

Session 7A :

**USER PROFILE
AND
CONTEXT AWARENESS**

REDUCING THE COMPLEXITIES OF SERVICE INVOCATION MANAGEMENT THROUGH USER DEFINED CONTEXT BASED RULES

Paul Foster, paul.w.foster@bt.com
Venura Mendis, venura.2.mendis@bt.com

BT GROUP PLC
Sirius House, Room 125
Adastral Park
Martlesham Heath
IP5 3RE
United Kingdom

1. Introduction

With the arrival of Next Generation Networks, future network based services will be tailored to the individual user's needs, giving them a real feeling of personalisation. Currently, most personalised applications are restricted to the use of preferences, for example voicemail can be delivered either as a text message or call back as pre-selected by the user. But as the number of services offered increase and become increasingly varied, managing the circumstances under which a service gets invoked and interacts with other services becomes as important as the service itself. As a consequence end users will demand greater flexibility and control based around selectable contexts i.e. how when and where services get invoked. This effectively moves the service invocation decision from a Service Provider to the End User.

Next Generation Networks, such as 3GPP's IMS [4] for instance, clearly acknowledge the need for service invocation components but as yet there are no detailed designs or standards. But it does suggest that these type of service selection decisions are made by a Service Capability Interaction Manager (SCIM). MSF's Service Architecture [7] suggests its Service Broker component provides a similar functionality.

Existing vendor solutions, in terms of managing service invocations, commonly utilise a Network Services Broker (NSB). These tend to rely on fixed policies or rules and offer little or no user control in the invocation decision making process. Our research is investigating an approach that augments NSBs with a context aware rule brokering capability that enables users to create their own context based rules for managing service invocations.

But of course as the number of services and rules used grows then managing them will become more and more difficult. Eventually there will come a point for each user, be they a teenager or grandmother, where the sheer number of rules and their complexity become too difficult to manage.

This paper looks at the use of context aware rules for network service invocation in terms of both runtime and user management perspectives.

2. Rule Invocation Types

The functionality offered by using rules could equally be implemented directly in service logic. Although this may offer some performance gains, modifying service behaviour is harder. This is especially relevant to a service provider's *Service Internal Rules* as described in Section 2.1.

In general a rule consists of a body and a head. The body contains a set of predicate expressions and values. The head contains an action. If the body contents are true then the head's action is asserted.

For the purposes of this paper a rule in its basic form is represented as

IF (*Trigger Event*) + (*Set of Context Values*)
THEN (*Perform Action*)

Trigger Event refers to network and application events, for example a call invite, application start-up, etc. *Context Values* denote such things as a user's location, calendar events, device type, time of day, other active services, etc. *Action* represents a service to invoke if the trigger and context values test true. For example,

IF (*somebody calls me*) AND (*I am in a meeting*)
THEN (*Invoke Voice Mail*).

Rules are considered to be a key component of a user profile in ETSI's STF-265 User Profile Management Guide [1]. It places considerable emphasis on a user's ability to specify conditional behaviour.

To control service invocation behaviour based on context there are four basic functional categories as shown in the following table.

Rule Type	Owner	User Relevance	Service Relevance	Service Invocation
Service Internal	Service Provider	Multiple Users	Specific Service	After
Service Specific	End User	Single User	Specific Service	After
Service Selection	End User	Single User	Multiple Services	Before
Service Intelligence	Service Provider	Multiple Users	Multiple Services	Before

Figure 1 Rule Types, Relationships, & Ownerships

2.1 Service Internal Rules

These are rules that are set up and owned by a service provider. They would not be stored as part of the user's profile but they could use data or preferences from it, given appropriate access rights. Internal rules would be used only after a service has been invoked and are common to all users of a service. For example, an "on-line train ticket purchasing" service may have a rule stating that if a user's current location is not their local train station then offer to purchase a ticket for them by pre-populating the "From-To" form fields with the train station they are currently at and the station nearest their home respectively.

Alternatively, this type of rule maybe implemented directly in service logic as mentioned earlier.

2.2 Service Specific Rules

These rules are set up and owned by a single user and stored in his/her user profile. They are specific to one service and only applicable after the service has been invoked. For example, a user might set-up a "Do Not Disturb" service (To connect the call callers are required to enter a pin) such that

IF (*pin=1111*) THEN (*Use ring tone "A"*)
 IF (*pin=2222*) THEN (*Use ring tone "B"*).

2.3 Service Selection Rules

These rules are set up and owned by a single user and stored in their user profile. They are used to determine which service(s) to invoke given a user's current context. For example, "Do not allow anyone to contact me after 6PM unless they are my family". This could be set up as

IF (*Somebody calls me*) AND (*Time is after 6PM*)
 AND NOT (*Caller is family*)
 THEN (*Invoke VoiceMail service*)

Service Selection rules are only applicable before the service has been invoked, unlike Service Specific rules.

2.4 Service Intelligence Rules

These type of rules fulfil two main requirements:

- To act as default rules when no user defined rules exist. For example,

IF (*User does not answer phone in 6 rings*)
 THEN (*Divert to Voice Mail*)

- To control service precedence and chaining in response to various network triggers or events, overriding any settings by the user as necessary.

Although there is similarity to Service Selection rules, Service Intelligence rules are actually owned and set up by a third party, usually the user's primary service provider or whoever they buy services from. Hence, they are not specific to a single user and are not stored in the user profile.

2.5 User and Service Rule Symmetry

There is symmetry between the rule types. The diagram below shows that Service Selection rules are the user equivalent of service provider's Service Intelligence rules. Similarly, Service Specific rules are the user equivalent of Service Internal rules.

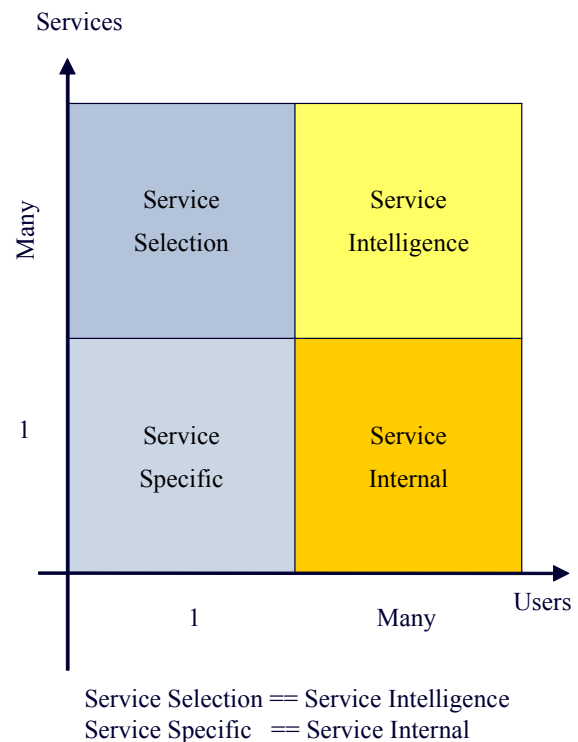


Figure 2 Symmetric Rule Type Equivalents

3. Context

In the field of context aware computing there is no exact definition of the term *context*. It can be many things; from a description of the physical surroundings (such as location and environment) to a representation of more

ethereal concepts such as why you are where you are and what you are doing.

Regardless of how context is defined, context aware systems use their knowledge of this contextual information to make decisions on how to react to requests or instructions so that their behaviour is tailored to current and relevant circumstances.

In the most basic situations context can be stated explicitly by the user, for example by choosing the most appropriate context from a pre-configured list, for example an "Instant Messenger" service where you are either available to use the service or not. For applications to offer more personalised context dependent behaviour this basic method becomes less practical. The level of context awareness needed in these will necessitate the user being able to define his/her own context definitions. For example,

User Defined Context Term of "Home" =
Location (*At Home*) AND Time (*18:00 – 06:00*).

