
**Do-it-Yourself Smart Experiences**
ITEA 2 project 08005

# SoA of applications, type of users, business models, and UCD

D1.1

**Editor:**
E. Marilly, Alcatel-Lucent Bell Labs France

**Reviewers:**
Zhenzhen. Zhao, Institut Telecom Sudparis, France
J. Pierson, VUB-SMIT, Belgium

**Contributors:**
E. Marilly, A. Outtagarts, Alcatel-Lucent Bell Labs France
J. F.. Rodelgo, M. A. Sanguino J. Lucea, F. J. Nieto, Atos Origin
D. De Roeck, C. van Nimwegen, CUO-KUL
N. Joncheere, Vrije Universiteit Brussel, Software Languages Lab
W.Van den Broeck, Karen Torben-Nielsen, Jo Pierson, An Jacobs, VUB-SMIT
M. Roelands, L. Trappeniers, J. Criel, M. Geerts, Alcatel-Lucent Bell Labs BE
J. Alonso, ESI-Tecnalia Spain
J. Santamaria, , S. Rodriguez, Robotiker-Tecnalia Spain
Zhenzhen. Zhao, Noel Crespi, Institut Telecom Sudparis, France
J. Kela, Finwe, Finland
Juan R. Velasco, Maite Lopez, UAH, Spain
Yacine Ghamri-Doudane, Ismail Salhi, ENSIIE, France
Jan De Moerloose, GeoSparc
Yan Tang, VUB Starlab
M. Niemelä, H. Lammi, T. Tuomisto, VTT, Finland
Rob Moonen, A.Iyer**,** Laurea UAS
University of OULO
G. Miranda Álamo, I&IMS, Spain
C. Valderrama, F. Vinci, J. Vachaudez, UMONS, Belgium

Security: <Public>
Version: <1.0>
Date: <April 01, 2010>
Number of pages: 131

# History

| Version | Date | Person, Partner | Comment |
|---|---|---|---|
| 0.1 | 31.07.2009 | E. Marilly, AL Bell Labs Fr | Creation of the document |
| 0.1.1 | 18.09.2009 | A. Outtagarts, AL Bell Labs FR | Modification of the Toc – more details |
| 0.1.2 | 22.09.2009 | J.F. Rodelgo, F.J. Nieto, Atos Origin | Modification of the ToC |
| 0.1.3 | 23.09.2009 | D. De Roeck, C. van Nimwegen, CUO-KUL | Modification of the ToC |
| 0.1.4 | 24.09.2009 | N. Joncheere, Vrije Universiteit Brussel, Software Languages Lab | Modification of the ToC |
| 0.1.4. | 24.09.2009 | W.Van den Broeck, K.Torben-Nielsen, J.Pierson | Modification of ToC |
| 0.1.5 | 24.09.2009 | M. Roelands, L. Trappeniers, J. Criel, M. Geerts, Alcatel-Lucent Bell Labs BE | Modification of the ToC |
| 0.1.7 | 24.09.2009 | J. Alonso, ESI-Tecnalia | Modification of the ToC |
| 0.1.8 | 25.09.2009 | J. Santamaria, Robotiker-Tecnalia | Modification of the ToC |
| 0.1.9 | | VUB Starlab | |
| 0.1.10 | 25.09.2009 | Zhenzhen. Zhao, Noel. Crespi, Institut Telecom Sudparis | Modification of the ToC |
| 0.1.11 | 25.09.2009 | J. Kela, Finwe | Modification of the ToC |
| 0.1.12 | 25.09.2009 | GeoSparc | |
| 0.1.13 | 27.09.2009 | Juan R. Velasco, UAH | Modification of the ToC |
| 0.1.14 | 27.09.2009 | Y. Ghamri-Doudane, ENSIIE | Modification of the ToC |
| 0.2 | 28.09.2009 | E. Marilly | ToC update based on plenary meeting |
| 0.2.1 | 07.10.2009 | G. Miranda Álamo, I&IMS | Added content to the ToC |
| 0.2.2 | 07.10.2009 | J.Alonso, ESI-Tecnalia | Modification of the ToC |
| 0.2.3 | 07.10.2009 | Zhenzhen.Zhao, Institut Telecom | Added content to the ToC |
| 0.2.4 | 09.10.2009 | N. Joncheere, Vrije Universiteit Brussel, Software Languages Lab | Added content |
| 0.2.5 | 12.10.2009 | Rob Moonen, Laurea UAS | Modification of the ToC |
| 0.2.6 | 22.10.2009 | A. Outtagarts, AL Bell Labs Fr | Modification of the Toc<br>Added content to the Toc |
| 0.3.0 | 22.10.2009 | E. Marilly | Modification of the Toc<br>Added content |
| 0.3.1 | 06.11.2009 | J.Alonso, ESI-Tecnalia | Section assignation |
| 0.3.2 | 17.11.2009 | E. Marilly | Added content |
| 0.3.3 | 18.11.2009 | Zhenzhen Zhao, ITS | Modification of content / Added references |
| 0.3.4 | 18.11.2009 | A. Outtagarts, AL Bell Labs Fr | Added content |
| 0.3.5 | 19.11.2009 | K.Torben Nielsen, W. Van Den Broeck, J. Pierson | Added content & added references |
| 0.3.6 | 23/11/2009 | G. Miranda Álamo, I&IMS | Added content |
| 0.3.7 | 24.11.2009 | S. Rodriguez, Robotiker-Tecnalia | Added content |
| 0.3.7 | 24/11/2009 | L.Bastida, M.Escalante, J.Alonso | Added content |
| 0.3.8 | 27/11/2009 | J. De Moerloose, Geosparc | Modification of the ToC/ Added content |
| 0.3.9 | 30/11/2009 | S.Rodriguez, Robotiker-Tecnalia | Added content |
| 0.4.0 | 01/12/2009 | E. Marilly, AL Bell Labs FR | Added content |
| 0.4.1 | 02/12/2009 | A. Outtagarts, AL Bell Labs FR | Review and remove figures |
| 0.4.2 | 04/12/2009 | M. Niemelä, H. Lammi, VTT FI | Added content, ToC update |
| 0.5 | 08/12/2009 | E. Marilly, AL Bell Labs FR | Added content + edition |
| 0.5.1 | 11/12/2009 | D. De Roeck, KUL CUO | Added content |
| 0.5.2 | 16/12/2009 | M. Geerts | Added content |
| 0.5.3 | 16/12/2009 | T. Tuomisto, VTT | Added content |
| 0.6 | | E. Marilly | Added content + Edition |
| 0.61 | 21/12/2009 | M.Niemelä, VTT | Added references |
| 0.6.2 | 21/12/09 | A.Iyer, Laurea | Added content |
| 0.6.3 | 22/12/2009 | Yan Tang, VUB STARLab | Added content |
| 0.6.4 | 23/12/2009 | J. F. Rodelgo, ATOS | Added content, Toc Update |
| 0.6.5 | 05/01/2010 | I. Salhi, ENSIIE | Added content |
| 0.7 | 05/01/2010 | E. Marilly, ALBLF | Edition |
| 0.7.1 | 07/01/2010 | D.De Roeck | Added content, Toc Update |
| 0.7.2 | 08/01/2010 | R.Moonen, Laurea | Toc Updated + edition |
| 0.7.3 | 07/01/2010 | M. Lopez | Added content |
| 0.7.4 | 15/01/2010 | J. F. Rodelgo | Merged content |
| 0.8 | 15/01/2010 | E. Marilly | Stabilized version – To Be reviewed |
| 0.8.1 | 01/02/2010 | J. Gilbert (Archos) | Added content |
| 0.8.2 | | VUB STARlab | |
| 0.8.3 | 18/02/2010 | F. Vinci, UMONS | Added hardware platforms |
| 0.8.4 | 29/03/2010 | Zhenzhen, ITS | First Review of the document |
| 0.8.4b | 02/04/2010 | J. Pierson, VUB-SMIT | Second review of document |
| 0.9 | 01/04/2010 | E. MARILLY | First Review integrated |
| 1.0 | 12/04/2010 | C. van Nimwegen (KUL-CUO) | Second review integrated |

## The DiYSE Project

The Do-it-Yourself Smart Experiences project (DiYSE) aims at enabling ordinary people to easily create, setup and control applications in their smart living environments as well as in the public Internet-of-Things space, allowing them to leverage aware services and smart objects for obtaining highly personalised, social, interactive, flowing experiences at home and in the city.

**www.dyse.org**

## The three State of the Art Deliverables.

The DiYSE project has three State of the Art (SOTA) Documents covering the different tools, techniques, methods and environments that may be used to provide a DiYSE platform. These documents present the same pool of elements from different points of view. Due to this SOTA partition, it will be needed to link some of the sections from one of the documents to other sections on some of the other two documents. This is really important in Section 3 of the D1.1 and D4.1 that will try to present the same topics from the requirements and the interfaces point of view: web technologies, mobile technologies, platforms, devices, etc…

In WP1 (Use cases and requirements), deliverable D1.1 will focus on which requirements will be covered by web technologies, mobile technologies, system platforms and toolsets, i.e. how the users will access the smart experiences by using these systems.
This document will present the SOTA of current applications, systems platforms and business models relating to DiYSE. This includes Ambient Experience applications, features of toolsets and the business models and ecosystems that are working at this moment in similar proposals.
The document also includes a PEST (Political, Economical, Social, Technological) analysis. It is important to know why people are motivated to produce and share services, devices, etc.

In WP2 (Interaction with the environment), deliverable D2.1 will focus on the State of the art of:

- electronic devices that can retrieve data from the users' environment and produce physical outputs,
- algorithms to extract information from them (such as identification or location) and the functionalities those devices can provide in DiYSE,
- networking technologies to interconnect them.

In particular, D2.1 will put a special emphasis on the following kinds of devices:

- existing ready-made devices available in the market,
- networks of tiny battery-powered programmable wireless sensors,
- open hardware platforms used by DiY hobbyists.

In WP4 (Interactive Experience Creation), deliverable D4.1 analyses similar elements to D1.1, but from the user/developer point of view: how the users will use the elements of the DiYSE ecosystem.
The review of existing application creation approaches will identify technologies that may be supportive for the envisioned creation of applications and services in smart spaces. It is expected that lessons learnt from methods empowering users in the

world wide web to contribute content or even applications to communities may provide a good base. Also the issue of actually do-it-yourself versus do-it-together (or have the community do it for you) and crowdsourcing will be addressed. The document reviews how to create interactive experiences.

## Abstract of this deliverable

SoA of current applications, systems platforms and business models relating to DIYSE. This include Ambient Experience applications, features of toolsets like those provided by ITEA-Amigo, Apple's iPhone, and Google's Android, and the business models and ecosystems such as Google marketplace. Also included is a PEST(Political, Economical, Social, Technological) analysis. It is important to know why people are motivated to produce and share. Next to this the SoA also explores the lead user and producing users of these technologies.

