of interest to the user and location and date of events. Moreover the user specifies a time schedule defining when he wants the results of the service to be delivered (see figure 4).

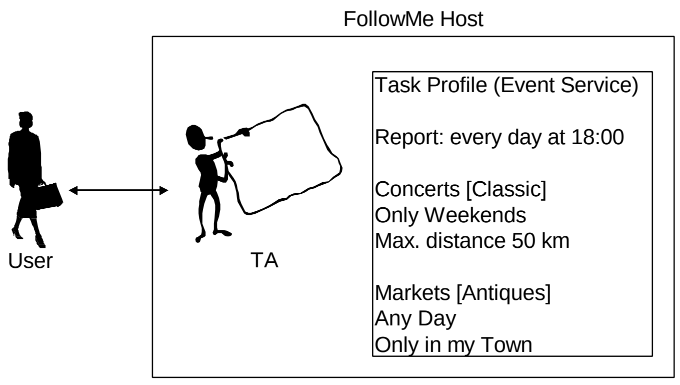


Figure 4: Defining a *task profile*

Step 4:

After specifying these input parameters the user may disconnect from the system. The specified service will execute automatically and in regular intervals according to the user defined schedule. The agent in charge of executing the respective task will contact another *broker* (*directory service*) that links it to *service interaction interfaces* at *content provider* sites relevant to the application domain of the *task agent* (in our example these are providers of information on regional events). All the agent needs to know is the type of interfaces it is capable of connecting to. There is no need to hard-code the addresses of *content providers* within the agent. This de-couples *service providers* (*agent providers*) from *content providers*. That way, new *content providers* could join the system by registering at the *directory service* without the need for changes to existing services. The same holds for the integration of new services operating on data offered by existing *content providers*. In our example the agent connects to servers providing information on regional events and queries these servers according to the user specified parameters.

The directory service holds meta-data for each service. Based on this meta-data a task agent can optimize the selection of content providers. In the domain of regional events the meta-data of a service may contain geographical information on what region is served. By this information the agent could rule-out content providers that do not serve events relevant for its query.

As described in the architectural framework, all objects and thus agents are potentially mobile. *Service interaction interfaces* are capable of providing an agent runtime environment (a FollowMe *place* – see architecture framework). Whether an agents makes use of these mobility features depends on the type of service the agent is representing. In applications where communication between a *task agent* and a *service interface* is very intensive (i.e. sophisticated negotiation processes) the agent could be designed to move to the interface instead of remotely connecting to it (see figure 5):

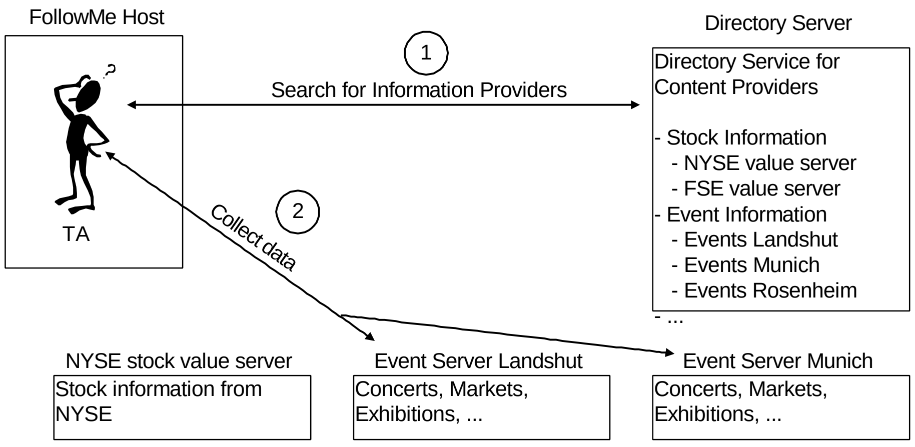


Figure 5: Task execution

Step 5:

After the *task agent* finished its task of information retrieval and refining, it stores the information in the user's *information space* for future reference either by the user or by the agent itself (see architecture framework). In addition the agent might be instructed to deliver reports on new information to its user. In our example the user instructed the agent to send a report every day at 18:00. Reporting might as well be triggered by specific changes to data values or other events (i.e. a stock portfolio management agent might be instructed to report to the user immediately when a share value exceeded a specified limit).

To send reports to its user, the *task agent* uses the *user access components*, which provide gateways to a variety of devices like mail boxes, phones, pagers or fax machines (see architecture framework). The kind of device to be used for report delivery is stated in the diary section of the user's *personal profile*. The *task agent* consults the *personal assistant* to obtain this information. In our example the appropriate device for delivering reports at 18:00 is a fax gateway (see figure 6).

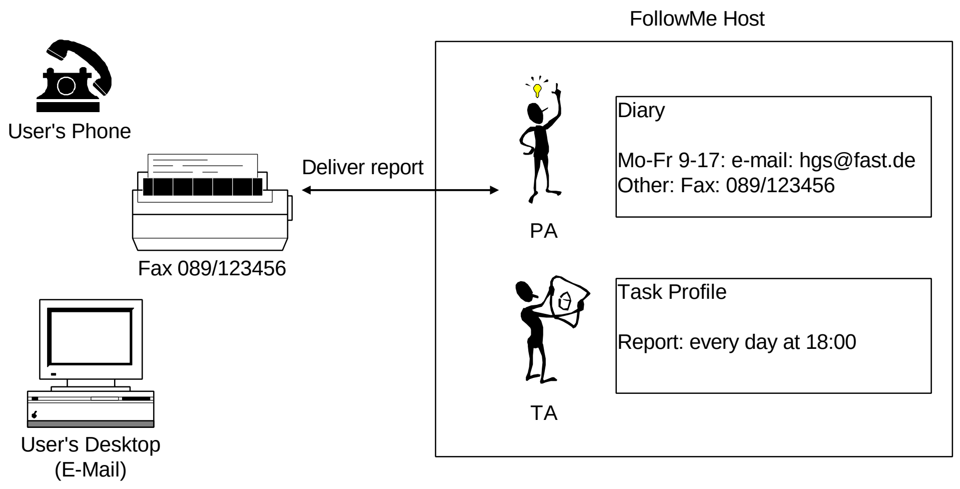


Figure 6: Delivering a report

1.3 System Components

With the previous scenario description in mind we now take a closer look at the system components and how they map onto the components provided by the technical work packages of the FollowMe project (see figure 7).

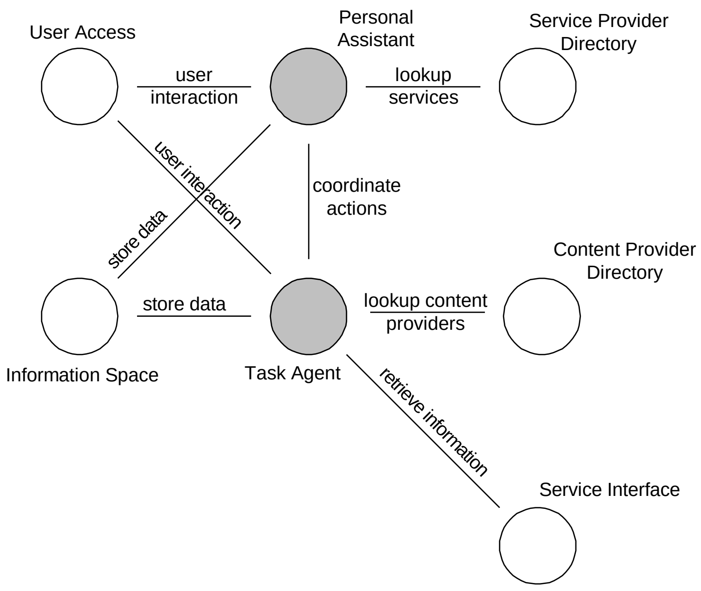


Figure 7: System components of the application framework

Personal Assistant:

The *personal assistant* in figure 7 is in fact composed of a number of components (figure 8).

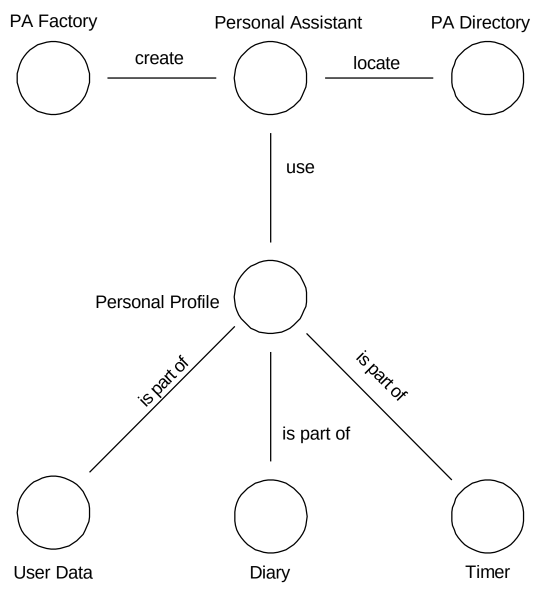


Figure 8: Personal Assistant Components

A *personal assistant* (*PA*) represents a single user of the FollowMe system. The role of the *PA* is to assist the user in organizing the usage of services. Attached to the *PA* is the user's *personal profile*. This profile is used to store persistent data about the user. Basically user related data consists of the user's name and address, the user's system access password and a list of the services the user is currently subscribed to. Since all persistent data is stored in XML (see architecture framework), profiles can be easily extended according to the needs of the evolving system.

Besides basic persistent data about the user, the *personal profile* provides a *diary* functionality that enables the user to specify how the system may contact him at different points in time (see description of the *user access component*).

Agents act on behalf of the user while the user is offline. The system therefore provides mechanisms that enable the scheduling of task execution and reporting. This mechanism is provided by a *timer* component as part of the *personal profile*.

New users might join the system by requesting the instantiation of a new *personal assistant*. A new *PA* is created by instantiating the respective Java classes from a (remote) *personal assistant factory*.

To allow users to address their *PA* by the *PA*'s name (i.e. *MyAgent Aladin*), there exists a *PA directory* mapping *PA* names to object references.

Task Agent:

The *task agent* in figure 7 is composed of components similar to those of the *PA* (see figure 9).

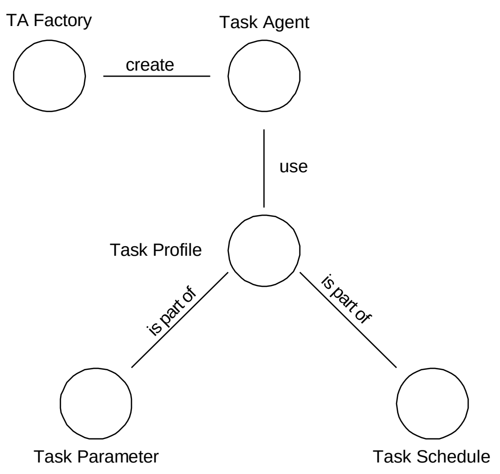


Figure 9: Task Agent Components

Task agents are agents offering application domain specific functionality. They are capable of using services with domain specific interfaces. The tasks of an agent can be described in terms of *missions* programmed in the scripting language of the FollowMe Agent Framework (see WP D, E and F) or coded in Java classes. *Task agents* can communicate with the *personal assistant* using the message passing protocols provided by the underlying *mobile object workbench* (see architecture framework).

Interfaces to content provider services (data-sources) are registered with a content provider directory service. The directory service may hold meta-data for each service. Based on this meta-data a task agent can optimize the selection of content providers. In example, these interface descriptions may provide information on update schedules of the *content providers*' databases (i.e. it doesn't make sense for an agent to query a *content provider* for new information every five minutes when the data sources are updated only once a day).

Attached to a *task agent* is a *task agent profile*. The profile contains user specified task parameters and scheduling information defining when to execute which tasks.

Task agents are created by downloading mission profiles and instantiating the respective Java classes from a (remote) *task agent factory* located at a *service provider* site.

Component Interaction:

As outlined in the scenario description, the user communicates with agents via the *user access* through a variety of different device types. User interaction is required to provide the agent system with personalized instructions on how and when to execute certain tasks on behalf of the user.

Both agent types, *personal assistants* and *task agents,* may store the results of their information retrieval activities (that is data objects retrieved from *content provider* sites) in a user's *information space*. Access to stored objects is via location transparent object references. The *information space* provides access control mechanisms to ensure that data can be accessed only by authorized agents.

The activities of a user's *task agents* are coordinated by the *personal assistant*. The *PA* keeps a record of all active *task agents*. Tasks are triggered by the *timer* component in the *personal profile*.

Personal assistants link their users to the directory of available services whereas *task agents* contact *content provider directories* to determine which data sources (in form of *service interfaces*) to use. This again demonstrates the concept of separating *service providers* from *content providers* and thus freeing the user from having to deal with widely distributed raw data.

2 Implemented Pilots

In the previous chapter we briefly described the application level architecture used to implement the pilot applications of WP I. Based on these generic models we implemented two specific services.

2.1 Identification of Knowledge Domains

To identify knowledge domains that are of specific interest to the targeted pilot application test users, we followed the following pre-survey procedures:

First we introduced the generic concepts of the application architecture to representatives of the Bavarian Bürgernetz organization and explained to them the key advantages the architecture offers compared to other existing information retrieval and management systems. In general they expected that their users would be especially interested in the concepts of autonomy, personalization and automated reporting.

Since the Bavarian Bürgernetze continuously have been monitoring their users' interest and information needs for several years, they were able to provide us with a set of knowledge domains they could foresee that their users would be especially interested in. Two major characteristics of potential knowledge domains could be identified:

- Domains dealing with information with high regional focus: Regional social events, marketplaces for purchasing used items (e.g. cars, furniture, sports equipment,...), real estate business
- Domains dealing with highly dynamic information with strong user interest for up-to-date information: Financial information services, daily news, weather information services

Secondly we identified potential information or content providers capable of providing access to raw data for the respective knowledge domain and queried them for their interest in participating in the pilot trials. It turned out that in specific domains (such as e.g. real estate business, used cars) providers are not (yet) interested in allowing external applications to browse their data and compare it to data from other providers. Therefore we had to narrow down the set of potential application domains.