The user's description of a context must be mapped to terms automatically determined by the network. For example, the user may define a location value as *Home* but the network's definition of a user's location may be specified using longitude/latitude co-ordinates.

It is probably most efficient for context updates to be event driven so that when something happens in the real world which affects some aspect of context, the context will be updated accordingly. The alternative to this would be for the network system to use artificial intelligence to ascertain the current context and change it only when it thinks it is needed.

For context information to be useful it must be readily available so that automated choices of which services to invoke can be made quickly. Exactly how the user's current context can be acquired, stored and retrieved is not the subject of this paper but could involve such techniques as:

- Location detection, where the user's location is determined automatically in some way such as through the use of Bluetooth, GPS, RFID, etc.
- Device detection when the user starts to use a particular device.
- Activity detection where the user starts to use a particular service or application.

4. Conflict Handling

In the service creation process the service provider needs to take into account two types of conflicts:

- *Provisioning Conflicts*. These refer to services that should not be provisioned for the user at the

same time, for example two voice mail applications. These are not relevant to this paper.

- *Runtime Conflicts*. These refer to situations when two or more services are invoked that are mutually exclusive i.e. Services that conflict with each other at runtime regardless of context, for example Voicemail and Call Divert services.

When a service provider introduces or modifies a service they may directly define default priorities to resolve conflicts with other services that are mutually exclusive. This can be implemented by using System Intelligence rules containing pair-wise relationships of conflicting services, for instance:

Service Selection Rule 1:

IF Location=(*At Work*)
AND Calendar=(*In a Meeting*)
THEN Action=(*Voice Mail*)

Service Selection Rule 2:

IF Location = (*At Work*)
AND Caller = (*My Boss Calls*)
THEN Action = (*Connect Call*)

System Intelligence Rule:

IF Service = (*Voice Mail*)
AND Service = (*Connect Call*)
THEN Action=(*Connect Call*)

If this user is *At Work* and *In a Meeting* and somebody calls them then their *Voice Mail* service is selected. On the other hand, if the user's *Boss Calls* whilst they are *At Work* AND *In a Meeting* then both the *Connect Call* and *Voice Mail* services are selected. In this case, the runtime conflict is resolved by the System Intelligence rule that gives precedence to *Connect Call*.

This approach is similar in concept to some existing B2B (Business To Business) rule engines for conflict handling (for example IBM's Common Rules [4]) and standards initiatives in rule definition languages (for example SCLP in ruleML [5])

To avoid giving the user the impression of unpredictable runtime behaviour it is advisable to expose potential runtime conflicts when the user is creating or modifying their rules. For example, they may then wish to change Service Selection Rule 2 above to avoid conflict i.e.

IF Location = (*At Work*)
AND Caller = (*My Boss Calls*)
AND NOT(Calendar = (*In a Meeting*))
THEN Action = (*Connect Call*)

In some cases a rule's body is unlikely to be in a complete canonical form where it contains an instance for each potential context parameter known to the

system. These missing context parameters effectively take on a *Don't Care* state as far as the rule is concerned. This can lead to unexpected service invocations, a problem if the services are mutually exclusive. The diagrams below show the nature of problem.

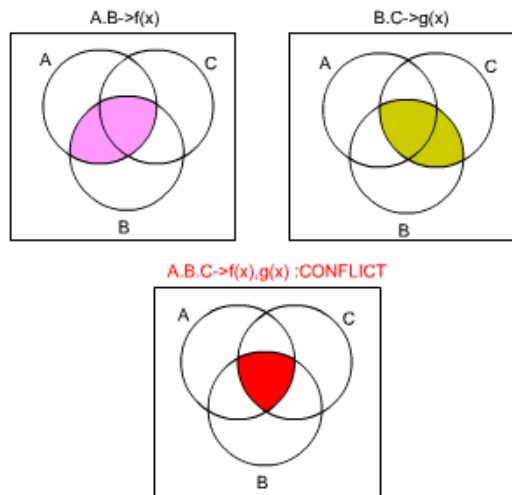


Figure 3 Potential Conflicts from “Don’t Care” States

$f(x)$ and $g(x)$ represent two mutually exclusive services. A, B and C represent all known context parameters.

$f(x)$ is invoked whenever context predicates A and B are both true i.e. $A.B \rightarrow f(x)$, represented as :

IF ($A=True$) AND ($B=True$) AND ($C=Don't\ Care$)
THEN Action= $f(x)$

$g(x)$ is invoked whenever context predicates B and C are both true i.e. $B.C \rightarrow g(x)$, represented as :

IF ($B=True$) AND ($C=True$) AND ($A=Don't\ Care$)
THEN Action= $g(x)$

Note: The ‘Don’t Care’ States are included in the above rules to illustrate conflict detection. They would not be included in the actual rules.

So if A, B and C are all true then both services will be invoked because the value of predicate C has no effect on the “A.B” rule. Similarly, Action $g(x)$ is also valid because predicate A has no effect on the “B.C” rule. If $f(x)$ is mutually exclusive with $g(x)$ then there is a conflict. This will be resolved by service intelligence rules at runtime but it must be represented to the end user on rule creation or modification.

5. Service Profiles

If a service is to be made available to a rule based invocation mechanism then the service’s capabilities and features must also be made available to it; ideally using a common or standard provisioning format. In

other words, service providers must provide a service profile with new services.

The information contained in a service profile could be quite extensive and used across many different processing domains. For service invocation purposes though, the information would primarily be used to enable System Intelligence rules to be updated in accordance with the new or modified service’s capabilities.

For example, for a new communications service the information contained in the profile could enable the system to

- Map it into an existing services hierarchy or ontology.
- Infer default trigger criteria .
- Compare new rules to existing system intelligence rules to identify conflicts or chaining with other services.
- Identify sharable information with other services

6. Rule Management Framework

The ability of a network based Rule Management Framework (RMF) to simplify rule management to a level that is acceptable by an individual user is critical. No user can be expected, or will want to, directly manage all the runtime rules for each and every service.

Figure 4, shows that just setting-up a simple “Do Not Disturb” concept for a single service can involve some quite specific rule creation. Then to duplicate this across multiple services, potentially in different domains with interaction conflicts, becomes a very complex and manual operation. Allowing the user to create generic concepts applicable across multiple services greatly simplifies user involvement. So, by extracting key information from a user given concept and applying it to the RMF’s knowledge base of service type information (capabilities, relationships, costs, etc), and taking into account any known user profile preferences, a significant portion of service selection rules for a specific service can be created automatically on the user’s behalf.

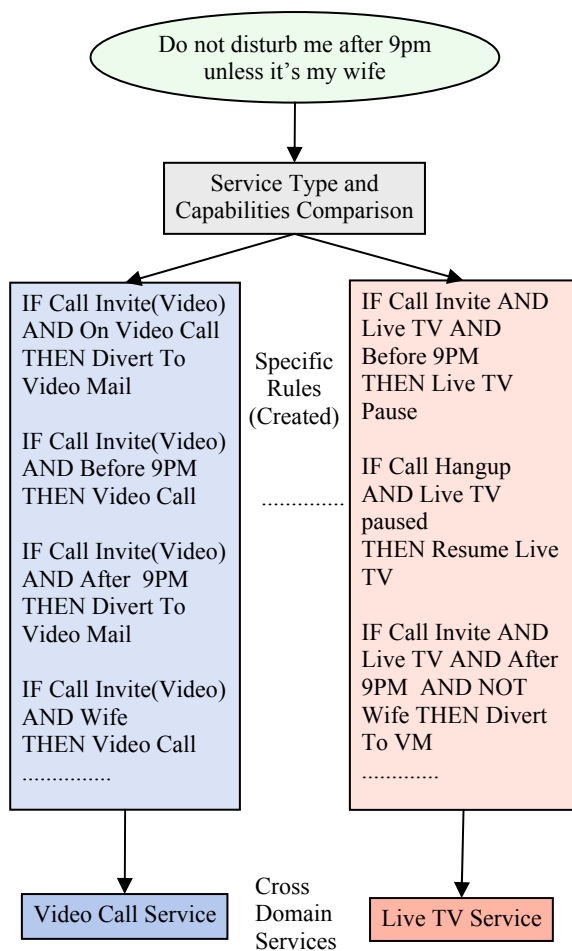


Figure 4 Specific Rules from a Single Rule Concept

Functionally, four key components of an RMF have been identified as shown in Figure 5.

