ICT-2009-248730

Florence

Multi Purpose Mobile Robot for Ambient Assisted Living

STREP
Contract Nr: 248730

Deliverable: D2.1 State of the Art of Multi-Purpose Robots and Privacy-Aware AAL Home Services

Due date of deliverable: (31-07-2010)
Actual submission date: (30-07-2010)

Start date of Project: 01 February 2010 Duration: 36 months

Responsible WP: Novay

Revision: final

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

0.1 Author

Author	Company	E-mail
Dirk-Jan van Dijk	Novay	dirk-jan.vandijk@novay.nl
Melvin Isken	Offis	melvin.isken@offis.de
Florian Winkler	NEC	florian.winkler@neclab.eu
Elena Cruz Martin	TID	mecm@tid.es
Anthony Remazeilles	Fatronik	aremazeilles@fatronik.com
Marc Laval	Wany	florence@wanyrobotics.com
Karin Gaustere	Wany	florence@wanyrobotics.com
Erwann Lavarec	Wany	florence@wanyrobotics.com
Dietwig Lowet	Philips	dietwig.lowet@philips.com
Stefan Gessler	NEC	gessler@neclab.eu

0.2 Documents history

Document version #	Date	Change
V0.1		Starting version, template
V0.2		Definition of ToC
V0.25		ToC revised (Gessler)
V0.30		1st complete draft
V0.4		Integrated version (send to WP members)
V0.5		Updated version (send PCP)
V0.6		Updated version (send to project internal reviewers)
Sign off		Signed off version (for approval to PMT members)
V1.0		Approved Version to be submitted to EU

0.3 Document data

Keywords	
Editor Address data	Name: Dirk-Jan van Dijk Partner: Novay Address: Brouwerijstraat 1, 7500 AN Enschede, The Netherlands Phone: +31 (0)53 48 50 349 Fax: +31 (0)53 45 50 349 E-mail: Dirk-Jan.vanDijk@novay.nl
Delivery date	

0.4 Distribution list

Date	Issue	E-mailer
	Consortium members	al_florence_all@natlab.research.philips.com
	Project Officer	Luiz.Santos@ec.europa.eu
	EC Archive	INFSO-ICT-248730@ec.europa.eu

Table of Contents

0 DOCUMENT INFO **2**
0.1 Author **2**
0.2 Documents history **2**
0.3 Document data **2**
0.4 Distribution list **2**
1 INTRODUCTION **6**
1.1 Objective **6**
1.2 Scope **6**
1.3 Outline **6**
2 SERVICE ORIENTED FRAMEWORK **7**
2.1 SOPRANO **7**
2.1.1 SOPRANO BUILDING BLOCKS 7
2.1.2 OPENAAL 9
2.2 Amigo **11**
2.2.1 BASE-MIDDLEWARE 12
2.2.2 INTELLIGENT USER SERVICES 14
2.2.3 PROGRAMMING AND DEPLOYMENT FRAMEWORK 16
2.3 ETSI M2M **17**
2.3.1 M2M ARCHITECTURE 17
2.3.2 AAL AS USE CASE FOR M2M 18
2.3.3 APPLICABILITY OF ETSI M2M TO FLORENCE 19
2.4 The SENSEI Real World Internet Architecture **20**
2.4.1 THE SENSEI SYSTEM MODEL 21
2.4.2 PRINCIPLES OF THE SENSEI ARCHITECTURE 22
2.4.3 SENSEI AAA AND PRIVACY SUPPORT 23
2.5 Conclusion **24**
3 MULTI PURPOSE ROBOTS **25**
3.1 Robots on the market **25**
3.1.1 R&D MARKET 25
3.1.1.1 Wheeled platforms 25
3.1.1.2 Humanoid platforms 27
3.1.1.3 Comments 27
3.1.2 CONSUMER ROBOTS 28
3.1.2.1 Domestic robots 28
3.1.2.2 Entertainment robots 28
3.1.2.3 Comments 28

3.2 Robotic platform integration in intelligent environment 29
3.2.1 COMPANIONABLE FRAMEWORK 29
3.2.2 ROBOCARE 29
3.3 Key components of service robots 30
3.3.1 ROBOT LOCALISATION 30
3.3.2 ROBOT NAVIGATION 31
3.3.3 HUMAN ROBOT INTERACTION 32
3.4 Programming frameworks for robots 33
3.4.1 FUNCTIONALITY SPECIFIC FRAMEWORKS 34
3.4.2 GENERIC FRAMEWORKS 34
3.5 Conclusion 36
4 TECHNOLOGIES FOR PRIVACY AWARE HOME SYSTEMS 38
4.1 Home Network Services 38
4.1.1 FEMTOCELLS 38
4.1.2 SET TOP BOX (STB) 39
4.1.3 HOME GATEWAY (HGW) 39
4.2 Set top boxes as Home Gateways 40
4.3 Privacy awareness and access control 41
4.3.1 PRIVACY / SECURITY FOR DATA COMMUNICATION WITHIN INTELLIGENT HOMES / ROBOTS 41
4.3.1.1 Privacy / Security for data storage 43
4.3.2 PRIVACY DEALING WITH MEDICAL DATA / INFORMATION 44
4.3.3 ID MANAGEMENT 44
4.3.3.1 Leading Identity Management Frameworks 46
4.4 Zero-Admin Execution Environment and OSGi 49
4.4.1 ZERO ADMIN TECHNOLOGY 49
4.4.2 OPEN SERVICES GATEWAY INITIATIVE (OSGI) 50
4.4.2.1 Main features of OSGi 53
4.4.2.2 OSGi for Mobile environments (M-OSGi) 54
4.5 Conclusions 55
5 CONCLUSIONS 56
6 REFERENCES 57

Table of Figures

Figure 2-1 SOPRANO architecture [Wolf2009] ... 8
Figure 2-2 SOPRANO sensor level ontology [Wolf2009] ... 9
Figure 2-3 openAAL architecture ... 10
Figure 2-4 openAAL as part of SOPRANO ... 10
Figure 2-5 Main Amigo Components ... 12
Figure 2-6 Amigo Base-Middleware ... 12
Figure 2-7 Amigo Driver ... 13
Figure 2-8 Amigo Intelligent User Services ... 14
Figure 2-9 Amigo Context Management Architecture ... 14
Figure 2-10 Amigo Interaction ... 15
Figure 2-11 Amigo Programming and Deployment Framework ... 16
Figure 2-12 Draft simple M2M Architecture (ETSI TC M2M) ... 17
Figure 2-13 Functional view on ETSI M2M, annotated ... 19
Figure 2-14 SENSEI Abstraction Level ... 21
Figure 2-15 Key entities in the SENSEI system model ... 21
Figure 2-16 SENSEI Support Services ... 22
Figure 3-1 Pioneer 3AT robot from MobileRobot Inc. ... 26
Figure 3-2 Wheeled robots for service: Kompai, Scitos A5, Care-O-bot and PR2 ... 26
Figure 3-3 Humanoid robots examples: Asimo, Reem-B, HRP2 and Nao ... 27
Figure 4-1 Femtocell Diagram ... 39
Figure 4-2 Home Gateway ... 40
Figure 4-3 Three Identity Management Models ... 46
Figure 4-4 Remote Administration of the Florence Robot ... 49
Figure 4-5 Areas where the OSGí service platform is used [OSGiWP] ... 50
Figure 4-6 Architecture of an OSGi-based system ... 51
Figure 4-7 OSGi ... 52
Figure 4-8 OSGi R4 Mobile specification ... 54

1 Introduction

1.1 Objective

This document provides an overview of the State of the Art in the topics relevant to Florence's Work Package 2 (WP2) "Architecture and Platform for Service-Robot@Home". The objective of the work package is to design the Florence Service Framework and the core infrastructure for the provisioning of Florence AAL and lifestyle services.

Within the work package the existing state of the art solutions will be used wherever possible and will be extended where needed in order to reach the requirements of Florence. We therefore start with a study to the current state of the art in order to lay a foundation for upcoming work within this work package.

1.2 Scope

The topics described in this document should be relevant to one of the tasks defined within this work package. There are three tasks defined in this work package apart from the state of the art research, one on the overall system architecture, one on the multipurpose robot platform and one task which deals with the privacy aware AAL home system.

1.3 Outline

This document is structured as follows: Section 2 provides an overview of service oriented frameworks, how existing (research) projects incorporate them and how they are structured. Section 3 takes a look at what kind of robots are available on the market today. An overview is given of core robotic principles and the most important programming frameworks for robots and how we can use them for Florence are discussed. In section 4, about technologies for privacy aware home systems, we have a look at home network services, privacy and access control, and zero-admin execution environment. Finally in section 5 we present our conclusion and envisioned future work.

2 Service Oriented Framework

In this section we describe frameworks, which provide technologies to integrate different domains (e.g., home domain, operator domain, service provider domain) and are defined across the network-, information- and service-layer. Both, the addressed domains as well as the considered layers, are most relevant for Florence.
All frameworks have been developed from a service point of view. They show architectural aspects from the viewpoints of the Independent Living use case, of smart homes, and of device interaction abstraction and its standardisation, and last not least of an architecture which is providing support services (in Florence called *enabling services*) to gain real world awareness.

2.1 SOPRANO

SOPRANO [SOPRANO] is a FP6 European Integrated Project which started in 2007 and will end in 2010. The main aim of SOPRANO is supporting Independent Living of older people in their own home. As it is stated in the public web site of the project the "*major objective of SOPRANO is to take a leap forward in the way users can interact with and take charge of their living environment, as well as to develop the way professional care personnel can support the SOPRANO users when called on to do so*."

The Soprano project intends, among other objectives, to design and develop a SW framework to cope with the main challenges that have to be overcome to build AAL system. That is: being flexible and adaptable, combine the expertise of the different stakeholders involved in the development of the system, being extensible (not just engineered for a single case), etc.

This architecture (which has been called *openAAL*) is being enriched and tested in several other research projects both at German and international level, as the FZI Living lab AAL [FZILL]. Besides, openAAL, as Florence, is OSGi-based.

The project takes special care of topics such as design and use of innovative evaluation methodologies for AAL-technologies tailored to AAL-domain specific requirements. Moreover, the technical approach of SOPRANO combines ontology-based techniques and service-oriented device architectures. In this sense, the core of the SOPRANO architecture has been released as open source under the name 'open AAL' [openAAL]. The rationale behind this architecture as well as its main building blocks will be presented in the following section.

Thus, Florence can benefit from the lessons learnt while developing SOPRANO/openAAL and explore its main components to integrate context information and decision making inside Florence.

2.1.1 SOPRANO building blocks

SOPRANO is based on an OSGi architecture, into where it integrates its different components and services. Figure 2-1 presents the general architecture for SOPRANO. As it will be presented later on this document, this architecture has evolved to what is known as openAAL.

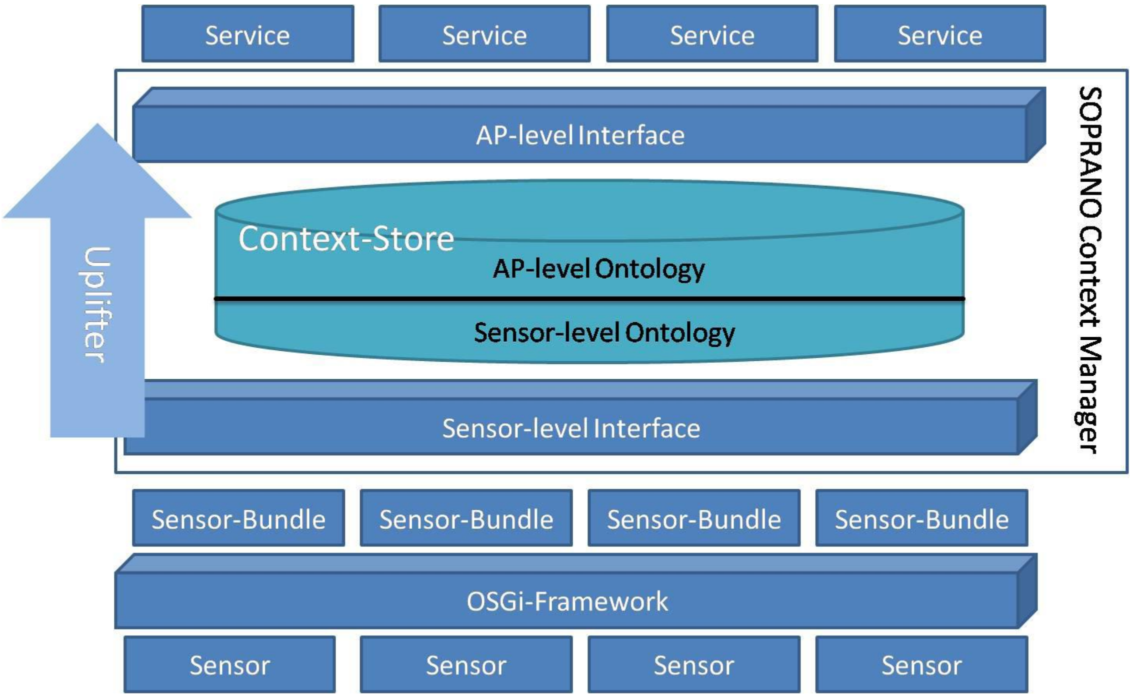

Figure 2-1 SOPRANO architecture [Wolf2009]

The main blocks of this system are:

- **Sensor-level interface:** mapping between the sensor-internal communication semantics and the SOPRANO semantics, achieved via the bundle that connects the sensor to the framework.
- **Sensor-level ontology:** models sensors and the states they can measure. See Figure 2-2 for more detail. . Sensors can transmit their information in form of context statements, which incorporate meta-information that can be exploited by augmentation and context-aware services.
- **AP (Assisted Person) level Ontology:** includes terms like activities, emergencies, location, etc. It is centred in the concept of person.
- **Interface for Context-aware services:** allows querying information in an easy way for the services.

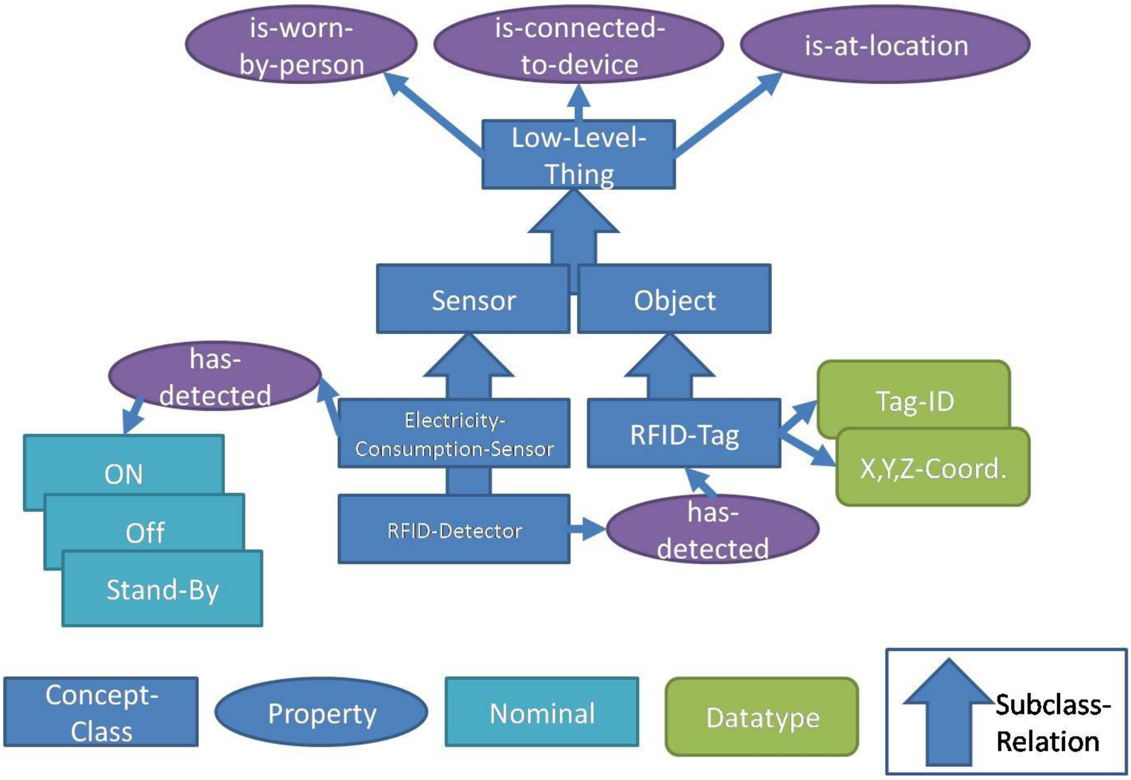

Figure 2-2 SOPRANO sensor level ontology [Wolf2009]

2.1.2 openAAL

The design and implementation of AAL solutions is usually a complex task, as the technical solutions proposed should be flexible and adaptable to the final user and changing needs. Thus, AAL solutions should be built upon platforms which enable the integration of different services, products and knowledge. In this sense, openAAL is an open source semantic middleware for Ambient Assisted Living, which has been created in order to enable flexible building of AAL products.

openAAL defines a framework on top of OSGi that allows the easy integration of and communication between services. Besides, it delivers a set of transversal services, such as workflow based specification of system behaviour and semantically-enabled service discovery that can be used to build more complex applications.

Figure 2-3 presents the openAAL architecture, which has been used into the SOPRANO project as a core component to be combined with an ontology in order to provide services to the different stakeholders involved in the care delivery process (see Figure 2-4).

openAAL uses OSGi as common layer for its main components. Thus, the interaction and communication between the modules is governed by the principles and methods of the OSGi framework.