## TABLE OF CONTENTS

# 1 Introduction

## 1.1 DIY Movement

### 1.1.1 Past

The earliest roots of the Do-It-Yourself (DIY) culture could be traced back to the eighteenth-century handcrafts, which formed a means self-expression and helped to avoid the sin of idleness.

An article in an American 1912 magazine encouraged homeowners to decorate their own homes. Starting from the mid-1920s, this DIY spirit spread to the middle-class. This success was mirrored in Great Britain. After the Second World War, DIY culture boomed in America, as it was an affordable alternative (due to the economic situation and the lack of tradesmen) for homeowners to realise the American dream of a modern and comfortable home [114] . More recently, Sennet [221] approached the issue from a sociological perspective, elaborating on "the special human condition of being engaged". In the modern world he explores what experiences of good work are shared by computer programmers, nurses and doctors, musicians, glassblowers, and cooks.

Two types of DIY-activities could be distinguished: on the one hand the 'soft DIY' (as e.g. knitting and painting), and on the other hand the 'hard DIY' (as e.g. 'home-improvement'). New power tools supported the latter [115] . With its 'Home Utility Line', the first series of electrical 'power tools', tool producer Black & Decker stimulated the emerging DIY culture even more.


**Figure 1: Cover Time Magazine (August 2, 1954) portraying the abundance of tasks for a DIY-homeowner**

### 1.1.2 Present & future

As DIY is still highly popular today, many researchers have investigated what makes it so attractive. Although several authors put forward different reasons, they nevertheless all highlight the satisfaction of one's own creativity – which is a main ingredient for all DIY-activities:

- DIY offers people pleasure and confirms their creative side [115]
- DIY gives people the feeling of 'being their own boss' [115]
- DIY provides the satisfaction from self-created things, that purchased things are unable to give [116]
- Self-made things have hidden meanings that others can not see [116]

To date, there appear some new tendencies in the classic DIY culture: Done for You (DfY), the rehabilitation of the plumber, the self-service society and co-creation.

- Done for You
  - Poulter [117] observes the rise of a new phenomenon, Done for You (DfY). Related to this, Powell [114] notes the emergence of the 'cash-rich and time-poor' (CRTP) people, who want to 'outsource' DIY activities. CRTP-people find DIY-tasks competing for their time with other activities. They chose for other activities or leisure time, instead of engaging in DIY themselves. Therefore, they outsource the DIY-activities to others, making it a 'Done for You'-process.

- The rehabilitation of the plumber
  - This outsourcing of DIY activities is fuelled on the one hand by the rise of cheap labour (from new member states that joined the European Union), and on the other hand by one's realisation that (s)he lacks the knowledge to carry out home maintenance [117]
    - *"DIY superstores abound, but that does not mean that we can all be instant plumbers. (...) The explosion in DIY has not made plumbers extinct: they tend to be called upon for the more complex jobs, or when DIY goes horribly wrong."* [118]

- The self-service society
  - In America the flipside of DIY is becoming clear as the self-serve society: companies keen on saving labour costs pass the 'workload' on to customers, who demand self-empowerment. Some customers would claim it beats dealing with 'incompetent workers' [119] .

- Co-creation
  - Hoftijzer regards co-creation as a modern sort of DIY, as e.g. toolkits enable the user to design products and services to his every wish [115] .

## 1.2 "Traditional" offline DIY and it's relation to the digital SoA

Much like it has been covered in chapter 1.1, there are several concepts from the analogue world of DIY that clearly can be 'transposed' to the digital realm. When comparing several "offline" DIY activities with "online" ones, there are several clear concepts emerging. The list below does not intend to be complete, but based on these comparisons it becomes clear that several aspects characterise a DIY activity.

- Reciprocity

A mutual exchange of information is a characteristic that happens both in online and offline DIY projects. Initiatives such as instructables.com take great advantage of this characteristic in order to enable people to share real and virtual world with each other and learn from each other.

- Fun

Both online and offline DIY projects are often driven by fun. Fun is often the result of an experience relating to flow (Csikszentmihalyi, 1990) or the concept of pottering (Taylor, 2008). By creating things themselves, people tend to get a lot of energy & joy out of the both the creation process as the result.

- Putting your stamp

In most DIY projects, the people involved want to include something of themselves into their creation. This is also closely related to gaining respect and 'standing' in a community. This aspect of putting your stamp can be a strong motivation for people to create DIY projects.

- Creativity

Often the question is why people create things in the first place (Gelb, 2005). 'Online' or 'Offline' DIY is in many cases a way for people to express this creativity. For some people this can be welding two pipes together, for others it can be creating a mashup of RSS feeds.

- Social aspects

The social aspects of DIY can be very diverse depending on the type of person. Traditionally creating things by yourself is often associated with terms like 'inventing' (Griffith 2007). People would be working on projects by themselves in their spare time. This stereotype is radically changing due to the influence of the internet, which has catalysed many DIY communities. DIY has evolved into a social trend and creating your own things is regarded as 'trendy'. (trendwatching.com, 2006)

## 1.3 DIY ICT culture

### 1.3.1 From the garage to the web

The Internet has given an enormous boost to the DIY-culture. On the one hand, the possibilities to learn more about all sorts of DIY projects became nearly endless with various DIY-websites. Two of the most renowned sites are Instructables and Ponoko. Instructables [120] is an online repository of people-driven tutorials on DIY projects, based on a structure of step-by-step collaboration. Ponoko [121] is an online starting place, to design, produce and put up for sale material products.
On the other hand does the Internet, and more specifically the Internet of Things (IoT) with objects that can communicate with each other and perform action, offer an abundance of possibilities for DIY-ers who want to enlarge their creative 'playground' to the virtual world.

#### *Importance of Internet of Things (IoT)*

The International Telecommunication Union ([122] ) commences with the paradigm shift from *Ubiquitous Computing* (ubicomp, a set of processes where information technology has been thoroughly integrated into everyday objects and activities; [124] ) to *Ambient Intelligence* (AmI, electronic systems that are sensitive and responsive to the presence of people; Aarts and de Ruyter, [123] ) in its vision on an Internet of Things. This vision can go as far as an "Information Society where humans will be surrounded by intelligent interfaces supported by computing and networking technology that is everywhere, embedded in everyday objects such as furniture, clothes, vehicles, roads and smart materials" [222] . It is a vision where computing capabilities are connected everywhere and always on, enabling people and devices to interact with each other and with the environment. It is envisaged that Ambient Intelligence will be aware of the specific characteristics of human presence and personality, and will be capable of meeting needs and responding intelligently to spoken or gestured wishes. It could even engage in intelligent dialogue.

The Internet of Things is hauled as: "*a technological revolution that represents the future of computing and communications, and its development depends on dynamic technical innovation in a number of important fields, from wireless sensors to nanotechnology"* [122] . This 'technological revolution' could be described as an evolution from a network of interconnected computers (the Internet as we know it) to a network of interconnected objects, as e.g. electronic appliances, food or cars. According to the European Commission, the Internet of Things could significantly alter how our societies function [125] .

## 1.4 DiY Design scene

Besides the increasing influence of the DIY phenomenon in consumer markets, there is a clear influence of the DIY idea in a variety of design disciplines. The most notable example are the emerging 'interaction design' and 'service design' disciplines. Both are similar and related ways to intertwine user research and technical development (also see section 2.2 in this document).
The major attitude change by people involved in design is that because of technology becoming more accessible people without an explicit technical background are becoming able to create medium or high fidelity working prototypes of design ideas. Whereas in the past 'designers' had to explicitly go to 'developers' to create a working hardware and/or software prototype of their ideas, it is nowadays a lot harder to identify this gap between both roles.
Catalysed by initiatives such as the Arduino platform for hardware or Processing for software, it becomes possible for people without education in software or hardware development to relatively quickly create prototypes of interfaces and devices. Also, these emerging tools facility a much better way to communicate between the developer, the designer, and whatever other roles that might be included in a project team.
Although the interpretation of DiY in the design community might be different compared to that of the consumer or end user, but the fact that people who were not able to create things in the past because of a technological gap is a very similar process to the activities going on in the consumer market.

# 2 Human centered research

## 2.1 Communities & mass collaboration

### 2.1.1 Concepts and models on mass collaboration

Mass collaboration has been defined by several authors and sources. The Wikipedia define mass collaboration as follows:

***"Mass collaboration** is a form of collective action that occurs when large numbers of people work independently on a single project, often modular in its nature. Such projects typically take place on the internet using social software and computer-supported collaboration tools such as wiki technologies, which provide a potentially infinite hypertextual substrate within which the collaboration may be situated.*
*A key aspect which distinguishes mass collaboration from other forms of large-scale collaboration is that the collaborative process is mediated by the content being created - as opposed to being mediated by direct social interaction as in other forms of collaboration."*

In [70] Mass Collaboration is defined as *"a collaboration model that based on collective actions that occurs and takes place while large number of contributors and participants work independently but collaboratively in a single project which is modular in its nature. Such projects typically take place in the Internet by the mean of Web based collaboration tools".*

Tapscott & William [70] developed seven models of mass collaboration. Each model represents a different ways of mass collaboration:

| | |
|---|---|
| Peer Producers | Peer production is described [71] as a new model of economic production in which the creative energy of large numbers of people is coordinated (usually with the aid of the Internet) into large, meaningful projects mostly without traditional hierarchical organization (and often, but not always, without or with decentralized financial compensation). Linux, Wikipedia, sourceForge, Amazon (peer reviews), are pioneering examples. |
| Ideagoras | A new global marketplace for ideas, solutions, innovations, creativity, based on places on the internet where large numbers of people and/or businesses gather to exchange ideas and solutions. This marketplace (Ideagoras) is booming world wide and outside boundaries on organization and firms. Organizations looking for improvements and enhancements can tap this global market place in order to achieve what they couldn't do internally and by their employees. |
| Prosumer Communities | Prosumer is a compound word formed by the words producer and consumer. It refers to new generations of customers (Producers-Consumers) who are co-producing and participating in the creation of goods and services by being engaged in products lifecycles, they put their own designs and customization to the products they will consume. Lego Mindstorm is an example of a prosumer community. |
| New Alexandrians | The Alexandrian Greeks were passionate about accumulating and aggregating knowledge and bringing it in one place. But in |

| | |
|---|---|
| | the recent past, firms relied heavily on closed, hierarchical approaches to producing and utilizing knowledge. Today, knowledge is being increasingly viewed as the product of networked people and organizations looking for new solutions to specific problems. Collaboration, publication, peer review and exchange of precompetitive information are becoming keys to success in the knowledge based economy. Leading scientists and observers expect more change in science than any time before; their expectations are based on the emerging of collaborative science which is a model of mass collaboration that is placed in the scientific community. Companies working in the scientific field are taking this opportunity and rethinking the way how they are doing science, and trying to achieve goals by this way of collaboration. |
| Platform for participation | Open platforms are greatly increasing mass collaboration. A platform might be a web service, an e commerce system or a developer ecosystem where a company opens its software services and databases via an application programming interface (API). For example, vibrant developer communities have formed around eBay, Google and Amazon.<br>Mass collaboration changes the attitudes and policies of different number of firms (e.g Amazon, SAP and Google) and instead of concentrating severely on protecting their trademarks and copyrights they open their products and infrastructure to thousands of innovators in the community to create vibrant business. |
| Global plant floor | A wave of digital fabrication technology looks all set to put the means of producing physical objects in the hands of every household and community. Thanks to such technology largely driven by modularity, ordinary people may be able to produce objects that were earlier produced only by large scale industrial manufacturers. Companies producing good (food, clothes, etc.) are becoming more aware of the global economy; they are making their goods internationally and deploying global capabilities and resources. In addition, the number of distributed networks (networks that distribute goods), multinational factories, and subsidiaries and business units which are distributed worldwide are rising in a global basis. All this revolution in the way of producing, marketing and distributing is a part of the evolution of mass collaboration. |
| Wiki Work place | The new web is also reshaping organizations and workplaces in a fundamental way. Increasingly, employees are using blogs, wikis and other tools to collaborate and form ad hoc communities spanning departmental and organizational boundaries<br>In today organizations employees are using blogs, wikis, social networks and other collaborative tools and services forming communities across organization boundaries. This new workplace fits with the nature of today's work, the work that becomes more complex, more team based, more reliant on technology and internet, more time pressured and less dependent on the location, place or geography. |

The building blocks of these models are **openness, peering, sharing** and **acting globally**.