Additionally we wanted to choose domains that offer the opportunity to build applications that fit well for demonstrating the value-added features of our information retrieval and delivery architecture.

With this information we pinned down the two application domains in which we planned to offer services to the users of the Bavarian Bürgernetze:

Domain of regional social events

The content providers in this domain offer information on social events with regional focus. Events advertised by the content providers might be for example local cinema programs, concerts, theater performances, flea markets, etc. This specific domain is likely to be of interest to potential users and demonstrates the value-added features of our information retrieval and delivery architecture due to the following core characteristics:

- The information has extremely regional focus and is valuable to all Bürgernetz-users.
- The contents are provided by the Bürgernetze themselves (thus getting access to the data is not a problem at all).
- Any application in this domain requires component personalization by the user.
- Automated and scheduled reporting is of great value (e.g. users state their interest in up-coming Mozart concerts, forget about it and will be automatically informed whenever such an event is advertised).

Domain of stock share value information

Information in this domain is on constantly updated share values for stocks traded at major European and US stock exchanges. Again we summarize the core characteristics of this specific domain related to the objectives of WP I:

- The information offered is highly dynamic (constantly changing share values).
- Personalization is required for maintaining individual portfolios.
- Automated, instant reporting is of great value (e.g. sending a SMS message to a mobile phone whenever a share value crosses a user specified limit).
- Publicly accessible content providers can be queried for the required information.

Finally we directly interviewed potential users. We introduced the general key featured of the planned applications (such as automated information retrieval, automated reporting to various device types, personalization) and asked them in how far applications offering these features would be helpful to satisfy their individual everyday information needs. Moreover we described the two application domains and asked the users whether they could think of frequently using applications with above features delivering information related to these domains.

All interviewed persons showed both interest in the general features of the application framework and interest in the proposed application domains.

The questionnaires used for the interviews and the detailed results have been published in a document delivered as a supplement to the WP I requirements deliverable.

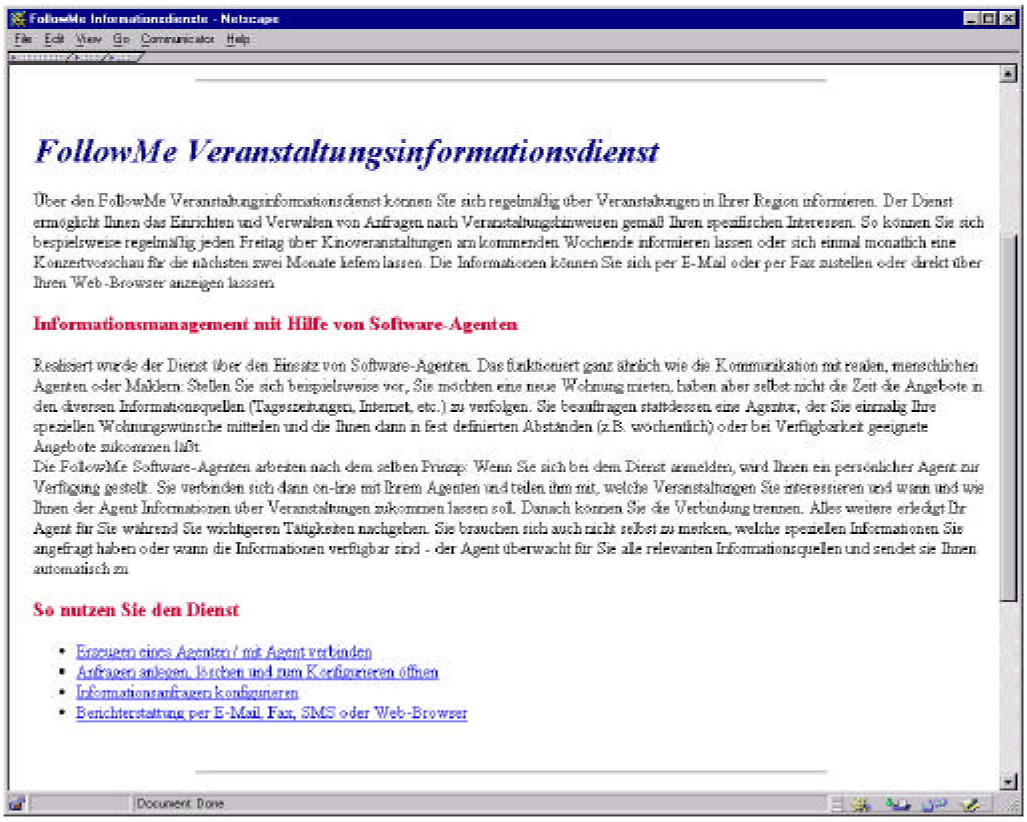
2.2 Application Domains and User Models

In this section we describe the domain and user models of the two domains we chose for implementing the pilot applications. The domain and user models are described in terms of actors and system use cases (for more details see the requirements and design documents of WP I). The use cases are illustrated by screen shots showing corresponding user interfaces.

Accessing the personal assistant:

Prior to actually describing the domains we illustrate the user interfaces to the personal assistant: Connecting to the application is via URL: pictor.fast.de:8888/servlet/fmeproxy/html/b-online.html.

This start page provides an general introduction to personalised information management using our software-agents. The button next to the FollowMe logo connects to the personal assistant creation and login page (see figure 10).


FollowMe Informationsdienste - Netscape
File Edit View Go Communicator Help
Document: Done

FollowMe Veranstaltungsinformationsdienst

Über den FollowMe Veranstaltungsinformationsdienst können Sie sich regelmäßig über Veranstaltungen in Ihrer Region informieren. Der Dienst ermöglicht Ihnen das Einrichten und Verwalten von Anfragen nach Veranstaltungshinweisen gemäß Ihren spezifischen Interessen. So können Sie sich beispielsweise regelmäßig jeden Freitag über Kinoveranstaltungen am kommenden Wochende informieren lassen oder sich einmal monatlich eine Konzertvorschau für die nächsten zwei Monate liefern lassen. Die Informationen können Sie sich per E-Mail oder per Fax zustellen oder direkt über Ihren Web-Browser anzeigen lassen.

Informationsmanagement mit Hilfe von Software-Agenten

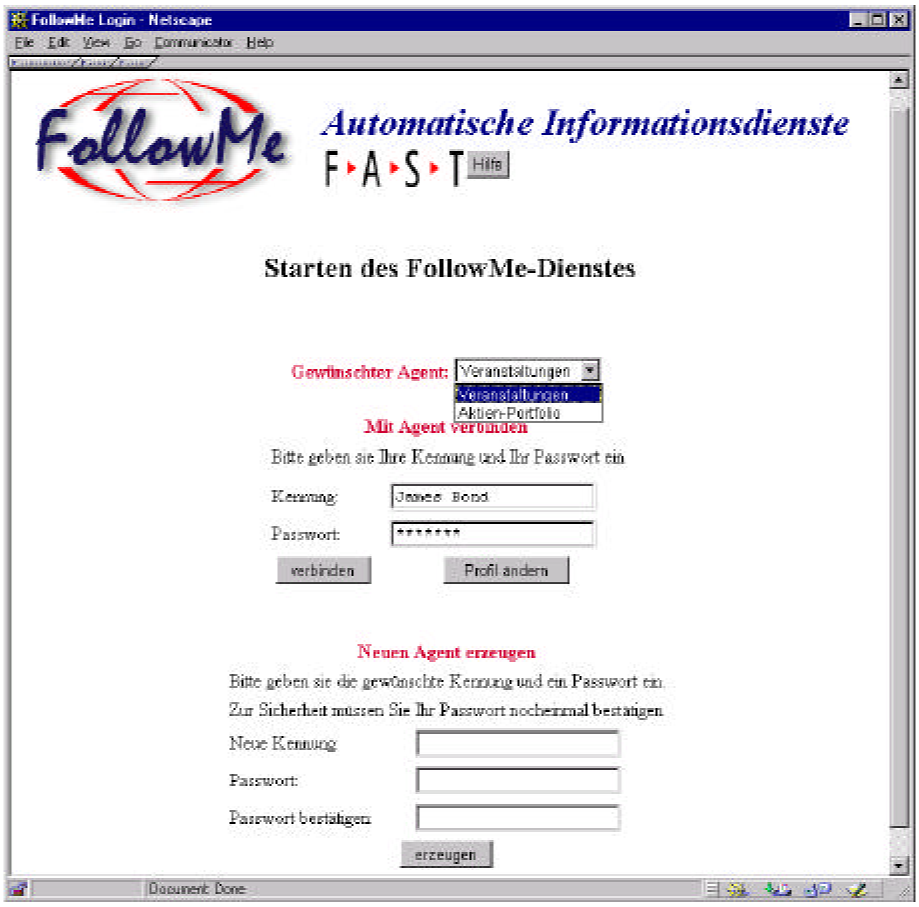
Realisiert wurde der Dienst über den Einsatz von Software-Agenten. Das funktioniert ganz ähnlich wie die Kommunikation mit realen, menschlichen Agenten oder Maklern: Stellen Sie sich beispielsweise vor, Sie möchten eine neue Wohnung mieten, haben aber selbst nicht die Zeit die Angebote in den diversen Informationsquellen (Tageszeitungen, Internet, etc.) zu verfolgen. Sie beauftragen stattdessen eine Agentur, der Sie einmalig Ihre speziellen Wohnungswünsche mitteilen und die Ihnen dann in fest definierten Abständen (z.B. wöchentlich) oder bei Verfügbarkeit geeignete Angebote zukommen läßt.
Die FollowMe Software-Agenten arbeiten nach dem selben Prinzip: Wenn Sie sich bei dem Dienst anmelden, wird Ihnen ein persönlicher Agent zur Verfügung gestellt. Sie verbinden sich dann on-line mit Ihrem Agenten und teilen ihm mit, welche Veranstaltungen Sie interessieren und wann und wie Ihnen der Agent Informationen über Veranstaltungen zukommen lassen soll. Danach können Sie die Verbindung trennen. Alles weitere erledigt Ihr Agent für Sie während Sie wichtigeren Tätigkeiten nachgehen. Sie brauchen sich auch nicht selbst zu merken, welche speziellen Informationen Sie angefragt haben oder wann die Informationen verfügbar sind - der Agent überwacht für Sie alle relevanten Informationsquellen und sendet sie Ihnen automatisch zu.

So nutzen Sie den Dienst

- Erzeugen eines Agenten / mit Agent verbinden
- Anfragen anlegen, löschen und zum Konfigurieren öffnen
- Informationsanfragen konfigurieren
- Berichterstattung per E-Mail, Fax, SMS oder Web-Browser

Figure 10: Pilot application start page

The personal assistant creation and login page is divided into two sections (see figure 11). New users may create a new personal assistant by providing a user name and a password of choice. Already registered users may connect to their personal assistant by entering their user name and their password. When creating a new personal assistant users are automatically prompted to initially edit their personal profile (see figure 12). Registered users may choose to edit their profile or to directly connect to a specific task agent (selectable via a list-box). After the personal profile has been edited and saved, the personal assistant connects the user with the task agents the user did selected from the list-box in the login dialog.


FollowMe Login - Netscape
File Edit View Go Communicator Help
FollowMe
Automatische Informationsdienste
F·A·S·T
Hilfe
Document: Done

Starten des FollowMe-Dienstes

Gewünschter Agent: Veranstaltungen
Veranstaltungen
Aktien-Portfolio

Mit Agent verbinden

Bitte geben sie Ihre Kennung und Ihr Passwort ein

Kennung:	James Bond
Passwort:	*******

verbinden Profil ändern

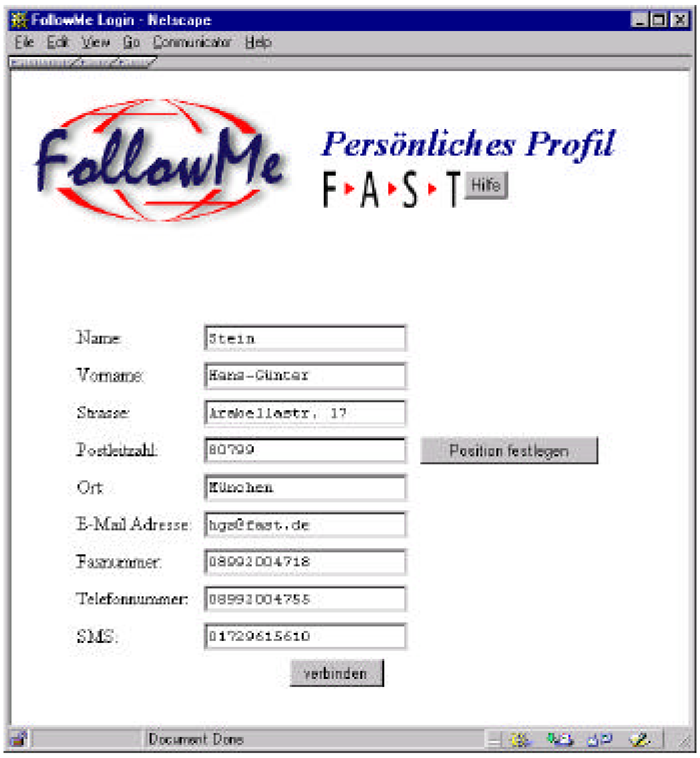
Neuen Agent erzeugen

Bitte geben sie die gewünschte Kennung und ein Passwort ein.
Zur Sicherheit müssen Sie Ihr Passwort nocheinmal bestätigen

Neue Kennung	
Passwort:	
Passwort bestätigen	

erzeugen

Figure 11: Login dialog


FollowMe Login - Netscape
File Edit View Go Communicator Help
FollowMe
Persönliches Profil
F·A·S·T
Hilfe
Document: Done

Name:	Stein
Vorname:	Hans-Günter
Strasse:	Arabellastr. 17
Postleitzahl:	80799
Ort:	München
E-Mail Adresse:	hgs@fast.de
Faxnummer:	08992004718
Telefonnummer:	08992004755
SMS:	01729615610

Position festlegen

verbinden

Figure 12: Profile dialog

The domain of regional event notification:

The Regional Event Notification System gathers and delivers information in the domain of social events (movies, concerts, theatre performances, special markets, etc.).