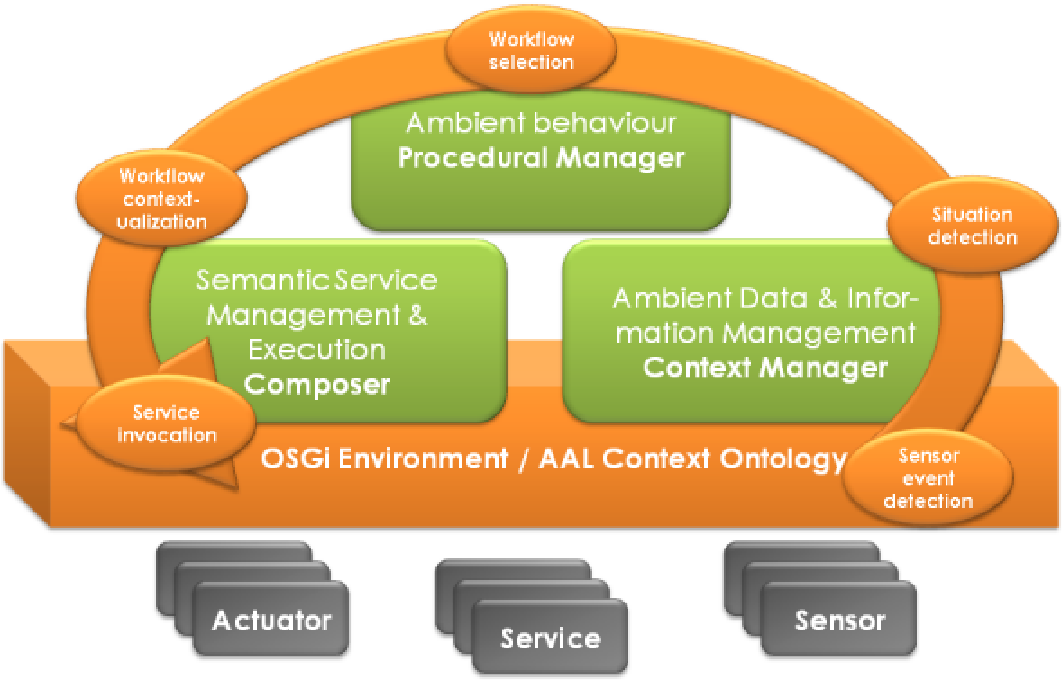

Figure 2-3 openAAL architecture

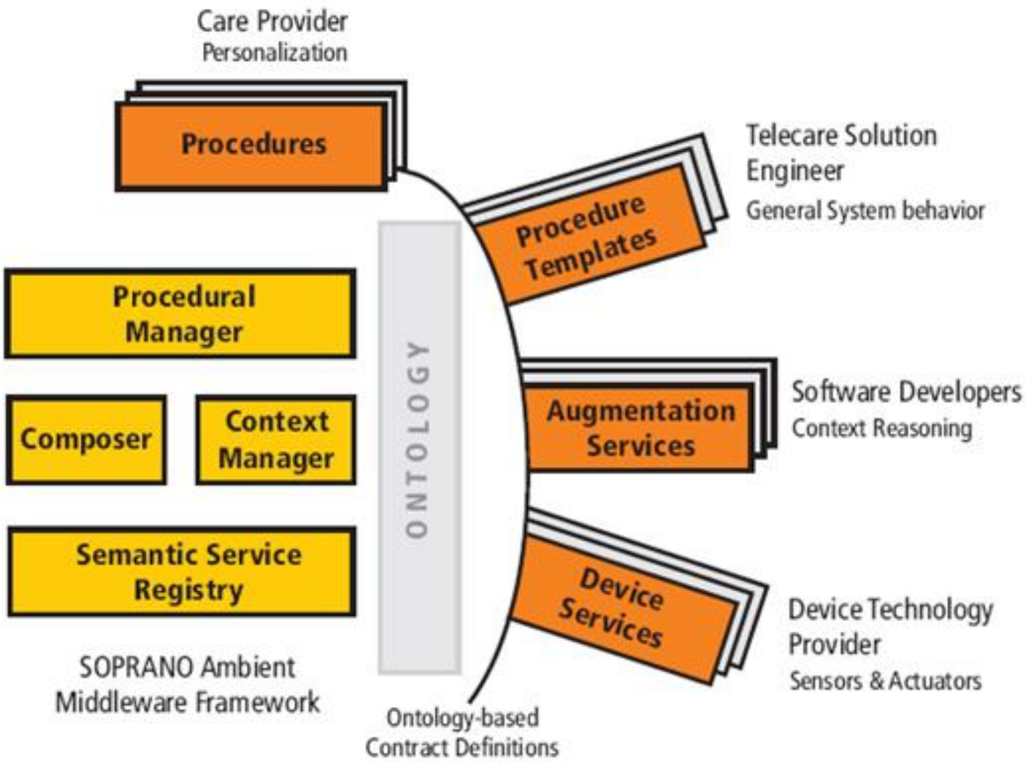

Figure 2-4 openAAL as part of SOPRANO

The main components of openAAL are [Wolf2010]:

- **Context Manager**, which provides ontology-based information storage used to capture information coming both from the sensors and the user. It allows deriving situations of interest based on different inference methods (such as Bayesian networks, logical reasoning, etc). The defined ontology supports context reasoning from a low-level sensor model (sensor, actuators and their states) to a high-level service oriented model, which describes the context of the assisted person.
  Besides, the Context Manager provides a *conflict resolution* component to enable complex situation detection and decision making. This module also provides a 'conflict-free' view for the different services that are offered. That is, it provides a consistent view on context at a specific point in time.
- **Procedural Manager**, to manage and execute different workflows which react to situations of interest. This module can communicate both in a synchronous and asynchronous way with the Context Manager via BPEL-based workflows.
  Workflows define the reaction of the system to given situations, which are identified by the Context Manager. Therefore, they can be used to resolve critical situations or to fix pre-defined behaviour patterns.
  The system reacts upon specific installation independent situations of interest instead of concrete sensor events, which means that the reaction is independent of hardware configuration.
- **Composer**, to analyse which services are available in a particular installation, selecting and combining those which are necessary to achieve the goals scheduled by the Procedural Manager. The matchmaking mechanism is based on the DIANE Service description framework [DIANE], but adding the notion of 'virtual services', which bridge the gap between abstract service requests and concrete service goals (describe the service as a set of configurable service states).

2.2 Amigo

Amigo tries to bridge the gap between home automation, consumer electronics, mobile communications and personal computing. Traditionally these are all separate domains, but by introducing the networked home, also called the connected home, all these domains should integrate together giving opportunity for new kind of services. Amigo was a FP6 project which started September 2004 and was finished in February 2008.

One of the key innovations that came out of the Amigo project is a feature rich and stable core middleware for developing smart home systems. The Amigo middleware offers uniform access to a set of heterogeneous hardware. The same underlying technique, OSGi, that was chosen for Amigo has been chosen by Florence which should help in reusing Amigo. Additional to the base middleware Amigo also offers some high level services, like context management or user interface services which we'll discuss in section 2.2.2, that might be of interest for Florence.

The Amigo project follows the paradigm of Service Orientation, which allows developing software as services that are delivered and consumed on demand. The benefit of this approach lies in the loose coupling of the software components that make up an application. Discovery mechanism can be used for finding and selecting the functionality that a client is looking for.

Figure 2-5 shows the main components of Amigo: base-middleware, intelligent user services and the programming and deployment framework. On top of these components "Amigo-Aware" applications can be build. We'll discuss each of these components in the following sections.

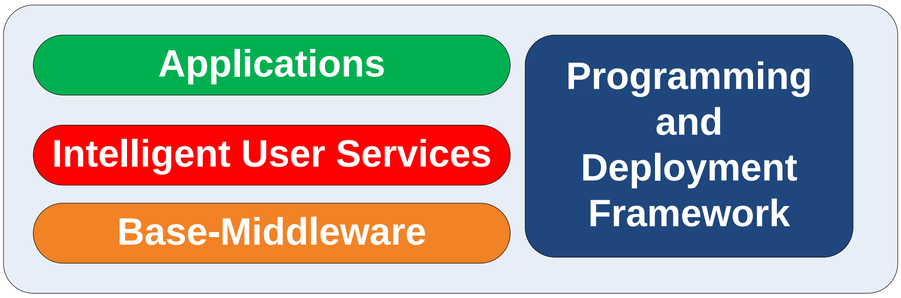

Figure 2-5 Main Amigo Components

2.2.1 Base-Middleware

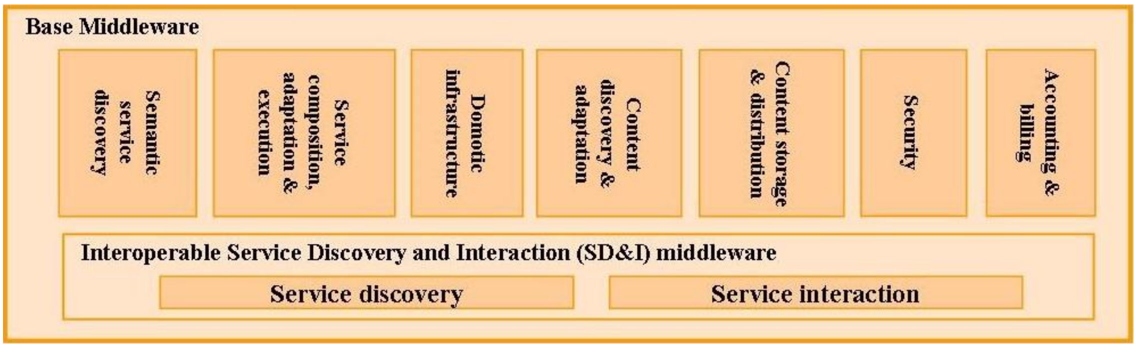

Figure 2-6 Amigo Base-Middleware

The Base-Middleware contains functionality to facilitate a networked environment.

At its core it provides means for discovering and interacting with services and devices in the network. Any Amigo-aware service or device can be discovered and invoked using a 'high-level driver'. This driver provides a Web Services or UPnP interface at the top regardless of the underlying technique used at the bottom. For example the Domotic Infrastructure component exposes a set of heterogeneous domotic services, like BDF and EID bus, and makes these available as Amigo Services.

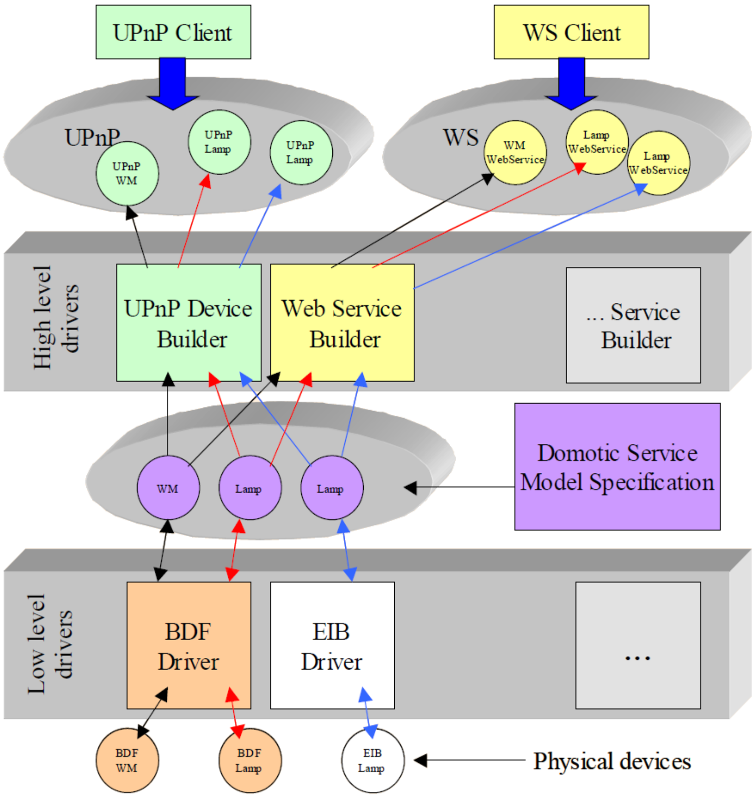

Figure 2-7 Amigo Driver

The Amigo networked home environment is supposed to be dynamic; meaning devices and services may enter and leave the environment at any given time so no guarantees on availability can be made. Furthermore the exact interfaces a service can offer or even its behavior is unknown on forehand. Therefore a semantic discovery mechanism is created. A service's capabilities are described using Amigo-S (a declarative language for semantic service description) and the Amigo Vocabulary Ontologies (a set of ontologies used for describing services, content, devices users and context). Upon requesting a service first a semantic service discovery is being executed. The results are being filtered on QoS and context properties as required. If no single service satisfies the request an attempt will be made to satisfy the request by using a composition of several services.

The content discovery, adaption, storage & distribution solution provides an overall interface to home content and related devices. Content in the Amigo home can be available from multiple sources and can be rendered by various devices. A content distribution mechanism is created to allow seamless playback between these sources and render points. A request for a content item is first passed to the content discovery component. This component locates the sources that can supply the requested item. The request is then forwarded to the multimedia session initiation component which then decides which device is best suited for playing back the item. If no suitable device can be found, due to for example the fact that no render device is strong/fast enough to playback the requested item, the request is sent further to the content adaption module which will transcode the content into the proper format.

Security mechanisms are also provided for authentication, authorization, and encryption. These allow for secure communication between the different services.

2.2.2 Intelligent User Services

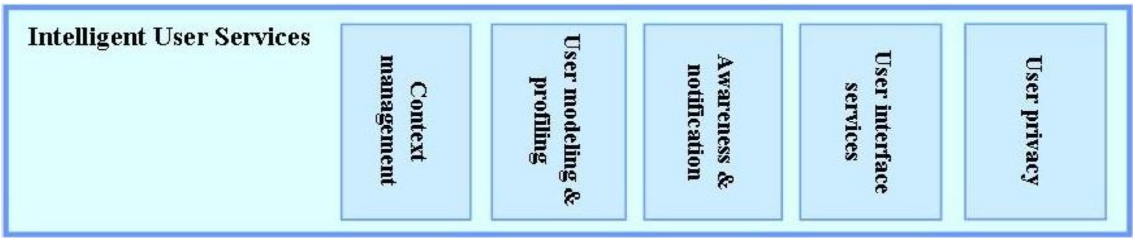

Figure 2-8 Amigo Intelligent User Services

The intelligent User Services (IUS) contain functionality that is needed to facilitate an ambient in-home network. They broker between users and services, and provide context information, combine multiple sources of information and make pattern-based predictions. Information is tailored to user profiles and adapts to the user's situation and changes in context.

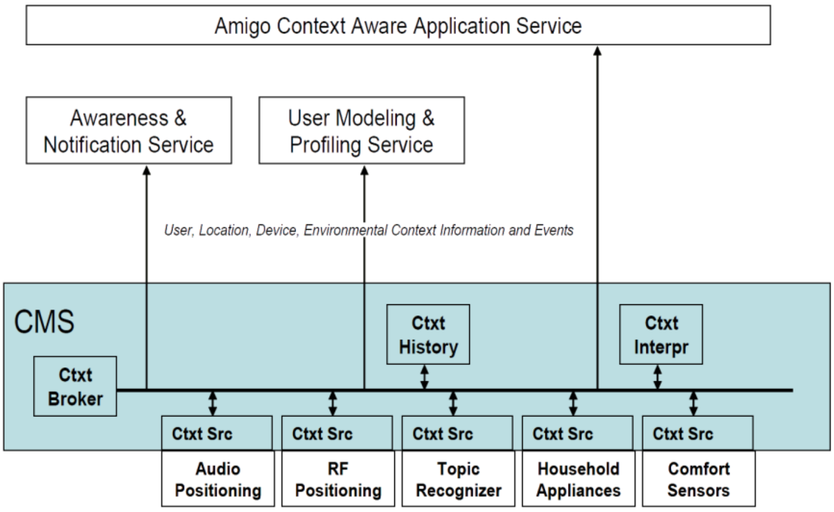

Figure 2-9 Amigo Context Management Architecture

The Amigo Context Management Service (CMS) provides an open infrastructure for managing context information. The CMS features a context broker where context sources are being registered. Context clients can ask the context broker to provide them with context sources which provide a certain type of context information. Every context source is designed as a web service providing a standard interface.

The CMS can acquire information coming from various sources, such as physical sensors, user activities, and applications. Subsequently the CMS can combine or abstract these pieces of information into high level context information. For instance the low level context sources might provide information that a user is: in the bedroom, the time is 22:30, TV and lights are off. A context interpreter can deduct that the user has most likely gone to sleep and provide this as new context information.

A high level language for exchanging context information between context consumers and context sources is being used.

Apart from providing context information to any Amigo aware application, the CMS is also used by other IUS like the Awareness & Notification Service and User Modeling &

Profiling Service (UMPS). The UMPS manages user profiles which can be used for personalization of Amigo services. UMPS takes context information into account when providing profile information. For instance a user's media preference might differ depending on his location (living, kitchen, etc) or presence information (if he's alone or together with his family).

The Awareness & Notification Service (ANS) makes it easier for developers to rapidly build applications that inform the users about their environment. Additionally it offers notifications tailored to the user's preferences and current context. Within Amigo the ANS rule language, based on Event-Condition-Action (ECA) rules [Dock2005], is defined. An ANS rule contains a context data event (e.g. a change in context), a condition which specifies what must hold prior to the execution of the action and the action itself which specifies a type of notification which should be fired.

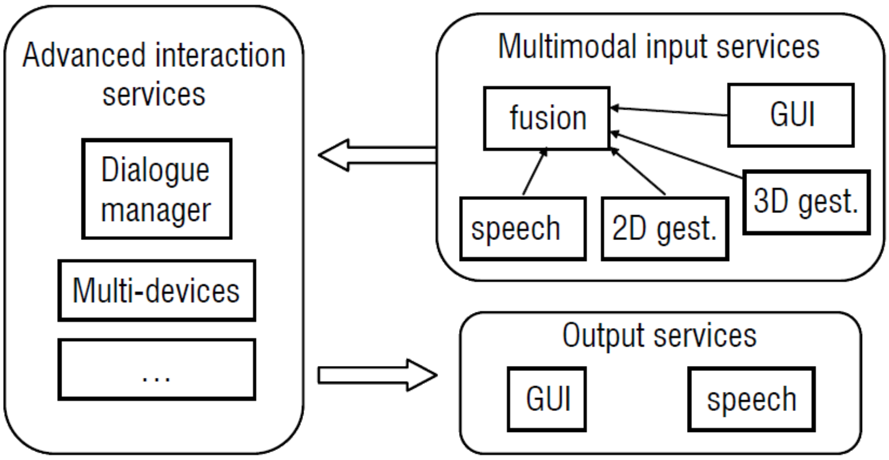

Figure 2-10 Amigo Interaction

Within the ambient home different modalities for interfacing with the system as well as multiple places for rendering UI might be available. The User Interface Service (UIS) tries to cope with this. Its purpose is threefold: (1) handle the interaction devices to present the contents, (2) manage multiple interaction modalities and their combinations, and (3) provide support for both explicit and implicit user interactions. First the Multimodal Input Services can provide multiple input modalities: speech, 2D gestures (typically realized with an electronic pen), 3D gestures (captured via a dedicated hand device) and Graphical User Interfaces. The input is captured, merged, encoded into Multimodal Interface Language (MMIL) and send to a dialogue manager, the application or to the CMS in case of implicit user inputs. The application can use the output services to send back information user either the GUI or Speech output service.