### 2.1.2 Role of user communities: tapping knowledge

User communities act as a 'helpline' where members reach out to each other in an effort to help one another. E.g. members post answers and questions back and forth, test out and comment on each other's creations, but also address the companies (e.g. to report a bug). When members share, free of charge, their 'innovations', others benefit from it, gaining access to 'fresh' and 'novel' content or features to the original product or service [126] . Also see notions on "warm experts" [223] and "local experts" [224] play a major role in getting people to feel comfortable in and to find use for the internet in everyday life.

Companies could perceive user communities as sources of costs (e.g. offering support), but also as opportunities, because consumer-to-consumer support could lower the amount of assistance that the firm would otherwise need to allocate to support [126] . *"Sharing of innovation is a key condition for firm-hosted user communities to succeed."* [126]

The idea of sharing 'innovations' leads to the notion of 'free-revealing' of information, which can be defined as: giving access to all interested parties, without expectation of financial compensation [127] . The question that arises is: *what drives users to share?*

Franke and Shah [128] suggest the following five motivations (the first four based on [127] ):

A Community members like to improve others

Assistance (support) could be seen as an important aspect to improve innovations. Communities offer 'innovators' (lead users) two vital benefits. First, other members offer their support. Second, members can transfer them to individuals on the 'outside' [128] . Franke and Sha note that the community members offering support where not driven out of an eagerness to use nor to improve the innovation, although this could be a result of their behaviour [128] .

Franke and Shah [128] report a high satisfaction from 'innovators' regarding the help received. Members who offer their assistance are perceived as creative and innovative, offering additional skills and helpful expertise. This shows that 'mutual help' within communities can operate as an effective support organism. The collaboration between members depends on the degree of ease with which their innovations can be passed on to each other [126] .

B Community members like to set standards

Once an innovation is adopted on large scale, it could become an informal standard. Being the first to reveal an innovation thus increases an innovator's chance to have his solution adopted, which could act as a motivation to be the first to reveal. However, unlike in professional contexts, there is no commercial motivation [128] .

C Community members like to reveal for free - in non-competitive conditions

'Free-revealing' takes place in both high and low competitive conditions. Jeppesen and Frederiksen [126] nevertheless remark that to users in a professional context, openness could conflict with the intentions to harvest the benefits of innovations. Community members participating as a hobby have nothing to lose.

With regards to this, Rosner and Bean [129] , noted in their research on 'IKEA hacking' that some users need to set aside their fear of 'individuality-robbing copycats', to be able to 'reveal' their creations. The notion of 'free-revealing' could thus be termed as 'non-concurrent collaboration' [129] .

D Community members like a to be recognised

'Reputation effects' seems to influence the decision to share innovation-related information less than mutual expectations do. The strongest motivations to support someone are social motivations (as e.g. 'helping others is the right thing to do'), and not personal benefits [128] .

Jeppesen and Frederiksen [126] found that 'peer recognition dynamics' do not seem to influence innovative users. They are more interested in the technical aspects of the innovation. Therefore, they situate themselves less on the level of their 'peers', and more on that of the 'firm-based developers'.

Companies could acknowledge innovations of community members by granting them recognition [126] , e.g. by exhibiting the best examples and credit the contributors for it. Recognition from the firm could than also lead to peer recognition.

E Community members like to feel satisfied

Community members do not see participation and contribution as a cost that needs to be compensated, but in fact as a pleasing action [128] .

### 2.1.3 Open source movement

The community attitude that is linked to most DiY projects engages people to share, remix and rebuilt based on the work of others. Most DiY communities release all their material using some kind of open source (opensource.org) or creative commons (creativecommons.org) licence. It is via this kind of licensing that many DiY related projects create some kind of foundation to base their activities on.
For the people involved in DiY communities it is important that these types of 'open' licenses are embraced and respected.
Share and learn is something that is very typical to open source projects. Looking back at section 1 of this document, a lot of parallels can be drawn between the ongoing open source movement on the internet and the 'roots' of DIY in the physical world.

## 2.2 Human centered design

### 2.2.1 Existing models & approaches

#### 2.2.1.1 Concepts

Human Centered Design (HCD), or User Centered Design (UCD), is all about designing products tailored to users. A product designed with HCD should hide the technique and details of how it works and be completely intuitive to use.
Human Centered Design is a methodology that maximizes the likelihood that a product will meet user's wants and needs, behave the way they expect it to, and provide them with a quality user experience that will not only satisfy, but also delight them.
It assumes that all the participants in the design process bring their own personal bias into the process and that the actual end-users are really the only participants who can even come close to providing objective input as, after all is said and done, they are the ones that are going to use the product.

Overall Principles of Human Centered Design:

- **Understand the Problem:** Determining the target market, intended users, and primary competition is central to all design and user participation. You MUST know who you're designing for.
- **Understand Users:** A commitment to understand and involve the intended user is essential to the design process. If you want a user to understand your product, you must first understand the user.
- **Assess the Competition:** Successful design requires ongoing awareness of the competition and its customers. Test your users tasks against the competition. You cannot design successfully in isolation.
- **Design the Total User Experience:** Everything a user sees and touches should be designed in concert.
- **Evaluate Designs:** User feedback is gathered early and often, using prototypes of widely ranging fidelity, and this feedback drives product design and development.
- **Manage by Continual User Observation:** Throughout the life of the product, continue to monitor and listen to your users, and let their feedback inform your responses to market changes and competitive activity.

Models of a user centered design process help software designers to fulfil the goal of a product engineered for their users. In these models, user requirements are considered right from the beginning and included into the whole product cycle. Their major characteristics are the active participation of real users, as well as an iteration of design solutions.
The different models are

- **Cooperative design** (see also 2.2.1.3): involving designers and users on an equal footing. This is the Scandinavian tradition of design of IT artefacts and it has been evolving since 1970.

- **Participatory design** (see also 2.2.1.3): [72] enables designers and developers to learn user needs and preferences through a facilitated group design process. With users in a participatory design session, product teams gain user input early in the process. The results are early design ideas that reflect user requirements. In participatory design, a team of people who represent the stakeholders in a product design effort—users, designers, and developers—work together to create product designs that reflect the way customers will actually use the product. Group members use common materials such as sticky notes and markers. The facilitator refers to a list of topics, issues, and questions in guiding the session. The facilitator stimulates discussion and supports a dialogue among all the participants, ensuring that both planned and impromptu topics are covered.
- **Contextual Design (CD)** [73] is a user-centered design process developed by Hugh Beyer and Karen Holtzblatt. It incorporates ethnographic methods for gathering data relevant to the product, field studies, rationalizing workflows, system and designing human-computer interfaces. In practice, this means that researchers aggregate data from customers in the field where people are living and applying these findings, into a final product. Contextual Design can be seen as an alternative to engineering and feature driven models of creating new systems.

The most relevant Standards for DIYSE Project are those related to development process:

- **ISO 13407:1999 Human-centred design processes for interactive systems [75]**
  This influential standard is "aimed at those managing the design process" and is now increasingly used to ensure software quality.
  The standard describes four principles of human-centered design:
  1. Active involvement of customers (or those who speak for them).
  2. Appropriate allocation of function (making sure human skill is used properly).
  3. Iteration of design solutions (therefore allow time in project planning).
  4. Multi-disciplinary design (but beware overly large design teams).

  And four key human-centered design activities
  1. Understand and specify the context of use (make it explicit – avoid assuming it is obvious).
  2. Specify user and socio-cultural requirements (note there will be a variety of different viewpoints and individuality).
  3. Produce design solutions (note plural, multiple designs encourage creativity).
  4. Evaluate designs against requirements (involves real customer testing not just convincing demonstrations).

  The standard itself is generic and can be applied to any system or product. Two completed projects part sponsored by the European Commission have produced sourcebooks of methods that can be used to implement the standard: The INUSE and TRUMP projects.
- **ISO/TR 16982: 2002 Ergonomics of human-system interaction -- Usability methods supporting human-centered design [76]**
  This standard provides information on human-centered usability methods which can be used for design and evaluation. It details the advantages, disadvantages and other factors relevant to using each usability method.
  It explains the implications of the stage of the life cycle and the individual project characteristics for the selection of usability methods and provides examples of usability methods in context.
  This technical report outlines the different types of usability methods that can be used to support user centered design.

### 2.2.1.2 User-centered design

User centered design is a design process whereby the user is central and plays a key role in the design process. [..] user-centered design (UCD) is a design philosophy and a process in which the needs, wants, and limitations of end users of an interface or document are given extensive attention at each stage of the design process.(User-centered design, Wikipedia [174] )

If we design for the users, we need to know about them. The product or service will be part of their lives and we need to know if and how the design will fit their lives and benefit the users. Therefore we need to involve the users in the design process, as they are the best experts of their own lives.

To be able to design for users, we need to understand their personality, social context, way of life, behaviour, attitudes, values, motivations, drivers and needs. This cannot be understood with traditional methods of research and moreover, the target users being special, it calls for radical ways of thinking and designing. To be able to design for the users, it is essential to involve them in the design process.

User or Human centered design could act as the umbrella term for a number of other design methods like Participatory design, Transformation design, Co-designing, Inclusive design, Universal design, Contextual design etc. The principle method behind all these methods is the same i.e, keep the user in the centre of the design process. Some design methods even describe how users are not just in the centre but involved throughout the process. Similar methods have different names and for the DIYSE project the most relevant methods have been identified.