Referring to the pattern of information latticework (see figure 1), actors in this system are:

- institutions advertising events they host,
- content providers that host databases at which above institutions may register their ads,
- a service provider providing user customisable task agents specialised in gathering and delivering information on social events,
- people interested in visiting events and thus use the system to obtain event advertisements.

The institutions register their events at content provider sites according to the following data model:

- Type of event (‘What?’): Data on social events is classified according to an extensible three layer category scheme.
- Geographical location of an event (‘Where?’): A social event takes place at a specific geographical location.
- Date of event (‘When?’): Events take place at specific moments in time. This information is coded in event start and event end dates.

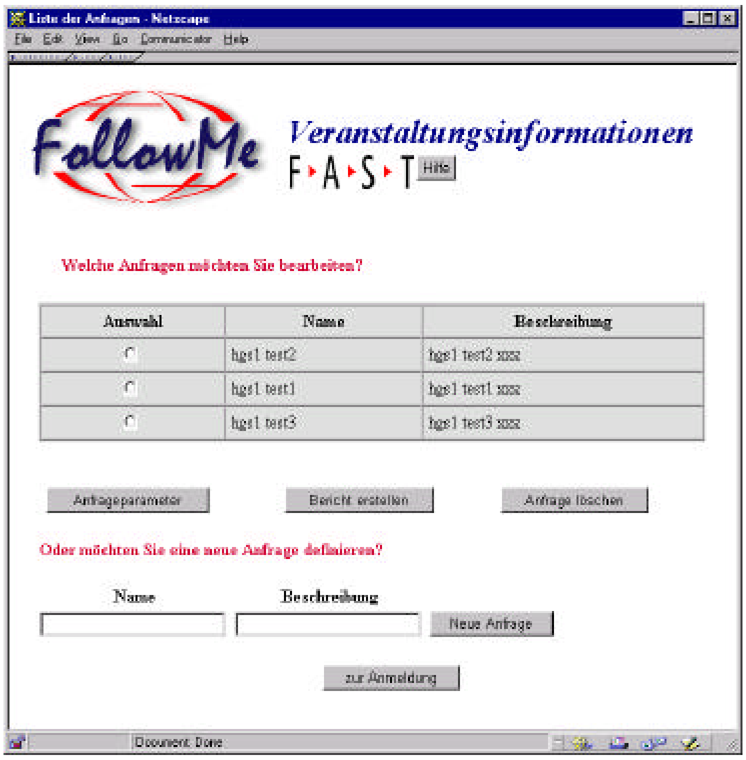
The content providers are not restricted to using a specific database model or a specific type of database to store the data. The only constraint is that they have to provide an interface that allows the task agents provided by the service provider to connect to an ip-port to place a query encoded in a well-defined SQL-statement. Once content providers have set up a database and a proper interface, they have to inform the service provider about the ip-address and ip-port number of the interface. The service provider will then register the interface with the content provider directory service. From there on task agents may include the new source of information in their information gathering processes.

In addition to ip-address and ip-port number, the content providers may provide the service provider with information defining the boundaries of the geographical area in which events advertised in their database take place. This information can be attached to the respective registry entry in the content provider directory service in form of service properties (see documentation of the FollowMe Agent Framework). The tasks agents may use this information to avoid querying content provider sites that hold information that is not of interest to the owners of the task agents.

The task agents specialised in the domain of regional social events provide their users with mechanisms to automate retrieval of information structured according to above data model. Users may declare and maintain an arbitrary number of queries on social events according to their individual interests. In addition to specifying parameters defining what kind of information users are looking for, they may declare how and when they want to access the information by providing report delivery schedules. Reports on upcoming events may be sent via fax or e-mail or accessed online via web-browsers. Due to their message size limitations SMS devices do not play a major role in this domain. Users may use SMS to receive a reminder stating that new event advertisements are available for view via browser. The actual event listings are not sent via SMS.

User interfaces of the event notification task agent :

After successful authentication (and profile editing) users may create a new query, select an existing query to either delete it or change its configuration or view an online report related to a selected query (figure 13).



Welche Anfragen möchten Sie bearbeiten?

Auswahl	Name	Beschreibung
○	hgs1 test2	hgs1 test2 xxx
○	hgs1 test1	hgs1 test1 xxx
○	hgs1 test3	hgs1 test3 xxx

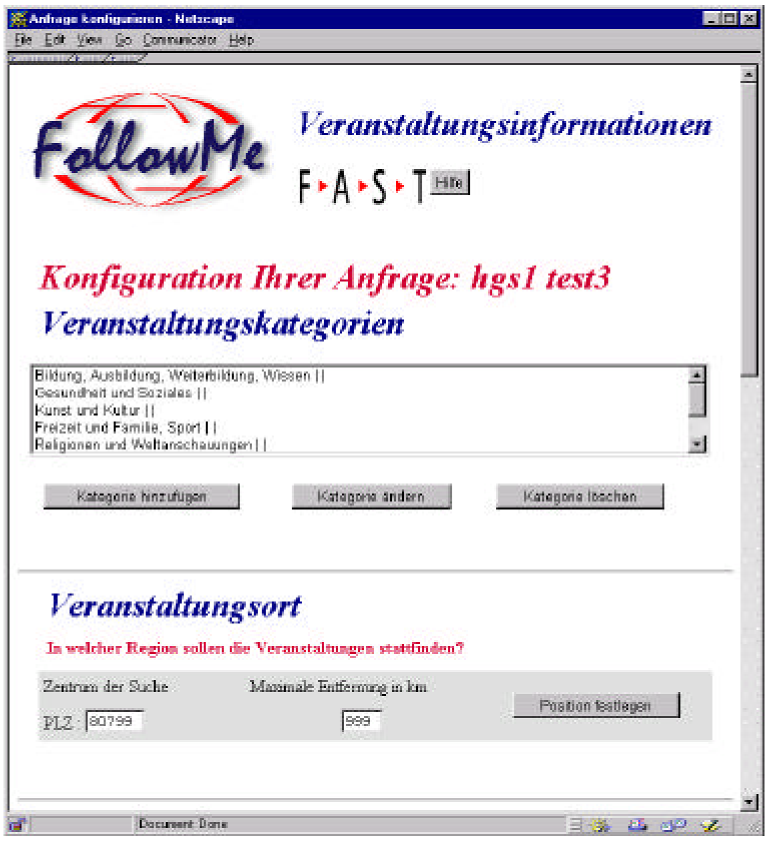
Oder möchten Sie eine neue Anfrage definieren?

Name	Beschreibung

Figure 13: Creation and selection of queries

In order to instruct their agent on what information it should gather and when this information should be delivered, users need to fill out a configuration form (see figures 14, 18 and 19) which is divided into four sections. The first three sections deal with the ‘what, where and when’ of events that should be returned by the query. The last sections deals with the report delivery schedule.

All events stored in the content provider databases are classified according to a three level category scheme. In the first section of the configuration form users may define a list of categories into which the requested information should fit.



Veranstaltungsinformationen

F▸A▸S▸T

Konfiguration Ihrer Anfrage: hgs1 test3

Veranstaltungskategorien

Bildung, Ausbildung, Weiterbildung, Wissen ||
Gesundheit und Soziales ||
Kunst und Kultur ||
Freizeit und Familie, Sport ||
Religionen und Weltanschauungen ||

Veranstaltungsort

In welcher Region sollen die Veranstaltungen stattfinden?

Zentrum der Suche

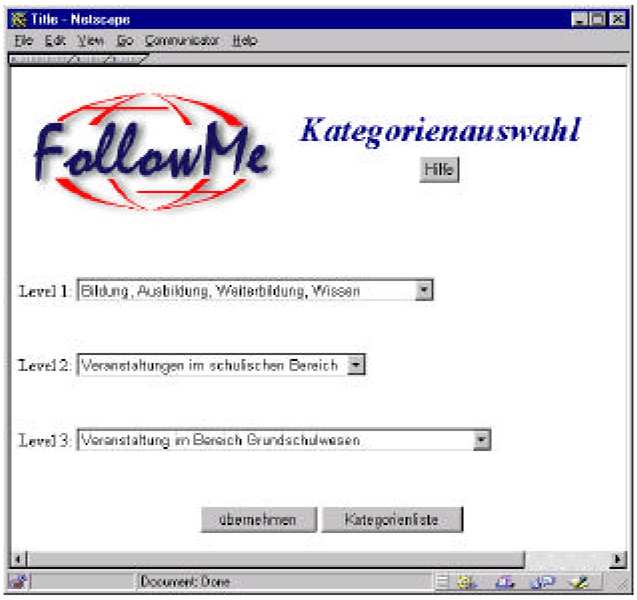
PLZ : 80799

Maximale Entfernung in km

999

Figure 14: Configuring a query – part 1

When adding to or changing categories in above list, users may either directly select the appropriate categories by using the form shown in figure 15 or select the categories of choice from a complete listing of all available categories as shown in figure 16.



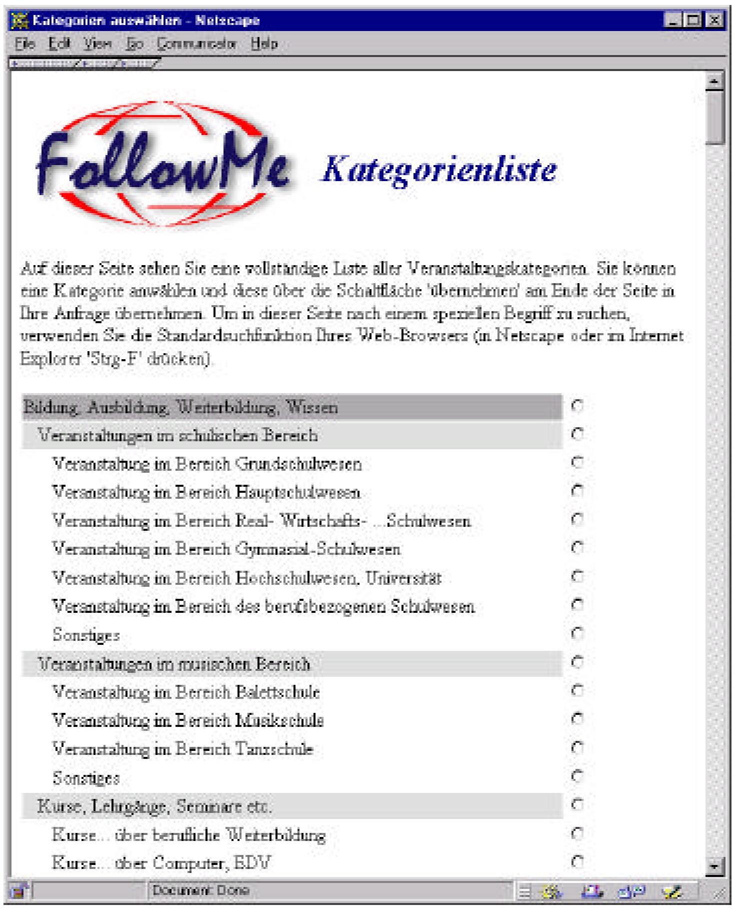
Kategorienauswahl

Level 1: Bildung, Ausbildung, Weiterbildung, Wissen

Level 2: Veranstaltungen im schulischen Bereich

Level 3: Veranstaltung im Bereich Grundschulwesen

Figure 15: Selection of event categories


Kategorien auswählen - Netscape
File Edit View Go Communicator Help
FollowMe
Kategorienliste
Document: Done

Auf dieser Seite sehen Sie eine vollständige Liste aller Veranstaltungskategorien. Sie können eine Kategorie anwählen und diese über die Schaltfläche 'übernehmen' am Ende der Seite in Ihre Anfrage übernehmen. Um in dieser Seite nach einem speziellen Begriff zu suchen, verwenden Sie die Standardsuchfunktion Ihres Web-Browsers (in Netscape oder im Internet Explorer 'Strg-F' drücken).

Bildung, Ausbildung, Weiterbildung, Wissen ○
Veranstaltungen im schulischen Bereich ○
Veranstaltung im Bereich Grundschulwesen ○
Veranstaltung im Bereich Hauptschulwesen ○
Veranstaltung im Bereich Real- Wirtschafts- ...Schulwesen ○
Veranstaltung im Bereich Gymnasial-Schulwesen ○
Veranstaltung im Bereich Hochschulwesen, Universität ○
Veranstaltung im Bereich des berufsbezogenen Schulwesen ○
Sonstiges ○
Veranstaltungen im musischen Bereich ○
Veranstaltung im Bereich Balettschule ○
Veranstaltung im Bereich Musikschule ○
Veranstaltung im Bereich Tanzschule ○
Sonstiges ○
Kurse, Lehrgänge, Seminare etc. ○
Kurse... über berufliche Weiterbildung ○
Kurse... über Computer, EDV ○

Figure 16: Complete category listing

In the next section of the configuration form (figure 14) the user has to specify the geographical area in which the requested events should take place. Geographical locations in the Event Notification System are specified in geo-coordinates via a third party geographical information system.

In order to specify the geographical area in which the requested events should take place the user has to provide geo-coordinates of the centre of the search along with a search radius. To pin down the coordinates the user specifies an area zip code and connects to the geographical information system which pops up with a map centred on the area of the specified zip code. From there the user may use zoom and pin functions to exactly pin down the precise location of the centre of the search.