2.2.3 Programming and Deployment Framework

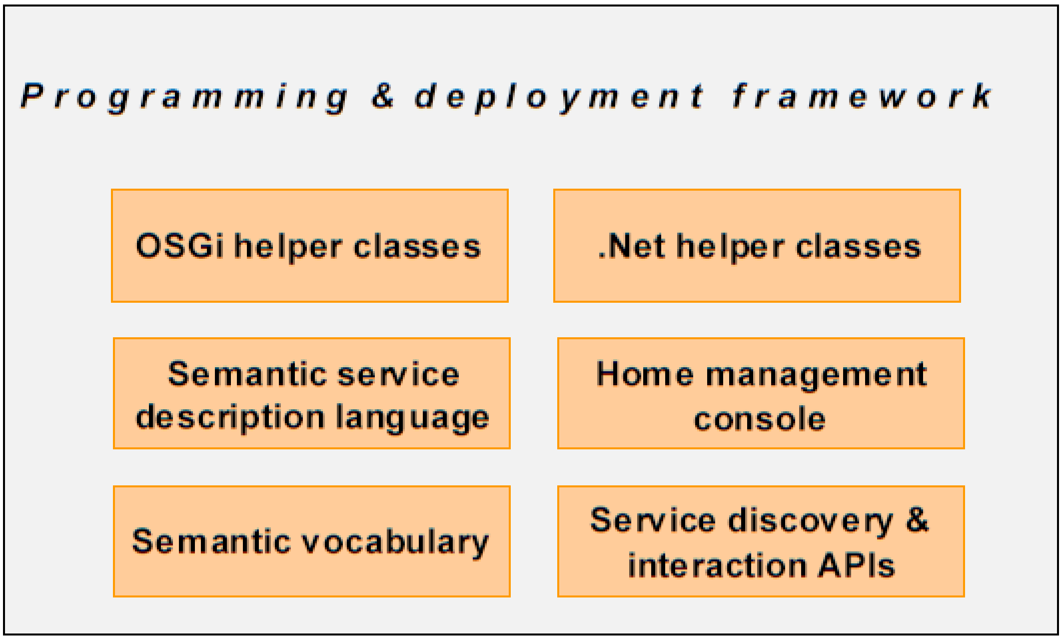

Figure 2-11 Amigo Programming and Deployment Framework

The Programming and Deployment Framework contains modules that facilitate the development of Amigo-aware services in .NET of Java as shown in Figure 2-11. Amigo abstracts over several protocols for discovery and communication. Developers can select the protocol of their choice while they can still access the functionality of services that are using different methods. Applications developed using either the .NET or OSGi framework can interoperate, thanks to the use of a common set of protocols for discovery, remote procedure calls and asynchronous event delivery.

The following protocols are being used in Amigo, all based on W3C standards:

- WS-discovery for publishing and discovering amigo services
- HTTP/soap for remote method invocation
- WS-eventing for subscription to event sources

Amigo services are described using the Amigo-S language and vocabulary, which allows services to be discovered, composed, adapted and executed within the Amigo home.

The Home Management Console provides a single point of control and diagnostics for the whole connected home. It is able to connect (remotely) to the different deployment platforms on the devices for control (software update) and diagnostic purposes).

On top of these components Amigo-aware applications can be built.

Amigo has the potential to offer a solid starting point for the development of Florence. The base middleware can help abstracting away from low level driver details, however we should investigate how well the Florence hardware is supported by the current set of drivers that Amigo offers and how much effort it takes to develop drivers for Florence specific hardware that's not supported yet. Some of the high level Amigo services like the CMS and UIS could be reused by Florence. However it is most likely that Florence specific requirements would need to be added. The gap between the Florence requirements and the features Amigo offers has to be explored in a reuse assessment.

2.3 ETSI M2M

ETSI [ETSI], the European Telecommunications Standards Institute, is a world-leading standards developing organisation for Information and Communication Technologies. With more than 700 members from 60+ countries from all over the world their standards do have global relevancy. One of the most prominent Standards of ETSI is GSM.

At the end of 2008, ETSI has installed the ETSI TC M2M, a so called Technical Committee on Machine-to-Machine (M2M), which is responsible to coordinate standardisation activities in the area of M2M and to define an architecture.
This work is ongoing, so any of the here provided information reflects the status until June 2010 and is based on publicly available material. Florence will track the progress of ETSI TC M2M and will reflect further developments within the work in WP7.

2.3.1 M2M Architecture

M2M involves communication without or with limited human intervention. However, the human still might be the target of an M2M service.

Figure 2-12 depicts an outline of the draft ETSI M2M Architecture.

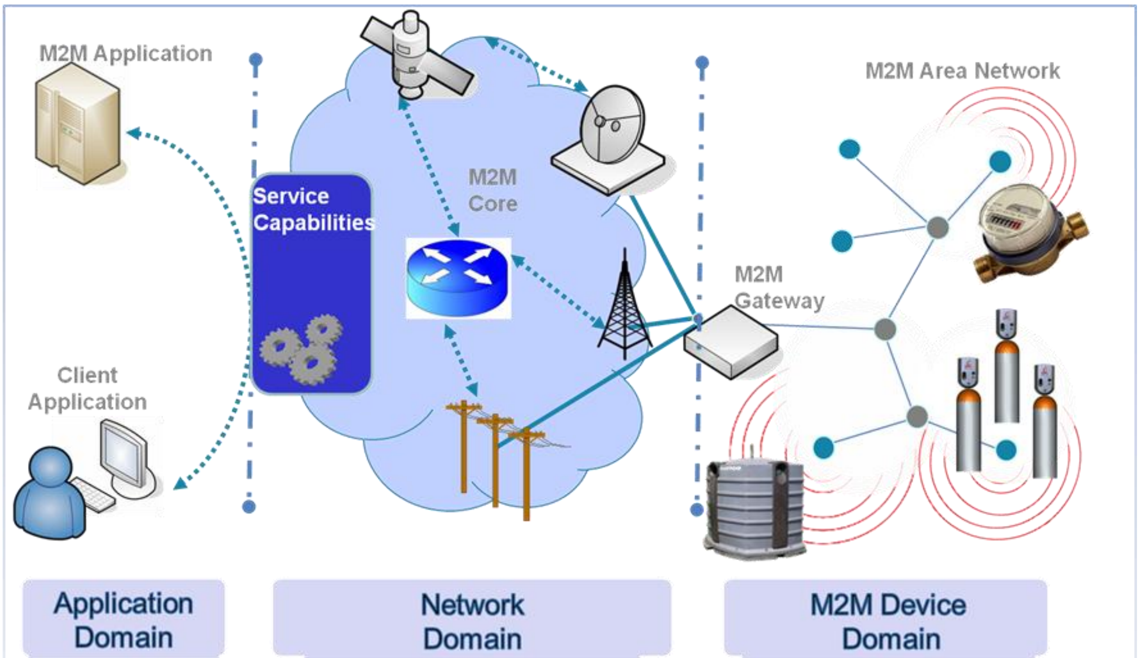

Figure 2-12 Draft simple M2M Architecture (ETSI TC M2M)

ETSI TC M2M defines six key elements for an M2M Architecture [ETSI2010-1]:

1. M2M Device
   A device that runs application(s) using M2M capabilities and network domain functions. A M2M Device is either connected straight to an Access Network or interfaced to M2M Gateways via an M2M Area Network.

2. M2M Area Network
   A M2M Area Network provides connectivity between M2M Devices and M2M Gateways. Examples of M2M Area Networks include: Personal Area Network technologies such as IEEE 802.15, UWB, Zigbee, Bluetooth, etc

3. M2M Gateways
   Equipment using M2M Capabilities to ensure M2M Devices interworking and interconnection to the Network and Application Domain. The M2M Gateway may also run M2M applications.

4. M2M Core
   Composed of Core Networks and Service Capabilities

5. Service Capabilities
   Provide functions that are shared by different applications. Expose functionalities through a set of open interfaces. Use Core Network functionalities and simplify and optimize applications development and deployment whilst hiding network specificities to applications. Examples include: Data Storage and Aggregation, Unicast and Multicast message delivery, etc.

6. M2M Applications (Server)
   Applications that run the service logic and use Service Capabilities accessible via open interfaces.

2.3.2 AAL as use case for M2M

ETSI TC M2M identified eHealth as one of the prominent use case areas for M2M [ETSI2010-2]. Acquiring information on the patient's health or fitness via sensor devices are here in the focus. Those sensor devices are supposed to be connected via short range networks to an aggregating device, which gateways to a back-end entity, that is supposed to store and react on the collected data. For the location of those sensors both options are considered: sensors worn by the user as well as sensors located somewhere in the environment of the patient.

Aging independently is seen as specific example for the eHealth Use Case area, especially in terms of remote monitoring, medication tracking and tracking the activity level of seniors to infer the overall health.
Those use cases are very close to the ones which are relevant for Florence. One of the objectives of Florence is to integrate service robots with intelligent home environments and addressing the sharing of contextual information to enable advanced AAL services.

The M2M System is also seen as a communication platform, which covers communication initiated by the Patient (either himself or a patient connected device) or by the Service Provider (again via real users which may also include family members, as well as without human involvement like autonomous services or devices). This again matches perfectly with the Florence view.

2.3.3 Applicability of ETSI M2M to Florence

The functional view of ETSI M2M is shown in Figure 2-13 Functional view on ETSI M2M, annotated. Mapping it on the Florence environment, both, the Robot and the home automation system with its sensors and actuators can be seen as M2M devices. Whereas the Sensors and actuators are simple devices connected to a Gateway, the Robot hosts also some M2M capabilities with M2M applications on top. However both are linked to an M2M Core Platform, which exposes Service Capabilities to the Application domain. Requirements of M2M do have their counterpart in Florence objectives, be it remote configuration (zero-configuration-toolbox), secure messaging system (ID based authentication) or Service Capability Entity (Resource management).

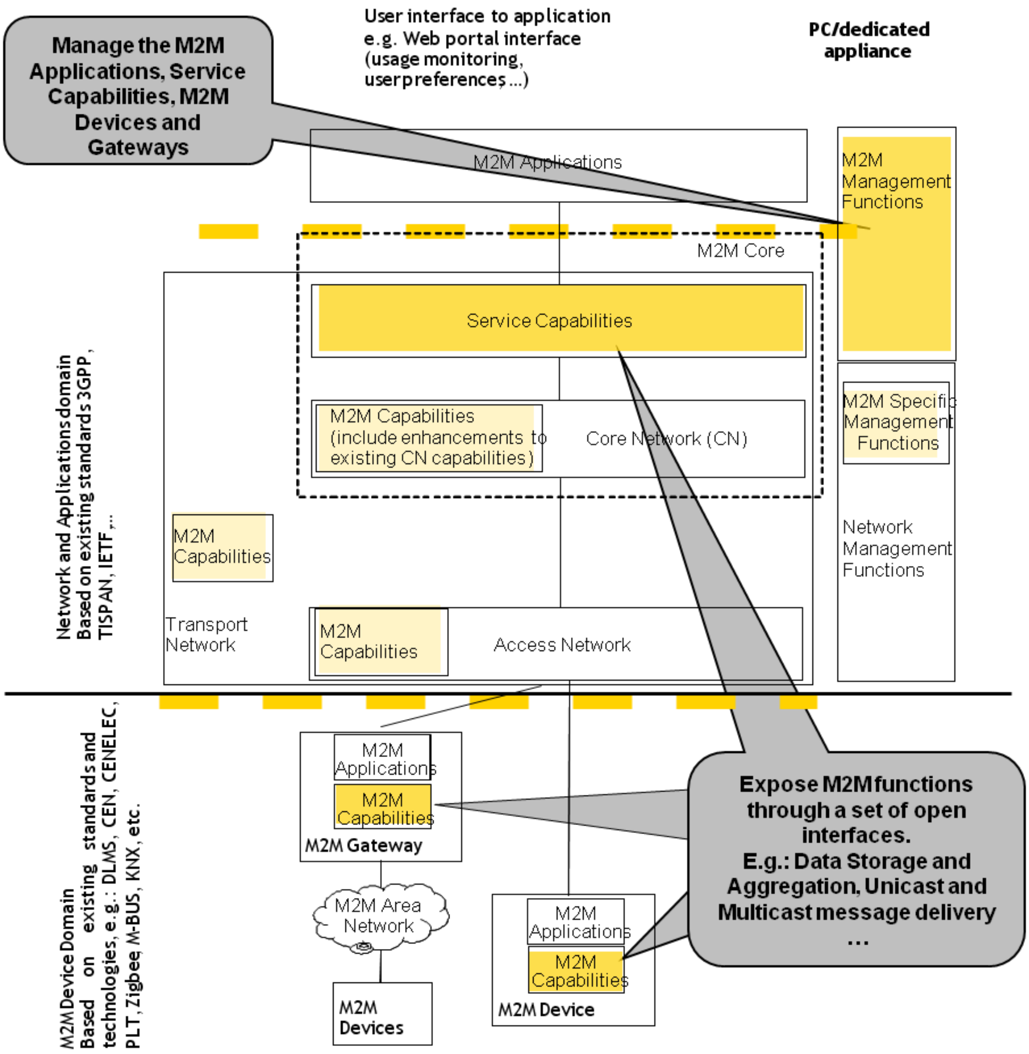

Figure 2-13 Functional view on ETSI M2M, annotated

Standardized architectures, interfaces and protocols, are essential for an economic success of products in a shared mass market. Therefore the initial results of ETSI TC

M2M and its further progress are considered to be relevant for the design of the Florence system and thus the further progress will be carefully monitored. The initial scope of ETSI TC M2M is quite open and the spans from low level network service capabilities to high level application service capabilities. However it is to be watched whether the activities on detailing the architecture will be restricted on the network level or whether the application area will indeed become relevant in the discussions. In the latter case, ETSI M2M might even become a target forum for Florence for active standardisation contributions.

2.4 The SENSEI Real World Internet Architecture

The EC co-funded project SENSEI [SENSEI] is currently working on a system architecture [SENSEI2010] which enables real world awareness. It leverages the concept of context-awareness towards the capability, to access and capture information from the physical world, transferring it seamlessly into the digital world, and enable sophisticated information processing. It realizes automated interaction with the real world through actuators. SENSEI support services provide means for automated resource configuration in a PnP like manner, as well as capabilities to compose complex information and mash-up services.

According to SENSEI the world that we live in is divided in a real and a digital world (see Figure 2-14). The real world consists of the physical environment that is instrumented with sensors, actuators and processing elements organized in Wireless Sensor and Actuator Networks (WS&AN) islands in order to monitor and interact with the physical entities that we are interested in: people, places and objects. The digital world consists of:

- Resources which are representations of the instruments (sensors, actuators, processing elements),
- Entities of Interest (EoI) which are representations of people, places and things and
- Resource Users which represent the physical people or application software that intends to interact with Resources and EoI (see Figure 2-14).

Bridging the physical and the digital world by allowing users/applications to interact with the Resources and EoI is the main contribution of SENSEI towards a Real World Internet.

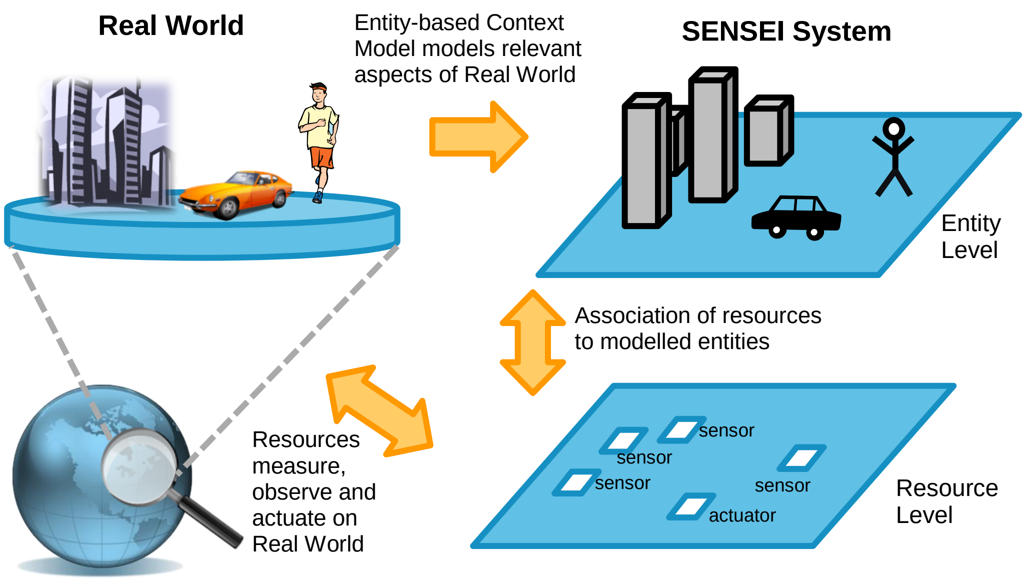

Figure 2-14 SENSEI Abstraction Level

2.4.1 The SENSEI System Model

At its core the SENSEI system model defines a set of entities and their corresponding relationships. The system model (see Figure 2-15) differentiates between the (physical) instances of system resources (Resource) and the software components implementing the interaction endpoints from user perspective (REP). Furthermore it differentiates between the devices hosting these resources (Resource host) and the network devices hosting the respective interaction end points (REP host). This separation allows the various system functions in the SENSEI system (explained later in the paper) to deal with real world dynamics in an efficient and adequate manner and allows for different deployment models of the system.

.

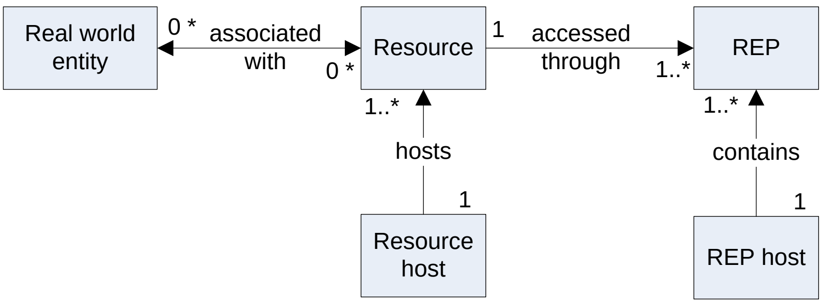

Figure 2-15 Key entities in the SENSEI system model

Unlike other systems, SENSEI considers also real world entities in its system model and manages the dynamic associations between real world entities and the sensors/actuators that can provide information about them/act upon them. Examples for real world entities (also known as Entities or Entities of Interest (EoI)) are persons, places, or objects of the real world that are considered relevant to provide a service to users. A SENSEI resource thus provides (context) information or interaction capabilities concerning associated real world entities.