### 2.2.1.3 Participatory and Co-design

Practicing User-centered design would mean designing with the users themselves. But if we were to exclude the word 'users' for a minute and think of all the people involved in a system, then it would involve various stakeholders as well.

"Every design research project must consider how the research participants are embedded in a larger social system and to study the network of key players who inform the user's values, beliefs, actions, lack of actions and stories." [162] .

To create a win-win-win (stakeholders-end-users-designers) situation, it would be important to involve the relevant representatives from the stakeholders in the design process too. Such a process would be called Participatory design or Co-designing.

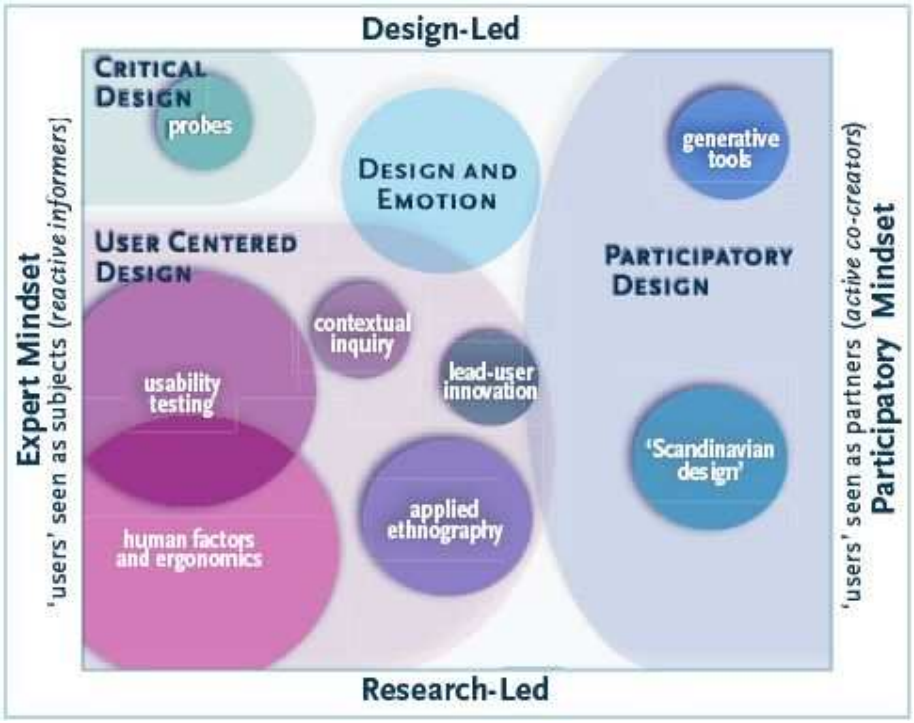



**Figure 2: Emerging trends in Design research [172] )**

Figure 2 shows different trends in design research based on the intensity of involvement of the end-users and stakeholders.

### 2.2.1.4 Inclusive design

The Finnish Consortium within DIYSE will be concentrating on different user groups such as intellectually disabled, elderly, and children. Therefore, even the experts who are involved with caring

for people with special needs will be essential in the process as participants. They also form part of the 'larger social system' of the target users.

Everyday design concepts might not work effectively on all kinds of users mentioned above. Hence the method of Universal design or Inclusive design would be essential to design a system that is not just suitable but also likable by the above groups. It would mean giving special considerations to every kind of user, with or without a different ability or cognitive capacity. There are some tools available (Goodman-Deane, Clarkson, Langdon, Waller) to practice Inclusive design, that simulate conditions for an able-bodied person to experience for themselves, certain disabilities. This will then give better insights and understanding of the pain points and problems of a person with impairments.

#### 2.2.1.4.1 Applications that address Inclusive design: apply to the disabled


**Figure 3:** TouchTone, [160]

TouchTone is an electronic musical instrument designed with a vision to develop musical ability, develop bimanual coordination and increase social participation of children with hemiplegic. It deliberately encourages physical engagement through tangible interface design to support musical creativity in children with hemiplegic cerebral palsy.

Music therapy is one of the most well-known therapies to help the mentally disabled with social and emotional aspects. It also requires the use of musical instruments that maybe difficult for the patients to use and control. This makes the therapy less effective. TouchTone aims at providing a usable musical instrument for the children to play and learn with.

Another example is a gadget imagined by Mac Funamizu, that could possibly be manufactured:

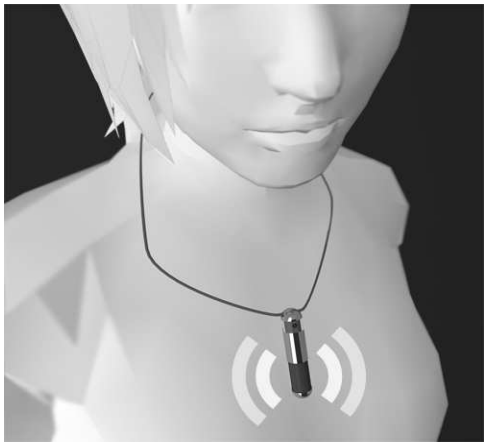


**Figure 4: Gadget 1**


**Figure 5: Gadget 2**

A small pendant-like gadget enables a vocally-challenged person to speak. The camera captures the motion image of the speaker's hand gesture. It translates it into an oral language and gives out the translated words.

Another concept in a similar area from Funamizu [164] , is a screen that shows conversations in the form of speech bubbles, so a person with hearing impairment can follow conversations easily.


**Figure 6: Screen bubbles**

### 2.2.2 Design process

"User-centered design is about understanding users, to design for them, with them." - [168] .
The International Standards Organization came up with a standard for Human-centered design process for interactive systems in 1999. The overview diagram of ISO 13407 (also mentioned earlier on page 14 and 16) is as follows:

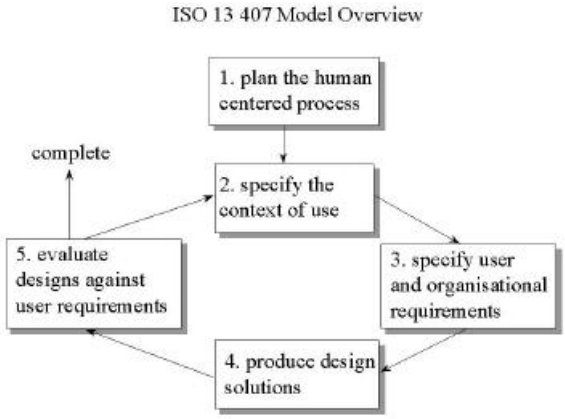



**Figure 7: ISO 13407**

Figure 7 indicates an iterative process, whereby the concepts and requirements are revisited after evaluation which would refine the design concept further.

Steps to involve users and the process for conducting UCD are described throughout this chapter. Different tools and methods are illustrated under each step.

### 2.2.3 Design methodology

#### 2.2.3.1 Background research

Literature reviews - gathering related existing written material on the concerned subject. This could also include 'benchmarking' to find the best solution-to-date.

Interviews (by phone or in person) - To get an initial idea of what the users are like and a rough understanding that is required in the beginning. This will help in forming a bond with the participants and aid with the planning of next steps.

### 2.2.3.2 Understanding *about current practices

Contextual inquiry – is an effective method of getting a peek into the users' lives. It is done by being over a specific period of time with the users, in their habitat or context of inquiry.

Shadowing – is a design-research method that is useful for users with particular needs (for example disabled people). This method involves silently observing, with or without a video camera to record all we see. A day in the lives of the users could give good insights that would confirm some notions discovered during other sessions and/or reveal new ones.

Design workshops - these are workshops that could include activities, group discussions, brainstorming etc. It helps in fostering the designer-user relationship for deeper understanding at a later stage. Elizabeth Sanders has a three tier simple approach to UCD - Say-Do-Make [173] see Figure 8. The 'say' part of the approach would be in this stage of the process, where we 'listen' to what the users have to say.

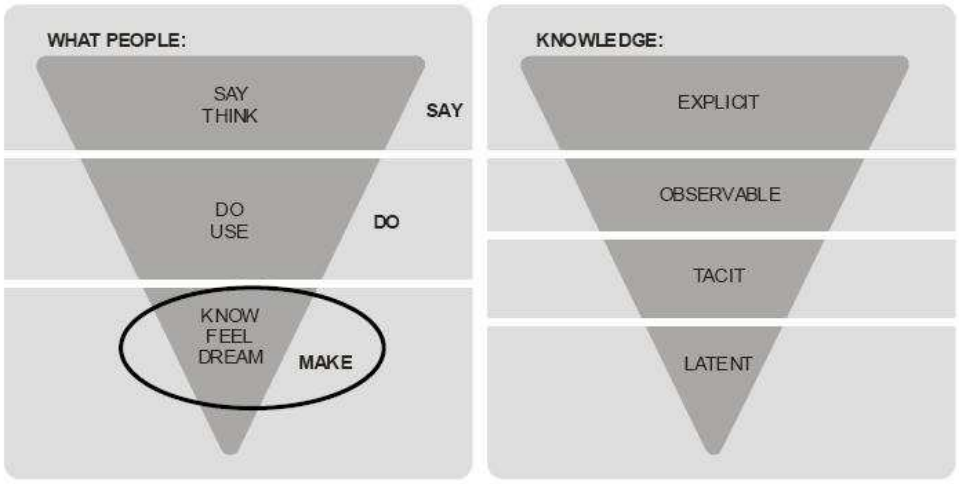



**Figure 8: Say Do Make diagram from Sanders, From User-centered to Participatory Design Approaches, 2002. [173]**

### 2.2.3.3 Deeper understanding *about wishes

Dreams - letting users talk about their dreams.
Personify an object - 1. to understand how they perceive it and the relationship between them, 2. embody emotional and social reactions.[163]

Probes - "Probes are based on user participation by means of self-documentation. Probes are a collection of assignments through which or inspired by which the users can record their experiences as well as express their thoughts and ideas. Secondly, probes look at the user's personal context and perceptions. The assignments focus the users' attention and record their daily lives including social, aesthetic and cultural environment, needs, feelings, values and attitudes." [168] . They can be cultural probes, domestic or empathic probes.

Games - using games during the co-design sessions allows to share the references in order to bridge different points of view: the games provide a common platform for the conversation between all the participants. (Designing Games)

In this stage we would make the users 'do', so we can 'observe' and learn.

#### 2.2.3.4 Concepts about future possibilities

Personas - Each distinct behavior pattern (that is observed through research) becomes the basis for a persona: a description of an archetypal user. (Modeling).
Personas are fictitious characters that represent a set of users with attributes like: from their social and demographic characteristics, to their own needs, desires, habits and cultural backgrounds.

Storytelling - The storytelling supports the exploration of the service idea. Through the use of simple words, the teller will illustrate the solution as it is a story. This allows the communication of the idea inside a group but also the preparation of the first sketches for the storyboard. The storytelling leave some blanks to be fill in by the suggestions of other stakeholders and users.