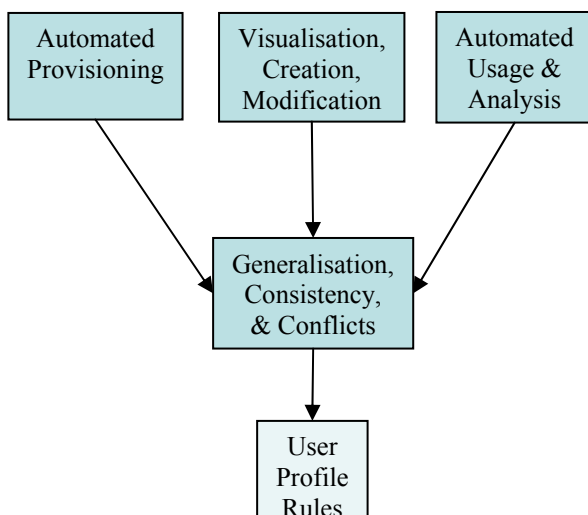


Figure 5 Rule Management Framework Components

6.1 Automated Provisioning

This component is required when a user first subscribes to a new service. It will automatically create a set of invocation rules specific to the new service. Its objective is to add a set of default rules for the new service based on the identified genre of the user (so they can begin using the service straight away). Two methods of rule pre-population that could be employed are

- Copy an existing ruleset from a similar service and use this as a basis for the new service rule set.
- Identify the new user's genre in a "Solicited" manner (e.g. a questionnaire) or "Unsolicited" manner (e.g. profile preferences or existing subscribed services) or a combination of both techniques. The user's rule base is then pre-populated based on this identified genre.

Although the lifecycle for rule creation and modification will usually start from here some users may prefer to create rules directly; in which case they would begin with the Visualisation Creation Modification component described next.

6.2 Visualisation, Creation and Modification

The way that rules are represented to a user is critical but is a double-edged sword. If it is too simple it may be impossible to convey the full meaning and scope of interactions between the rules. This will lead to misunderstandings and unpredictable behaviour. On the other hand if it is too complex then some users may not understand the meaning behind the information being displayed to them. Either way, they will eventually become frustrated and give up.

Whatever visualisation method is used it will need to perform such tasks as:

- Represent rules in both a generalised and service specific form as preferred by the user.
- Convey information regarding rule conflicts and inconsistencies between rules of similar or same service types to the user.
- View or filter existing rules that are relevant to a given user context (a combination of location, contacts, events, etc).
- Provide a preview mode to allow the user to try out new or modified rules without committing them.
- Prevent the user from creating invalid rules.

There is no such thing as a standard user, each of us is different. What may seem simple to one user can be quite complex to another. So, trying to design a single instance of a common visualisation interface that

represents rules to each and every user for each and every service in a way that meets their desired needs for simplicity, usability and capability is impossible.

Representing complex information and relationships to users is an ongoing research area. MIT's Scratch [2] is an interesting example of how programming logic is represented to non technical users. It offers a flexible and highly visual way to represent and validate a computer program through different shaped and coloured blocks, similar to the Lego® principle. The shapes and colours allow the user to see, and indeed preview, which types of code snippets and logic work together. It essentially lets the user visually validate their code. Although MIT's Scratch is intended for children to help them appreciate the logic behind computer programs we believe a GUI based on these principles and functionality consisting of interlocking context and logic building blocks could be an effective means for many different types of user to manage their rule sets and contexts.

6.3 Automated Usage and Analysis

This component would essentially be continuously working in the background to automatically modify the user's existing rules by analysing trends from services and network usage, for example if the system detects that the user rejects all business calls to his mobile on Sundays, then it could create a rule that automates this action.

6.4 Generalisation, Consistency and Conflicts

This component provides common functionality required by the other three components. For instance, by

- Providing a management layer or gateway between the other three components and the user's profile for rule storage.
- Checking new rules and amendments for inconsistencies in their declared predicates that invoke the same or similar services. For example, the Automated Usage component attempts to write a new service specific rule "Incoming business voice calls to mobile during lunch hour should be diverted". But this is inconsistent with an existing rule that states "Incoming business video telephony calls should be diverted when in a meeting".
- Making use of System Intelligence rules to check new user rules and amendments for conflicts between mutually exclusive services selected at the same time.
- Making information on similarities between services or existing rules available to the other components.

7. Conclusions

Personalised service invocation management is as important as service personalisation, especially with the inevitable growth of new services as networks offer more capabilities and user control. The features and flexibility offered by a context based rules solution seems to be a promising approach to take.

End users cannot be expected to understand the intricacies of network service interactions. So, in addition to any user defined rules some System Intelligence rules owned and managed by the network provider will be required to deal with runtime conflicts and default behaviours.

The use of an intelligent Rule Management Framework to simplify rule management from the user's perspective is critical. It must be capable of automatically creating a set of default rules when the user subscribes to a new service. It must provide an intuitive interface for the user to manually modify and view rules, and it must automatically update rules based on how users use their services.

8. References

1. ETSI Specialist Task Force 265: User Profile Management, <http://portal.etsi.org/stfs/hf/STF265.asp>
2. MIT Scratch Programming Toolkit, <http://weblogs.media.mit.edu/llk/scratch/>
3. RuleML, The Rule Mark Up Initiative, <http://www.ruleml.org/>
4. The 3rd Generation Partnership Project (3GPP), Internet Multimedia Subsystem (IMS) Network Architecture Release 6 (3GPP TS 23.002)
5. IBM Common Rules, <http://www.alphaworks.ibm.com/tech/commonrules>
6. RuleML Situated Courteous Logic Programming (SCLP), "Representing E-Commerce Rules Via Situated Courteous Logic Programs in RuleML". Benjamin N. Grosz, ECRA journal, 2004.
7. Multiservice Switching Forum, MSF R2 Service Architecture, June 2003, <http://www.msforum.org/techinfo/reports/MSF-TR-ARCH-002-FINAL.pdf>

A Location and Privacy Middleware for Context-aware Mobile Applications

Gerald Eichler, Ralf Karge

{gerald.eichler, r.karge}@t-systems.com

T-Systems Enterprise Services GmbH, Technologiezentrum ENPS
Deutsche-Telekom-Allee 7, D-64295 Darmstadt, Germany

Summary. This article proposes a common middleware architecture to combine the flexibility of location based services with the protection of the private sphere of their users. A unified localisation offers sufficient position information in each situation, while context awareness extends this basic approach to satisfy individual needs and gives room for a wide range of mobile applications. Personal profiles that were matched on the fly allow the user to pre-characterise interests before and when travelling around. Third party application developers will be given a framework which supports portability for various mobile devices.

1 Two Theses for Introduction

1.1 The New Mobility

Availability of smart mobile communication devices like smart phones – a combination of PDA and mobile phone – at a feasible price, decreasing data communication transfer costs as well as social trends lead private and business customers (users) to use mobile applications more and more. The general demand for information is traditionally satisfied by web surfing at home or in the office. This has two draw-backs:

- information is not available at the right time in the right place, and
- information has usually overhead regarding individual needs in the current situation. Mostly a small sub-set is required.