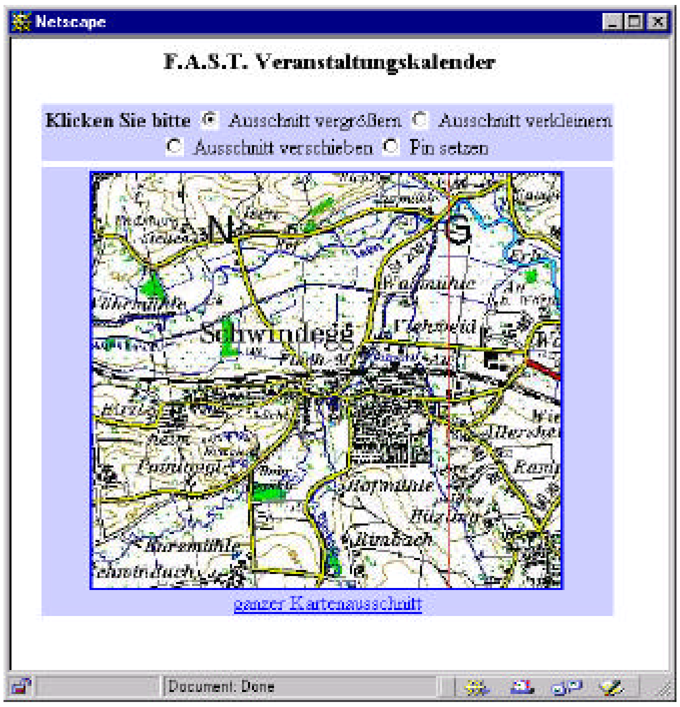
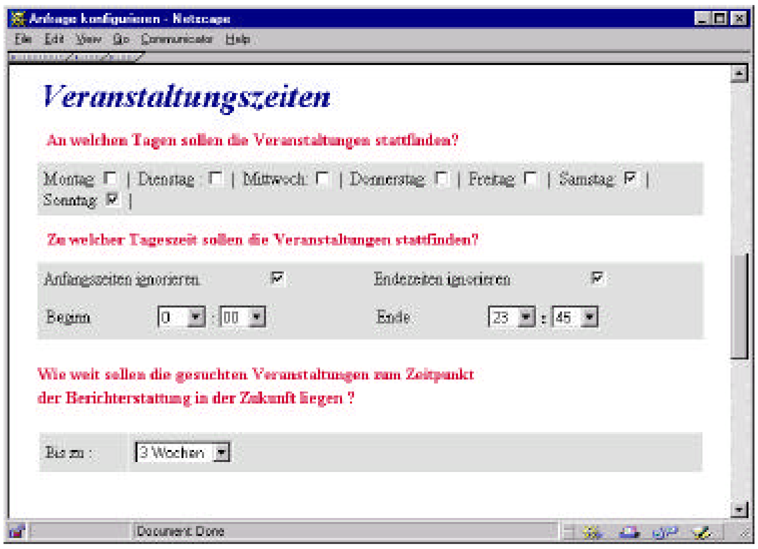


Figure 17: The geographical information system

Next, the user has to specify on which weekdays and at which day-times the requested events should take place. Event start times and event end times can optionally be ignored. If, for example, users are interested in events starting after 18:00 but don't care for event end times, they would specify 18:00 as event start and check the 'ignore event end time' checkbox.

Additionally the user must specify the maximum time interval allowed between time of reporting and occurrence of the event (figure 18).


Anfrage konfigurieren - Netscape
File Edit View Go Communicator Help
Document: Done

Veranstaltungszeiten

An welchen Tagen sollen die Veranstaltungen stattfinden?

Montag ☐ | Dienstag ☐ | Mittwoch ☐ | Donnerstag ☐ | Freitag ☐ | Samstag ☑ | Sonntag ☑ |

Zu welcher Tageszeit sollen die Veranstaltungen stattfinden?

Anfangszeiten ignorieren ☑ Endezeiten ignorieren ☑

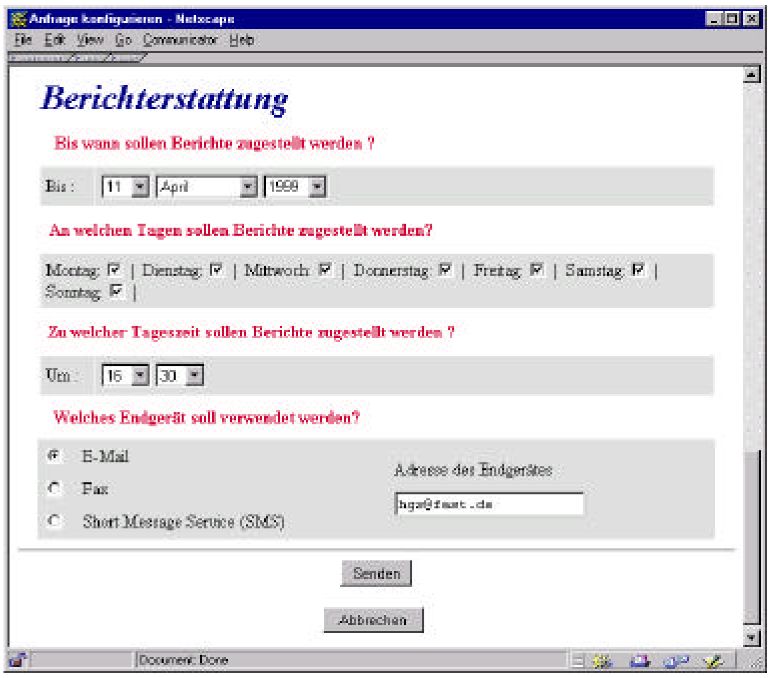
Beginn 0 : 00 Ende 23 : 45

Wie weit sollen die gesuchten Veranstaltungen zum Zeitpunkt der Berichterstattung in der Zukunft liegen ?

Bis zu : 3 Wochen

Figure 18: Configuring a query – part 2

In the last section the user provides information on when (in which intervals) reports on information about regional events should be sent. To do so, the user specifies a date up to which reports should be sent on a regular basis (after this date reporting will stop). Next the user specifies at which week-days and at which day time the reports should be sent. Finally the user specifies an output device which may be either an e-mail address, a fax number or a SMS number.


Anfrage konfigurieren - Netscape
File Edit View Go Communicator Help
Document: Done

Berichterstattung

Bis wann sollen Berichte zugestellt werden ?

Bis : 11 April 1999

An welchen Tagen sollen Berichte zugestellt werden?

Montag: ✓ | Dienstag: ✓ | Mittwoch: ✓ | Donnerstag: ✓ | Freitag: ✓ | Samstag: ✓ | Sonntag: ✓ |

Zu welcher Tageszeit sollen Berichte zugestellt werden ?

Um : 16 30

Welches Endgerät soll verwendet werden?

[✓ E-Mail]
Fax
Short Message Service (SMS)

Adresse des Endgerätes: hgz@fast.de

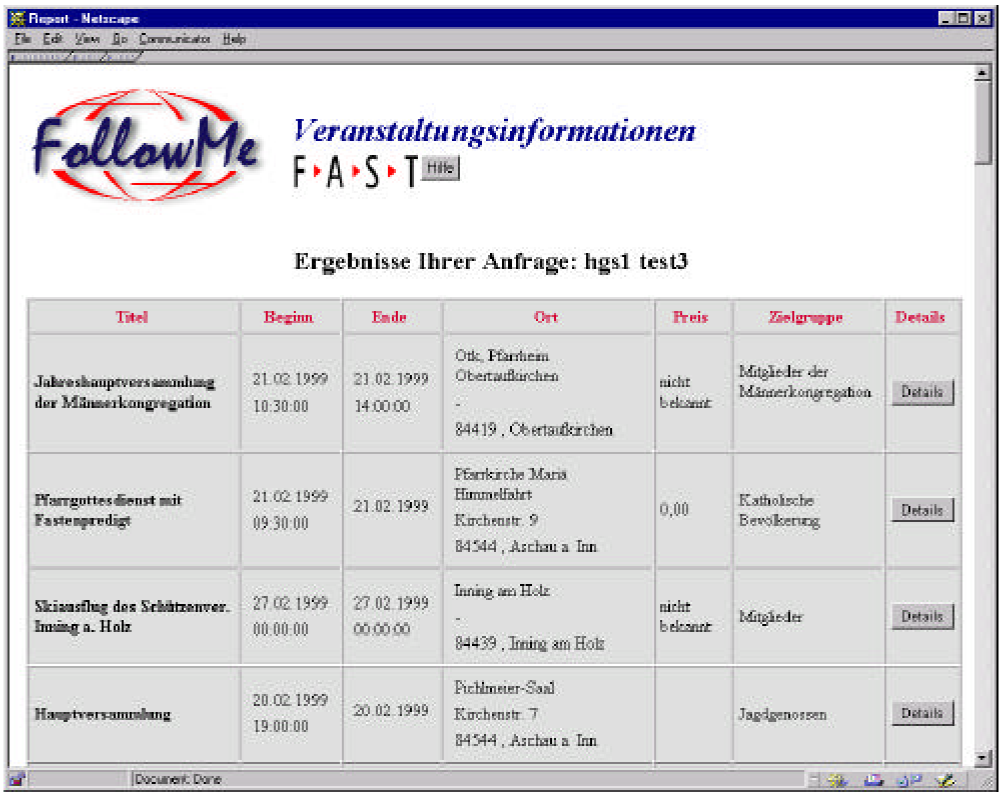
Senden

Abbrechen

Figure 19: Configuring a query – part 3

After submitting the configuration parameters the user may disconnect from the system. Reports on regional events will be sent automatically according to the specified parameters.

Reports are sent to either fax, SMS or e-mail gateways (where SMS is used only to send a short message stating that new information had been gathered and could be reviewed via web-browser). In addition to viewing reports via these output media, users may choose to generate reports online using their web-browsers. These reports do not triggered new database queries. Instead the results of the most recent already executed query are used to produce the online report.

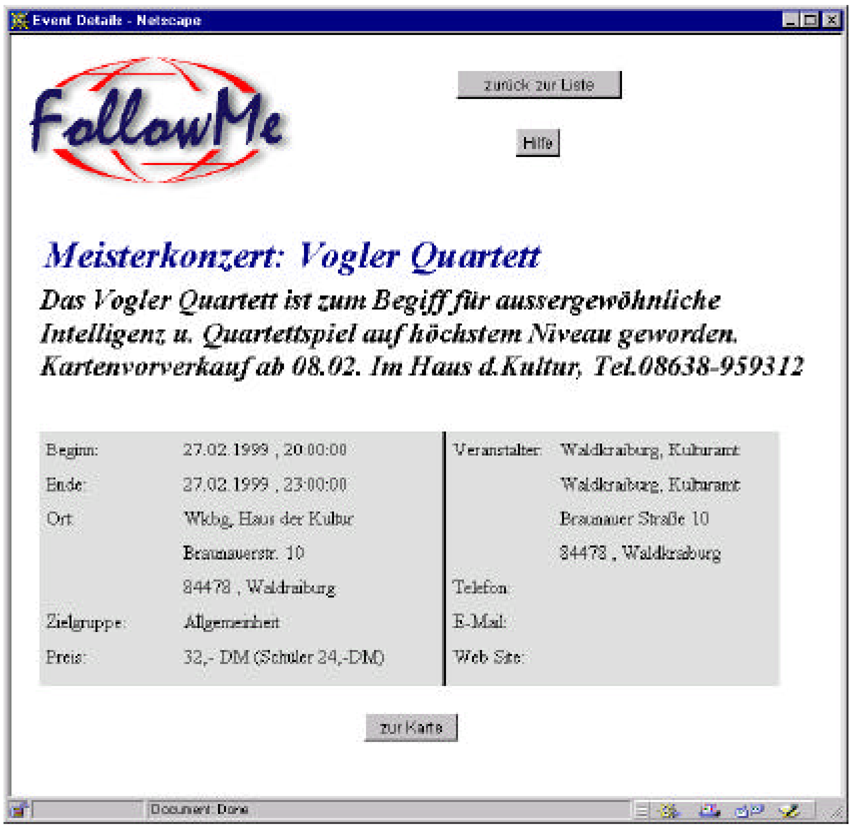

Report - Netscape
File Edit View Go Communicator Help
FollowMe
Veranstaltungsinformationen
F·A·S·T
Hilfe
Document: Done

Ergebnisse Ihrer Anfrage: hgs1 test3

Titel	Beginn	Ende	Ort	Preis	Zielgruppe	Details
Jahreshauptversammlung der Männerkongregation	21.02.1999 10:30:00	21.02.1999 14:00:00	Otk, Pfarrheim Obertaufkirchen - 84419 , Obertaufkirchen	nicht bekannt	Mitglieder der Männerkongregation	Details
Pfarrgottesdienst mit Fastenpredigt	21.02.1999 09:30:00	21.02.1999	Pfarrkirche Maria Himmelfahrt Kirchenstr. 9 84544 , Aschau a. Inn	0,00	Katholische Bevölkerung	Details
Skiausflug des Schützenver. Inning a. Holz	27.02.1999 00:00:00	27.02.1999 00:00:00	Inning am Holz - 84439 , Inning am Holz	nicht bekannt	Mitglieder	Details
Hauptversammlung	20.02.1999 19:00:00	20.02.1999	Pichlmeier-Saal Kirchenstr. 7 84544 , Aschau a. Inn		Jagdgenossen	Details

Figure 20: Online reporting

The most important facts on the retrieved data sets are displayed in a table sorted in chronological order. When requesting more details on a specific data set a new window pops up displaying additional information (event organiser, web-link) along with a button connecting to the geographical information system. Pressing this button will bring up a map displaying the location of the event.



Meisterkonzert: Vogler Quartett

Das Vogler Quartett ist zum Begiff für aussergewöhnliche Intelligenz u. Quartettspiel auf höchstem Niveau geworden. Kartenvorverkauf ab 08.02. Im Haus d.Kultur, Tel.08638-959312

Beginn:	27.02.1999 , 20:00:00	Veranstalter:	Waldkraiburg, Kulturamt
Ende:	27.02.1999 , 23:00:00		Waldkraiburg, Kulturamt
Ort:	Wkbg, Haus der Kultur		Braunauer Straße 10
	Braunauerstr. 10		84478 , Waldkraiburg
	84478 , Waldraiburg	Telefon:	
Zielgruppe:	Allgemeinheit	E-Mail:	
Preis:	32,- DM (Schüler 24,-DM)	Web Site:	

Figure 21: Detailed information on events

The domain of share value information:

The stock portfolio management system provides a constant feed of share values for company stocks traded at stock exchanges for which value providers have been registered.