2.4.2 Principles of the SENSEI Architecture

The SENSEI framework provides a set of support services for direct, rendezvous and context/entity based interaction, which are called SENSEI Support Services (see Figure 2-16).

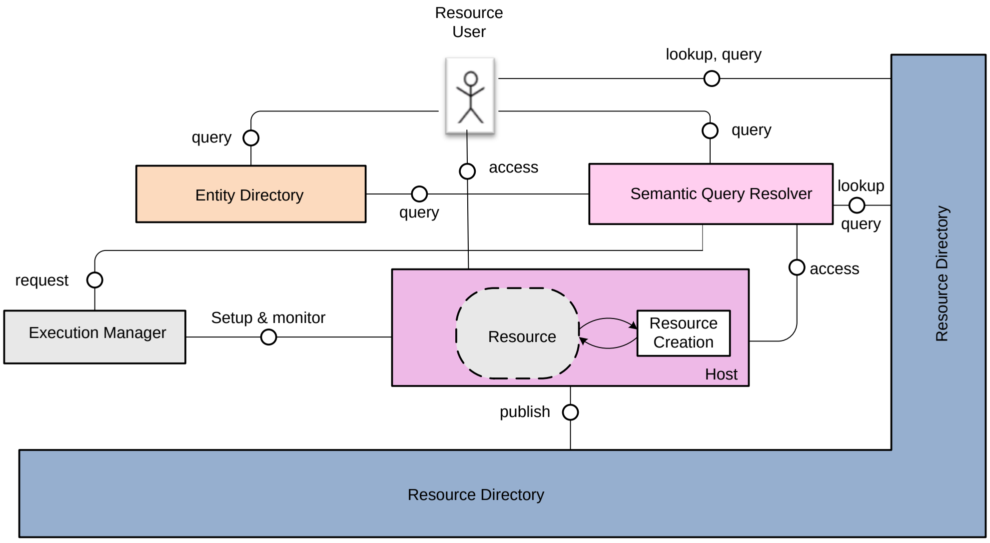

Figure 2-16 SENSEI Support Services

Rendezvous Function for basic Interaction

The first need that a user of the SENSEI framework may have is to find what information or interaction capabilities the framework can provide. To support this requirement SENSEI includes a basic *rendezvous* function which is split in two components:

- The *Resource Directory (RD)* that stores the descriptions of the resources, and can provide a list of resources that meet the users' criteria.
- The *Entity Directory (ED)* that maintains the associations between the properties of the EoI and the resources supplying information or interaction capabilities related to these EoIs.

The Entity Directory complements the Resource Directory. The focus of the latter is on the Resources while the Entity Directory focuses on the associations between the Entities and the registered Resources in the RD.
With these two components the user is able to find resources that provide some information and interact directly with them. However, these components just support basic search criteria (e.g. one type of criteria is simple keywords), they are not able to reason, combine or compose existing resources to create new ones based on the user needs.

Advanced Rendezvous Function and Service Composition

For these more advanced interactions SENSEI has designed the *Semantic Query Resolver (SQR)*. The SQR includes an advanced rendezvous function as well as advanced resource tasking and service composition. The SQR is able to receive queries that contain semantic information through a declarative language, analyze the queries and find the appropriate set of resources to satisfy the query using the basic rendezvous components (Resource and Entity Directory).

If the query includes any form of aggregation of information or interaction resources, the SQR will construct a plan involving a sequence of resources. If required resources do not exist, a resource provider will attempt to create and deploy them. An example query formulated in English is “Provide the average temperature of Stockholm”. The SQR will receive this query in a SENSEI formatted request and deduce that the entity of interest is a place (Stockholm) and the desirable property of the EoI is temperature. The SQR will consult the entity directory to find out the temperature resources associated with the entity Stockholm and according to a plan of action for such types of queries (average temperature), it will compose an aggregation (averaging) tree to produce the result.

Direct and Session-based Interaction

SENSEI provides different types of Interactions between users and the framework depending on the goal of the Interaction.

For direct usage of Resources the user can request simply a list of available resources that meet some criteria, in order to direct access those resources afterwards. An entity based interaction would address information about an entity of interest. Here the framework (either Entity directory or the SQR, see paragraph above) will identify an appropriate resource or a complete task and use them for acquiring the requested information, which is then delivered to the user.

Also the lifecycle of an interaction can divert. There is the option of one-shot interaction (where the user asks for information and gets it back) and the one for a long term interaction (where the resource can provide additional responses over the time).

Long term interactions, also called sessions, are supported by the *Execution Manager (EM)* (see Figure 2-16). The EM handles permanent delivery of information, events (i.e. notify the user when the temperature reaches 25ºC) and even more complicated functionality like the resource adaption (i.e. dynamically change the resource that provides the information back to the user due to a failure without any user interaction).

2.4.3 SENSEI AAA and Privacy support

SENSEI Architecture is providing a flexible framework to adapt the system depending on the deployment scenario. It allows for a tightly managed system with a centralized trust model as well as a dynamic and loosely managed system with multiple trusted parties involved. SENSEI AAA Architecture (*AAA: Authentication, Authorization, Accounting)* encloses a Security Token Service complementing normal AAA services.

The Security Token Service (STS) is an AAA support service, which provides the accessing Entity (AE, e.g. users) with the security assertions they will need to access the Access Controlled Entities (ACE, e.g. resources). The nature of the tokens being provided is dependent on the type of STS. The STS provides interfaces for token transfer within one administrative domain or between STS across administrative domains (federation).

Once Tokens have been acquired by the AE and provided with a service request to the ACE, this must establish whether or not the request should be granted. This decision is performed by the AAA service, and the answer returned to the ACE.

As the *SENSEI Framework* is mainly about sensing and communicating information, it is natural that privacy concerns arise. SENSEI identifies two categories of privacy issues in its system:

- **Real world privacy issues:** Real-world privacy issues include the privacy of personal information collected by sensors. Access to this information via the SENSEI framework is controlled by use of the AAA architecture to ensure that only those who are authorised to see and/or use the data can do so.
- **Electronic privacy issues:** Electronic privacy issues include people leaving digital traces of their movement and actions in various places. Privacy protecting systems aim at concealing the connection between the traces and the people who caused them. While traceability is necessary under certain considerations (legal proceedings, business process accountability, etc), this capability must in general be restricted to specific entities at specific times. The management of this capability is central to the flexibility of the overall privacy framework within the system. The AAA architecture provides a range of features to allow users to control how difficult it is to link their traces to them, such as use of pseudonyms or attributed instead of recognisable identities, single use tokens, etc.

SENSEI is a still running project, and is considered to be the project which paves the way for currently starting Internet of Things research. It has a remarkable history of predecessor projects in the sensor-, context- and security-area. Its research results are forming the global forefront of the State-of-the-Art in its area. Florence can benefit of the advanced knowledge on data models to describe heterogeneous information resources, the framework for processing of context information, the handling of actuation requests and finally of the sensor data security mechanisms. The approach of providing supportive services (enabling services) as essential part of a service framework, which enables the creation of mash-up services and which performs their execution can be exemplarily for the Florence Service Platform.

2.5 Conclusion

This chapter described relevant Service oriented Frameworks which reflected aspects of the Florence use case, smart home environment, device interaction and its standardisation, and an overall framework, which involves enabling services, privacy concepts and more. All frameworks provide means of context management and decouples information sources, information processing and information consumption. They give valuable advice how to design the overall architecture of Florence, which is suitable to enable the seamless integration of service robots, home environment, and (remote) service provider domain through the provisioning of supportive services like context management, automated configurability, actuation, privacy and interaction between stakeholder.

The presented approaches are valuable sources for Florence to learn from and to reuse relevant experiences, achievements or even developments. This helps Florence to focus on its specific objectives, rather than spending time in reinventing the wheel.

3 Multi Purpose Robots

This section is focusing onto the state of the art related to the use of a robotic system within the project. First of all, a brief overview of the characteristics of the robot available onto the market will be done in part 3.1. The following section will describe the robot that are already involved within larger architecture such as smart environment (as we will see, so far this embedment is mainly active within research projects, and have not yet reached the market). After presenting the basic components required to get a robot able to move and interact with the Human, the different software developments platform that are currently available to implement these functionalities will be presented.

3.1 Robots on the market

The robots available onto the market can be categorized by the end-user these systems are targeting. Three main categories can be identified:

- Systems that are so far mainly developed for investigation purposes,
- Systems already targeting personal or domestic use, and
- Systems dedicated to industrial applications.

We describe below the two first sections that are directly related with this project.

3.1.1 R&D market

In this segment, robots are considered as platforms onto which researchers can implement and validate their solutions. Most of the available R&D robots were originally designed by scientists and then made available on the market. R&D robots can be separated into two main categories: wheeled platforms and humanoid robots.

3.1.1.1 Wheeled platforms

Wheeled based mobile robots shown themselves to be the easiest way to build robots able to evolve in complex environments such as homes or factories but also outdoor, where the environment can rarely be controlled.

The availability of these platforms is dating back to the 70's and their design has evolved with the progresses made in various areas such as embedded computing, actuators and sensors.

The main regions of development for R&D platforms have been Europe, North America and more recently Japan and Korea.

MobileRobots Inc, formerly known as ActiveMedia Robotics, designed and made available several mobile robots platforms that have been used by researchers to demonstrate their projects, including human interaction applications.

Figure 3-1 Pioneer 3AT robot from MobileRobot Inc

In Europe, similar platforms are available and several companies developed robots with their own specificities. The Swiss company K-Team developed Koala that is a mid-size robot that can be used indoor and outdoor. In France, Robosoft with the RobuLAB platforms (see the Kompaï system in Figure 3-2) and Wany Robotics with the successive Pekee mobile robots iterations have been proposing R&D mobile platforms to researchers for many years. Both companies are investigating into new concepts of robots specifically targeting the prototyping and development of human interaction robots. The German Metralabs [METRALABS] company has designed the Scitos A5 robot [Muller2007], which is currently used within the companionAble project [companionable, Pastor2009]. The recent progress in object grasping and manipulation also influenced the robotic system design and production. As an illustration, in Germany Fraunhofer IPA is working on a mobile robot assistant aiming to assist human in their daily tasks. This goal is also shared by Willow Garage, an American company that designed the Personal Robot 2 (PR2) with two arms (note that this company also realized the telepresence system and Texas Alpha Telerobot).

Figure 3-2 Wheeled robots for service: Kompai, Scitos A5, Care-O-bot and PR2

As no hardware and software standardization has clearly emerged over the past years, all the available R&D robots rely on different architectures with a part of proprietary elements (mechanical and electronic design, low level software) but a general trend is the development of compatibility layers for the major software platforms and an harmonization of some of the hardware equipments of these robots. Indeed, the purpose of these platforms is to provide to the researcher a system onto which it is easy to implement some behaviors, while not spending too much time onto other issues that may not be the focus of their research. This point can be noticed in the effort of proposing some open software architecture, into which the integration of new behaviors is simplified. As an example, The Kompaï robot from Robosoft is provided with the robuBox open Software that is based onto Microsoft Robotic Developer Studio, the Scitos A5 is compatible with Player, the Care-O-bot robot can be programmed with ROS [ROS] or OROCOS [Orocos], and the PR2 is also controlled with the ROS environment. We can also note that The Pioneer 3At comes with its own library, ARIA, but the full source is provided, under the GNU General public license.

3.1.1.2 Humanoid platforms

A consequent part of the investigation on human-robot interaction is based onto the use of human-like robotic system. The complexity of such system (just to handle the stability and the walking capabilities) has delayed the production of such system, but their presence onto the market right now demonstrates that a higher maturity has been achieved.

Figure 3-3 Humanoid robots examples: Asimo, Reem-B, HRP2 and Nao

One of the most famous humanoid robots might be the ASIMO created by Honda, and which first prototype was produced in 1987. The last contributions were mainly focusing onto the integration of some capabilities to better interact, like the recognition of objects, human gestures and faces, as well as environments. The REEM-B is developed by a Spanish company, PAL technology member of a UAE firm. REEM-B is developed to be a service robot. So far, no delivery date has been announced. The HRP is a series of robot developed by the Japanese Kawada Industries. Once more, such system is directly available onto the market, but may be acquired by laboratories through some collaboration with the Japanese research organization AIST (like the joint AIST-CNRS laboratory between France and Japan). Another example of humanoid robot is the Nao from Aldebarran. The particularity of this system is that this 60cm height robot has been designed while having a strong concern about the production cost, so that one version can be obtained for less than €15.000, which is much more affordable for smaller research laboratories than the other humanoids. This latest system is also provided with a software development script, based on the Urbi architecture, and researchers have ported it to the ROS framework.

3.1.1.3 Comments

Most of the R&D robot providers have understood the interest of using some open programming platforms, to get an easy and generic management of their robot, to ease the use of these systems, and to take benefit from the development of the researchers. The application in which these robots are used strongly depends on their motion capabilities and on the activities of the underlying laboratories. Wheel-based robotics platform are frequently used to perform some autonomous localization and navigation proofs of concept, while humanoid-like robots are mainly used to demonstrate grasping capabilities or physical human-robot interaction. Nevertheless, this distinction is quite small and frequently crossed, especially with systems like Care-o-bot and PR2 that are equipped with arms and can also realize some physical interaction with its surrounding. It is also interesting to note that these advanced systems usually provide some built-in solutions for human-robot interaction. As an example, the Kompaï robot is equipped with some natural voice interaction, as well as some daily-life management (waking up, shopping list, calendar and meeting management, etc).

3.1.2 Consumer robots

If the more advanced robotic systems are so far only available for the research community, it has to be noticed that robot with lower capabilities are already available for the public.

3.1.2.1 Domestic robots

A first group of robots have been designed to take in charge some unwanted household chores like vacuum cleaning, swimming pool cleaning and lawn moving. The Roomba vacuum cleaner from iRobot is clearly the most famous system [iRobot]. Basically, the robot starts by scanning the room size (with distance sensor like infra-red ones), and then start to navigate within this place. Depending on the version (and the price) the navigation method is more or less developed. The line of home robot of iRobot is very successful; they have surpassed the 5 million units since 2002. This outstanding success has produced the emergence of a consequent concurrence, among which we can cite Neato robot that will soon be released its particularity being the embedment of a laser-based localization and mapping solution.

3.1.2.2 Entertainment robots

Entertainment robots or robotic pets have also known a great development in the first decade of 2000. The Sony's Aibo is a concrete example of such robot [Aibo]. Equipped of a set of touch and distance sensors, as well as some microphone, he is able to interact with the person, understand specific vocal commands. In the latest version, the camera embedded into the nose is used to detect the docking station.
The Aibo system is just an illustration of the large production of low-cost robots. Apart from the pet-like robot, other ones present a humanoid-like appearance, like the Robosapien [Robosapien]. This type of system provides some basic capacities integrated (infrared sensor, color recognition, some low-level vision-based capabilities, audio order recognition, etc) that gives them sufficient interaction capabilities and for an affordable price (of the magnitude of hundreds to thousands of Euros). Some systems also provide some telepresence capabilities, like Aibo, or the Spykee telepresence robot [Spykee].
It is interesting to note that even if these robots are numerous and frequently mentioned in blogs, and even if the underlying market seems promising, several lines of product have been stopped: Aibo's production was discontinued in 2006; the company initially producing the dinosaur Pleo, Ugobe filed for bankruptcy.

3.1.2.3 Comments

Consumer robots have known a great success in the domestic area. One of the conditions of such success might be a combination of an affordable price and the good-execution of a specific task. If the strong concurrence in the field has a tendency to make the prices go down, it may also limit the behavior extension on such platform (that would require some processing and sensing increase).
The entertainment robots take benefit of the nowadays low-price of sensors (infrared, touch, camera sensor) to embed them in well-designed systems to get some basic but still present interaction frameworks. The connection to internet is also a way to extend the robot capabilities through the network. If some of these systems provide programming tools to extend the behavior of the robot, these capabilities are usually quite limited in term of interaction. If some system presents higher capabilities, they do it at a price that could take them out of the domestic group.

3.2 Robotic platform integration in intelligent environment

It has to be said here that very few robots available in the market are embedded within a larger infrastructure. Most of the systems are indeed rather autonomous, in term of decision making and processing location. Most of the telepresence robots are directly connected to the Internet through a Wi-Fi connection, as if it was a private and classical computer (like the Carebot robot from GeckoSystem, or Kompai from Robosoft). So far, the main illustrations of robots embedded into an intelligent platform can mainly be observed within the investigation side. We comment here about two different approaches, as they are presented within the projects CompanionAble and Robocare.

3.2.1 Companionable Framework

In CompanionAble, the robotic system is considered as an element equivalent to the smart home environment [CompanionAble]. Both architectures present the same organization, with equivalent layers. The two elements are synchronized to share and diffuse the information so that they have a common understanding of the environment. Raw data and situation and context awareness are shared through blackboards. The system integration module uses the symbolic data of the two entities to deduce the current situation. Note that the smart home environment and the robot control environment are completed with a central server that takes care of some specific services and application, but also handles some tools to support the services.

3.2.2 Robocare

In the Robocare project [Robocare], the robot is considered as an *active interactor*, with respect to the *silent observer* that gathers all the fixed sensors data used to recognize the person's activities in the home. On a software side, the heterogeneity of the sensor's type and location (fixed or on the robot) has been managed through a Multi-Agent Coordination formalism (as a distributed Constraint Optimization Problem, as explained in [Cesta2007]) and is implemented onto a fixed server.

The robot is not considered as an entity different from the environment. In the distributed multi agent architecture used, the emphasis is put onto the software agents that provide services, whether they are located onto the robot or fixed into the home. These agents are considered as generic intelligent subsystems and enable to handle applications like activity monitoring, the dialogue manager (onto the robot screen or with a PDA), the mobile platform control, and the person localization and tracking for each of the fixed cameras.. Each agent can request an access to the services of another agent; an event manager handles this inter-agent communication (the communication is implemented using the e-service paradigm, i.e. with web services). This formalism enables the overall system to realize an active monitoring of the person action. Depending on the activity monitoring results, and according to a list of requirements per activity (like *take the pill after dinner*), decides if the robots needs to start an interaction procedure with the human.