Context awareness in combination with protection of the private sphere might help to fill this gap.

1.2 The Re-invention of Middleware

The idea of common middleware approaches is well known in telecommunication technologies. E.g., the Common Object Request Broker Architecture (CORBA) is known as a platform independent, object-oriented function suite for service development. In the past such architectures were seen as being inflexible, low performing and too expensive. Today there are three good reasons to reinvent such ideas:

1. **Protocol Variety.** Multiple protocols compete to serve the same purpose. Even a single protocol offers so many options that a straight forward implementation by different vendors would not imply interoperability between different systems.

Interaction at the protocol level is replaced by basic service offerings, e.g. web services.

2. **Performant Hardware.** Some years ago software implementations of protocols required a system near programming and compact code. Object oriented approaches using modern methods of software engineering like design pattern did not behave performant enough. Nowadays even mobile devices contain fast processors and offer reasonable memory capacities.
3. **Liberalised Markets.** As multiple, independent providers enter the market and create service offers, a new level of successful intercommunication has to be defined.

2 Context Awareness

Context awareness covers a wide range of situation related aspects. Best known, the location enables information about points of interest (PoI) nearby, which are hosted in a “geo-information database”. In combination with additional facts like time information and previous events a typical behaviour of a user can be derived using learning statistics.

2.1 Unified Localisation

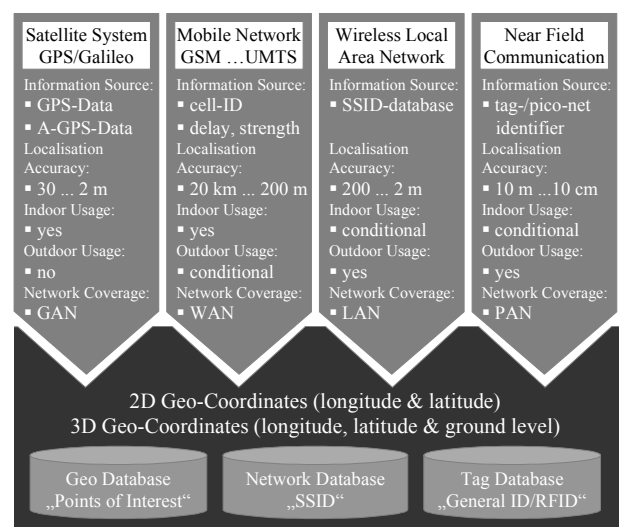


Figure 1: Unified Localisation using different systems, networks and tags.

There are multiple ways to get information about the current position (location) of a user due to its mobile device equipped with several external communication interfaces e.g., GSM/GPRS/UMTS, WLAN, Bluetooth, near field communication GPS/A-GPS or other sensors [1]. Some of them can be independently used for both, communication and localisation in a seamless manner.

Figure 1 provides an overview of four different methods for localisation, classified according to the coverage of underlying network infrastructure from GAN over WAN and LAN to PAN.

By means of public or private databases 2D (longitude and latitude) or 3D (plus absolute height or height over ground) geo-coordinates can be applied as common exchange format, which is a pre-requisite for graphical location representation on maps.

2.1.1 GSM Network based Localisation

GSM including GPRS and UMTS mobile network operators (MNO) already have information about the cell, a mobile phone is logged in at a certain time. This allows a rough localisation, which is sufficient for several purposes.

Even the mobile phone itself recognises neighbour cells and their signal strengths and delays, which will help to increase the localisation accuracy from some kilometres to several hundreds of meters by triangulation, depending on the base station density and the cell diameter. The closer the PoI are to each other, the smaller the cells usually are, like in a city centre. Such environment is looking mostly for more accurate localisation than rural areas. MNOs can do even more accurate triangulation as they know about their transmission power and antenna characteristics.

2.1.2 Tag and ID driven Localisation

The detection of active WLAN networks (access points) in combination with a fairly stable SSID database already allows resolutions of some 10 meters. This does not require an active registration with such a network. Naturally, there is a different applicability for indoor and outdoor localisation. Private indoor WLANs usually have a lower coverage than public hotspots. That allows more precise localisation but the network infrastructure changes permanently by replacing, moving or adding WLAN access points.

Information tags like 2D visual codes [2] or RFID transponders at fixed places can be directly used as trigger for location based events, assuming mobile devices are equipped with appropriate receivers, like a camera or an RFID reader. This technology requires the active handling of the user by pulling the tag data. While doing this, multiple alternative actions could be selected e.g., information download, reservation of

tickets, activation of transactions, interaction with other users; even ad-hoc short-term mobile gaming.

2.2 Context aware Information

2.2.1 Support of Applications

While localisation already enables a wide range of applications like

- mobile find service (PoI; places of service offering; other people known as buddies),
- alerting service (pollution limits; allergy causing substances; reminder functions),
- mobile couponing service (sales discount; customer binding programs), or
- distributed resource planning service (fleet service or resource management),

additional and detailed knowledge about the user and its preferences and repugnances can provide him or her with more appropriate and easy to catch information in a certain situation. This individual context is not only driven by the location but by several other factors, called context.

2.2.2 Context Classification

Figure 2 gives a classification of context relevant issues, enriched by some typical examples. It is divided into user and situation specific context in one dimension as well as semi-static and dynamic context in the other.

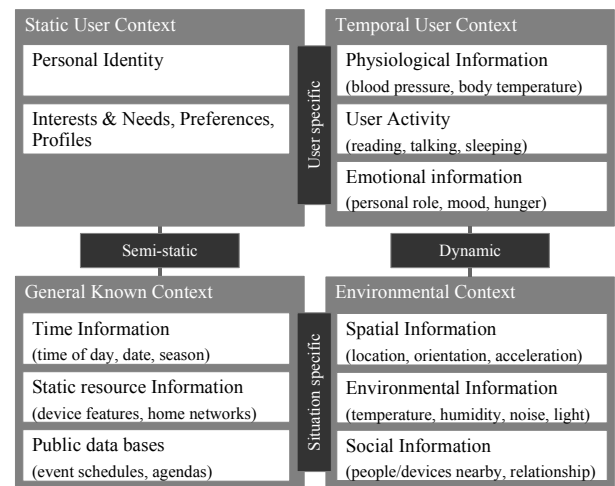


Figure 2: General classification of context with examples.

2.2.3 Context Personalisation

Multimedia contents usually contain a lot of irrelevant parts for the current situation of a certain user. Personalized information increases the acceptance of services and overcomes the bandwidth and transmission cost limitations for today's mobile services.

An important factor of personal context is the time aspect. Using the knowledge about the user's individual planning deterministic schedules can be derived. By means of the evaluation of known habits, rules can be derived, which lead to heuristic predictions, as described in Figure 3.

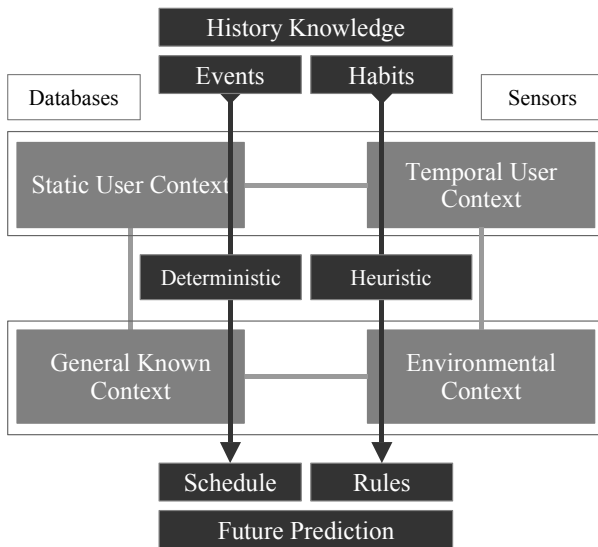


Figure 3: The time aspect in context classification.