The model of distribution of system components and actors slightly differs from the one used in the event notification system. Since share value information is publicly accessible, there are no content providers that actively want to advertise specific pieces of information and thus register themselves at a content provider directory service hosted by a service provider. Instead the service provider itself picks suitable publicly accessible share value providers and currency converters to feed share values to portfolio management task agents. To reduce network traffic by using caching and request bundling techniques, the service provider components in the stock portfolio management system are distributed among several hosts as outlined in figure 22 (see design documents for more details).

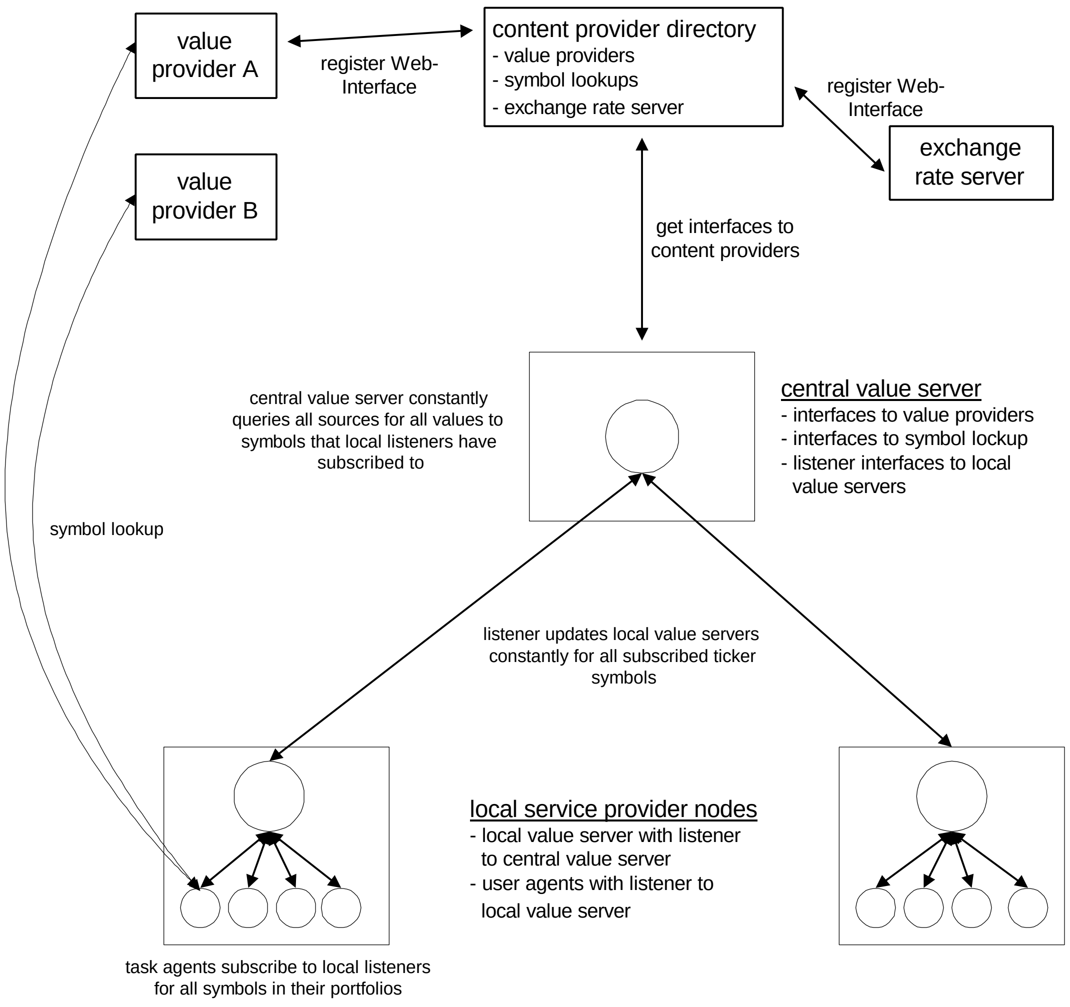

value provider A
register Web-Interface
content provider directory
- value providers
- symbol lookups
- exchange rate server
register Web-Interface
exchange rate server
value provider B
get interfaces to content providers
central value server constantly queries all sources for all values to symbols that local listeners have subscribed to
central value server
- interfaces to value providers
- interfaces to symbol lockup
- listener interfaces to local value servers
symbol lookup
listener updates local value servers constantly for all subscribed ticker symbols
local service provider nodes
- local value server with listener to central value server
- user agents with listener to local value server
task agents subscribe to local listeners for all symbols in their portfolios

Figure 22: Component model in stock portfolio management system

Thus actors in this system are:

- providers of share values feeds, foreign currency converters and stock symbol lookups (content providers),
- a central service provider hosting a central value server and a content provider directory service (additionally the central service provider provides a package composed of local value server modules and user customizable task agents specialized in maintaining a users stock portfolios; this package can be obtained by local service providers; see below for details)
- local service providers hosting local value servers and providing access to the stock portfolio management task agents,
- people dealing with stocks and thus use the system to maintain their stock portfolios.

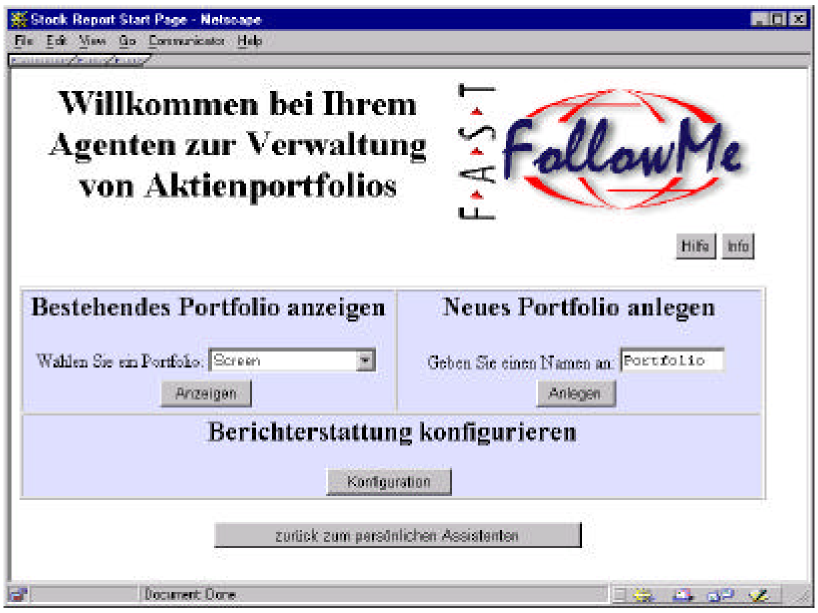
In the current implementation of the stock portfolio management system three types of publicly accessible content providers register interfaces at the content provider directory service hosted by the central service provider: share value providers, currency exchange rate servers and stock symbol lookup servers. These interfaces are registered by type using the concepts of the Agent Framework trader (see FollowMe Agent Framework documentation). Thus other components may retrieve lists of interfaces of specific type when performing specific tasks (e.g. convert currencies, retrieve share values or connect to a symbol lookup interface). Specific types of interfaces are registered with additional information (service properties). Share value providers, in example, register the business hours of the stock exchanges they service. This enables other components to only query for share values during stock exchange trading hours (share values do not change when there is no trading). New provider interfaces of existing type could be integrated without changes to other components. New interface types could be added to be used by newer versions of service provider components with extended functionality (e.g. add providers of financial news services).

Task agents in the domain of share value information enable their users to maintain an arbitrary number of stock portfolios, deposit accounts and transaction logs to trace transactions (buying and selling shares) between portfolios and deposit accounts. All share values retrieved from the service provider are automatically converted to the users local currency. Users may periodically receive reports on the status of their portfolios and deposit accounts via e-mail or fax or review their portfolios online via Web-browser. They may attach limits to each share position they hold. Whenever such a limit is crossed the agent sends an instant message to the user either via e-mail, fax or mobile phone (SMS).

To reduce network traffic service provider components have been split into central and local components (relative to the location of the task agents – see figure 22). Local providers bundle share value feed request from all task agents running on their host. Thus share values of a given stock symbol are only transferred once across the network between the central provider and a local provider even if more than one agent requests the respective value.

User interfaces of the stock portfolio management task agent:

The first screen send to a user by a portfolio management task agent offers the user to either create a new stock portfolio, to maintain an existing portfolio (selectable via a list-box) or to specify a reporting schedule to define when and how (via which device) the agent should deliver reports.


Stock Report Start Page - Netscape
File Edit View Go Communicator Help
F-A-S-T
FollowMe
Hilfe
Info
Anzeigen
Anlegen
Konfiguration
zurück zum persönlichen Assistenten
Document: Done

Willkommen bei Ihrem Agenten zur Verwaltung von Aktienportfolios

Bestehendes Portfolio anzeigen

Wählen Sie ein Portfolio: Screen

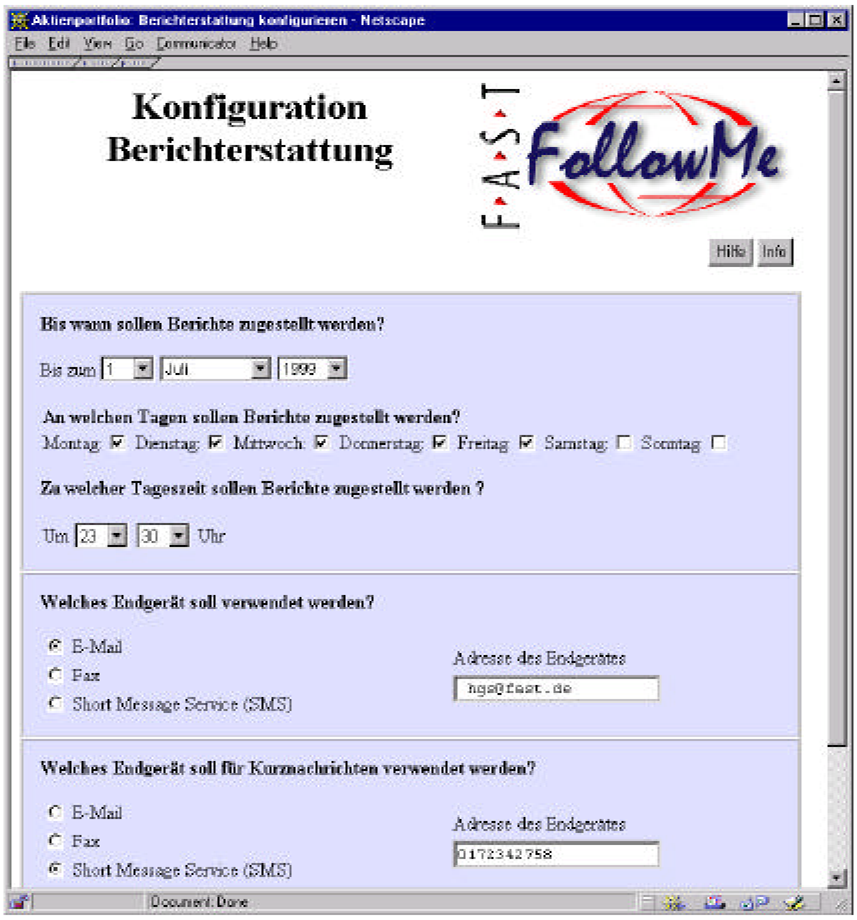
Neues Portfolio anlegen

Geben Sie einen Namen an: Portfolio

Berichterstattung konfigurieren

Figure 23: Portfolio management agent main dialog

The configuration dialog for defining a task agent reporting schedule is similar to one used in the event notification system. In addition to defining a schedule and a device for delivering standard reports on portfolios and deposit accounts the user may specify the device to which the agent should send notification on exceeded share value limits.


Aktienportfolio: Berichterstattung konfigurieren - Netscape
File Edit View Go Communicator Help
F·A·S·T
FollowMe
Hilfe
Info
Document: Done

Konfiguration Berichterstattung

Bis wann sollen Berichte zugestellt werden?

Bis zum 1 Juli 1999

An welchen Tagen sollen Berichte zugestellt werden?

Montag ☑ Dienstag ☑ Mittwoch ☑ Donnerstag ☑ Freitag ☑ Samstag ☐ Sonntag ☐

Zu welcher Tageszeit sollen Berichte zugestellt werden ?

Um 23 30 Uhr

Welches Endgerät soll verwendet werden?

[✓ E-Mail] Fax Short Message Service (SMS)

Adresse des Endgerätes: hgs@fast.de

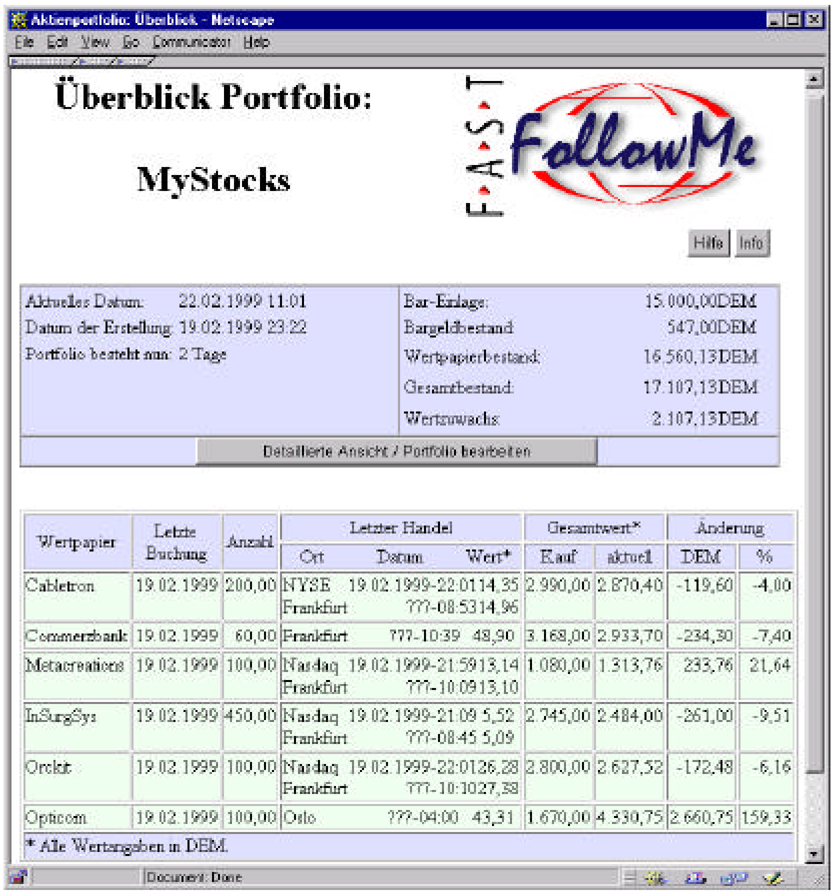
Welches Endgerät soll für Kurznachrichten verwendet werden?