3.3 Key components of service robots

Within their strategic research agenda for Robotics in Europe, EUROP[1] explicitly mentions the opportunities of robotics within the emerging service robotics sector: "[...] products will impact on our everyday lives by contributing high-value added services and providing safer working conditions" [Europ2009]. Like it is the case in our Florence project, it is said also that: "The aging population will drive the application of robotic technologies that improve the quality of life and assist people to live longer and more comfortably in their own homes."
This reference document provides a list of key technologies that are to be improved within the ongoing research to attain the expected the impact of robotics onto markets. We will here present some of these technologies mentioned, focusing on those that seem to us important in order to make the “robot at home” concept come true. In this context, a successful mobile system has to provide:

- A *localisation scheme* to estimate the position and attitude of the robot with respect to a reference frame.
- A *navigation solution*, to determinate the motions to realize to reach a goal position, while taking care of the obstacles encountered during the navigation.
- Some *interaction media* to provide information to the person.

3.3.1 Robot localisation

Laser sensors may be considered as the first sensing technology that enabled to perform some precise indoor-localization. It is already embedded in several robotic platforms (as an example, see the one proposed by Neobotix [Neobotix] It has also been the sensor initially used to define the famous technique of SLAM (for *Simultaneous Localization and Mapping*, see [OpenSLAM] or [Thrun2005]), that enables, with statistical techniques, to progressively build a map of the environment from the information given by the sensors, while using this map to localize at the same time the robot. Such technology is embedded within the promising Neato vacuum cleaner [Neatorobotics]. The outstanding advances on computers and algorithms made possible to reuse such methodology with vision sensors, and thus enabling to go from 2D to 3D maps [Neira2008].

If the use of stereo camera eases the reconstruction of an environment or the localization of a robotic system [Comport2010], the arrival of 3-D TOF camera (for *Time Of Flight*) may also have some impact onto the localization process, since this

[1] The European Robotics Technology Platform (EUROP) is an industry driven platform comprising the main stakeholders in robotics. Its goal is to strengthen Europe's competitiveness in robotic research and development and global markets, as well to improve the quality of life of European citizens.

sensor, through the combination of a camera with an illumination unit (LEDS or laser diodes) offers a high resolution 3D image of the environment [May2009].

When considering indoor environments, the localization scheme can also be simplified and improved by using some markers well placed within the scene and easily recognizable by the robot. The advantage of these markers is that they can also provide some semantic information about the associated location. The Northstar system, from Evolution Robotics is an example of solution based on IR-marker [NorthStar]. A device simply plugged onto the wall projects IR light spots with unique signature onto the ceiling, and the robot uses them to estimate its position and heading. The Stargazer system from Hagisonic uses specific infra-red reflecting landmarks to perform localization, by projecting infra-red light from the robot onto the ceiling where these reflectors are fixed [Hagisonic]. This system is used by several commercial robots in Asia (UROBOT, Dooly, iRobi-Q and Sentinel 3), and it is the localization tool for navigation that is using our Pekee robot.

Perspectives

As mentioned within [europ2008], the objective in the following years is mainly to improve the current available technologies, to lower their price and increase their shift from laboratories to industrial and personal environment. Within the domestic framework (vacuum cleaner, lawn-mowers), it is expected to move towards some topological representation of the scene (i.e. with placed labelled like living-room, bedroom, etc). SLAM and Visual Slam approaches are to be able to manage large environments (few kilometres), and also to handle dynamical (moving) objects and environments. Within a larger horizon, the localization systems should be able to handle unconstrained environments, and also to attach some symbolic interpretation to the object representation. It is also expected that the localization process will take benefit from the use of 3D sensors that will become more accurate and affordable. Furthermore, effort will be put to improve the efficiency and velocity of vision processes, by taking advantages of specific optical developments, and of MEMS (Micro-Electro-Mechanical Systems) technology to improve the integration of algorithms directly onto the sensor board.

3.3.2 Robot Navigation

Navigation consists in defining the robot motions needed to reach a desired configuration. Naturally, this operation depends on the level of knowledge the system has of its surrounding environment. A large part of the investigation in this field makes the assumption that the robot knows perfectly the scene (*i.e.* a map of the environment is available, and both current and desired position are located within this map). These methods are generally gathered into the keyword motion planning [Latombe1991]. A classical approach to achieve this planning is the potential field algorithm [Khatib1986], in which the robot is associated to a particle and the desired position to an attractive field, while the obstacles produce repulsive field. Nevertheless, this method is known to be very sensitive to local minima that could block the robot far from its desired configuration.
Another group of method, more explicit, consists in estimating a discrete collision-free trajectory to the desired position. This can be done either by generating some roadmaps, corresponding to a set of safe displacements within the environment (through visibility graph, Voronoï diagram, or Roadmap, onto which a simple shortest path can be used to get a path), or by decomposing the environment into free

navigation grids or cells (the famous A* algorithm can then be used to perform the path search [Hart1968]). To reduce the time of exploration, another alternative is to use some sampling based algorithm in which safe path are only searched between random robot configurations.

Recently, the outstanding innovation on vision machine and computer processing has enabled to consider another approach to navigation, in which the knowledge of a global map is not necessary. On one hand, three-dimensional solutions propose to automatically estimate the 3d structure of the environment during a learning stage in which a camera is moved within the considered scene. These teaching motions are then considered as reference trajectory the robot has then to reproduce [Royer2007]. On the other hand, topological techniques do not require to reconstruct the environment, but just use a set of reference sensor input (or images) to define not only the environment for localization, but also to provide the needed information for performing the navigation [Goedemé2007, Courbon2008]. Visual servoing maybe then used to realize the motion between two known images defining the environment [Chaumette2007]. Note to finish that the separation in between topological and three-dimensional solution is not complete; several works propose indeed a hybrid system using advantages from both approaches [Segvic2008].

Naturally, mobile robots have to take into account the fact that the environment they move into may differ from the knowledge they have of it, and, more than anything, they have to take into account the potential presence of unexpected obstacles. The detection of such obstacles can be done with different sensors, like infrared sensors, ultrasonic sensors, laser range, or even vision sensors. After some data processing the mobile robot gets a distance map of elements in front of the mobile. The robot needs then to modify its motion to keep a certain distance from them.

Several techniques have been proposed to avoid obstacles. Among others, one can cite the classical artificial potential field approach [Khatib1986], in which a repulsive potential field is associated to any element that is considered to be too close to the robot. The Vector Field algorithm [Borenstein1991] consists in generating a polar histogram of the space occupancy around the robot and selecting then the sector with the lower polar density as the driving direction. The Elastic band approach [Quinlan1993] is based on the definition of a deformable collision-free path. Its initial shape corresponds to a free path generated by a planner. When an obstacle is detected, the path is deformed according to an artificial force, in order to avoid the obstacle while keeping a smooth trajectory. We can also mention that some approaches take into account not only the velocity of the mobile robot (like the Dynamic Window Approach, see [Fox1997], or Global Dynamic Window Approach [Brock1999]), but also the velocity of the moving obstacle (Velocity Obstacle Approach in [Prassler1999]). To finish, it has to be noticed that the selection of the obstacle avoidance procedure is related to the sensors used, as well as to the navigation strategy of the robot.

3.3.3 Human Robot Interaction

Since the state of the art of WP4 is focused onto the interaction, we advice the interested reader to consider that document to get more precise information. This section just gives a brief overview of the topic.

The interaction in between the human and the robot is an active investigation field. Naturally, most of the robotic system targeting at interacting with the person are equipped of touch screen, speech synthesiser and recognizer. The use of touch screen is less important in humanoid robots, as we can notice onto Figure 3-3, which increases the importance of the speech interaction. This point might be problematic, and more especially for application targeted at seniors because, due to physical and cognitive decline associated with the aging process, seniors may not only have difficulty understanding simple sentences, but also producing an answer sufficiently well-articulated for the computer system to understand it [Montemerlo2002].

Since the body expression is also an important component of the human-human communication, several projects propose to equip robotic systems with some non verbal interaction media. In this context, human-like facial expressions are easily identified and recognized by human users [Brezeal2002], so that several robotic systems equipped with a screen are providing a human-face that can be used to express some information [Rich2009]. In the project companionAble system, the “virtual eyes” of the robot are two small screens, and different eyes configuration are used to express the importance or to attach some attributes to the message delivered to the person [CompanionAble]. The use of real character (i.e. not a virtual avatar onto a screen) to provide continuous feedback to the person is also studied by different projects. Onto the Cero robot [Hüttenrauch2002], a small robot with few degrees of freedom for the head and the arms is used to inform the person about the speech process state (active, parsing, realized, failed, etc.). The arms of the Mamoru are used also to attract the attention of the person [Mamoru]. The importance of these real gestures for interaction is also demonstrated by the development of several coaching robots for motivating people to practice exercise: the arm’s motion of the Taizo from General Robotix [GeneralRobotix] are essential since they corresponds to the motion the person has to replicate, within a fitness activity. The replication by Bandit of the human arm gestures is shown to encourage the user to better practice [Fasola2010][2].

Note to finish that the human motions can also be an input of information for the robot. For instance, in the Cogniron project the vision is used to identify objects that are referenced by the user through pointing gestures [Haash2005]. In the Mobile Robotics Lab from Coimbra, the Laban notation, a metric used to describe dance motion, is used within a probabilistic framework to enable the recognition of specific body motions (like “come here”, “good-bye”, etc.) [Rett2010].

3.4 Programming frameworks for robots

First of all it has to be said that there exist a substantial quantity of programming frameworks for robots, so that it is impossible to describe them all in few pages. In this section we will just highlight some of them, gathering these frameworks on the level of abstraction they provide to the programmer: in the first subsection, we will focus on framework aiming at solving some specific but bounded problematic, like navigation, while the second will introduce some framework aiming at providing a generic programming tool, whatever application is implemented.

[2] Mamoru, Taizo and Bandi are described within the Deliverable D5.1, please refer to this document to get more details.

3.4.1 Functionality specific frameworks

Initially, the development of software packages was mainly motivated by the need to solve the navigation problem for mobile robots. Indeed, this capability is required for any application running onto a mobile platform, even if the navigation may not be the core of the application. As an example, we can mention the CARMEN toolkit, an open-source collection of software developed by the University of Carnegie Mellon [Carmen]. This toolkit provides indeed the basic navigation primitives like base and sensor control, obstacle avoidance localization, path planning, and mapping.
In a short time frameworks delivered have considered other problematic than the navigation itself. In this sense, the OpenSLAM initiative gathers a set of programs dealing with the Self Localization and Mapping of Robot. This web platform enables any researcher to publish there their programs. This demonstrates another interest for providing some framework to the community: it enables to ease the comparison and benchmarking of the different solutions proposed.

Since a mobile platform is soon asked to interact with its environment, several frameworks have also been proposed in this context. For example, the software package Gatmo [GATMO], which is developed by Intel and Carnegie Mellon University that enables to perform a SLAM (self-localization and Mapping), while detecting, labeling and tracking static objects (like walls) and mobile one (like people) [Gallager2009]. The Open Computer Vision library [OpenCV] provides also an impressive set of programming functions for real time computer vision, and is now used in numerous research works (the system was used on the Stanley car that won the 2005 DARPA Grand Challenge)[Bradsky2008].
The design of robots with a large number of degrees of freedom has also been considered by specific libraries, like OpenRAVE [Diankov2008]. This programming environment has been especially designed to handle complex interaction with the environment (such as object manipulation) or to control humanoids robots. While the lower hardware level can be handled by a classical controller (like ROS or Player that are described latter), OpenRAVE provides well-adapted motion planning tools and high-level scripting tools.

Naturally, in parallel to the open-source philosophy, some mature software are also converted into successful products, which supposes to get a more reliable, documented and robust system. As an example, the Karto [Karto] software development kit, developed by SRI International, provides high-level functionalities such as localization, mapping, exploration, path planning, obstacle avoidance and also some network communication tools.

3.4.2 Generic frameworks

As we have seen in the previous section, the functionalities envisioned for robotic platforms may require the mutual use of several libraries. Another common issue relies on the management of the specific hardware used on each robot, that can easily change from one platform to another (as we mentioned before, for example, the obstacle detection may be handled with infrared, ultrasonic, laser or vision sensors, and each of these sensors are provided with numerous hardware solutions). This strong heterogeneity results in a costly integration of programs onto a given platform, sometimes just due to the complexity of communicating correctly with the embedded sensors or actuators. In a previous section we have seen that the research community is providing some development physical platforms that are prepared to ease the time of this integration. We can cite there of course the Pekee robot of Wany, but also the PR2 robot from Willow Garage, Care-O-Bot from Fraunhofer or the Nao system from Aldebaran.

On the software side, a set of frameworks provide an abstraction level enabling to be less dependent to the hardware used, and to reuse the drivers implemented by others. As an example, the Player [Player] framework can be considered as an interface, or API layer, in between the user code and the robotic components that are completely abstracted. The advantage of such approach is that the higher-level process becomes more independent to the lower level layers, and thus to the hardware components.

In France, the Cognitive Robotics Laboratory of ENSTA has developed the framework Universal Real Time Behavior Interface [URBI], which is now further developed by the Gostaï Company. Partially based on Player, URBI keeps a client/server approach and provide a scripting language that can be interfaced with all the popular programming languages. The CPU-intensive algorithmic is handled by classical code, while the behavioral scripting part is managed by the script language *urbiscript*, enabling some dynamic interaction during the program execution.

Different to the client/server framework of Player and URBI, Robot Operating System [Quigley2009, ROS] proposes a peer-to-peer approach, presenting the advantage, natively, of enabling the distribution of services onto different hosts, using a messaging layer based on XML-RPC. Basically, in ROS an application is defined by a set of nodes that is a set of processes performing computation. The communication is performed by “publishing it” in a “topic”, enabling then all the subscribed nodes to be informed. ROS presents the advantage to be platform, language and implementation independent.

The OpenRDK framework, developed within the University of Sapienza, Italy [Calisi2009], is an illustration of an agent-based structure, in which an agent is associated to a software process. The communication is realized with a blackboard-like object within a same process while inter-communication is handled with sockets. A current strong limitation of this framework is that it is mainly developed for the Linux environment. A similar limitation is encountered with the JDERobot framework developed by the Spanish University of Rey Juan Carlos [Cañas2007].

Simulation is also an important phase within the development and validation of robots. The hardware abstraction enables to validate with simulated data a code meant to be embedded onto the real platform. Player has been extended with some simulation capabilities, through its components Stage and Gazebo (2D and 3D robot simulators respectively). Gostaï sells the Webots framework that can be used for simulation within URBI, and generally speaking, all the programming frameworks provide tools to connect to simulation modules.

Naturally commercial solutions have also been proposed. One of the most famous is the Microsoft Robotics Developer Studio [MS-Robotics]. This framework uses a component-based or service-based approach, in which a runtime library is used for orchestrating the services (DSS, Decentralized Software Services) and another for managing the asynchronous, parallel tasks using message-passing (CCR, Concurrency and Coordination Runtime). A nice element provided by this framework is the visual programming environment, enabling to define graphically a program, using the usual drag and drop mechanism to instantiate and organize blocks corresponding to services.

3.5 Conclusion

The current attractiveness of multi-purpose robots is demonstrated by different evident factors. First of all, the great investigation effort within the service robotic sector proves that both industrial and academic structures foresee a potentially high impact of such systems onto people everyday-life. During the last years, quite an impressive quantity of new robotic systems aiming at providing assistance to the person, like the Kompaï, Scitos or Care-o-Bot robots have been developed. The design of humanoid robots pushes forward the similarity with the human, while also putting more emphasis onto the manipulation capabilities.

Nevertheless, so far, very few, if not none, of these systems have really reached the personal market. The robots currently available for the non-researchers are mainly dedicated to single "simple" chores (vacuum cleaner, swimming pool cleaner). Some other systems are targeting the interaction with the person, but they are still labeled and designed as robots for leisure or entertainment. Nevertheless we can imagine that the great market expected[3] will contribute to stimulate the competition, which will not only produce a cost reduction but also require a higher product differentiation, through the integration of innovative capabilities and functionalities.

Clearly, several issues need to be addressed to make service robots reach the personal market. Among others, a key issue clearly relies onto the functionalities provided by the system. Without mentioning here the upper layers related to the services we envision to produce within the Work Package 5, the robotic systems must provide some basic capabilities like the ones we have identified here, i.e. localization system, navigation tool and interaction medium with the human (the physical interaction with the environment and or the person has not been studied here due to the focus of the project). Current advances here concern mainly the elaboration of more robust and generic solutions, involving relatively low-cost but reliable sensors.

Another key issue relies onto the integration of such services within the robotic platform. As we have seen, this need is addressed by the proposal of various software platforms providing some level of encapsulation to ease the re-use of modules or enablers, and therefore speed up the development onto these robotics platforms. This abstraction can be at the level of the physical component, like the generic drivers provided by Player, or at a higher level of enablers, like what we can benefit with all the libraries developed with ROS. The gain can also be obtained at the level of the communication abstraction between the different modules like the script-based programming proposed by URBI or OpenRAVE.

To finish, we can note that if several architectures models have been thought for the robot itself, its integration within a higher architecture has been less studied so far. We have identified two main projects providing an explicit embedment of a mobile platform

[3] Up to the end of 2008, the entertainment robot sales represented 655 millions of dollar, while vacuuming-like robots represented 655 millions of dollar. But the two markets were described with a similar forecast for the period of 2009-2012 of 1000 millions of dollars [IFR2009].

within a larger scheme, CompanionAble and RoboCare. However, like a good internal architecture is a key issue to get an efficient multi-purpose robot, a good integration and or communication with a larger framework is a must to fulfill the objectives of an aware and communicant service robot, as we are here envisioning within Florence.

4 Technologies for Privacy Aware Home systems

The first part of this section describes some technologies for the home network and home infrastructure. The Florence robot should integrate as seamless as possible into the services supplied by the intelligent home environment. In the first case, these services are for data connection. There are different technologies like WLAN or GSM through which the robot could connect to the network services.

Another aspect is the security of the complete system, the privacy rights of the user have the highest priority. Some technologies for securing data processing, communication and user management are presented in the second part.

Since the Florence system will be based on the Java OSGi platform, this technology is also described. A main aspect is the Zero Admin functionality that Florence should provide to allow easy setup for the user and technical assistance via remote access.

4.1 Home Network Services

In this section we'll have a look at different technologies to enable network services within the home.

4.1.1 Femtocells

In telecommunications, a femtocell [Femto] is a small cellular base station, typically designed for use in a home or small business. It connects to the service provider's network via broadband (such as DSL or cable); current designs typically support 2 to 4 active mobile phones in a residential setting, and 8 to 16 active mobile phones in enterprise settings. A femtocell allows service providers to extend service coverage indoors, especially where access would otherwise be limited or unavailable. Although much attention is focused on WCDMA, the concept is applicable to all standards, including GSM, CDMA2000, TD-SCDMA, WiMAX and LTE solutions.