2.3 Personal Profiles

User profiles, stored in an ontology, will help to specify the user's interests on the required granularity. They can be used as both, selection criteria for relevant PoI from a list and filters against given facts belonging to a certain PoI.

There are several ways to configure profiles:

- manual step by step evaluation of a hierarchical list of given facts,
- selection of stereotypes by answering a questionnaire, or
- automatic adaptation of a basic profile over time by so called learning objects.

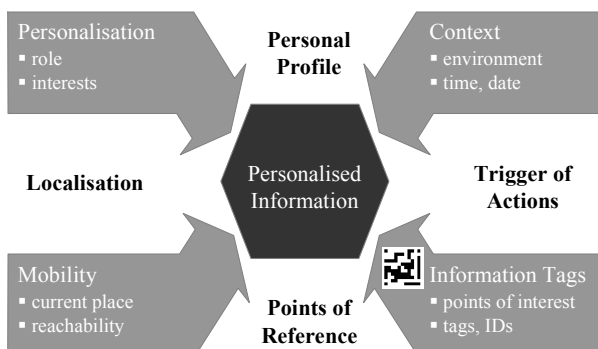


Figure 4: Factors for mobile and context aware information services.

Figure 4 summarises the factors, which lead to meaningful personalised information.

3 Permission and Privacy

Localisation touches the private sphere of users. It is quite important to clarify, which information is needed for which purpose.

3.1 Role of Mobile Operators

MNO already have location information (associated with cells) of mobile phones available to run their networks. Emergency services make use of it, provided that the user agrees to be tracked. This works as long as the mobile network operation and service offering belongs to one hand.

The market liberalisation brings new players into the game. But most users don't like multiple contractual relations. A native solution would be the establishment of an infrastructure, which allows third parties to offer their services without handling personalised data. Location data are considered to be "personal" as long as a mapping to an individual user is possible. Randomly assigned pseudonyms will help to overcome this issue.

A clear differentiation between pull and push services will help to provide the user with more transparency in the world of mobile services. While push services are initiated by the MNO, pull services allow full user control.

3.2 Tracking and Tracing

"Tracking" allows a current localisation of individual users or user groups. "Tracing" adds the time dimension as correlation of single tracks. It allows a prediction of locations, which probably will be accessed next [3].

History functions support the detection of statistic correlations in multiple dimensions. Finally, geo-databases are required to translate pure location information into a current context.

An example will illustrate this: By tracing and checking against a geo-database it could be detected that a user is taking the train. In combination with the user's schedule the destination is known. Therefore the order for a taxi or the presentation of the time table of local public transport could be applied, depending on user's preferences and expected arrival time.

3.3 Profile Management Engine

Personal interests and preferences are collected in user profiles. These profiles are stored at a secure level within a repository, which is part of the middleware. For configuration purposes an applet is provided, which allows fine granular justification by means of a

web browser, preferably used on a fixed device. Multiple profiles can be stored and activated in parallel depending on subscribed application services.

The *Profile Management Engine* with its core component, the “Profile Matching Engine”, compares personal profiles, called interest profiles against, either the profile of a resource, which represents given offers, or another active interest profile. As a result only a list of resources characterized by a high degree of matching similarity is delivered to third party services like “Resource Finder” or “Contact Builder”.

3.4 A Common Framework

A trustable middleware is proposed, which decouples the collected localisation information from the required service function to establish independent third party applications. The middleware framework contains two main components

- the Permission and Privacy Gateway, and
- the Profile Management Engine.

Common basic functions are designed to fulfil storage, handling, accounting and security functions, see Figure 5. Additional modules can be introduced on demand. Third party application developers get granular access by means of open APIs. Granular means to bundle functions in protected packages, which could be offered to a third party application development or not. The preferred module inter-communication makes use of XML to structure and exchange messages.

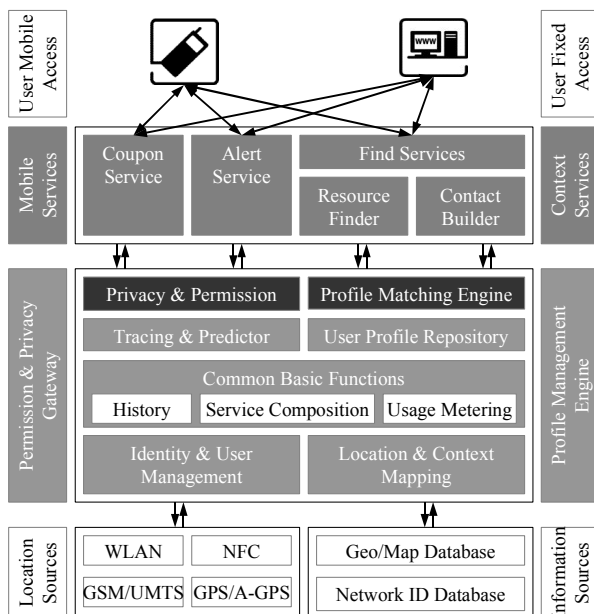


Figure 5: Common Platform for Permission & Privacy Gateway and Profile Management Engine.

User profiles are stored centrally for two reasons: firstly, to perform an efficient matching function and secondly, to protect profiles against third party access.

The modification of profiles by the user itself requires protected access to the system and encrypted transmission technologies.

4 Mobile Applications

Generally, access to the services is possible from any fixed and mobile device as long as it has network connectivity and a web browser, allowing HTTP and HTTPS, is up and running. Due to limitations of the display size of mobile devices it is recommended to do most configuration e.g., profile definition and activation, via the fixed access. Mobile access should be limited to a minimum of interaction.

4.1 Business Focus

Mobile applications address several domains [4]. Within first evaluation trials business focus was given to single domains:

- mobile travel support,
- mobile transaction support, and
- mobile decision support.

Figure 6 shows how these domains can be seen as a combination of two out of the three actions: localisation, information retrieval and consumption offering.

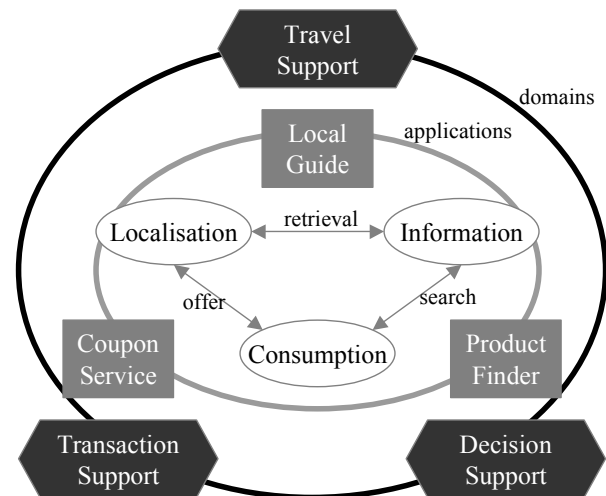


Figure 6: Domains of application.

Detailed business plans are not in the focus of this contribution. Up to now such evaluation were taken only for single applications and not for the proposed entire framework.

4.2 Service and Application Creation

The mentioned applications

- Local Guide
- Coupon Service, and
- Product Finder

are three pilot examples developed in two R&D innovation projects [5]. They should demonstrate, how to make use of localisation, personal profiling and privacy protection. Underlying basic services will be generalised and encapsulated as part of the middleware Parts of applications, which become of general interest or require special protection, will be shifted to the middleware framework step by step.

5 Conclusions and Further Work

5.1 Customer's View

Incumbent telecommunication providers are brought into the position to offer value added basic and network services for third party applications in the areas of

- convergence of fixed and mobile communication,
- unified localisation combined with information databases,
- trusted profile hosting, and
- personal information filtering and services.