Use cases - The use cases are traditionally used in the interaction design projects for the development of the interaction flows. They are a means of roughing out the functionality of a product or of a service.

Story board - The storyboard is a tool derived from the cinematographic tradition; it is the representation of use cases through a series of drawings or pictures, put together in a narrative sequence.

#### 2.2.3.5 Prototyping and Testing

Wizard-of-Oz prototyping - It is a technique derived from the information technology that is used in order to test a product or a service in a detailed way by observing the interaction of a potential user with the object without revealing the evaluator's presence. (Wizard-of-Oz)

Experience prototyping - The experience prototype is a simulation of the service experience that foresees some of its performances through the use of the specific physical touchpoints involved. The experience prototype allows designers to show and test the solution through an active participation of the users. (Experience Prototyping)

Using drama/Informances - Eric Dishman and team, constructed another name, Informance, for what could also be experience prototyping. Informance stands for informing performance. It was coined when they built a set that represented an old age home. The designers and researchers enacted roles of elderly patients in the home and how they could interact with each other and the system that they had envisioned and prototyped. (ElderSpace project at Interval Research, 1993) The same method was used again in 2003, in an Intel ProHealth project.
They used this method and the audience for their performance were elderly who gave criticism and feedback about the concept.
There was however a drawback according to Dishman; elderly could not very well envision the future concepts. They could not think beyond what they currently experienced in their lives. Another method might be to involve the users themselves in the role-play and experience prototyping!

#### 2.2.3.6 Revisiting

According to the feedback that we get from the prototyping and testing, the design brief and requirements might need revisiting. This would ensure a sound design that erases as many design flaws as possible. How these methods can be used depends on the kind of users, project timeline and approach. Figure 8 includes a graph that maps various research methods.
"Modern day user experience research methods can now answer a wide range of questions. Knowing when to use each method can be understood by mapping them in 3 key dimensions and across typical product development phases." [171] .

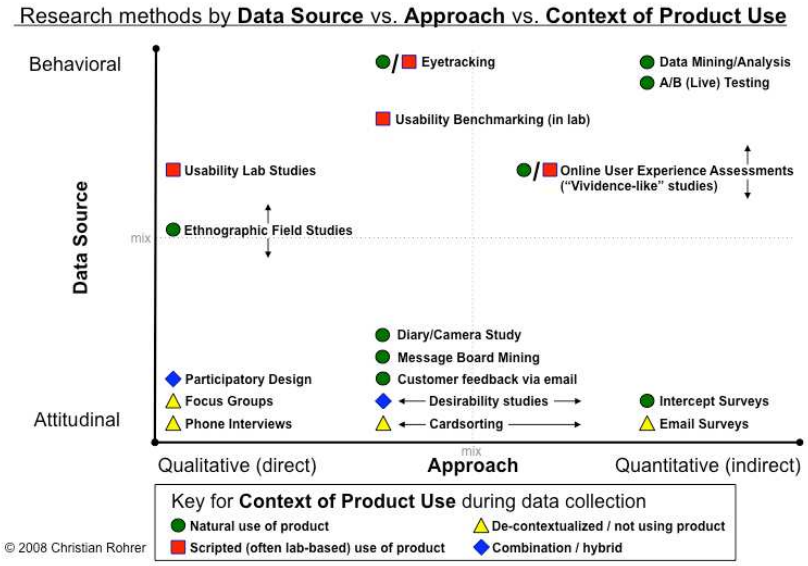



**Figure 9: When to Use Which User Experience Research Methods**

### 2.2.3.7 Challenges and Cautionary notes

In case of certain specific end-users, for example people with some sort of disability etc., it is very important to bear their welfare, feelings and reactions in mind when we carry out research in context.
The ideal situation would be to have a research strategy with a mix of all the above methods. When using only for example videotaping which is proven to be long and tedious, as there would be hours of tape that one has to go through and there might be just a few minutes of important footage it is important to find a mixture of the right methods. Videotaping might give very cold insights to the happenings, therefore videotaping must be supplemented with contextual interviewing and observations. (Schuler & Namioka, 1993, 131-139.)

It is very important to connect and create a good relation with the people to do research with. They need to trust you and feel comfortable to interact with you. One way to do this is by being genuine, honest, friendly and never give the impression you're judgmental even if you are not in reality. One big factor in getting their deep thoughts is to let them know that nothing is right or wrong. They are the experts on their own lives! It is important to never go for the research with preconceived ideas and thoughts. Also they key rule is to 'listen' and not try to fill in words or prompt the person while they're trying to answer or narrate. Another important thing to remember in addition, is to look out for what is *not* being said. Although reading into the silence or reading between lines should not be overdone, instead should be found out through probes and workshops. (Daina, during her lecture on Research questionnaires, 2009.)

Concerning videotaping activities and spaces, it is highly important that the people are notified and they're fully aware of why it is there and what's going on. Depending on trials, the informal interviewing can take place first, followed by videotaping and/or contextual observation and then finish with probes and workshops. This would create a gradual trust-building with the participants.

Finally, patience is of utmost importance in such research, as it can be a tedious process in itself, and results can take time to surface. Participants might be varied because of the variety of complexes and characteristics. It is essential to deal with all kinds of people, for which empathy is a key characteristic a researcher should posses. Along with this, unbiased curiosity and an analytical mind is essential to generate interesting.

### 2.2.3.8 Analysis

Gathering insights can be considered a psychological activity. Keywords that emerge out of interactions can be put up on a wall. Picking out those keywords is an important task. They could be about feelings, like/dislikes, actions, habits, behaviours, attitudes. Sometimes someone has to look

beyond the obvious to extract the insight behind the words or actions. Intuition will also play a role here.

Once those words are picked out on post-its and stuck on a wall, as a group one can think of how to group them and create what is known as an Affinity diagram. Similar topics/words go together. There might be problems as some words may fit into more than one group. This can be resolved by either making another group for the in-betweens or firmly deciding as a group where the keyword should belong. This grouping will give rise to themes and main focus or problem areas.

Once these themes emerge, ideas can be thought about that try to solve each theme. In this way various concepts would be generated. Ideally a session can take place where this grouping and themes is done with caretakers and others involved.

Some names and definitions of analysis methods from Steve Baty, Johnny Holland magazine (an open collective talking, sharing and finding answers about the interaction between people and products, systems or processes, [159] ): :

- Deconstruction: breaking observations down into component pieces. This is the classical definition of analysis.
- Manipulation: re-sorting, rearranging and otherwise moving your research data, without fundamentally changing it. This is used both as a preparatory technique - i.e. as a precursor to some other activity - or as a means of exploring the data as an analytic tool in its own right.
- Transformation: Processing the data to arrive at some new representation of the observations. Unlike manipulation, transformation has the effect of changing the data.
- Summarization: collating similar observations together and treating them collectively. This is a standard technique in many quantitative analysis methods.
- Aggregation: closely related to summarization, this technique draws together data from multiple sources. Such collections typically represent a “higher-level” view made up from the underlying individual data sets. Aggregate data is used frequently in quantitative analysis.
- Generalization: taking specific data from our observations and creating general statements or rules.
- Abstraction: the process of stripping out the particulars - information that relates to a specific example - so that more general characteristics come to the fore.
- Synthesis: The process of drawing together concepts, ideas, objects and other qualitative data in new configurations, or to create something entirely new.

Data collected on video can be analyzed using the video card game method, whereby clips are made of relevant and important material and distributed among participants. These participants could be stakeholders and researchers. Each participant writes down his observation on a card that matches the clip watched. Every participant then reads aloud and shares his observations about his set of clips with the other members. After which, the cards are grouped according to similar notes or observations and categories are made. These categories could be the arising themes for the design process. In this way, the task of analyzing video becomes simpler due to distribution and also involves stakeholders in the process.

#### 2.2.3.9 Data representation and visualization

It is important to visually represent processed information, be it the gathered data that is analyzed or a concept that has risen from it. By visually representing the information, all the participants involved and stakeholders get a clear understanding of the situation on viewing it besides getting more insights than one would normally get from textual data. Communicating the right information visually is helpful for the progress of the project. Some representation techniques for service design (from www.servicedesigntools.org) are as follows:

System map – this is a visual map of the service solution which shows the different actors involved, flows of material, energy etc. The system map (Figure 10) could be used as a starting point for building the scenario.

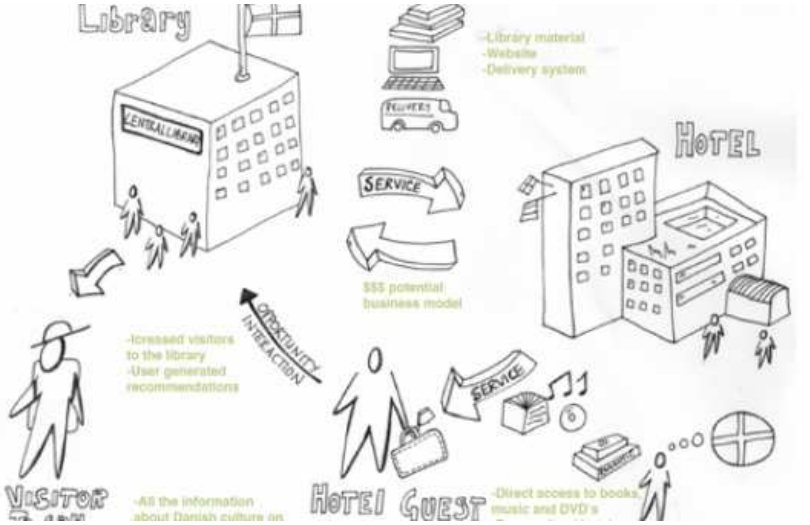



**Figure 10: system map**

The blueprint is an operational tool that describes the nature and the characteristics of the service interaction in enough detail to verify, implement and maintain it.

The touchpoint matrix provides a visual framework that enables the designer to “connect the dots of the user experience”.

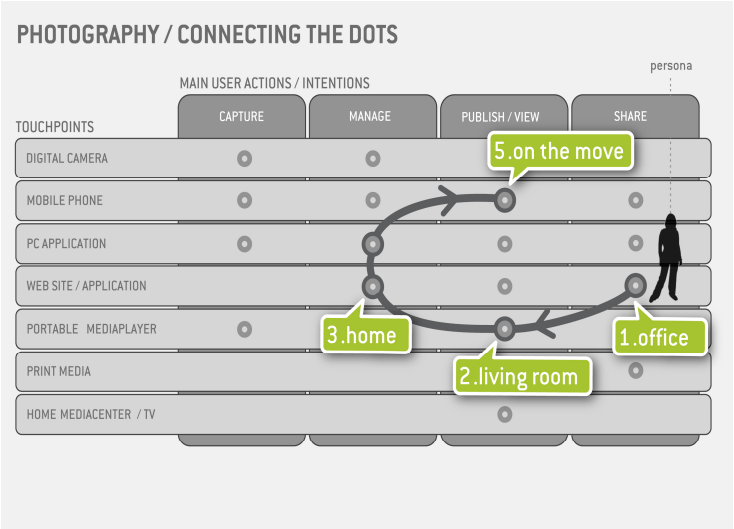



**Figure 11: Touchpoint matrix**

The ecology map is a graph representing the system of actors with their mutual relations. It provides a systemic view of the service and of its context.