For a mobile operator, the attractions of a femtocell are improvements to both coverage and capacity, especially indoors. This can reduce both capital expenditure and operating expense. Providing a better service to end-users in turn reduces churn. There may also be opportunity for new services. Consumers benefit from improved coverage and potentially better voice quality and battery life. Depending on the carrier they may also be offered more attractive tariffs e.g. discounted calls from home.

Femtocells are an alternative way to deliver the benefits of fixed-mobile convergence. The distinction is that most FMC architectures require a new (dual-mode) handset which works with existing unlicensed spectrum home/enterprise wireless access points, while a femtocell-based deployment will work with existing handsets but requires installation of a new access point that uses licensed spectrum.

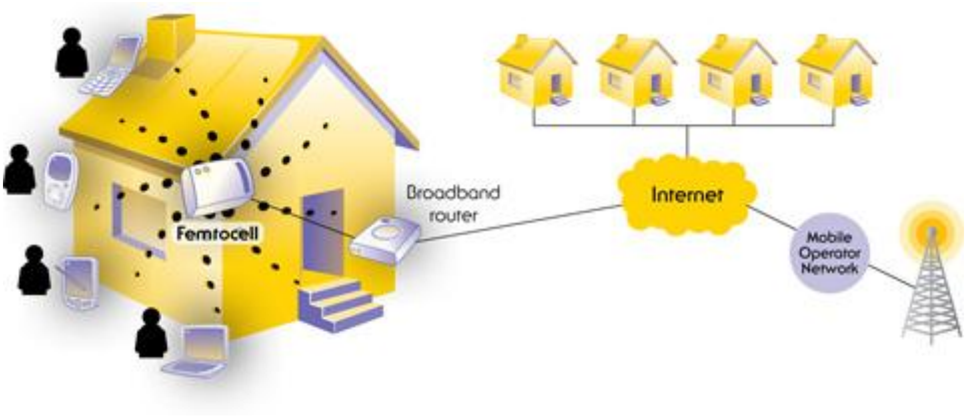

Figure 4-1 Femtocell Diagram

A femtocell can provide open or closed access. A closed access femtocell has a fixed set of subscribed home users - for privacy and security - that are licensed to use the femtocell. Open access femtocells provide service to macrocell users, if they pass nearby.

Ethical/legal dilemmas can arise on whether a femtocell should service macrocell users with poor outdoor coverage for making emergency calls, if they are located within its radio range. In open access networks, this problem can be solved by handoff. Closed access femtocells should be provisioned to allow communication with unsubscribed users in the event of emergencies. [Chan2008]

4.1.2 Set Top Box (STB)

Digital Set Top Box (STB) is a consumer device connected to a TV set to provide the services like Digital High Definition Television, Content Decryption (based on consumer subscription of pay channels), Personal Video Recorder, Electronic Programming Guide etc. Optionally it provides Web browsing and interactive television features. STB gets the content feed (channel programming) from digital cable, terrestrial or satellite broadcast and the return path for the interactive TV up-clicks could be either through a dialup or broadband modem which could be built-in or connected externally.

4.1.3 Home GateWay (HGW)

Home gateway [HomeGateway], also called residential gateway is defined as an intelligent network interface device located at the consumer premises. It provides the means for the residential user to access the Internet services delivered to home and also to access the different services offered by the various smart devices located within home. Essentially the home gateway device provides the necessary connectivity features to enable the consumer to exploit the advantages of a networked home. In technical terms, a home gateway device does the bridging/routing, protocol and address translation between external broadband network and the internal home networks. It acts as a secure firewall, and it is also the focal point for applications such as Voice/Video Over IP, home automation etc. Home gateway device allows the residential users to access their home networks and to control various devices from a remote location through Internet.

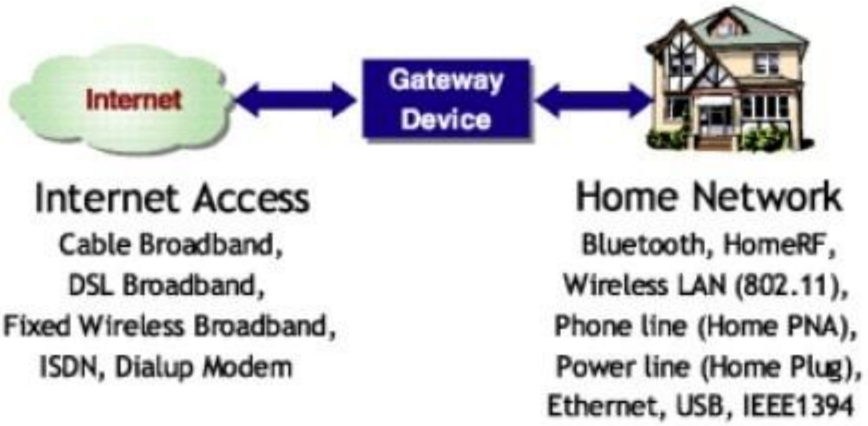

Figure 4-2 Home Gateway

There are multiple devices that have been described as "residential gateways" each with a different function. Each type of device allows the connection of a LAN (used in the home) to a WAN. The WAN can often be the Internet or can merely be a larger LAN of which the home is a part (such as a municipal WAN that provides connectivity to the residences within the municipality). WAN connectivity may be provided through DSL, cable modem, a broadband mobile phone network, or other connections.

The term "residential gateway" was originally used to distinguish the inexpensive networking devices designated for use in the home from similar devices used in corporate LAN environments (which generally offered a greater array of capabilities). In recent years, however, the less expensive "residential gateways" have gained many of the capabilities of corporate gateways and the distinctions are fewer. Many home LANs now are able to provide most of the functions of small corporate LANs.

Therefore the term "residential gateway" was becoming obsolete and merely implies a less expensive, lower capability networking device.

Nowadays, the home gateway tends to have abundant interfaces, powerful functions and a more user-friendly interface. It is a manageable terminal with auto-configuration, and multi-service perceiving and bearing. The home gateway provides Quality of Service to simultaneously support different types of services. As a part of the carrier network, the home gateway shall support remote control, detection and configuration.

4.2 Set top boxes as Home Gateways

The Digital Set Top Box device is fast evolving in to a Set Top Box home gateway, where in it caters to the data, voice and entertainment connectivity needs of the entire household and does not limit the connectivity to the TV set alone. This is attributed to the factors described below. [Gupta2004]

- Consumers demand to see the media content /programming and streaming audio/ video on multiple TV sets/display devices
- Advances in home networking technologies (especially Wireless LAN)
- The availability of multiple PCs, information appliances (like PDAs, Web Pads etc.), intelligent/smart home devices and increase of telecommuters

In addition to the traditional functionality of digital/HDTV, Content Decryption, Pay Per View, PVR, EPG, and Interactive TV, the STB home gateway device provides the below mentioned additional consumer benefits.

- High broadband connectivity to the multiple PCs, and the information appliances in the household.
- High speed Wireless LAN access to anywhere in the house.
- In-home file/print/device sharing
- Firewall security, parental protection and VPN connectivity
- Show the content programming (directly received through satellite, cable, terrestrial or stored in the STB PVR hard disk) on to multiple TVs and display devices located in various rooms of the house
- Streaming content and IP Video on Demand (received through broadband Internet connectivity or stored in the STB PVR hard disk) on to multiple TVs and display devices located in various rooms of the house
- Enables remote health monitoring and security surveillance applications

4.3 Privacy awareness and access control

In this section we'll provide a look at the privacy aspects of intelligent home, robot, remote, mobile platforms. Which state of the art solutions are available for secure communication between the home, robot, etc; which methods are available for allowing visitors within such an environment to connect and use the available services yet without giving up their privacy and which identity management solutions are there to secure the sensitive date that's being gathered in such systems.

4.3.1 Privacy / Security for data communication within intelligent homes / robots

Since the platform and the intelligent home communicate mostly with wireless connections, it is theoretically possible to disturb the system with attacks on the intelligent home infrastructure or the robot system. This section deals with these problems.

According to Philip Brey [Brey2005], "of all ethical issues that could be raised in connection to Ambient Intelligence (AmI), the issue of privacy has by far received the most attention". Proponents and critics of AmI agree that it has a significant potential to violate personal privacy. As Bohn et al. write [Bohn2005], ''Intelligent fridges, pay-per-use scenarios, and dynamic insurance rates paint a future in which all of our moves, actions, and decisions are recorded by tireless electronic devices, from the kitchen and living room of our homes to our weekend trips in our cars.'' (p. 9). And Langheinrich [Langheinrich2001] observes: ''With a densely populated world of smart and intelligent but invisible communication and computation devices, no single part of our lives will per default be able to seclude itself from digitization. Everything we say, do, or even feel, could be digitized, stored, and retrieved anytime later.'' (p. 280). Critics of AmI have condemned it for its alleged ability to create a Big Brother society in which every human activity is recorded and smart devices even try to probe people's thoughts. Proponents have admitted that privacy issues require the utmost attention in the design of AmI, and that a basic trust that it is protective of privacy is vital for its acceptance by the public.

Marc Langheinrich has asked this question in relation to Ubiquitous Computing and has argued that it has four special properties [Langheinrich 2001]:

1. Ubiquity. Ubiquitous computing is supposed to be everywhere, and computers will therefore be omnipresent and impact every part of our lives. Because of the pervasive role in our lives that Ubiquitous Computing is supposed to have, its privacy issues may affect us more deeply than those associated with other forms of information technology.
2. Invisibility. In Ubiquitous Computing, computers are supposed to disappear from view. Consequently, its users will not always know they are present, and will not always know what they are doing. If what they are doing is collecting and disseminating personal data, users may often not know this.
3. Sensing. Ubiquitous Computing makes use of sensors that perceive aspects of the environment. These sensors will be increasingly sophisticated, and may in the future allow high quality audio and video feeds from cameras and microphones smaller than buttons. They may also sense emotions, like stress fear and excitement.
4. Memory amplification. Ubiquitous Computing may allow for future applications that ‘‘continuously and unobtrusively record every action, utterance and movement of ourselves and our surroundings, feeding them into a sophisticated back-end system that uses video and speech processing to allow us browsing and searching through our past.’’ (p. 279). Ubiquitous Computing has the unique potential to be used to create a rather complete record of someone’s past, which has been called a life-log.

AmI adds to Ubiquitous Computing the technologies of IUI’s and Ubiquitous Communication. These technologies add two properties that are important in relation to privacy:

5. Profiling. Smart objects in AmI contain, construct and use unique profiles of users, including their unique characteristics, preferences, and behavioral patterns.
6. Connectedness. Smart objects have to be able to communicate with other devices, whether local or remote. Ideally, they will be able to form wireless and ad hoc networks with other devices, over which data is exchanged. Unless special safeguards are put in place, personal information may move freely over networks in this kind of architecture.”

So it is important for Florence not to increase this fear of permanent observation which can’t be controlled any more. First of all, within the user evaluations (focus groups, wizard of oz tests) these questions will be tackled. We want to know which functions the users would accept to increase their comfort or safety and which functions are not desired even if they would increase comfort/security. Based on these results, the Florence software will be designed for the user to have control on the “big brother”-level of the system.

Within the communication itself, the Florence software framework will use the highest available security standards (like WPA2 for wireless LAN) to assure no data could be misused by third party people.

4.3.1.1 Privacy / Security for data storage

Data that has been gathered by the Florence system may have to be stored on the device itself or other data mediums (USB Sticks and the like) so this gives an overview on existing systems for storing this data securely.

Most important for all transportable data volumes is data encryption of its contents. Data encryption schemes generally fall in two categories: symmetric and asymmetric. AES, DES and Blowfish use symmetric key algorithms. Each system uses a key which is shared among the sender and the recipient. This key has the ability to encrypt and decrypt the data. With asymmetric encryption such as Diffie-Hellman and RSA, a pair of keys is created and assigned: a private key and a public key. The public key can be known by anyone and used to encrypt data that will be sent to the owner. Once the message is encrypted, it can only be decrypted by the owner of the private key. Asymmetric encryption is said to be somewhat more secure than symmetric encryption as the private key is not to be shared. Asymmetric algorithms however do require more computation power. Securely transporting a shared symmetric key is a common problem which is often solved by using an asymmetric algorithm as bootstrap for setting up a secure channel. The asymmetric algorithm is used for sharing the symmetric key. Once this key has been transported both parties can use the symmetric key for the remainder of their communication. Another use for asymmetric algorithms is to prove the authenticity of data. In this case the message is encrypted with the private key allowing everyone to decrypt the message with the public key. Authenticity is proven since only the owner of the private key would have been able to encrypt the message this way.

Data can be encrypted on hard disks, CD/DVD discs, USB sticks or more general on any type of storage medium. Usually symmetric algorithms are used when storing data since they perform better and impose fewer requirements on the hardware. There are several operation modes when storing encrypted data ranging from only encrypting a file's content to encrypting an entire hard disk. An advantage when using full disk encryption is that for an outsider not only the contents of the files are hidden, but since the entire disk is encrypted an outsider also can't see the file names or even the number of files present on a disk. A full disk encryption system called BitLocker [Bitlocker] has been part of Windows since Vista. TrueCrypt [TrueCrypt] is another application which can do full disk encryption, other operating modes are also supported, and is available for Windows, Mac OS and Linux.

Encryption techniques in itself, symmetric or asymmetric, are considered safe nowadays if one uses a strong enough key. The biggest problem is storing the key in a safe place since secure keys are too long to be memorized by users a hardware token can be used to store the private key. In its most simple form the encryption key is for instance stored on a plain USB stick. So in order to get access to encrypted data the USB stick has to be inserted to obtain the key. In this simple case however anyone can use the key and it's easy to make a copy of the key. In a more sophisticated solution a Smartcard can be used where a private key is stored in a piece of protected memory which can only be accessed by an on board chip and requires a pin code. Even after entering the pin the private key will remain on the smartcard and the chip will only allow asymmetric operations which then can be used to decrypt an encrypted symmetric key or sign hash values of data which has to be signed. This way the private key will never have to leave the smartcard and even if a symmetric key is confiscated the smartcard can still be used in further operations. Entering pin codes can be done through a wired or wireless terminal, but since sometimes the terminal cannot be trusted there are also hardware tokens that contain their own interface for authenticating to the chip.

4.3.2 Privacy dealing with medical data / information

Here the context of medical data is considered. Most medical data has special security requirements to handle it because of the very personal information that is included in this data. So there are a lot of regulations dealing with different kind of medical data. Not only medical data but all personal data has to be protected. The article 8 of the “Charter of Fundamental Rights of the European Union” states therefore:

1. Everyone has the right to the protection of personal data concerning him or her.
2. Such data must be processed fairly for specified purposes and on the basis of the consent of the person concerned or some other legitimate basis laid down by law. Everyone has the right of access to data which has been collected concerning him or her, and the right to have it rectified.
3. Compliance with these rules shall be subject to control by an independent authority.

Directive 95/46/EC was adopted to harmonise national provisions on protection of individuals in processing and free movement of personal data. The directive has been implemented in all EU countries. This directive includes a detailed description of which data to protect and how.

As already stated, medical data is highly personal data which has to be protected against abuse. As the special needs of different medical areas are different, there are also different standards to deal with these data. For example analysing blood generates different data than imaging technologies like MRT.

There are additional policies which deal with medical data protection and development of medical devices which include data storage.

- Directive 2002/58/EC: Processing of personal data and the protection of privacy in the electronic communications sector
- Charter of Fundamental Rights of the European Union
- Medical device Directive (MDD 93/42/EEC)
- Directive 83/570/EEC, Regulation or administrative action relating to proprietary medicinal products
- Directive 2001/20/EC or Clinical Trials Directive of Implementation of good clinical practice in the conduct of clinical trials on medicinal products for human use
- World Medical Association Ethics

One of the outcomes of these policies is that patient data has to be anonymized. Only if the application can’t work without direct personal data the collection and storing is allowed.

4.3.3 ID management

Identity refers to the collection of information about an individual subject, typically a user. The role of Identity Management (IdM) is to manage and control the use of the identity related information over different services and systems and strives to do that in an easy-to-use yet secure way.

Identity Management is becoming more important due to the diversity of the services with personalisation as well as the prominence of security and privacy protection.

The user related information is used by Service Providers to personalise their services according to user profiles. The information is distributed as account information across different Service Providers for their own use which means that users need to manage their information across different accounts. This requires a lot of effort by the user, creating a demand for an easy way of managing distributed account information. At the same time, users are becoming increasingly concerned about their privacy. Moreover, regulatory and societal requirements on security and privacy protection are increasing, and there is a huge demand for identity information management solutions on the internet already today.

Figure 4-3 shows the three models in Identity Management:

- The independent model
- The user-centric model
- The federation model

The independent model is mainly used in legacy systems, where a single identity is handled by a single Service Provider.

In the user centric model, a user has multiple identities which are used by different Service Providers individually. Users can manage their identities by using an identity selector, providing a consistent user experience for authentication through user interfaces that allow users to choose the identity with which they interact with a service, while there are no relationships between the different identities from a Service Provider's point of view. CardSpace [CardSpace] and OpenID [OpenID] are based on this model.

In the federation model, the focus is on the management of identity that takes place between multiple administrative domains. In this model, a central component for the user, called an Identity Provider (IdP), is used to resolve the identity in one domain to the identity in another domain. Entities from different domains share a *circle of trust* with each other and the Identity Provider. This circle of trust can be established by contractual bindings between the business entities or by trust models that allow the establishment of the circle of trust on the fly. A circle of trust defines which entities can be trusted and considered to be not harmful.

The coupling of domains through a central component allows an identity to be authenticated and authorised in a domain in which it is not administered. Moreover, it allows the cross-exchange of profile information with other domains without disclosing the full identity of the user. Liberty Alliance [Liberty] Identity Federation is based on this model.