The user can benefit from this combination, while getting secure, mobile, location based services for several domains.

5.2 Technical View

The contribution emphasises re-invitation of a middleware solution to make applications secure for private sphere protection in a mobile context.

The appliance of web technology using HTML and HTTP allows a simple portability to a large selection of mobile devices. Basic services of the framework offer their functions through XML based API calls.

5.3 Next Steps and Ideas

Field Trials. First usability tests showed that users look for a high amount of flexibility, while spending a minimum of effort for configuration. Predefined profiles overcome this trade-off. More focussed acceptance tests will be carried out to optimise the mobile application design, followed by field trials.

Business Models. Multi-party business models have to be developed to penetrate the mass markets. Currently, tracing applications are too expensive as single localisation is priced with some 10 cents by MNOs.

Database Maintenance. The proposed applications make use of localisation data. Furthermore, devices supporting multiple communication technologies can be used actively to verify and maintain fast changing geo database entries and to keep them up to date.

Context of Submission. When making a mobile phone call the typical first question is 'where are you?'. Context information is more than the location. By

adding an identified context to a service request or message transmission, most message contents become much more meaningful.

List of used Abbreviations

2D	Two Dimensional
3D	Three Dimensional
A-GPS	Assisted Global Positioning System
API	Application Programming Interface
CORBA	Common Object Request Broker Architecture
GAN	Global Area Network
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communication
ID	Identifier
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
HTTPS	Secure Hypertext Transport Protocol
LAN	Local Area Network
LBS	Location Based Service
MNO	Mobile Network Operator
NFC	Near Field Communication
PAN	Personal Area Network
PDA	Personal Digital Assistant
Pol	Point of Interest
RFID	Radio Frequency Identifier
SSID	Service Set Identifier
WLAN	Wireless Local Area Network
XML	Extended Markup Language

Acknowledgement

Contextual Information to Go (Contigo) and *Permission Tracking and Tracing Project* (PTTP) are two independent innovation projects [5] funded by the *Deutsche Telekom Laboratories*. Ideas for the common middleware approach are taken from both.

References

- [1] Eichler, Gerald: "Von der Lokalisation zum personalisierten Kontext". Presentation at 14. Treffen der VDE/ITG Fachgruppe 5.2.4, Stuttgart D, 9/10.06.2005. URL: <http://www.ikr.uni-stuttgart.de/Content/itg/fg524/Meetings/2005-06-10-Stuttgart/>
- [2] Rohs, Michael et al: "Visual Code Recognition for Camera-Equipped Mobile Phones". ETH Zürich, Dpt. of Computer Science, Inst. f. Pervasive Computing 2005. URL: <http://www.inf.ethz.ch/personal/rohs/visualcodes/>
- [3] Eichler, Gerald; Böhm, Andreas: "Einheitliche Einzellokalisierung – Voraussetzung für Track & Trace Mehrwertdienste". Contribution to Praxis der Informationsverarbeitung und Kommunikation (PIK), 29. Jg., Heft 1/2006, S. 4 ff., K. G. Saur Verlag, 2006.
- [4] Küpper, Axel: "Location-based services: fundamentals and operation". John Wiley & Sons Ltd., 2005.
- [5] Deutsche Telekom Laboratories: "Areas of Innovation and Research" URL: <http://www.deutsche-telekom-laboratories.de/english/content/2-areas/index.html>

Enabling context-aware applications

Per-Oddvar Osland*, Brynjar Viken*, Fredrik Solsvik*,
Gaute Nygreen*, Jan Wedvik**, Stein Erik Myklebust***

* Telenor R&D. E-mail: {per-oddvar.osland, brynjar.viken, fredrik.solsvik, gaute.nygreen}@telenor.com
Address: Otto Nilsensv 12, 7004 Trondheim, Norway

** Sun Microsystems. E-mail: jan.wedvik@gmail.com Address: Haakon VII's gt. 7B, 7485 Trondheim, Norway

*** Gintel AS. E-mail: stein-erik.myklebust@gintel.no Address: Otto Nilsensv 12, 7004 Trondheim, Norway

Abstract

The idea of context-aware applications is to take into account the situation of the user when performing actions. If this can be achieved automatically and without demanding attention from the user, it will serve as an enabler for pervasive computing. In this paper we present the design and implementation of Context Manager, a component for acquisition, storage, and distribution of context information. Context Manager has been tested in realisation of a scenario related to eHealth. The applicability for such a component is vast, and it may serve as an enabler for a wide range of context-aware applications.

1. Introduction

The objective of pervasive computing [Weis92] is to hide technology from the user by seamless integration into everyday life. The focus is on services that allow the user to solve his tasks while devices and technology fade away into the background. Context-aware systems are a vital part of pervasive computing. Context information gives applications entirely new opportunities to offer new services and adapt their behaviour according to the situation of the mobile user. These applications, which make out context-aware systems, increase the usability by reducing the demands on the end-user. The pervasive system minimizes the need for user attention.

An example of such a service is the transfer of an ongoing communication session as a more suitable device becomes available. This could be the case e.g. for a girl who is in a conversation using her mobile phone. As she arrives at her room, the session is transferred to her PC, where she can use videoconferencing as well as chat in addition to voice. Gathering of her context information is a precondition for this example, including presence (in conversation), location (in her room), and service discovery (PC with camera available). Moreover, application of her personal profile (use best devices available for conversation) is necessary as well.

User-oriented context information is relevant to a large number of applications. Obviously, it is not desirable that every application implements mechanisms to gather and process specific context information; rather

a generic context platform/infrastructure that supports applications by gathering, processing and managing context information is needed. Such a generic context infrastructure is an essential component of pervasive computing.

The work presented in this paper is carried out within the Akogrimo project¹. In the project a set of scenarios have been presented, where user context is used to enhance the provided application [Loos05]. These include several scenarios within eLearning, eHealth, and crisis handling and management. We present a basic context model and an infrastructure for context handling. As proof of concept, we have designed, implemented, and tested Context Manager. It is a part of the Mobile Network Middleware Layer, one of the four layers in the Akogrimo architecture.

1.1. Background

The dictionary defines context as “The interrelated conditions in which something exists or occurs”. In the literature many definitions and examples of context information are found. [Dey01] reviewed previous context definitions, and provided the following general definition of context: “Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”

Schilit et al [SAW94] state that the important aspects of context are: where the user is, who the user is with, and what resources are nearby. Context attributes can also be classified as internal (logical) and external (physical) [BaDu04]. External context may be detected by sensors and includes light, movement, temperature

¹ Akogrimo, Access to Knowledge through the Grid in a mobile World, is a project funded by the EC under the FP6-IST programme PRIORITY IST-2002-2.3.1.18 (Proposal/Contract no.: 004293). The Akogrimo project team comprises of totally 14 European organizations, and the vision is a world in which grid services, pervasively available, are eventually meeting the needs of fixed, nomadic and mobile citizens in the ‘everywhere at every time in any context’ paradigm. Network and service operators are viewed as contractors that are able to develop new business activities and to provide profitable services in such an integrated world based on grid and mobile communications concepts. See www.akogrimo.org for more information.

etc. Internal context is mostly specified by the user or captured monitoring the user's interaction; e.g. tasks, work status, presence etc.

A range of different definitions and models exist for context, including [OOK05] and the ones mentioned above. An objective is often to define and model context generic and suitable for most purposes. However, in most cases the modelling approach will be influenced by the work and situation that originated the need for dealing with context. This is the case in the work presented here, where our objective is to use context for Akogrimo services in a beneficial way, and not to define a new modelling approach.

The remainder of this paper is organised as follows: In the next section we briefly describe some scenarios, while Section 3 shows the basic context model derived from these scenarios. Section 4 shows an infrastructure for context handling, while Section 5 describes design and practical experiences. Finally, the last section concludes the paper.