### 2.2.3.10 User Experience

*User experience* (UX) is a holistic concept used to describe the overall experience that the user has of using a product or a system. The user experience research focuses on the interactions between people and (mostly technological) products/services, and the experience resulting from the interaction. The experience evolves over time as user’s previous experiences affect the overall experience of the product.

UX is often understood to cover more or less all the aspects of end-users' interaction with a system. UX is defined as "a result of a motivated action in a certain context. User's previous experiences and expectations influence the present experience; this present experience leads to more experiences and modified expectations". [157]

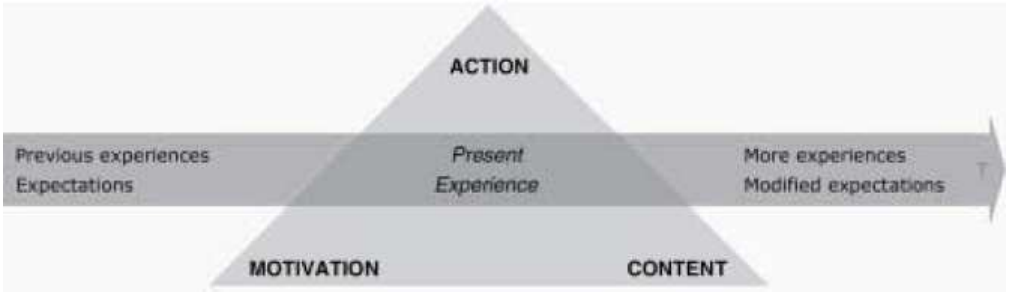



**Figure 12: Conceptual model of user experience [157] .**

Furthermore, UX is described as "a consequence of a *user's internal state* (e.g. predispositions, expectations, needs, motivation, mood), the *characteristics of the designed system* (e.g. complexity, purpose, usability, functionality, etc.) and the *context* (or the environment) within which the interaction occurs (e.g. organizational/social setting, meaningfulness of the activity, voluntariness of use, etc.)". [155]

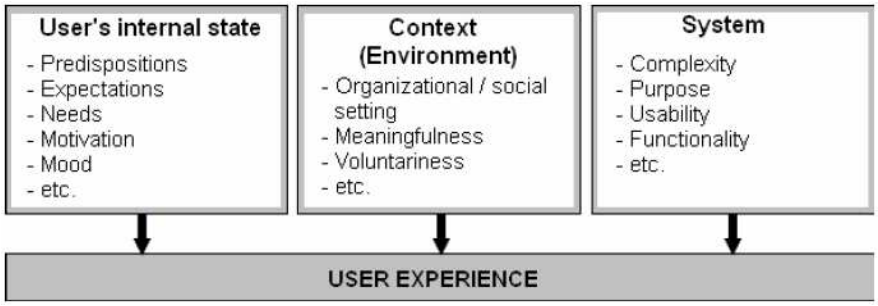



**Figure 13: Framework for user experience [155] .**

It should be noticed how multi-faceted phenomenon user experience truly is. It is important to move from traditional usability issues of human-computer-interaction (HCI) research towards more diverse aspects such as **enjoyment**, **pleasure** and **aesthetics**. Because of this holistic nature of user experience, aspects such as joy and hedonism are also brought into discussion [154] . These values have been complement by introducing also social and cultural factors as elements that affect the overall experience [152] . All in all, UX is seen to be influenced by a multitude of factors, both external and internal in relation to the user.

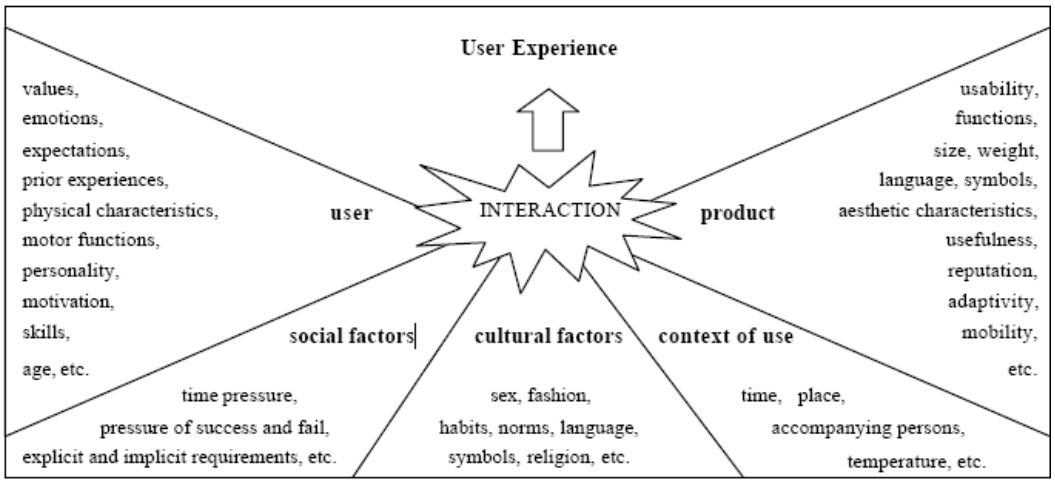



**Figure 14: Factors influencing the experience in user-product interaction (Arhippainen and Tähti, 2003) [152] .**

However, an experience can not be designed but with sensitive and skilled way of understanding the users, it is possible to design *for* experience. *"Design for experience requires the designer to have ways of seeing experience, to talk about it, to analyse the relations between its parts and to understand how technology does or could participate to make that experience satisfying"* [158] .

### 2.2.3.11 New art media approaches

New media art is a type of art that makes use of emerging technologies as an artform [206] . These technologies are often very experimental or combine existing technologies in refreshing ways. The reason why it is referred to as being 'art' is mostly because the people involved start from a certain passion for a subject, a message they want to bring across or a thought they want to express. New media artpieces are typically referred to as being 'interactive installations'.
The relevancy in the context of DiY related concepts it twofold:

1. The methods and tools used in new media art cross fertilise a lot with related disciplines such as interaction and experience design. As an example, people with a relation to new media art support lots of open source technologies and initiatives. The best know example is the Arduino board, which was first adopted by many digital artists creating interactive products and applications with it. When designers, academic researchers and companies saw the potential of this 'old technology in a new jacket' many other people hopped on.
2. People involved in new media art like to experiment with technology, creating and tinkering with devices and technology. This attitude clearly enables a DiY way of thinking and therefore becomes an interesting field to be involved in more.

During the past few years it is also becoming clear that research labs around the world are getting involved in media art. The best know example here is MIT Media Lab who is doing cutting edge research based on tools and methods often enabled via new media art. The strength of cross-fertilising art with science and technology is clearly showing it's potential in creating innovative combinations of old and new technologies and brining it to the market in new ways.

### 2.2.3.12 Service design methods

"Service design is the design of intangible experiences that reach people through many different touch-points, and that happen over the time" [170] . Many of the tools and methods used in User centred and Human centered design are naturally used within the service design practise as well.
Service design offers a holistic approach by designing a complete service concept taking the actual needs of the end-users, technical providers and external service providers into consideration.

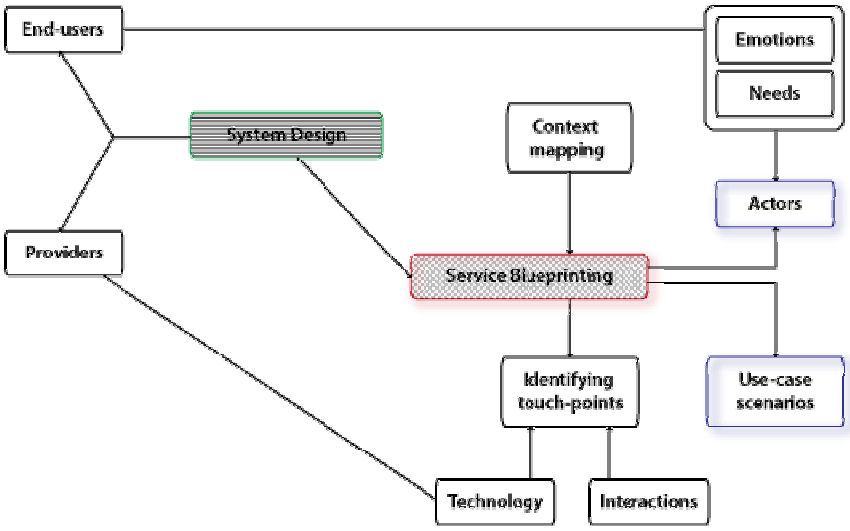



**Figure 15: Laurea, DIYSE funding application**

Figure 15 shows an example of Laurea's service design model for the DIYSE project. Besides identifying the specific needs of the different stakeholders in the service design process, also the business feasibility is addressed. This could potentially include the development of the business logic that either adds value to the service concept or saves costs.

#### 2.2.3.13 Design-for-All (DfA)

Design for All (DfA) is an approach to design in which designers ensure that their products and services address the needs of the widest possible audience, irrespective of age or ability without the need for adaptation. According to the European Commission, it "encourages manufacturers and service providers to produce new technologies for everyone: technologies that are suitable for the elderly and people with disabilities, as much as the teenage techno wizard." [153] Dfa can be seen as a wider concept than the earlier mentioned "Inclusive design", since it does not focus on *specific groups of users or specific needs of people* that are unable to use products.

It is stated, that "in a fair society, all individuals would have equal opportunity to participate in, or benefit from, the use of computer resources regardless of race, sex, religion, age, disability, national origin, or other such similar factors." [151]

There are a large number of people who are excluded (non financially) from using products or services. This includes people with problems with e.g. hearing, vision or dexterity. Nowadays there's an array of specialist products available. However, mainstream products can deliver improved access. For example, standard video phones can be used to allow deaf people to communicate.

Mobile communication has been recognised as a key enabler for connecting previously excluded groups, ranging from those with physical impairments to rural communities in developing countries. These markets require a design-for-all approach to the design of smart environments.

In principal, services should be designed in such a way that they are suitable as such for all users. The service should be adaptable so it can be adjustable to the needs of those users who cannot use it as such. Designing specific products or services especially for certain user groups should be only a final option.

#### 2.2.3.1 Conclusion

There are myriads of methods and ways of carrying out design research in a project, but it varies with the target groups, the stakeholders involved and nature of the project itself. Therefore, for the DiYSE project, it is important to filter from the relevant methods to suit the timeline and end-users of the project. Following methods will be considered.