E-Mail Fax [✓ Short Message Service (SMS)]

Adresse des Endgerätes: 0172342758

Figure 24: Defining a reporting schedule

After selecting a portfolio name from the list-box in the main dialog, the agent displays an overview on the portfolio contents and the status of the deposit account. From here the user may choose to go on to a more detailed view (see below), to return to the main page or to reconnect to the personal assistant.


Aktienportfolio: Überblick - Netscape
File Edit View Go Communicator Help
F·A·S·T
FollowMe
Hilfe
Info
Detaillierte Ansicht / Portfolio bearbeiten
Document: Done

Überblick Portfolio:

MyStocks

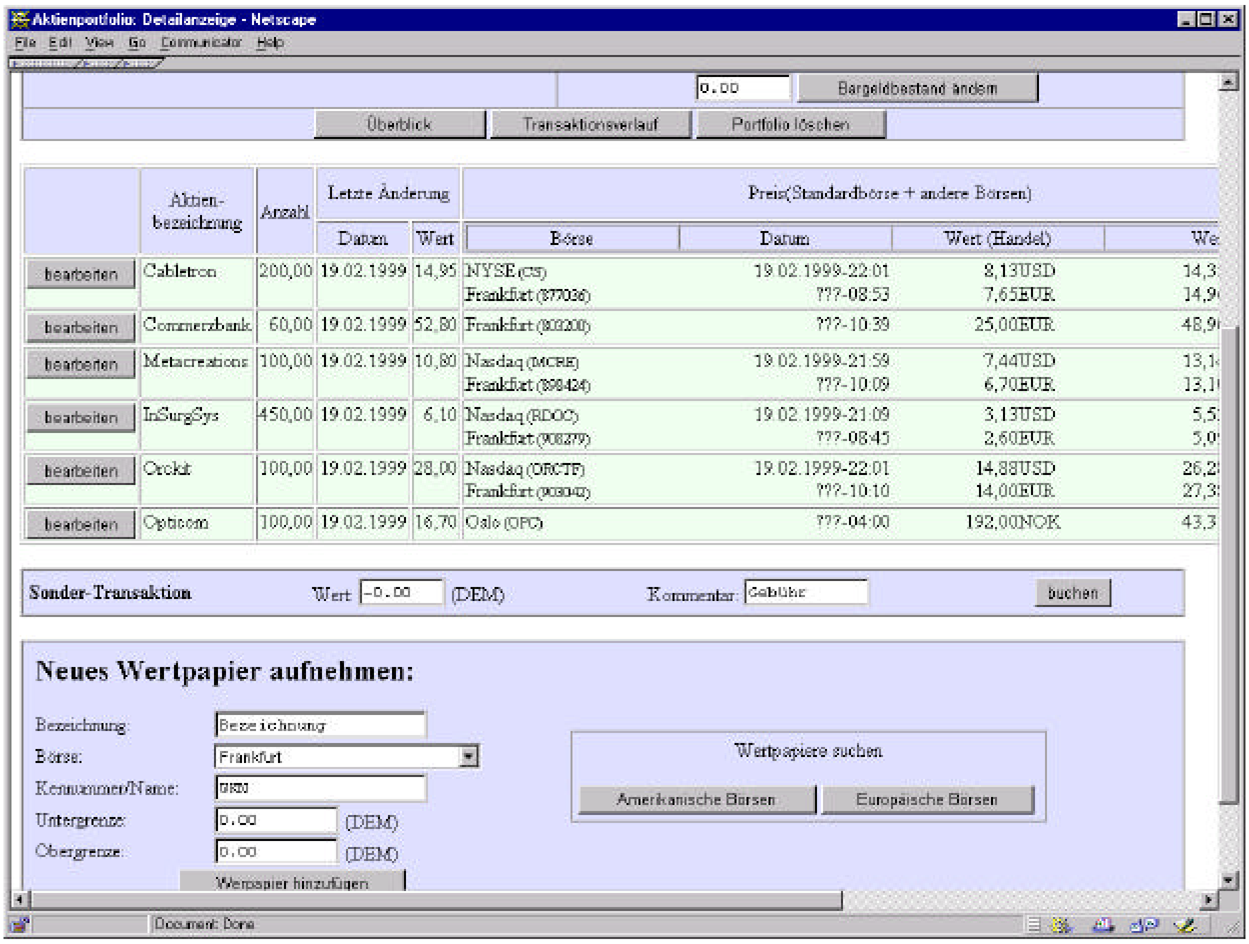
Aktuelles Datum:	22.02.1999 11:01	Bar-Einlage:	15.000,00DEM
Datum der Erstellung:	19.02.1999 23:22	Bargeldbestand	547,00DEM
Portfolio besteht nun:	2 Tage	Wertpapierbestand:	16.560,13DEM
		Gesamtbestand:	17.107,13DEM
		Wertzuwachs:	2.107,13DEM

Wertpapier	Letzte Buchung	Anzahl	Letzter Handel			Gesamtwert*		Änderung	
			Ort	Datum	Wert*	Kauf	aktuell	DEM	%
Cabletron	19.02.1999	200,00	NYSE Frankfurt	19.02.1999-22:01 ???-08:53	14,35 14,96	2.990,00	2.870,40	-119,60	-4,00
Commerzbank	19.02.1999	60,00	Frankfurt	???-10:39	48,90	3.168,00	2.933,70	-234,30	-7,40
Metacreations	19.02.1999	100,00	Nasdaq Frankfurt	19.02.1999-21:59 ???-10:09	13,14 13,10	1.080,00	1.313,76	233,76	21,64
InSurgSys	19.02.1999	450,00	Nasdaq Frankfurt	19.02.1999-21:09 ???-08:45	5,52 5,09	2.745,00	2.484,00	-261,00	-9,51
Orckit	19.02.1999	100,00	Nasdaq Frankfurt	19.02.1999-22:01 ???-10:10	26,28 27,38	2.800,00	2.627,52	-172,48	-6,16
Opticom	19.02.1999	100,00	Oslo	???-04:00	43,31	1.670,00	4.330,75	2.660,75	159,33

* Alle Wertangaben in DEM.

Figure 25: Portfolio overview

The detailed portfolio view (figure 26) allows the user to actually maintain the portfolio and the deposit account (the detailed view is also displayed when a new portfolio is created). The user may deposit or withdraw money from the deposit account, protocol special transactions (e.g. share transaction fees), review the transaction history (figure 27) or buy or sell shares. To assist users in buying shares the agent offers links to stock symbol lookup interfaces. The user may attach more than one value feed to a given stock position in the portfolio (e.g. value of Oracle shares as traded in New York and Frankfurt). Limits to stock positions may be defined or adjusted anytime.


Aktienportfolio: Detailanzeige - Netscape
File Edit View Go Communicator Help
0.00
Bargeldbestand ändern
Überblick
Transaktionsverlauf
Portfolio löschen

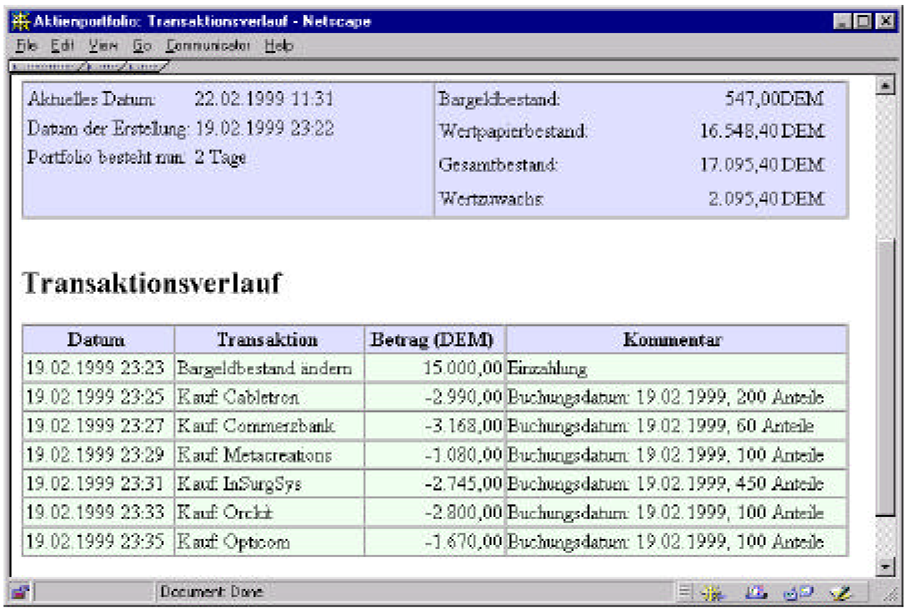
	Aktien-bezeichnung	Anzahl	Letzte Änderung		Preis(Standardbörse + andere Börsen)			
			Datum	Wert	Börse	Datum	Wert (Handel)	We
bearbeiten	Cabletron	200,00	19.02.1999	14,95	NYSE (CS) Frankfurt (877036)	19.02.1999-22:01 ???-08:53	8,13USD 7,65EUR	14,3 14,9
bearbeiten	Commerzbank	60,00	19.02.1999	52,80	Frankfurt (803200)	???-10:39	25,00EUR	48,9
bearbeiten	Metacreations	100,00	19.02.1999	10,80	Nasdaq (MCRE) Frankfurt (898424)	19.02.1999-21:59 ???-10:09	7,44USD 6,70EUR	13,1 13,1
bearbeiten	InSurgSys	450,00	19.02.1999	6,10	Nasdaq (RDOC) Frankfurt (908279)	19.02.1999-21:09 ???-08:45	3,13USD 2,60EUR	5,5 5,0
bearbeiten	Orckit	100,00	19.02.1999	28,00	Nasdaq (ORCTF) Frankfurt (903042)	19.02.1999-22:01 ???-10:10	14,88USD 14,00EUR	26,2 27,3
bearbeiten	Opticom	100,00	19.02.1999	16,70	Oslo (OPC)	???-04:00	192,00NOK	43,3

Sonder-Transaktion Wert -0.00 (DEM) Kommentar: Gebühr buchen

Neues Wertpapier aufnehmen:

Bezeichnung:	Bezeichnung
Börse:	Frankfurt
Kennnummer/Name:	WKN
Untergrenze:	0.00 (DEM)
Obergrenze:	0.00 (DEM)

Figure 26: Detailed portfolio view


Aktienportfolio: Transaktionsverlauf - Netscape
File Edit View Go Communicator Help

Aktuelles Datum:	22.02.1999 11:31	Bargeldbestand:	547,00DEM
Datum der Erstellung:	19.02.1999 23:22	Wertpapierbestand:	16.548,40 DEM
Portfolio besteht nun:	2 Tage	Gesamtbestand:	17.095,40 DEM
		Wertzuwachs:	2.095,40 DEM

Transaktionsverlauf

Datum	Transaktion	Betrag (DEM)	Kommentar
19.02.1999 23:23	Bargeldbestand ändern	15.000,00	Einzahlung
19.02.1999 23:25	Kauf Cabletron	-2.990,00	Buchungsdatum: 19.02.1999, 200 Anteile
19.02.1999 23:27	Kauf Commerzbank	-3.168,00	Buchungsdatum: 19.02.1999, 60 Anteile
19.02.1999 23:29	Kauf Metacreations	-1.080,00	Buchungsdatum: 19.02.1999, 100 Anteile
19.02.1999 23:31	Kauf InSurgSys	-2.745,00	Buchungsdatum: 19.02.1999, 450 Anteile
19.02.1999 23:33	Kauf Orckit	-2.800,00	Buchungsdatum: 19.02.1999, 100 Anteile
19.02.1999 23:35	Kauf Opticom	-1.670,00	Buchungsdatum: 19.02.1999, 100 Anteile

Figure 27: Transaction history

3 Pilot Deployment and Evaluation

In this section we report on the deployment of the pilot applications in the test field and on the evaluation methodology and results.

Right from the beginning of the project five pre-selected Bavaria Online nodes were involved in planning, testing and deployment of the pilot applications (a sixth node only recently joined the trial phase). Early releases of the software were made available to technicians at the Bürgernetze. Their bug reports and feedback on system usability served as input for WP I software developers to enhance functionality and user interfaces. This iterative process of testing and programming led to the release of version 1.0 of the event notification application in mid January 99. The first version of the stock portfolio application was finalised and made available to the public at the end of February 99. This means that by the time this document has been compiled, the event notification system has been accessible to Bürgernetz users for 6 weeks. The stock portfolio application has just been published so there is no real user feedback available yet.

3.1 Application Deployment

The Bürgernetz organisations operate on the basis of voluntary co-workers who maintain the Bürgernetz systems during their spare time. Therefore and for maintenance reasons the strategy for deployment of system components of the first releases of the pilot applications was to host as many of the components as possible at FAST.