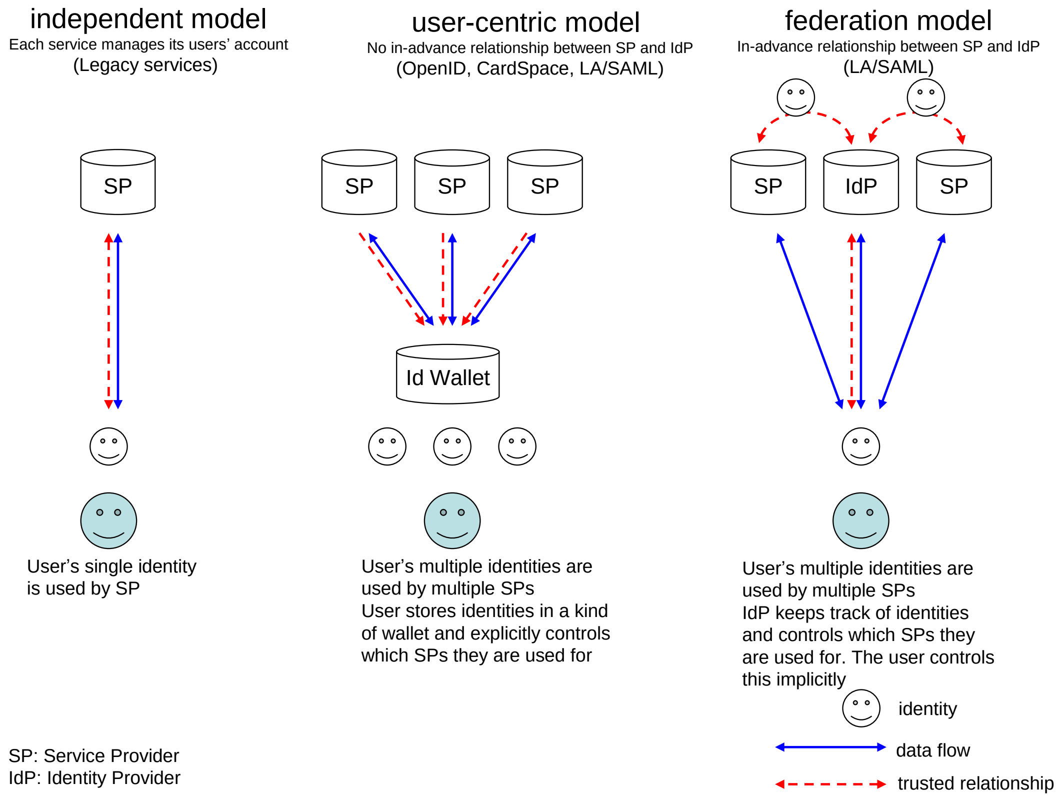

Figure 4-3 Three Identity Management Models

4.3.3.1 Leading Identity Management Frameworks

Liberty Alliance / Kantara Initiative

Liberty Alliance [Liberty] has been an alliance of a number of organisations. Its main focus was the establishment of open standards and development of guidelines and best practices for federated Identity Management. It promoted single-sign on, single logout, privacy and anonymity. The Liberty Alliance Project is discontinued giving all the work and related materials over to the Kantara Initiative [Kantara]

One particular standard that was influenced by Liberty Alliance is SAML 2.0 [SAML]. During its first phase, Liberty Alliance focused primarily on the Identity Federation Framework (ID-FF). In federated identity architectures, the user's personal information is not required to be stored centrally. It therefore requires user handles used by different parties (possibly from different domains) to be linked together (account linking in Liberty Alliance terms) without resolving to a global identity.

The second phase dealt with federation of Web Services (ID-WSF) and the communication of identity information in these Web Services.

Liberty Alliance also specified a protocol for the secure exchange of information about a user (i.e. a principal, in Liberty Alliance terms) which can be used in the case of federated identity information. The Data Services Template (DST) [LibertyDST] is an XML-based protocol for the exchange and management of user information that is distributed over several authorities. Additionally it provides mechanisms that allow a consumer of user data to subscribe to changes of that data. The providing authority can then notify the consumer of such changes.

Windows CardSpace

Windows CardSpace [CardSpace] is a framework developed by Microsoft which is based on the user centric model of Identity Management. Windows CardSpace takes the role of an identity meta-system - a unified, secure and interoperable identity layer for the Internet. It securely stores digital identities of a person and provides a unified interface for choosing the identity for a particular transaction, such as logging in to a website.

Windows CardSpace focuses on the “identity selector” as a user front-end as well as an Identity Management system for the backend. The user interface enables a user to manage and choose its digital identities depending on the context of their application. It handles both self-issued and managed identities which are identities that are confirmed by a trusted third party within the network. Based on those user attributes that have been disclosed, authorisation decisions on resources are done by the services offering access to the requested resources.

Windows CardSpace is built on top of the Web Services Protocol Stack, which is an open set of XML-based protocols including WS-Security, WS-Trust, WS-MetadataExchange, WS-SecurityPolicy, while instructions are provided about how to deploy CardSpace enabled Web Services. It is therefore compatible with any Identity Management technology that is supporting the WS-* protocols.

Windows CardSpace is integrated in Windows Vista, Internet Explorer 7 and Microsoft’s .NET Framework. Application examples of Windows CardSpace are Windows Live ID [LiveID] and the “Otto online” [Otto] shop in Germany.

OpenID

OpenID [OpenID] is a user centric Identity Management solution. It allows a user to identify itself to a website using a URL. This URL can be used by the website to determine where to authenticate the user. In order to achieve this, the website the user identifies itself to (the *relying party*) needs to be registered with an OpenID *Identity Provider* (OP). For example, AOL user Alice could gain access to an OpenID enabled website simply by identifying herself as alice@aol.com. The website would then contact AOL’s OP to have Alice identified.

The main advantage of OpenID is to allow a user to use a single password for multiple sites: the user authenticates to the OPs but not to the registered relying parties. It is light-weight and easy to implement. However, it neither provides single-sign on nor does it allow the usage of privacy-enhancing techniques such as pseudonyms.

The exchange of attributes is not supported in version 1.1 of OpenID. Version 2.0 supports a way of storing attributes centrally at the OP. Since the user chooses which OP to use, the user has the control over which attributes are released. OpenID is gaining popularity and is adopted by large organisations like AOL, BBC, Google, IBM, Microsoft, and others. OpenID is agnostic with respect to the authentication mechanism, and could for example work together with Windows CardSpace.

WS-*

The standards summarised as WS-* have been specified in several companies and standardisation bodies including OASIS [OASIS], WS-I [WS-I], IETF [IETF] and W3C [W3C]. They cover a variety of specifications associated with Web Services in varying degrees of maturity and are maintained or supported by various standardisation bodies and entities.

Among those specifications, the most relevant for Identity Management and federations are the OASIS specifications [OASIS] combined in WS-Federation for Identity Federation. WS-Security, WS-Trust, and WS-SecurityPolicy provide a basic model for federation between Identity Providers and Service Providers and define mechanisms to protect and authorise Web Service requests using policies and security tokens. WS-Federation extends this foundation to enable richer trust relationships and advanced federation of services. A good overview of WS-Federation can be found in [WS-FED], which shows that all the aspects of WS-Federation together enable new scenarios where for example “authorised access to resources managed in one domain can be provided to users whose identities and attributes are managed in other domains”, while “protecting the privacy of these [editorial: identity, attribute exchange, authentication and authorisation] claims across organisational boundaries”.

Others

Both Bandit [Bandit] and Higgins [Higgins] projects have created a reference application that showcases identity services that interoperate with the Windows CardSpace Identity Management system.

Higgins is an Identity Management framework that enables users and systems to integrate identity, profile and social relationship information across multiple sites, applications and devices and is compatible with the user-centric “Information Card”-based (or “i-card”-based) approach.

Bandit is also an Identity Management framework that focuses on defining a common identity framework related to the enterprise usage, to provide authentication and credential caching and enable common roles and authorisation.

NetMesh LightWeight Identity (LID) [NetMesh] is a technology that enables users to keep control over and manage their identities. It is the original URL-based user centric Identity Management technology and is decentralised. Additionally it supports multiple underlying protocols such as OpenID, Yadis and others.

Yadis [Yadis] is a service discovery system for Identity Management allowing Service Providers to use identifiers for authentication, accountability, privacy-controlled data exchange, to determine automatically, without end-user intervention, the most appropriate protocol to use.

4.4 Zero-Admin Execution Environment and OSGi

4.4.1 Zero Admin Technology

The Zero Administration Technology is already a known technology for managing different computers from one central place. A quite common one is the Zero Administration Kit from Microsoft for their Windows NT Workstations. With the help of this software kit an administrator can install and configure lots of workstations by using its own desktop.

OSGi's architecture of services enables similar functions for the developer. It is possible to start and stop services at runtime, so a management of the system from outside is possible.

System Services from OSGi provide horizontal functions that are necessary in virtually every system. The Log Service, Configuration Admin Service, Device Access Service, User Admin Service, IO Connector Service and Preferences Service are examples of system services. This includes the *Configuration Admin Service* – This service provides a flexible and dynamic model to set and get configuration information.

With being able to control the system from outside (e.g. via network, Figure 4-4) most configuration and runtime errors can be solved by remote access. The user doesn't have to deal with difficult software problems or log files etc.

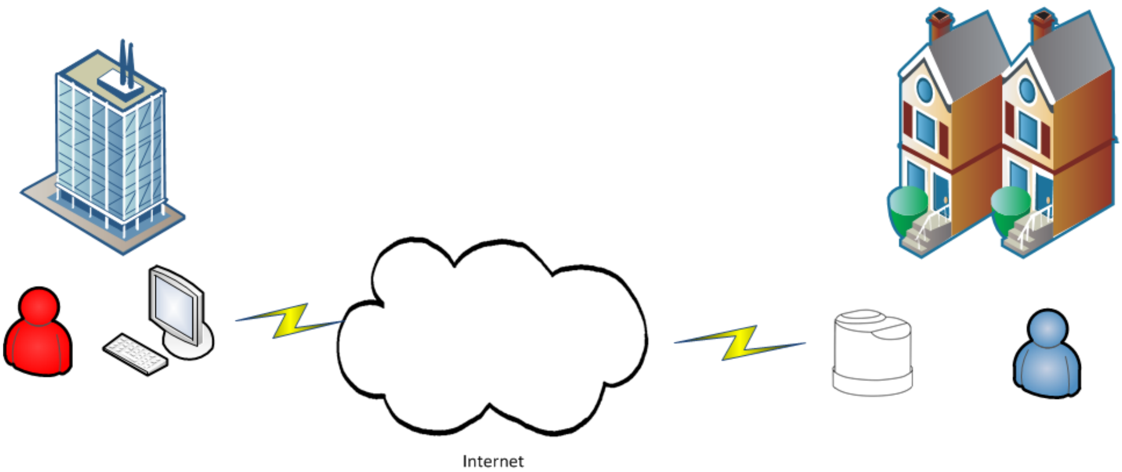

Figure 4-4 Remote Administration of the Florence Robot

Within this project, this technology will be used to update the system without the need of direct user interaction (of the elderly). In the case that a new device is added to the home environment it is possible to update the drivers for this device. The elderly doesn't have to care about any technical configuration. This could be a new heart rate meter or ECG. So the Florence system can be up-to-date all the time and catch up with new developments in medical device or home networking technology.

4.4.2 Open Services Gateway Initiative (OSGi)

The OSGi Alliance [OSGI] was formed in 1999 with the objective of specifying and promoting the adoption of an open service platform for the delivery and management of multiple applications and services using networked services. OSGi appeared, in this sense, as an Open Services Gateway initiative.

The first aim was to focus it on residential environments, for Internet gateways with home automation applications. However, this technology has soon attracted the attention of other markets.

For example, Siemens is developing some household components for home automation using the OSGi technology to add some innovative features to them, such as the functionalities to update and add drivers, interfaces and scenarios after the product is installed in the final customer's home. Also, some digital mobile phones include OSGi technology: Smart phones from Nokia and Motorola use this scalable, flexible and reliable unified service platform as a mechanism to place new deployed applications. Furthermore, automobile industry tends to convert vehicles into integral element within a network of information and services, connecting cars with the outside world and road traffic processes to provide drivers with information services for an individualized and ergonomic way of driving. In this sense, BMW uses the OSGi specifications as the base technology in the 5 series models, to centralize functionality with a non proprietary technology and manage upgrade functionally in an easy way. Finally, Eclipse, a popular integrated development environment (IDE), uses OSGi specifications in order to deploy and update plug-ins on the fly without restarting.

Figure 4-5 Areas where the OSGí service platform is used [OSGiWP]

OSGi technology is a dynamic module system for Java™. The OSGi Service Platform provides functionality to Java that makes Java the premier environment for software integration and thus for development. Java provides the portability that is required to support products on many different platforms. The OSGi technology provides the standardized primitives that allow applications to be constructed from small, reusable and collaborative components. These components can be composed into an application and deployed.

The OSGi Service Platform provides the functions to change the composition dynamically on the device of a variety of networks, without requiring restarts. This means that software components can be installed, updated, or removed *on-the-fly* without ever having to disrupt neither the operation of the device into the OSGi framework, nor the framework itself. To minimize the coupling, as well as make these couplings managed, the OSGi technology provides a service-oriented architecture that enables these components to dynamically discover each other for collaboration. The OSGi Alliance has developed many standard component interfaces for common functions like HTTP servers, configuration, logging, security, user administration, XML and many more. Plug-compatible implementations of these components can be obtained from different vendors with different optimizations and costs. However, service interfaces can also be developed on a proprietary basis.

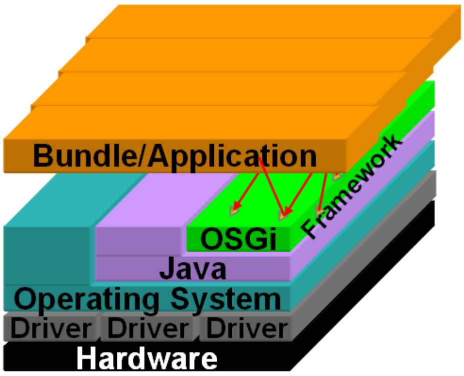

Figure 4-6 Architecture of an OSGi-based system

As depicted in Figure 4-6, the OSGi platform is developed on top of Java. It forms a small layer that allows multiple Java-based components to efficiently cooperate in a single Java Virtual Machine (JVM). It provides efficient and fast interconnection capabilities to components, and an extensive security model, so that components can run in a shielded environment where they can be reused and cooperate. Unlike other Java application environments, the OSGi framework gives the necessary tools for the management of these cooperative services and applications. The usage of Java as programming language allows OSGi to be portable and platform independent.

The core component of the OSGi Specifications is the OSGi Framework. The Framework provides a standardized environment to applications (called bundles). The Framework is divided in a number of layers.

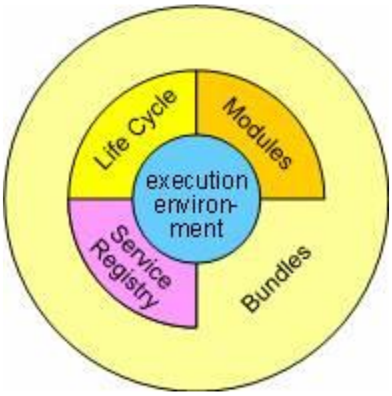

Figure 4-7 OSGi

- L0: Execution Environment
- L1: Modules
- L2: Life Cycle Management
- L3: Service Registry
- A ubiquitous security system which is deeply intertwined with all the layers.

Security is based on Java and the Java 2 security model. The language by design limits many possible constructs. For example, buffer overflows used in viruses are impossible. Access modifiers in the language restrict the visibility of the code to other programmers. The OSGi platform extends this model by allowing private classes, a mechanism that is not available in a standard way in Java. The Java 2 security model provides a comprehensive model to check access by code to resources. The OSGi platform adds full dynamic management of the permissions.

Security

One of the goals of the OSGi Service Platform is to run applications from a variety of sources under strict control of a management system. A comprehensive security model, present in all parts of the system, is therefore a necessity. The OSGi specifications use the following mechanisms:

- Java 2 Code Security
- Minimized bundle content exposure
- Managed communication links between bundles

Java 2 Code Security provides the concept of Permissions that protect resources from specific actions. For example, files can be protected with File Permission classes. Permission classes take the name of the resource and a number of actions as parameters. Each bundle has a set of permissions. This set of permissions can be changed on the fly. New permissions are immediately effective once set. However, this permission assignment can also be done prior to, or just in time during the install phase.

Besides, these permissions are roughly used as follows: When a class wants to protect a resource, it asks the Java Security Manager to check if there is permission to perform an action on that resource. The Security Manager then ensures that all callers on the stack have the required permissions. Checking the callers on the stack protects the resource from attackers. For example, if M calls T and T accesses a protected resource, both M and T need to have access to the resource. This is a very secure strategy that prevents much mayhem.

The access modifiers in Java classes are used to protect access to code. Classes, methods, and fields can be made private, package private (accessible only by classes in the same package), protected (accessible by sub-classes) or public. The OSGi Service Platform adds an extra level of module privacy by making packages only visible within the bundle. These packages are freely accessible by other classes inside the bundle, but they are not visible to other bundles.

Package-sharing between bundles is a possible attack route for malicious bundles. The OSGi specifications therefore contain Package Permissions to limit exports and imports to trusted bundles. Package Permission is a fine-grained permission that allows importing or exporting for a specific package, or for all packages.

Another security mechanism is Service Permission. This permission gives bundles the possibility to register, or get a service, from the Service Registry. Service Permissions are extensively used to ensure that only the appropriate bundles can provide or use certain services.

The security mechanisms in the OSGi Service Platform provide the operator of the Service Platform with powerful tools to control the security operation of the device [OSGI2007].

4.4.2.1 Main features of OSGi

OSGi provides a standardized environment for applications (which are called *bundles* into the OSGi terminology) where they can also be managed. The environment offers a list of properties that are enumerated next:

- OSGi framework is a standard technology: As a non-proprietary service platform, the OSGi specifications are an open standard available for everyone.
- It provides a powerful model where multiple applications or bundles can co-exist and cooperate into the same environment. In this way, applications can discover and use services provided by other applications running inside the same OSGi platform.
- OSGi framework facilitates the management of services: It allows installing, starting, stopping and updating components without having to stop and reinitialize the complete system. It is an environment that controls the whole life-cycle of applications or bundles.
- Off-the-shelf services and functions are already available into the OSGi platform. A large number of standardized services can be optionally used (HTTP server, SML, UPnP…)
- Due to easiness to incorporate services or applications to the platform, bundles are really reusable components.
- The applications are portable.
- It provides modularity based on a good component model, which reduces complexity in large projects.
- The development of the different software components can be made in parallel.
- Instead of trying to impose its specification, it complements other existing standards.

- Finally, and as it has been mentioned, OSGi specifications offer a comprehensive security model.

All these characteristics yield a reduction of the time and cost of the development of applications in many different fields.

The OSGi technology includes also some other features, that are interesting from the execution point of view:

- OSGi is based in Java technology, so it can run in almost every existing platform.
- It allows dynamic installation, update and also removal of components.
- As it is a dynamic environment, it provides capabilities to offer new services that can come from different service providers or vendors and that can be incorporated into the OSGi framework at run time, without having to stop any device.
- Allows remote configuration and administration of devices. The management of a large number of service platforms from a single management domain can be done.