## 2.3 Ecological Approach to design of Smart Environments (EASE)

Ubiquitous computing brings considerable challenges to the user-centred design (UCD) approach. UCD should extend beyond single products and services to networks of interoperable products with multimodal and tangible interfaces, context-aware and learning systems etc. The Ecological Approach for Smart Environments (EASE) provides a holistic perspective of designing smart environments. The approach emphasises that human-technology interaction is a complex, multi-level and multifaceted phenomenon which affects far beyond the user interface. The following presentation of EASE is based on the Finnish book "*Älykkäiden ympäristöjen suunnittelu: kohti ekologista systeemiajattelua*" (*Design of smart environments: towards ecological systemic thinking*), edited by two VTT researchers Eija Kaasinen and Leena Norros (2007) [156] .

EASE is based on a systemic view on smart environments. According to EASE, humans, technology, and environment make a system. Smartness means the ability of this system to appropriately adapt itself in a way that serves to reaching goals and creating new possibilities for activity. Thus, smart environment arises when the possibilities in the environment are such and in such a form that a human being can use and utilise them.

The main principles of EASE are:

1. The focus of design is environments in which people live and act
2. The design is based on understanding the human being and her/his practices in the environment
3. Environments can be analysed from different perspectives of activity for better understanding
4. Technologies of smart environments should be in a supportive role for human practices
5. Design can take place at several levels, from product and local design to enabling design
6. Evaluation of smart environments should be multi-dimensional

EASE does not determine any specific methods to be used in design or evaluation. Instead, the method or methods should be selected according the users, the environment and related specific needs.

In the following subchapters, the main principles of EASE are described in more detail.

### 2.3.1 The focus of design is environments in which people live and act

The advanced technologies underlying the concept of "smart environment" enable novel digital applications, services and interactions, some of them directly embedded to physical objects and surroundings, to users. These applications and services will be interconnected and interdependent with each other, building up a ubiquitous, invisible digital environment, parallel to our physical environments. To be able to understand and utilise the possibilities of this technological advancement, designing individual applications, services or their user interfaces is not enough. Instead, whole environments in which people live and act should be the focus of design.

### 2.3.2 The design is based on understanding the human being and human practices in the environment

The EASE design concept aims at improving practices – patterns of human activity and usage of tools and possibilities, manifested by persons and communities.

UCD is not sufficient for appropriate design but human orientation and intention should be understood in a larger frame. The concept of practice directs focus of attention to continuity of activity instead of separate actions and consequences. Practice allows us to study activity as a process. It is of importance *how* people act and what is the *meaning* of activity in addition to what actually they do and what results they achieve.

The concept of practice enables studying the specific environment from the perspective of generic possibilities as well as usage of technology. By including the users and their activities in the design process, it is possible to get closer to the actual use of technology.

### 2.3.3 Environments can be analysed from different perspectives of activity for better understanding

To better understand the needs and the goals of design in a specific environment, EASE presents a simple framework of analysis. Any concrete operational environment (e.g. a factory or a city area) can be, as a smart environment, considered from different perspectives that direct attention into different activities and inter-relationships within that environment. These angles justifiably include at least *living*, *service* and *production environment*.

**Living environment** From the living environment viewpoint, a smart environment manifests itself as simultaneous and intertwined human activities and attempts to control the different facets of life. The interrelationships between people are complex. Activities require controlling time and timing, as well as interruptions. People appear nomadic, they move and change environments continuously.

The user administers his/her life and its numerous simultaneously ongoing processes within the framework of personal environment. This living environment does not relate to some certain physical location. The living environment includes all the places where the user acts during his/her day.

For instance, home, work, public space and car provide good examples for identifying typical features of living environment.

**Service environment** From the viewpoint of services, a smart environment appears as a wide variety of services provided for masses of people. Producing the services is complex and requires advanced connections, arrangements, technology and infrastructure. Still the services feel simple, usable, transparent, and trustable for each single user. A smart services environment is studied as relationships of mutual services, interdependence and confidence, in addition to the viewpoint of division of labour.

For instance, movement and traffic related services represent the service environment.

**Production environment** A smart production environment refers to a goal-oriented environment, where coordinating the cooperation of several actors is emphasized. The environment is based on abstracted and networked basic information. The complexity of production environment results from the fact that the focus is aimed to perceive, administer and revise to achieve the intended results.

For instance, management of a complicated process can be considered as a production environment.

### 2.3.4 Technologies of smart environments are in a supportive role for human practices

We can define three paths of technological development towards smart environment: ubiquitous computing, advanced interaction and algorithmic intelligence.

*Ubiquitous computing* refers to information and communication technology embedded everywhere to our physical surroundings, connecting all physical objects and spaces to an enormous digital-physical network. With mobile technology, this network is (partly) wireless and can be accesses wirelessly and seamlessly even when moving between environments.

*Advanced interaction* brings us natural, intuitive user interfaces utilising several senses (multimodality) as well as gestures, expressions and speech. This development requires "teaching" the computer to understand high-level concepts to be able to communicate with humans in a way as close to natural language and human-human interaction as possible. High-level concepts may also facilitate the communication between computers in different contexts. Easy, intuitive interaction in smart environments is crucial due to their incremental development. Any smart environment will be in a continuous and necessary process of change and development similar to any physical environment: homes are brushed up again and again, cities are under construction of buildings, areas and roads all time.

*Algorithmic intelligence* enables computers to better serve the human needs by being more sensitive and responsive to the context in which individuals carry out their activities. For a human being, "context" is not a short list of simple parameters such as place (e.g., home, work, shop), time and people nearby. For instance, a context for an individual's activity includes her/his mental contents,

personal history for the activity, and her/his goals, which are multifaceted and even unconscious. These "parameters" are impossible to grab by a computer. Advanced algorithms however enable computers and thus the smart environment to learn complex activity and interaction patterns after being exposed to repeating activities and corrections of wrong guesses. A learning environment can at its best even learn to anticipate the needs of the individual before s/he even becomes conscious of them.

When designing a smart environment, EASE emphasises the human practices and activities to be carried out in that environment. The design of technology should be only subordinate to the practices. Otherwise design will too easily end up with fancy technologies for no real user need, low user acceptance and thus low level of adoption.

With thorough, need-sensitive design of technologies and technical arrangements, the environment becomes a *reserve* for individuals. Different reserves in the environment are perceived as *affordances* - possibilities in relation to the activity and goal of the individual.

Reminding that the same environment can be analysed from different perspectives of activity, the same reserves appear different affordances for living, service and production environments.

### 2.3.5 Design can take place at several levels, from product and local design to enabling design

To be able to deal with the complexity of smart environments, the design space must extend to different directions. Traditional product or system design is not sufficient to cover all different dimensions of smart environments. The ecological approach provides two new concepts to the design space: *local design* and *enabling design*.

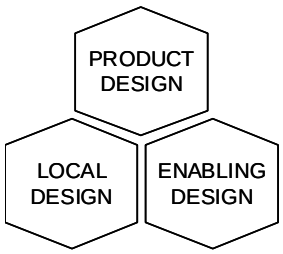



**Figure 16: Dimensions of design in EASE**

**Local design** focuses to immediate solving of problems where they have risen and working with (local) people. Local design builds on the local practices, which are the actual design matter; design is specific and situated. Local participants are active in the design process and the design proceeds on their terms. Technology is taken as an enabler and it must fit to the existing infrastructures. Design methods and practices are participatory and transparent to participants. The main participants are users and designers that are acquainted with local practices, possibilities provided by technology and methods of co- and participatory design. Sometimes the design can be based on mere activity of a user community.

**Enabling design** extends further on the time dimension and generalises over specific design matters. Enabling design generates the conceptual, informational, and methodological ground for other practices of design at different levels. The subject of design, technology development, strives for longer-term infrastructures, and directs new technologies to connect the existing ones. The goal of enabling design is to produce generic solutions that are widely applicable and enable different possibilities – with no direct value to end users (without application). Enabling design is engaged with developing technical infrastructures and societal decision-making. The actors (stakeholders) of enabling design are business, research institutes and responsible bodies of standardisation, legislation and other political steering, as well as media through its influence on public opinions.

**Product design** is still needed in addition to local design and enabling design. The challenges of product design are to build effective interfaces to local design and enabling design, as well as to

technical and business needs. The role of product design can be to be responsible for producing components and interoperable system parts for limited sub-needs.

### 2.3.6 Evaluation of smart environments is multi-dimensional

The EASE design concept aims at developing practices leaning on advanced technology. Thus the evaluation of the design product should be multi-dimensional. First, the environment should be evaluated at *individual, social and organisational levels*. Second, the change and development in the environment should be assessed in time (*temporal dimension*). Third, the evaluation should take three quality dimensions into account: *instrumental, psychological and communicative quality*. These qualities arise from the instrumental, psychological and communicative functions of a smart environment, which are explained in more detail below.

**Instrumental function** refers to supporting the ability of individual to influence her/his environment. The instrumental quality is measured as *easiness and intuitiveness of interaction*.

**Psychological function** means the support for coordination of activity and collaboration between individual, environment and tools. The psychological quality is defined as *seamlessness of co-operation*.

**Communicative function** refers to supporting mediation of meanings and creation of common consciousness. The communicative quality is measured as *meaningfulness of being and acting*.

The following table presents a summary of evaluation criteria for practices in a smart environment. The table is arranged according to the three quality dimensions. In addition, the criteria include considerations at individual, social and organisational levels, and the temporal dimension. The criteria can be used as a starting point when defining specific evaluation criteria for a certain smart environment.

| **EASY AND INTUITIVE INTERACTION** | |
|---|---|
| **Effectiveness** | Individual achieves her/his goals |
| **Efficiency** | Individual achieves her/his goals with reasonable use of resources |
| **Accessibility** | Individual achieves her/his goals regardless of her/his disabilities |
| **Availability** | Environment is operational and services available when individual needs them |
| **Comprehensibility** | Individual understands the course of events in the environment, is able to predict the influences of her/his behaviour and is able to correctly interpret the information and functions provided by the environment |
| **Affordances** | The possibilities provided by the environment are easily perceivable and easy to grab |
| **Adoption and removal** | Taking new services into use as well as discarding dispensable services is easy |
| **SEAMLESS CO-OPERATION** | |
| **Appropriate division of labour** | Activity is appropriately distributed between human and technology |
| **Effort** | Environment encumbers individual physically and mentally not too much, not too little |
| **Safety** | Mental or physical safety of individual is not in danger |
| **Privacy** | Environment protects the privacy of individual |
| **Data security** | Personal data is mediated only to directions that the individual has accepted |
| **Control** | Individual has the feeling of being in control and really is in control of the environment |
| **Trust** | individual trusts the environment as a collaborative partner |
| **Proactivity** | Environment is adequately proactive |
| **Adaptation** | Environment adequately adapts to the user and context |
| **MEANINGFUL BEING AND ACTING** | |
| **Meaningfulness** | Environment is important to individual because it carries meaning in avail, fun, or other |

| | issue |
|---|---|
| **Identity and values** | Environment adequately supports and presents identity and values of individual |
| **Community** | Environment supports the role of individual as a member of community |
| **Pleasure** | Being in the environment is pleasant |
| **Emotions** | Environment invokes desired emotions and feelings in individual |
| **Aesthetics** | The implementation of environment touches emotions, sense and yearning for beauty in individual |
| **Invisibility** | Technology is adequately embedded into the environment |
| **Respect** | Environment treats individual with respect and consideration |
| **Skills and challenge** | Environment provides individual with adequate challenges and experiences of success |

**Table 1: summary of evaluation criteria for practices in a smart environment**

## 2.4 DIY user roles

### 2.4.1 Supporting technology

#### 2.4.1.1 Pro-Am

Charles Leadbeater and Paul Miller [130] introduced the term 'Pro-Am' (Professional-Amateur), which is defined as an amateur that pursues activities, out of the love for it, but sets a professional standard, as it is being practiced with the same dedication and commitment. It is possible that this hobby takes a portion of their income. This new group of amateurs has made an uprise during the last two decades. Leadbeater an Miller [130] states that the fundamental shift in knowledge (from centralized to distributed) that was initiated by new technologies, allowed amateurs to play an increasingly important role in shaping our societies and economies. Amateurs can be knowledgeable, educated, committed and networked.