2. Usage scenarios

In this section we present some basic scenarios that have been defined within the Akogrimo project. These represent everyday situations where humans, enterprises and organisations can benefit from use of grid technology and mobile communications for complex problem solving. More detailed descriptions may be found in [Loos05].

The basic finding in the following scenarios is that information about involved personnel must be taken into account in order to handle the situation in a best possible manner. This includes their availability, location, access to resources, etc. This is the starting point for our approach to designing and implementing a system for context management.

2.1. eHealth – emergency handling

In this scenario, a patient's physical health is being monitored by medical equipment. If an exceptional situation occurs, e.g. heart beat rhythm shows abnormal patterns, a range of actions are taken. These are some of the actions:

- An alarm centre is involved for handling the case
- A medical expert is contacted for analysing the medical data
- The closest ambulance is located
- The closest hospital with available and qualified personnel is located

In this setting, context plays a vital role: The patient's medical state and location, the medical expert's availability, the location of the ambulance, etc.

2.2. eLearning

University students and staff, although not co-located, may need to communicate and interact as in a learning situation. One example is when archaeology students go to a field trip and need to consult findings with fellow students or supervisors. Moreover, students on a field trip may also need information associated with the location he/she is at.

Context in this scenario includes the location of the student at field trip, availability of supervisors and fellow students, and access to equipment necessary for analysing field trip findings.

This scenario may be extended to a more general case where people in an enterprise/organisation cooperate from remote locations. The situation and needs will more or less be the same as for eLearning, and hence the definition of context may be reused.

3. Basic context model

The definition of context is open-ended; the amount of context information available is enormous and a context information model cannot be completely specified for all purposes. The approach is to define a context information model that captures information common among multiple applications e.g. focusing at communications, user guidance, and user assistance. Based on requirements from Akogrimo scenarios, our information model focuses on the situation of a user. The main entities are user and devices/local services, as shown in Figure 1. Location is presented as an important context attribute. This model is inspired by Henriksen et al [HIR02].

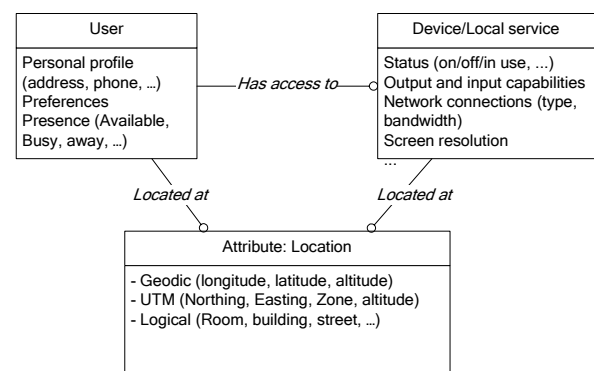


Figure 1 User-oriented context model.

In the following we describe user context with focus on presence, location, and device availability.

3.1. Presence

Presence information is the basis of instant messaging (IM) and refers to information about the state of users such as availability, reachability and other information set by the user (e.g. mood, interests, etc.). Presence

information is dynamic and may change frequently. The changes can occur manually by user interaction or automatically based on available context information (e.g. location, user talk on the phone etc.).

Currently, there are multiple protocols for handling presence information such as:

- Parlay PAM
- Parlay X Presence
- OMA Wireless Village
- IETF Working groups, including IMPP (Instant Messaging and Presence Protocol) and SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions).

3.2. Location

Location systems deliver geographic coordinates of people (things and devices). The geographical coordinates can be used to infer details about where the user is, e.g. room, street, at home, in the car. There exists a multitude of technologies for locating and tracking individuals (based on GPS, WLAN, GSM, RFID, active badges etc) that are becoming increasingly more common.

There is no common protocol for exchange of location information, examples of available protocols are Parlay (X) Location and OMA Mobile Location Protocol [MLP04]

3.3. Devices and local services

Devices and local services in the immediate proximity of a user are often an integral part of the application provided to the user. Hence attributes of devices and local services may be an important part of the user's context.

Devices and localizable services become available to users in two ways:

- Directly via logging on: User is logged onto a device. This is the case for a mobile phone or for a PC where the user is using a SIP User Agent. By logging on to a SIP UA installed on a device, we assume that the user has direct access to and control over that device.
- Indirectly via location: A user has local services available within a reasonable range. The size of such a range, and the usability of the local services within that range, depends on the application.

The discovery of "local services" may be accomplished by means of protocols like Service Location Protocol (SLP), UPnP, Zeroconf and so forth. Local services will be described with a set of attributes. Location should be included in the service description. User location, which is obtained by means of e.g. RFID positioning technology, may in turn be used when searching for services in the surrounding of a user.

The capabilities of the terminal a user is logged onto are also needed to provide general services, for example:

- Screen Size ++ (pixels, colour depth etc.)
- Operating System (version)
- Memory
- Network connections
- Browser
- Installed SW and players

4. An infrastructure for context management

In a general setting, context data is obtained from a number of sources, and requested by interested parties, i.e. context-aware services. In the trivial case, sources and consumers would interact directly with each other. However, we suggest a mediator infrastructure in the data flow, known as the context manager (see Figure 2). The context manager would address issues such as:

- Filtering out relevant data and forwarding those to the interested parties.
- Converting heterogeneous context data to a uniform format.
- Resolving situations where context data are incomplete or inconsistent.
- Inferring higher-level context data from basic data, e.g. by mapping the location of a user onto a map of a building.

Introducing a context manager hence improves the overall scalability of the context system, and also factors out functionality that would otherwise have to be duplicated in each context-aware service.

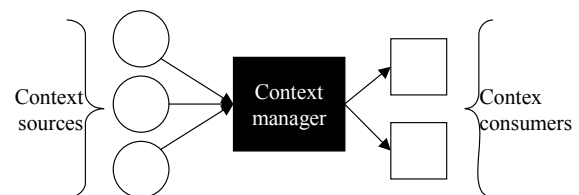


Figure 2 Generic flow of context data.

In the following, some issues that are of particular importance for the design of context management are discussed. The separation of basic context gathering and processing from the applications has several advantages such as:

- Hiding low level-details from applications
- Accessing standardized APIs facilitates the usage of context by applications
- Applications can add domain specific context handling as needed
- Context data sources are reused across applications

- End-systems (e.g. mobile terminals) do not have to perform context processing

Performance and scalability are two major concerns for the design of a context management infrastructure. The factors contributing to the problem is:

- The highly dynamic nature of context information,
- Large number of mobile users
- The huge amount of context data collected and processed from a variety of sources
- Near real-time requirements for delivery of context info to consumers.

Dynamic context information is useless for context consumers if it is out-of-date or incorrect, e.g. a location update or a “busy in meeting” 3 hours ago. If the information about a user is not fresh, it’s not as usable as user context.

5. A multi-purpose Context Manager

In this section we describe Context Manager, an infrastructure that has been developed as part of the Akogrimo project. The implementation is in line with the theory presented in the previous sections. However, its scope is limited to experimental use; hence context gathering focuses on a smaller set of context sources. A more detailed description may be found in Section 5 of [Osl05].