4.4.2.2 OSGi for Mobile environments (M-OSGi)

The mobile market is characterised by rapid changes and a large number of different devices, architectures, etc. This means that the reusability rate for the applications developed is relatively low (e.g. a service developed for iPhone cannot be used in an Android-based phone). In this sense, OSGi means a good opportunity to minimize the development effort by providing a common environment to deploy cross-platform services and applications. Moreover, the remote management capabilities of OSGi would enable a dynamic update system for terminals, allowing the installation of new features for the users seamlessly.

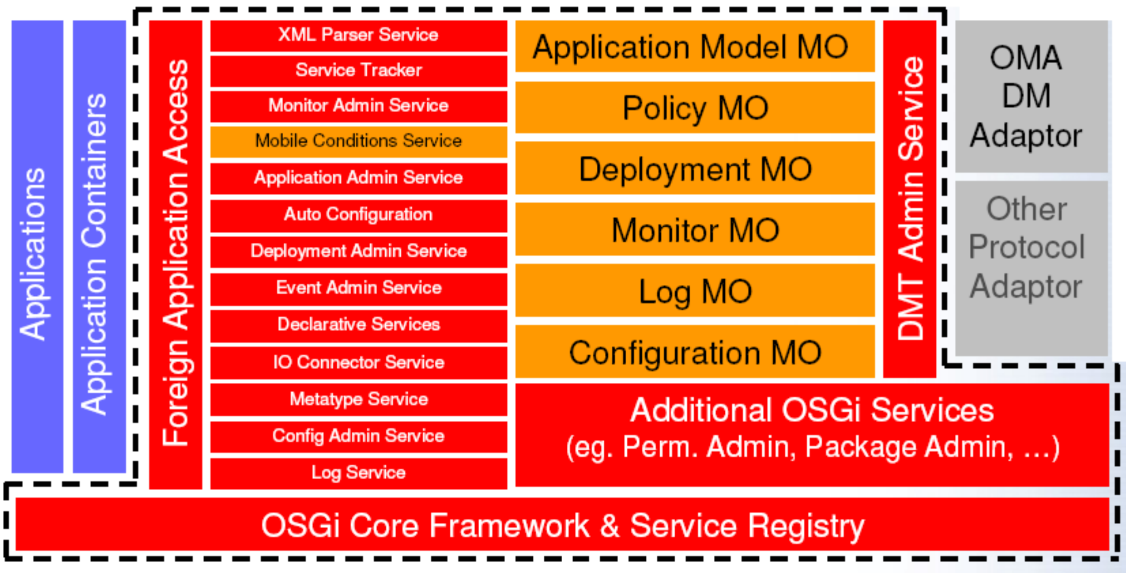

Figure 4-8 OSGi R4 Mobile specification

The main features of the 'fixed' OSGi platforms remain in the Mobile framework, this, remote management and security capabilities can be also used in this environment. In this sense, OSGi has also followed the OMA-DM protocol for remote management of devices (DMT) in:

- Application life-cycle
- Log Service & Monitoring
- Configuration Admin
- Policy management

The DMT can also be extended *on-the-fly*.

There are several frameworks already available for Mobile applications, such as eRCP [eRCP], provided by Eclipse or mBS, provided by Prosyst [mBS].

4.5 Conclusions

Security and privacy are major aspects when dealing with user data and networking technologies. The Florence consortium is aware of such issues and will try to reach a security level that is as high as possible. It is not planned to invent new security technologies but to use state of the art technology. The presented methods like ID management will be integrated into the Florence system. The admin services provided by the OSGi-Framework will be used to develop a zero administration component so that the Florence system can be maintained by external administrators (the elderly itself doesn't have to care about that). This also enables the possibility to quickly adopt new devices or features to the system.

The home network is another important part of the Florence system as it has to integrate into existing infrastructures as seamless as possible. Florence will make use of the existing home networking technologies. This is also needed for the system to be reachable from outside to give an access to caregivers etc. This again has to be secured so no abuse of personal data is activated.

5 CONCLUSIONS

We've taken different views on robots and AAL home services in this document. A lot of research is being conducted in the robotic sector. As a result robots are developing at a very fast pace. Despite the fast rate at which improvements are being made there are only few robots reaching the consumer in their everyday live.

As many different robots as there are, almost an equal number of robotic software platforms exist, each with their own strength and weaknesses. Apart from the robot platform we've also taken a look at different service oriented frameworks which are relevant to Florence. Here we see some commonalities, like context management and decoupling information sources, processing and consumption.

Despite these different frameworks which are available for the robot and for the in home services we have not found a lot on the integration of the robot within such an environment. Only two projects, Companionable and Robocare, were found. Both projects mention how they embed their robot, but there's no public available information on their architecture and which, if any, already existing frameworks they might have used.

At the consumer's homes we also see a lot of new developments. More network enabled devices enter the user's homes, both devices that use the network (STB) and those that provide the network (FemtoCells). Connecting the robot together with the home infrastructure should therefore be a viable approach and way to go.

Having a robot and smart house monitoring people's lives introduces numerous privacy concerns. Especially if you want to make some of these services cooperate with the outside world. Luckily an enormous amount of research has already been done to these privacy issues and several techniques for safely allowing accessing privacy sensitive data are present nowadays.

Like said at this moment we could not find an existing platform which takes into account the robot, the services, the home environment and privacy issues. As a next step within the Florence project we should decide upon which frameworks from the state of the art research we should reuse and integrate to enable the Florence scenarios.

6 References

[Aibo] http://support.sony-europe.com/AIBO/

[Bandit] Bandit, http://www.bandit.org/

[Bitlocker] http://windows.microsoft.com/en-US/windows-vista/BitLocker-Drive-Encryption-Overview

[Bohn2005] J. Bohn, V. Coroama, M. Langheinrich, F. Mattern and M. Rohs. Social, Economic, and Ethical Implications of Ambient Intelligence and Ubiquitous Computing. In W. Weber, J. Rabaey and E. Aarts editors, Ambient Intelligence, pp. 5–29. Springer, Berlin, Heidelberg, New York, Tokyo, 2005.

[Borenstein1991] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," IEEE Transaction on Robotics and Automation, vol. 7, no. 3, pp. 278 – 288, 1991.

[Breazeal2002] C. Breazeal. 2002. Designing Sociable Robots. MIT Press.

[Bradsky2008] G. Bradsky and A. Kaehler. Learning OpenCV: Computer Vision with the OpenCV library. O'Reilly Media Inc, 2008.

[Brey2005] P. Brey, Freedom and privacy in Ambient Intelligence, Ethics and Information Technology (2005) 7:157-166, Springer 2006

[Brock1999] Brock, O. and Khatib, O., 1999, "High speed navigation using the global dynamic window approach", Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, MI, pp. 341-346.

[Calisi2008] D. Calisi, A. Censi, L. Iocchi and D. Nardi. OpenRDK: a modular framework for robotic software development. In Proc. of International Conference on Intelligent Robots and Systems (IROS 2008), pages 1827-1877, September 2008.

[Calisi2009] D. Calisi and A. Censi. Robotic software development and interoperability using the OpenRDK framework. ICAR'09 Workshop on "Rapid Application Development in Robotics: On the role of re-use and adaptation of system components, middleware, and control architectures". Munich, Germany. June, 2009.

[Cañas2007] J. Cañas, D. Lobato and P. Barrera. Jde+: an open-source schema-based framework for robotic applications. Proceedings of Second International Workshop on Software Development and Integration in Robotics SDIR 2007 affiliated with ICRA, Rome, April 2007.

[CardSpace] Microsoft Windows CardSpace
http://www.microsoft.com/windows/products/winfamily/cardspace/default.mspx

[Carmen] http://carmen.sourceforge.net/

[Cesta2007] A. Cesta, G. Cortellessa, F. Pecora, and R. Rasconi. Supporting Interaction in the RoboCare Intelligent Assistive Environment. In Proccedings of AAAI Spring Symposium on Interaction Challenges for Intelligent Assistants, 2007.

[Chan2008] Vikram Chandrasekhar and Jeffrey G. Andrews and Alan Gatherer, 2008, Femtocell Networks: A Survey, June 28, 2008

[Chaumette2007] F. Chaumette and S. Hutchinson. Visual servo control, Part II: advanced approach. IEEE Robotics and Automation Magazine, Vol. 14, [ages 109—118, march 2007.

[CompanionAble] http://www.companionable.net/

[Comport2010] A. Comport, E. Malis and P. Rives. Real-time quadrifocal visual odometry. International Journal Robotics Research, Vol. 29, Issue 2-3, February 2010.

[Courbon2008] J. Courbon, Y. Mezouar and P. Martinet. Indoor navigation of a non-holonomic mobile robot using a visual memory. Autonomous Robots, Vol. 25, Issue 3, pages 253—266, 2008

[DIANE] http://hnsp.inf-bb.uni-jena.de/DIANE/en/

[Diankov2008] R. Diankov and J. Kuffner. OpenRAVE: a planning architecture for autonomous robotics. Tech Report CMU-RI-TR-08-34, Robotics Institute, Carnegie Mellon University, July 2008.

[Dock2005] Dockhorn Costa, P. Pires, F. Sinderen, M., “Architectural Patterns for Context-Aware Service Platforms” in Proceedings of the Second International Workshop on Ubiquitous Computing (IWUC 2005), Miamy, May 2005, pp 3-19

[eRCP] http://www.eclipse.org/ercp

[ETSI] http://www.etsi.org

[ETSI2010-1] Draft ETSI TS 102 690 V<0.5.1>, “Machine-to-Machine communications (m2M); Functional Architecture”, 2010

[ETSI2010-2] Draft ETSI TS 102 732 V<0.3.1>, “Machine-to-Machine communications (m2M); Use cases of M2M applications for eHealth”, 2010

[Europ2009] Robotic Vision to 2020 and beyond, the strategic research agenda for Robotics in Europe. July 2009. http://www.robotics-platform.eu/sra

[Fasola2010] J. Fasola, M. Mataric. Robot exercise instructor: a socially assistive robot system to monitor and encourage physical exercise for the elderly. IEEE International Symposium in Robot abd Human Interactive Communication, RO-MAN, Viareggio, Italy, Sep. 2010.

[Femto] http://femtoforum.org

[Fox1997] Fox, D., Burgard, W. and Thrun, S., 1997, “The Dynamic Window Approach to Collision Avoidance”, IEEE Robotics & Automation Magazine, March 1997, pp. 23-33.

[FZILL] http://aal.fzi.de/

[Gallagher2009] G. Gallagher, S. Srinivasa, D. Bagnell and D. Ferguson. Gatmo: a generalized approach to tracking movable objects. IEEE International Conference on Robotics and Automation (ICRA'09), May 2009.

[GATMO] http://www.gatmo.org/

[GeneralRobotix] http://www.generalrobotix.com/en/index.htm

[Goedemé2007] T. Goedemé, M. Nuttin, T. Tuytelaars and L. van Gool. Omnidirectional vision based topological navigation, International journal of Computer Vision, Vol. 74, Issue 3, pages 219-236, Sep. 2007.

[Gupta2004] S. Gupta, Home Gateway White Paper of Wipro

[Haash2005] A. Haasch, N. Hofemann, J. Firtsch and G. Sagerer. A multi-modal object attention system for a mobile robot, IEEE/ RSJ International Conference on Intelligent Robots and Systems, IROS, Edmonton, Canada, August 2005.

[Hagisonic] http://www.hagisonic.com/

[Hart1968] P. Hart, N. Nilsson and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 2, pages 100-107, July 1968.

[Higgins] Higgins Project, http://www.eclipse.org/higgins/

[HomeGateway] http://en.wikipedia.org/wiki/Home_gateway

[Hüttenrauch2002] H. Hüttenrauch, K. Severinson-Eklundh. Fetch and carry with CERO: observations from a long-term user study with a service robot. Proceedings of IEEE International Workshop on Robot and Human Interactive Communication, Roman'2002, pages 158-163, September 2002

[IETF] IETF, http://www.ietf.org/

[IFR2009] World robotics – service robots 2009. Statistics, Market Analysis, Forecasts and case studies. IFR Statistical Department, 2009.

[iRobot] http://www.irobot.com/

[Khatib1986] O. Khatib, "Real-time obstacle avoidance for manipulators and mobi le robots," International Journal on Robotic Research., vol. 5, no. 1, pp. 90–98, 1986.

[Langheinrich2001] M. Langheinrich. Privacy by Design – Principles of Privacy-Aware Ubiquitous Systems. In Lecture Notes In Computer Science; Vol. 2201 Archive. Proceedings of the 3rd international conference on Ubiquitous Computing, Atlanta, Georgia Georgia, USA, pp. 273–291. Springer-Verlag, London, 2001.

[Latombe1991] J.-C. Latombe Robot motion planning, 1991, Kluwer Academic Publishers

[Liberty] The Liberty Alliance, http://www.projectliberty.org/

[LibertyDST] Liberty Alliance Data Services Template v2.1, http://www.projectliberty.org/liberty/content/download/879/6213/file/liberty-idwsf-dst-v2.1.pdf

[LiveID] http://en.wikipedia.org/wiki/Windows_Live_ID

[Kantara] http://kantarainitiative.org

[Karto] http://www.kartorobotics.com/

[Mamoru] http://www.jsk.t.u-tokyo.ac.jp/research/irt/mamoru.html

[May2009] S. May, D. Droeschel, D. Holz, S. Fushs, E. Malis, A. Nüchter A. and J. Hertzberg. Three dimensional mapping with time-of-flight cameras. Journal of Field Robotics, Volume 26, Issue 11-12, pages 934-965, September 2009.

[mBS] http://dz.prosyst.com/devzone/Mobile

[METRALABS] http://www.metralabs.com/

[Montemerlo2002] Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada, 2002.

[MS-Robotics] http://www.microsoft.com/robotics/

[Muller2007] S. Muller, E. Schaffernicht, A. Scheidig, H.J..Boehme and H.-M. Gross,. Are you still following me? Proceedings of the 3rd European Conference on Mobile Robots (ECMR), Page 211–216, 2007.

[Neatorobotics] http://www.neatorobotics.com/

[Neira2008] J. Neira, A. Davison and J. Leonard. Guest editorial, special Issue on Visual SLAM, IEEE Transactions on Robotics, Vol.24, n. 5, October 2008

[NetMesh] NetMesh LightWeight Identity, http://lid.netmesh.org/

[Neobotix] http://www.neobotix.de/en/index.html)

[NorthStar] http://www.evolution.com/products/northstar/

[OASIS] OASIS, http://www.oasis-open.org/

[openAAL] http://openaal.org

[OpenCV] http://www.willowgarage.com/pages/software/opencv

[OpenID] OpenID http://openid.net/

[OpenSLAM] http://www.openslam.org/

[Orocos] http://www.orocos.org/

[OSGI] http://www.osgi.org

[OSGI2007] OSGi Alliance, About the OSGi Service Platform, Whitepaper of the OSGi Alliance

[OSGiWP] *"Technical Whitepaper: About the OSGi Service Platform"*, OSGi Alliance, revision 4.1, 7 June 2007. (http://www.osgi.org/documents/index.asp?section=2)

[Otto] http://www.otto.de

[Pastor2009] C. Pastor, G. Gaminde, A. Renteria. COMPANIONABLE: Integrated cognitive assistive & domotic companion robotic systems for ability & security. International Symposium on robotics, 2009, Barcelona, Spain.

[Quigley2009] M. Quigley, K. Conley, B. Gerkey, J. Faus, T. Foote, J. Leibs, R. Wheeler and A. Ng. ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software, May 2009.

[Player] http://playerstage.sourceforge.net/

[Prassler1999] E. Prassler, J. Scholz, M. Strobel and P. Fiorini. An Intelligent (Semi-) Autonomous Passenger Transportation System", IEEE International Conference on Intelligent Transportation Systems, pp. 374-379, 1999.

[Quigley2009] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng. ROS: an open-source Robot Operating System , Proc. Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA), 2009.

[Quinlan1993] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in Proceedings of the IEEE Conference on Robotics and Automation, pp. 802 – 807, 1993.

[Rett2010] J. Rett J. Dias and J.-M. Ahuactzin. Bayesian reasoning for Laban Movement Analysis used in Human machine interaction. International Journal on Reasoning-based Intelligent Systems, Vol.2, N.1, pages 13-35, 2010.

[Rich2009] C. Rich and C. Sidner. Robots and avatars as hosts, advisors, companions and jesters. AI Magazine, Vol. 30, No. 1, 2009

[Ritter2008] Ritter, J. "What OSGi offers to mobile enterprise developers", OSGi Alliance Community event, June 2008, Berlin.

[Robocare] http://robocare.istc.cnr.it/

[Robosapien] http://www.therobosapien.com/

[ROS] http://www.ros.org

[Royer2007] E. Royer, M. Lhuillier, M. Dhome and J.M. Lavest. Monocular Vision for mobile robot localization and autonomous navigation, International Journal of Computer Vision, Vol. 74, Issue 3, pages 237-260, Sep. 2007.

[SAML] http://www.oasis-open.org/specs/#samlv2.0

[SENSEI] SENSEI - Integrating the Physical with the Digital World of the Network of the Future, FP7 ICT-2007.1.1-215923, Jan2008–Dec2010, www.ict-sensei.org

[SENSEI2010] SENSEI Consortium, PF7 Project Number 215923, “The SENSEI Real World Internet Architecture”, Project Whitepaper Version 1.1, March 2010, http://www.ict-sensei.org

[Segvic2008] S. Segvic, A. Remazeilles, A. Diosi and F. Chaumette. A mapping and localization framework for scalable appearance-based navigation, Computer Vision and Image Understanding, Volume 113, pages 172-187, August 2008.

[SOPRANO] http://www.soprano-ip.org

[Spykee] http://www.spykee.org/

[Thrun2005] S. Thrun, W. Burgard and D. Fox. Probabilistic robotics, Mit Press, 2005

[TrueCrypt] http://www.truecrypt.org

[URBI] http://www.urbiforge.org/

[W3C] W3C, http://www.w3.org/

[Wolf2009] Wolf P., Scmidt, A., and Klein, M., “Applying Semantic Technologies for Context-Aware AAL Services: What we can learn from SOPRANO”, *Workshop on Applications of Semantic Technologies 09, Informatik 2009*, Lecture Notes in Informatics vol. , GI, 2009

[Wolf2010] Wolf P., Scmidt, A., Parada Otte, J. Klein M., Rollwage S., Köning-Ries, B., Torsten, D. and Gabdulkhakova, A., “openAAL – the open source middleware for ambient-assisted living (AAL)”, Málaga, 2010.

[WS-FED] http://msdn.microsoft.com/en-us/library/bb498017.aspx#wsfedver1_topic1

[WS-I] WS-I, http://www.ws-i.org/

[Yadis] Yadis, http://yadis.org/