Event notification system:

For the event notification system all service provider components, all directory services, the third party geographical information system, all user agents and device gateways were therefore deployed on hosts in the FAST internal network. The Bürgernetze currently only act as content providers and thus host event databases and database interfaces.

To assists the Bürgernetze in setting up these components, we provided them with a Microsoft Access database with a web-based front-end for registering event advertisements (implemented using Microsoft active server pages). We also provided them with a Java-based database interface.

Due to the above mentioned organisational structure of the Bürgernetze, the deployment of these components did progress slower than expected, so that at present the status of the deployment is as follows:

- Three nodes are connected and registered with the content provider directory service using the example Microsoft Access database. Two of these use the Java database connector provided by FAST. The other node implemented their own interface in C++.
- The remaining three nodes already host event databases but need to adjust or wrap their data models to meet the requirements of our system.
- One additional content provider for specific events (cinemas in Munich) is currently under preparation.

The three already connected databases currently provide information on approximately 700 regional events taking place between 15th of February and 30th of April 1999. The Bürgernetze have established a process for constantly feeding new information into their databases.

Stock portfolio management system:

Due to the experiences made during deployment of the event notification system components we decided to deploy all components of the first release of the stock portfolio management application on a set of hosts within the FAST-internal network. Thus it had been possible to grant access to the application to all Bürgernetz nodes (not only the six nodes selected for the pilot trials). Following this strategy we expect to obtain feedback from a large number of users of this application within the near future.

Components currently hosted by FAST are distributed over several machines thus proving that physical location of components is not an issue at application level (see FollowMe architecture concepts documented in WP A results). Therefore other components such as user agents or device gateways could be moved to and deployed at Bürgernetz nodes anytime a Bürgernetz volunteers to host them (see chapter 6: ‘Way forward’ for information on future plans for deployment of components at Bürgernetz hosts).

3.2 Evaluation Methodology

To prepare the evaluation of the results of the pilot trials we monitored the usage of the two pilot applications and collected direct feedback from application users. Both activities are ongoing. This report presents the intermediate results which line out basic tendencies about the degree of user acceptance of the new services. However, the size of the samples of both the usage monitoring and the feedback from the questionnaire are not large enough to provide a statistical sound basis to precisely measure the degree of acceptance of the new services by the Bürgernetz users.

Usage monitoring:

In order to trace usage of the pilot applications we continuously log the following set of system events together with their respective timestamps:

- Creation of a new personal assistant,
- creation of or connection to a task agent of a specific type (event notification or stock portfolio),
- delivery of scheduled reports by a specific type of task agent to a specific type of device (e.g. event notification agents sends e-mail report),
- delivery of event triggered messages (only in share value domain when limits are exceeded).

All events are logged in combination with the user defined name of the corresponding personal assistant. Thus we can e.g. measure how often users re-visited their agents during the evaluation period or whether users usually use only one of the offered task agents.

All logged events created by our test accounts were removed from the log files. Thus all statistics only reflect activities of real users.

User feedback:

In addition to the usage monitoring we set up a web-based questionnaire to collect direct feedback from the pilot application users. The questionnaire was publicly announced at the web-sites of the participating Bürgernetze four weeks after the event notification system had been opened to the public. Additionally we e-mailed the owners of all personal assistants using e-mail addresses retrieved from the users' personal profiles and pointed them to the web-based questionnaire.

The questionnaire basically consists of two parts. The first part is designed to figure out the users' general interest in the features of our application framework. We asked the user to value the relevance of automated information retrieval, personalization and scheduled reporting via various devices. We used a four stages ranking with values 'important', 'interesting but not a major issue', 'less important' and 'unimportant'.

The second part of the questionnaire relates more specifically to the features of the event notification system. We asked the users whether the service assists them in obtaining information on regional social events and in how far they started using it in addition to other information sources like daily newspapers and other regional press publications. Moreover we asked the users to recommend enhancements that would make the service more attractive. We wanted to know whether the new service adds value to their local Bürgernetz and whether their Bürgernetz should continue spending resources for further enhancement of the service or for development of new services based on the same application framework. Again we used a three to four stages ranking in combination with text fields for suggestions on enhancements and new types of services.

3.3 Evaluation Results

Here we present the statistics derived from the log-files and the results of the user questionnaires.

Usage monitoring

Within the evaluation period of 6 weeks a total of 60 personal assistants has been created. After the usual peak personal assistant creation events due to the public announcement, the number of new users did steadily increase during the whole evaluation period (see figure 28).

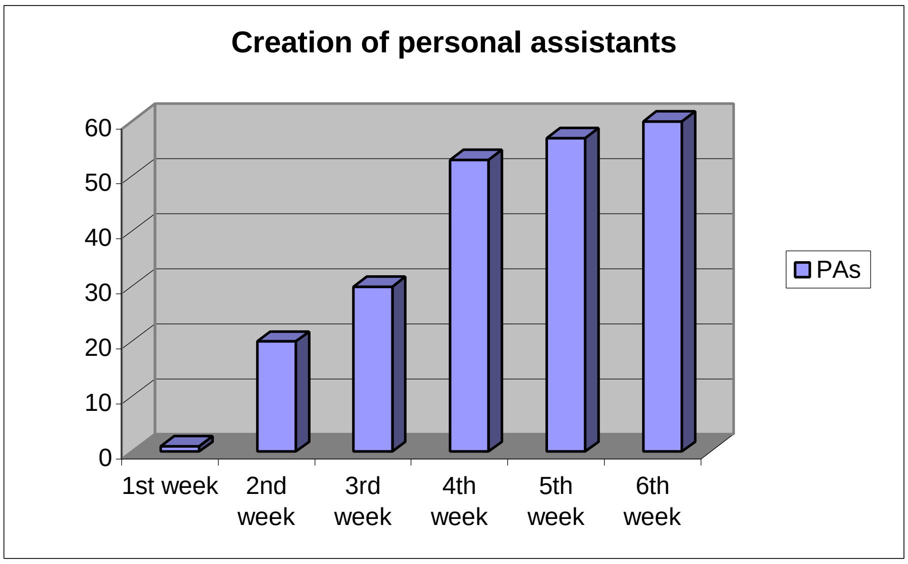


Figure 28: Creation of PAs over time

When creating a new personal assistant or connecting to an existing one, the user pre-selects the service he intends to use (see user interface descriptions in chapter 2). At present this may either be the event notification or the portfolio management service. After creating or connecting to a personal assistant, the user is automatically linked to his task agent of respective type (if for the first time, a respective task agent is created). Therefore, since portfolio task agents had not yet been available during the evaluation period, creating or connecting to a personal assistant has been equivalent to creating or connecting to an event notification task agent. The set of observables during the evaluation period is therefore reduced to the following three system events:

- Creation of a personal assistant and an event notification task agent (labeled 'createAgent'),
- login to a previously created personal assistant and task agent (re-visits; labeled 'login'),
- event notification task agent sending report to a specific device (labeled 'sendReport', 'device').

Due to the basic characteristics of the application framework and the implemented services, there are several aspects of user behavior that cannot be directly measured. The task agents are designed to gather information autonomously and deliver reports according to user defined schedules. Thus a user who only logged on to the system once but received reports from his agents periodically may either really use the information delivered to him or may have decided not to use the system and ignore the messages. With this in mind we now start evaluating the log-files:

Of the 60 registered personal assistants 41 (68 %) delivered reports to their owners. For the following interpretations we subtracted the remaining 19 inactive personal assistants/task agents from the total of 60 so that the following percentage calculations relate to a total of 41 active personal assistants/task agents.

The most important indicator to measure in how far users actively use the new service(s) is the number of re-visits (see figure 29). 16 of the 41 owners of active agents (which is 39 %) re-contacted the system to either define additional information gathering tasks (add queries), to re-adjust the tasks they defined when they created their agent (re-configure queries), to stop previously defined tasks (delete queries) or to review results of information gathering tasks via their web-browser. By explicitly tracing the activities of these agents we observed that only two of the 16 users with re-visits logged in to stop their agent from reporting (probably because they didn't find the results delivered by their agent useful).

users with re-visits	total #	percentage out of 41
0 additional logins	25	61 %
1 additional login	9	22 %
2 additional logins	4	10 %
3 additional logins	1	2 %
> 3 additional logins	2	5 %

Figure 29: Users with re-visits (additional logins)

Finally we take look at the amount of reports delivered by the agents and the device types used for delivery. During the 6 weeks of usage monitoring a total of 382 reports had been send out by the event notification agents. Related to the 41 active agents this is 9,3 reports per agents/user during the 6 weeks or 1,6 reports per agent/user per week, but nearly half of the users (19 out of 41) received less than 6 reports, that is less than one per week (see figure 30).

# of reports sent	holds for # of users	percentage out of 41
1-5	19	46 %
6-10	9	22 %
11-20	6	15 %
> 21	7	17 %

Figure 30: Number of reports received by the users

Of the 382 reports only 7 were sent to fax devices. The remaining 375 were sent via e-mail (see figures 31 and 32).

reports sent	total #	percentage
total #	382	100 %
sent via e-mail	375	98 %
sent via fax	7	2 %

Figure 31: Devices used to deliver reports

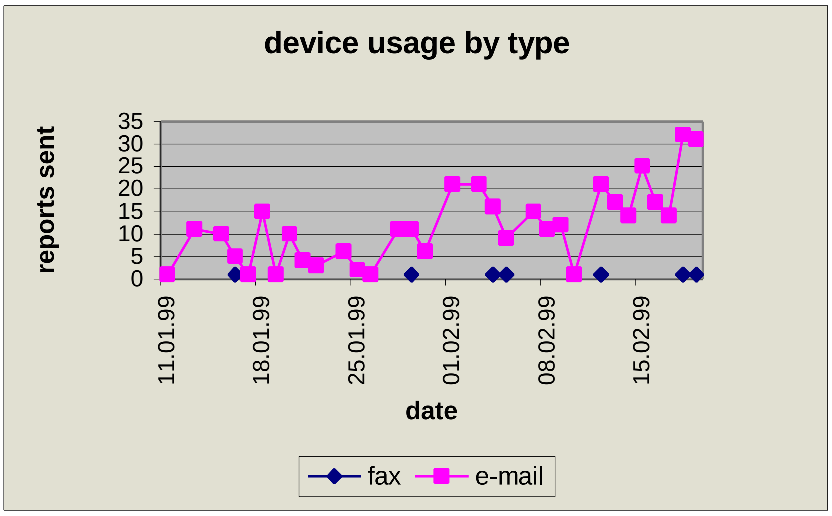


Figure 32: Reports delivered per day

Delivery to fax devices was only used by two users. One of them used fax exclusively, the other one also received e-mail reports. The obvious conclusion is that internet users have very little need for report delivery to fax devices, which is consistent with the feedback we received from users via our web-based questionnaire (see below). This result is not surprising since the owners of the personal assistants are presumably those Bürgernetz users that are the most advanced in using the internet and electronic communication. Fax delivery is likely to be more interesting for users that do not frequently connect to the internet or within application domains were instant notification (even when the user is not online and thus not able to get notified of incoming e-mails) is more relevant.

As previously mentioned, 19 of the 60 personal assistants never sent a single report to their owners even after multiple user log-ins (see figure 33). We could think of the following reasons for this:

- The users did not manage to save a valid query, or
- the users did not want to save a valid query, or
- the users saved a valid query but created their agent during the final week of evaluation and the agent was not yet triggered to send a report, or
- the users saved a valid query but deleted it before it would have been executed for the first time.

The third reason may hold for only two of these users (due to personal assistant creation date). The most likely of these reasons is the first one. Users who managed to correctly configure their agent would probably at least wait for the agent delivering one report before they quit using it. The most obvious conclusion out of this is that the user interfaces are not yet user-friendly enough and the online help files do not provide enough guidance. This is also consistent with the results of the user questionnaires (see below).

no sendReport events	total #	percentage
total #	19	32 % of 60
0 additional login	13	68 % of 19
1 additional login	4	21 % of 19
2 additional logins	2	11 % of 19

Figure 33: Agents that did not deliver reports

User feedback

22 users returned the web-based questionnaire published four weeks after the initial publishing of the event notification system. Compared to the number of personal assistants created (60; see above) this means that approximately 37% of the system users did provided feedback by filling out the questionnaire. This is quite a high rate of feedback which, as a first result, demonstrates the general interest of the users in the new application. The results listed below outline basic tendencies about the degree of user acceptance of the new service. These tendencies will serve as major input for decisions on future development of the application.

The detailed results of the questionnaire feedback are summarized in the following tables:

Question 1: What are the your most important reasons to use the service?

Question 1.1: Autonomous information gathering by software agents: You don't have to search information sources yourself but instruct your agent to do so on your behalf.

	# of answers	percentage
total # of answers	22	100 %
important	13	59 %
interesting	6	27 %
less important	1	5 %
not important	2	9 %

Question 1.2: Personalization: You can customize your agent to search for events that are of personal interest for you.

	# of answers	percentage
total # of answers	22	100 %
important	13	59 %
interesting	7	32 %
less important	2	9 %
not important	0	0 %

Question 1.3: Automated reporting: You automatically receive scheduled reports from your agent.

	# of answers	percentage
total # of answers	22	100 %
important	14	63 %
interesting	3	14 %
less important	3	14 %
not important	2	9 %

Question 1.4: Report delivery to various device types: You may instruct your agent to either deliver reports to fax or e-mail. In addition you may use your browser to review the reports.

Question 1.4.1: Overall relevance of having the ability to choose from various device types.

	# of answers	percentage
total # of answers	21	100 %
important – all device types should be supported	8	38 %
interesting	8	38 %
less important – one device type is sufficient	4	19 %
not important – reviewing in browser in sufficient	1	5 %

Question 1.4.2: Relevance of device type fax.

	# of answers	percentage
total # of answers	21	100 %
important	5	24 %
less important	8	38 %
not important	6	29 %
I don’t have a fax	2	9 %

Question 1.4.3: Relevance of device type e-mail.