5.1. System design

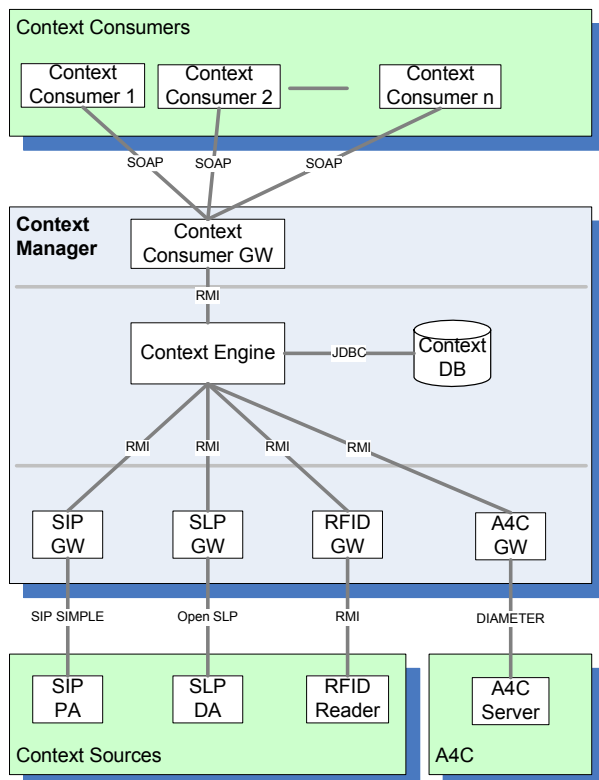


Figure 3 Context Manager architecture

The Context Manager, which is shown in Figure 3, has three major parts:

- Context gathering gateways, which are responsible for handling communications with the context sources. Currently the following gateways have been implemented:
 - SIP Presence GW: Acts as a SIP watcher, i.e. it subscribes to presence information for a presentity.
 - RFID GW, where position is based on RFID technology (Radio Frequency IDentification)
 - SLP GW (Service Location Protocol). SLP may be requested for services of a specific type. Or, once the user position is known, SLP is searched for services in a given area around the user. In Figure 4 three devices are registered with location parameters.
 - A4C (Authentication, Authorization, Accounting, Auditing and Charging) gateway is used to perform authorization operations, e.g. verify that a context consumer has right to access context for a specific user. Hence, A4C is not a context source like the others. A4C has been developed by project partners at the University of Zurich, and is described in Section 4 of [Osl05].
- Context Engine (including persistent storage), responsible for
 - Initiating context request towards context gateways
 - Storing context information
 - Keeps track of which consumers should receive notification about updated context
- Context Consumer GW
 - Provides a Web Service interface towards context consumers: It receives subscriptions and returns notifications when updates occur. Subscriptions specify user and scope, e.g. how much context is desired. In the example in Figure 4 the subscription is requested to devices of type “plasma screen”.

Figure 4 depicts a case where the context consumer, a context-aware application, is interested in knowing which users are close to a large display. The application may e.g. use this information to decide which user should be included in a communication session, or to offer a user better means for communication.

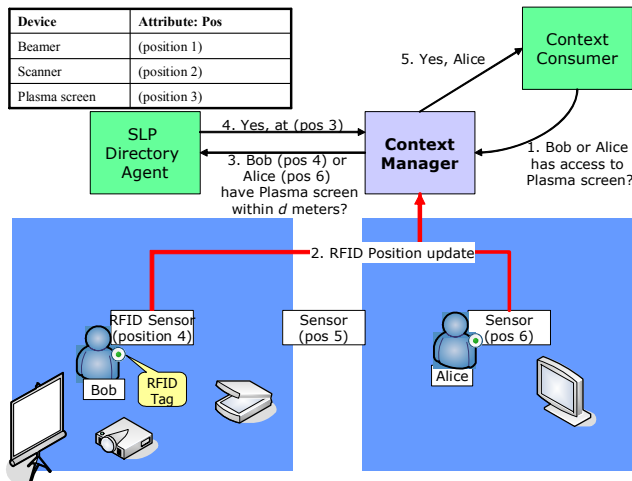


Figure 4 RFID and SLP as context sources

5.2. Practical experiences

In the Akogrimo Project, the scenario “eHealth – emergency handling” described in Section 2.1 was selected for demonstration purposes. Concerning the part of the service handling context, the following has been tested in practice (also see Figure 4):

- Patient (Bob) and doctor (Alice) were located using RFID tags. Services and resources in their surroundings are identified
- The context consumer, which executes the essential business logic in the service, has subscribed to context changes. The context consumer is a Grid application realised using Business Process Execution Language (BPEL).
- In the demonstration, Alice and Bob are communicating using voice and video. As Alice moves to a room with a larger display, her context is updated. Business logic decides to offer her the possibility to continue the conversation using this display. She accepts, and the ongoing session is transferred.

A similar service has been designed and implemented in [OOK05]. In our case the Context Manager differs from this as it is an independent component that provides access to context updates through a Web Service interface.

The Context Manager was found to work well and to provide close to real-time performance. Although currently used for a very specific task, the applicability of the Context Manager is vast. It is currently limited to gather context from a small set of sources, but it may still serve a wide range of context-aware applications. Moreover, the Web Service interface makes it easy to use for context consumers.

6. Conclusions and further work

A basic context model and an infrastructure for context handling have been presented. Based on this, we have implemented a Context Manager for experimental purposes. A context-aware application using this component has been tested successfully, and is under further development. The Context Manager currently serves as part of the Akogrimo architecture.

The applicability of the Context Manager is vast, and it may serve a wide range of context-aware applications.

Further work includes

- To design and build more context-aware applications
- To develop an ontology-based context model and request interface for context consumers
- To add more context sources, e.g. to use Bluetooth on the user’s terminal for discovering Bluetooth-enabled services around the user
- Address challenges related to scalability, privacy and security.

Abbreviations

A4C	Authentication, Authorization, Accounting, Auditing and Charging
CM	Context Manager
DA	Directory Agent
GW	Gateway
IMPP	Instant Messaging and Presence Protocol
MLP	Mobile Location Protocol
OMA	Open Mobile Alliance
PA	Presence Agent
RFID	Radio Frequency Identification
SIMPLE	SIP (Session Initiation Protocol) for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SLP	Service Location Protocol

References

- [BaDu04] M. Baldauf & S. Dustdar, *A Survey on Context-aware systems*, technical report, Distributed Systems Group, Technical University of Vienna, 2004
- [Dey01] A. K. Dey, *Understanding and Using context*, Personal and Ubiquitous Computing Journal, Volume 5 (1), 2001, pp. 4-7.
- [HIR02] K. Henriksen, J. Indulska, and A. Rakotonirainy, *Modeling Context Information in Pervasive Computing Systems*, Proceedings of the First International Conference on Pervasive Computing, 2002.

[Loos05] Christian Loos et al: *Testbed description*, Akogrimo deliverable 2.3.1. February 2005. URL: www.akogrimo.org -> Downloads -> Deliverables -> Akogrimo Testbed Description. URL last visited 31.1.06

[MLP04] Open Mobile Alliance, *Mobile Location Protocol*, 3.2Draft Version 2004-11-02, http://member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/OMA-MLP-Spec-V3_2-20041102-D.zip. URL last visited 31.1.06

[Most03] S. Kouadri Mostéfaoui, G. Kouadri Mostéfaoui, *Towards A Contextualisation of Service Discovery and Composition for Pervasive Environments*, University of Fribourg, 2003.

[OOK05] Østhus, Egil C., Osland, P.O., Kristiansen, Lill, "ENME: An ENriched MEDIA application utilizing context for session mobility; technical and human issues" (accepted for publication in proceedings UISW in LNCS, Japan Dec. 2005)

[Osl05] Osland et al: *Final Integrated Services Design and Implementation Report*. Akogrimo Deliverable D4.2.2. November 2005. URL: www.akogrimo.org -> Downloads -> Deliverables -> Final Integrated Services Design and Implementation Report. URL last visited 31.1.06

[SAW94] Schilit B., Adams, N. & Want, R. (1994). *Context-aware computing applications*, Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications, 85-90. Los Alamitos, CA: IEEE.

[Weis92] M. Weiser, *The Computer for the Twenty-First Century*, Scientific American, pp. 94-10, September 1991.