	# of answers	percentage
total # of answers	20	100 %
important	18	90 %
less important	1	5 %
not important	1	5 %

Question 1.4.4: Relevance of device type web-browser.

	# of answers	percentage
total # of answers	21	100 %
important	15	71 %
less important	5	24 %
not important	1	5 %

The feedback from this first block of questions shows that users are equally interested in agent autonomy, personalization and automated reporting. The users’ need for availability of various device types for report delivery is not as high as we expected due to the results of the pre-survey. As is also reflected by the results of the usage monitoring, delivery to fax devices is not a major issue.

Question 2: Does the new service assist you in your search for information on social events taking place in your region?

	# of answers	percentage
total # of answers	21	100 %
yes	9	43 %
a little	10	48 %
no	2	9 %

Question 3: Did the introduction of the new service change your habits in information search? Which sources did/do you mainly use to search for information before/after the new FollowMe service has been introduced?

	before FM		after FM	
daily newspaper	#	%	#	%
total # of answers	22	100 %	21	100 %
very important	11	50 %	7	33 %
important	4	18 %	7	33 %
less important	4	18 %	6	29 %
not important	3	14 %	1	5 %

	before FM		after FM	
regional brochures	#	%	#	%
total # of answers	20	100 %	19	100 %
very important	1	5 %	1	5 %
important	6	30 %	8	42 %
less important	12	60 %	8	42 %
not important	1	5 %	2	11 %

	before FM		after FM	
municipality news	#	%	#	%
total # of answers	21	100 %	21	100 %
very important	4	19 %	2	9 %
important	5	24 %	6	29 %
less important	8	38 %	7	33 %
not important	4	19 %	6	29 %

	before FM		after FM	
online internet queries	#	%	#	%
total # of answers	21	100 %	20	100 %
very important	4	19 %	5	25 %
important	8	38 %	6	30 %
less important	6	29 %	6	30 %
not important	3	14 %	3	15 %

FollowMe agent service	# of answers	percentage
total # of answers	22	100 %
very important	5	23 %
important	11	50 %
less important	5	23 %
not important	1	4 %

Question 4: What needs to be improved to make the service more attractive?

	# of answers	percentage
total # of answers	21	100 %
user interfaces	7	33 %
more data (content providers)	14	67 %

Here the users had the option to provide other recommendations in a text box. The most frequent recommendations where to *enhance the layout of the e-mail reports* and to *simplify the category schemes* used to defined the type of events an agent should look for.

According to the feedback to question 3 the new service already slightly reduced the users' need for other information sources. However, as derived from feedback to question 2 and 3, users do not yet use the service as their major source for information on regional social events. This is probably due to the fact that up to now there is a lack of available content (see feedback to question 4). The participating Bürgernetze need to feed more data into their databases and more Bürgernetze need to join the system to provide a wider area of coverage.

Question 5.1: Does the new service make your Bürgernetz more attractive?

	# of answers	percentage
total # of answers	22	100 %
yes	17	77 %
no	5	23 %

Question 5.2: Would the service become even more attractive when all Bürgernetze in Bavaria would provide data on events within their region?

	# of answers	percentage
total # of answers	22	100 %
yes	21	95 %
no	1	5 %

Question 6: Ongoing development of agent based services.

Question 6.1: Should your Bürgernetz invest further resources to improve the service?

	# of answers	percentage
total # of answers	22	100 %
yes	21	95 %
no	1	5 %

Question 6.2: Should your Bürgernetz invest in developing other agent based services?

	# of answers	percentage
total # of answers	22	100 %
yes	19	86 %
no	3	14 %

Here the users were asked to provide recommendations for other application domains. They proposed to develop services for *travel assistance*, *flea markets*, *markets for cars*, *apartment rental*, and *computer equipment*, *call for tenders*, *mediation of e-mail contacts* to people with similar interests and *news services for politics, stocks*, *sports* and *economics*.

Feedback on questions 5 and 6 clearly indicates that users welcome the new service and its basic concepts. They know and accept that the current release is still a prototype but believe that it will become more user friendly and more useful within future releases. Since they are well aware of the limited resources of their Bürgernetz operators, especially the positive results of question 6 indicate that the users have confidence in the new type of services. The recommendations for new types of agent based services show that users have well understood the basic concepts of the application framework and its applicability to all domains that involve information brokerage, information gathering and information filtering on the basis of a user’s personal interests.

4 Additional Lab-Tests

4.1 Pilot application implementations and the Agent Framework

Development of the modules that make up the component-ware tool box implementing the FollowMe framework has been an ongoing process until the very end of the project. Due to the narrow time schedule of the project the technical work packages applied a gradual release plan with extending functionality. In order to release the pilot applications and to start the field tests on schedule, the pilot application work package adopted a two tiered implementation strategy:

- an implementation of the pilot application targeted for the field tests, that used the available stabilized components of the other work packages and implemented simplified versions of the missing functionality
- lab tests that exploited the full power of the gradually developing technical framework

Pilot application releases used for the field tests are built on the core Agent Framework components of personal profiles, agent traders and diary alarms. They are integrated with the user access component to communicate with various device gateways. Information space functionality is used to ensure the capability to restart after system failure.

4.2 Lab tests with latest Agent Framework releases

In addition to maintaining the releases used for the field tests, pilot application developers started a series of additional (still ongoing) lab tests with the latest release (version 1.4) of the Agent Framework. The core objective of these lab tests is to integrate the domain specific parts (Java libraries) of the pilot applications into the Agent Framework’s agent creation and maintenance environment which involves the following tasks:

- wrapping of application specific libraries in scripted agent missions,
- registering these missions with the Agent Framework mission trader,
- implementation of XML/XSL-based user interfaces to the personal assistant factory and the personal assistants (including launch options for missions registered with the trader),

- passing ‘contact task agent’ request from the mission to the application specific libraries so the user can interact with the task agents.

After this integration process has been successfully completed, mobility features that come with the Agent Framework and allow to move personal assistants and task agents will be tested. The tests will conclude with a robustness and scalability analysis of the Agent Framework.

5 Validation of Success Criteria

The role of the pilot application work packages within the FollowMe project was to measure the applicability of both the FollowMe architectural concepts and their implementations by the technical work packages. We therefore now summarize work package I success criteria and discuss in how far these criteria have been met by work package I results.

The work package I pilot applications are considered successful if:

1. they match the application demands of the users in the test field (Bavaria Online users),
2. they were designed using the FollowMe architecture framework (work package A),
3. their implementations build on modules/libraries developed in the technical WPs, as there are:
   - mobility of code (while preserving state) – WP B,
   - security of private data – WP B,
   - persistent storage of Java objects – WP C,
   - agent framework concepts – WP DEF:
     - de-coupling of user specific and service specific components (personal assistants and task agents),
     - autonomous agents acting against services (while user may be offline),
     - personalized agent missions,
     - scripting model for agent missions,
   - mobility of users – WP H,
   - monitoring and data mining facilities – WP G,
4. feedback from pilot applications design and development to technical work packages assisted in enhancing quality of the latter (e.g. with respect to scalability and robustness),
5. there is evidence that the FollowMe architectural framework and the libraries provided by the technical work packages simplified application design and development (compared to other engineering frameworks).

Success criteria 1: applications meet user demands

As outlined in chapter 3 first feedback from users of the event notification system showed that these users are highly interested in the concepts of the application framework (automated, user customizable information retrieval and scheduled report delivery). The value-added features of the event notification system are considered useful for satisfying the users' every day information needs and the users are looking forward to access more information services of the same type. The major complaints about the first releases of the pilot applications were that user interfaces need to be enhanced and that content providers need to offer more data. This criticism is not related to the basic features and core characteristics of the applications. Thus we can conclude that the work package I pilot applications provide the Bavaria Online users with access to a new type of services with characteristics considered useful by most of the users and thus meet the users' application demands.

Success criteria 2: FollowMe architecture concepts were used to design the applications

The application framework developed in work package I (see chapter 1) is designed to build information services in dynamically changing distributed scenarios: Information sources (content providers), service components (service providers, task agent factories) and user specific components (personal assistants, user profiles) might be added to the system at runtime regardless of their physical location.

Location transparency at application component level is a basic concept of the protocol layer (part of work package B) of the FollowMe architecture. The concept of de-coupling components related to users, service providers and content providers is directly inherited from the agent framework layer (work packages D, E and F) of the FollowMe architecture. The same holds for the concepts of autonomous, user customizable task executors that act against service interfaces while their owner/user might be off-line. In fact the concepts and the early design of the application framework served as important input for the agent framework architecture.

Users of work package I applications may receive information via various devices. This concept is derived from the user access part (work package H) of the FollowMe architecture.

Success criteria 3: Work package I used code libraries from FollowMe technical work packages

We now step through all technical work packages and describe in how far (and for what reasons) work package I applications have been implemented using FollowMe code libraries.

Work package B (MOW) provides a protocol layer mobile object system that allows encapsulation of Java objects in clusters that can be moved around a network while preserving state and references to or by other objects/clusters. Although the pilot applications use MOW's models for remote method invocations (instead of e.g. Java RMI) there is no direct need for code mobility. The currently publicly accessible versions of the applications do not require personal assistants or task agents to be mobile. However, code mobility might proof helpful to implement load balancing strategies when the number of Bürgernetz users increases. Prototypic versions allowing mobility of agents are currently tested in the lab (see chapter 4).

Work package C (Information Space) provides persistent storage facilities for any Java objects. This is extensively used in the pilot applications to store and retrieve all kind of objects holding user specific data (e.g. personal profiles, task agent configurations, objects encapsulating results from information retrieval activities). Since personalized components like personal assistants and task agents are usually long lived (task agents may be instructed to continuously gather information and deliver reports for month), reliable persistent storage is a major issue in our applications.

Work package D, E and F provide an agent development framework composed of a number of individual components of which some have been used in the currently publicly available releases of the applications. The pilot applications' personal profiles build on work package E class libraries. To trigger time dependent agent activities work package D's alarm implementation has been used. The agent framework trader has been used to register interfaces to content provider sites along with provider specific interface properties (e.g. information of area of coverage in event notification system – see chapter 2). Due to the fact that a fully operational and complete version of the agent framework became available very late in the project, a full integration of the application specific components into the complete agent framework could only be tested in the lab (see chapter 4 for lab tests and chapter 6 for a description on plans for future releases of FollowMe based information services in the context of Bavaria Online).

Work package G (Service Deployment) provides usage monitoring and load prediction facilities that were not used in work package I pilot applications.

Work package H (User Access) provides gateways to a variety of different I/O-devices. The interfaces to these devices allow other components to receive and send messages in a standardized way using latest XML and XSL specifications. Our pilot applications use these interfaces to communicate with the users. By using the User Access components work package I developers were relieved of having to deal with all kind of different output formats (e.g. fax, e-mail, SMS, html).

Success criteria 4: Feedback from pilot applications design and development helped to enhance quality of technical work packages

During the whole project, workshops and e-mail communication were used to constantly feed back work package I experiences to the developers of the code libraries of the technical work packages to help enhance the functionality and robustness of their products. To name but a few examples of such feedback processes:

- Results of the requirement analysis of work package I were fed back to the developers of the agent framework. Concepts developed as part of the application framework of work package I were integrated into the agent framework architecture.
- The first release of the MOW only allowed to pass messages of a maximum size of 8 kb. We soon discovered that we could not send our XML/XSL based reports from agents to user access components or from agents to the information space with this restriction. The restriction has been removed with release 2.0 of the MOW.
- The pilot application trials in the field test only recently showed some limitations in the number of objects that can be stored in a single store within the information space. Information Space developers then provided a solution to that problem.
- Limitations in the MOW 2.0 release of the information space didn't allow task agents created by the agent framework to restore their state after a system failure. This was a serious problem in our scenarios were agents are usually long-lived and recovery of state after system failure is essential. The problem will be solved with the upcoming release of the Agent Framework that will build on MOW 2.1

Success criteria 5: FollowMe technology simplifies application development

The FollowMe framework is build around a set of independent components that can be used in a variety of different contexts. For the development of the work package I pilot applications the following modules proved to be most valuable from application developers point of view:

1. The User Access modules relieved our application developers of having to deal with a variety of different device types and file formats themselves. By using the User Access modules and device gateways they just had to program their agent components to encapsulate the data in pre-defined XML structures, define report layouts in static XSL files and send these to the User Access along with instructions on which device to send the report to.
2. The Information Space modules provided application developers with easy to use interfaces to persistently store (and retrieve) any Java object. This relieved them from having to deal with object (de-)serialization themselves.

3. Location transparency on application level provided by the underlying MOW protocol layer allowed application developers to distribute objects and components over the network without having to deal with the details of object locations when handling object references.

6 Way Forward

As detailed in previous sections, the deployment of the two pilot application services in the Bürgernetz-environment has only just began. The services have been successfully introduced at three Bürgernetz nodes. The early feedback from the users will be used to enhance the quality of the services in future versions. A new major release is scheduled for April 99. At the same time, the services will be made available to Bürgernetz nodes all over Bavaria. The service has already been demonstrated in an exhibition at the Bavaria Online user congress and the feedback from Bürgernetz users as well as operators and officials from the local governmental institutions has been very positive.

By request of the head organization of Bavarian Bürgernetz nodes FAST will host and maintain the services for at least one more year. The Bürgernetze themselves declared strong interest in working on enhancements especially on the user interface parts.

As outlined in chapter 3 users already made suggestions for a variety of additional services in other domains. The operators of the Bürgernetz nodes are currently evaluating in which of these domains they could build up a network of content providers similar to the one in the event notification scenario. From a technical viewpoint the implementation of new services would only require few adjustments to the existing software modules. Main effort would be the definition of the domain specific data models and the implementation of suitable data bases.

In addition, FAST started with the design of a billing protocol that could be plugged into the application architecture so that services could bill agents for the information they offer.

In co-operation with another providers of agent-based applications FAST is working on combining the event notification system with a route-planning agent. The communication will be based on the FIPA standard ACL. The results of this project shall then be generalized towards building agent brokers for heterogeneous agent scenarios.