#### 2.4.1.2 Lead User

Eric von Hippel [131] introduced the concept of the 'lead user', and provides a two folded definition: 'lead users have two distinguishing characteristics:

1. They are at the leading edge of an important market trend, and so are currently experiencing needs that will later be experienced by many users in that market.
2. They anticipate relatively high benefits from obtaining a solution to their needs, and so may innovate.'

The most interesting user-developed innovations and modifications, are often coming from lead users [131] . Therefore, it is frequent for lead users to be engaged into the development processes. Inviting them as advisors or beta-testers Cooperation can offer companies an opening to discover how their products or services are being used.

#### 2.4.1.3 Bricoleur

Claude Lévi-Strauss introduced the notion of the 'bricoleur'. This user type 'uses all the concrete materials he encounters in everyday life, and all the earlier experiences of himself and others around him, to make sense of the world he is living in, and to find solutions for the problems he is confronted with in everyday life. He thus creatively and intuitively combines and recombines the bits and pieces that are available in the 'treasury' of his everyday surroundings' [132] .

#### 2.4.1.4 Local Warm Expert

Maria Bakardjieva coined the term 'warm expert' which can be defined as: 'an Internet/computer technology expert in the professional sense, or simply in a relative sense, vis-à-vis the less knowledgeable other' [133] . The warm expert has following characteristics:

1. 'he/she possesses knowledge and skills gained in the System world of technology',
2. 'he/she 'can operate in this world, but at the same time he or she is immediately accessible in the user's lifeworld as a fellowman/woman' [133] [134] .

In other words, this warm expert is often a close friend, who knows (slightly) more about the technology. The warm expert is able to relate to the 'novice user's' local situation, needs and background. He is driven through intrinsic motivation. The reward (next to a lunch) could be seen as the 'warm' feeling arising out of the gratification of spending time with a friend [135] .

### 2.4.2 Altering technology

#### 2.4.2.1 Product hacker

Stephen Flowers [136] introduced the term 'product hacker'. This is a person who: *'will seek, in some form, to change hardware or software products by either using or developing specialised modifications. In order to develop such modifications product hackers will often reverse-engineer existing products or systems, potentially violating the manufacturer's IP or violating other copy-protection laws in the process'*. The hacker's goal to alter the boundaries, could not be in line with the manufacturer's intention. Although the gaming business for example engaged with the trend, not all industries are that eager [136] .

Product hacking' thus can have a positive as negative connotation as well.

On the negative side, it fits in the larger concept of 'outlaw users'. These are *'users who, either individually or as part of a group, actively oppose or ignore the limitations imposed on them by proposed or established technical standards, products, systems or legal frameworks'* [136] .

On the positive side, there is the so-called 'Ikea hacking'. Rosner and Bean [129] introduce the notion of 'Ikea hacking', which refers to a growing interest in tinkering and customization of standard products, supported through the online world. This changes the meaning of 'hacking', as it moves from a digital to a material world.

### 2.4.3 Not using technology

#### 2.4.3.1 Rejecters

Rejecters are those people *'who have stopped using the Internet voluntarily, perhaps because they find it boring, or expensive, or because they have perfectly adequate alternative sources of information and communication'* [137] . Wyatt notes that 'voluntarily' rejection conflicts with the assumption that non-use just relates to inequality and deprivation. Some users could in fact discard Internet use, because they have access to alternative information sources and/or communication forms suitable for their needs, or consider it as difficult and expensive [137] .

#### 2.4.3.2 Resisters

Resisters are those people *'who have never used the Internet because they do not want to' [137]* . Wyatt notes that it should be accepted that some people in the end never will use the Internet.

#### 2.4.3.3 Excluded

Excluded are those people *'who have never used the Internet because they cannot access'* [137] . These users can be regarded as as 'excluded' on a social and technical level [137] . The Pew Research Center's Internet & American Life Project classifies these people as 'off the network'. In America, this group consists out of 15 percent of the Americans. Demographics learn us that these are older people with a low income. However, this group contains the heaviest users of 'old media'.

#### 2.4.3.4 Dropouts

Dropouts are those people who are 'former users'. Wyatt [137] describes various reasons for 'dropping out', these can be different for older people (e.g. the cost and difficulties to use it), and younger people (e.g. lack of interest or access). This group could be related to the 'rejects' and the 'excluded'. 'Non-users' could sooner or later, return to the Internet, due to an increased income or because of regained access. Wyatt [137] stresses the importance of finding out the reasons for 'dropping out'.

#### 2.4.3.5 Expelled

Expelled are those people *'who have stopped using it, [the Internet,] involuntarily, either because of cost or because they have lost institutional access'* [137] .

# 3 State of the art relating to DIY Smart Experiences technologies and research platforms

## 3.1 DiY Technologies

### 3.1.1 "Web" technologies

#### 3.1.1.1 Web Service

Web service [77] is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically Web Services Description Language WSDL). Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards. Web services are frequently just Internet Application Programming Interfaces (API) that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

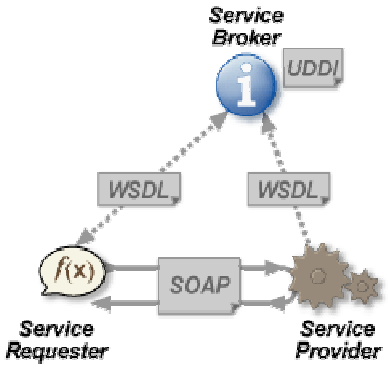



**Figure 17: Web services architecture**

Following a description of the three main elements which appear in the Figure 17

**WSDL** [78] (Web Services Description Language) is regarded as the standard for XML-based service description. In particular its purpose is to define the interface and mechanics of a services interaction. WSDL is typically built on top of a SOAP that provides the actual communication support between service provider and service requestor.

The focus of WSDL is on describing the technical aspects of utilizing a service. These include, for example, what methods the service provides, how these methods are called, what parameters are needed, etc. WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate. WSDL does not provide support for describing a service in terms of its business context, business relationships, etc. However WSDL can be used along side other service description documents that can provide this information (for example, UDDI).

A typical WSDL document is comprised of four key elements:

- <portType> - the operations performed by the web service. This can be compared to a functional library within a traditional programming language
- <message> - the messages used by the web service. This can be compared to the parameters of a function call in a traditional programming language
- <types> - the data types used by the web service. WSDL uses XML Schema to define data types.
- <binding> - the communication protocols used by the web service. For example, this could be bindings to the SOAP layer.

WSDL provides a meta-language for describing the low level aspects of a service - the operations it provides, the inputs and outputs to these operations and how to access them. In this respect WSDL plays a vital role in supporting the binding of a Service Consumer with a Service Provider. However, WSDL does not provide support for capturing details with regards to issues like what the service actually does, what processes are involved, etc. This would need to be provided by different service description documents such as UDDI, or OWL-S. Being based in XML does mean that WSDL can be extended to reflect additional properties - however the main drawback is that these would not be standardised anywhere

| | |
|---|---|
| Service interface | WSDL supports this directly. in WSDL a service may be defined as several interfaces which can be bound to different endpoints (network addresses). |
| Behavioural specification | WSDL does not support the representation of service state information or internal workflow. |
| Service invariant | WSDL does not support the specification of service invariant information. |
| Exception specification | WSDL only provides very simple support for representing exceptions, in that the message format for error messages can be specified. There is no support for failure modes however, nor for specifying what conditions will make the exceptions occur. |
| QoS specification | WSDL does not directly support the specification of QoS properties. |
| Testing support | WSDL provides limited support for run-time monitoring, in that it provides the possibility of the inputs and outputs to the service to be monitored during its execution.<br>WSDL does not provide any addition support for testing a service aside from access to the interface details |

**Table 2 WSDL –Criterion Assessment**

**SOAP** [79] (Simple Object Access Protocol): is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols; however, the only bindings defined in this document describe how to use SOAP in combination with HTTP and HTTP Extension Framework.
SOAP messages are fundamentally one-way transmissions from a sender to a receiver, but SOAP messages are often combined to implement patterns such as request/response.
Regardless of the protocol to which SOAP is bound, messages are routed along a so-called "message path", which allows for processing at one or more intermediate nodes in addition to the ultimate destination.
A SOAP application receiving a SOAP message must process that message by performing the following actions in the order listed below:

- Identify all parts of the SOAP message intended for that application
- Verify that all mandatory parts identified in step 1 are supported by the application for this message and process them accordingly. If this is not the case then discard the message .The processor May ignore optional parts identified in step 1 without affecting the outcome of the processing.
- If the SOAP application is not the ultimate destination of the message then remove all parts identified in step 1 before forwarding the message.

Processing a message or a part of a message requires that the SOAP processor understands, among other things, the exchange pattern being used (one way, request/response, multicast, etc.), the role of the recipient in that pattern, the employment (if any) of RPC mechanisms as well as other semantics necessary for correct processing.
While attributes such as the SOAP encodingStyle attribute can be used to describe certain aspects of a message, this specification does not mandate a particular means by which the recipient makes such determinations in general.

**UDDI** (Universal Description, Discovery and Integration) is a description protocol for distributed Web-based information registries of Web services – essentially repositories that store information about available web services. Service Providers can register information about the Web services they offer with these registries, and this information can then be discovered and accessed by Service Requestors.
A key concept within UDDI is UDDI business registration - an XML based file used to describe a business entity and its Web services. Information captured within this file includes contact based information (business address, identifiers etc), categorizations of the business and services (using taxonomies, etc), and technical information (specifications of the services, etc). This information can then be used to help service requestors locate relevant services.
UDDI can be considered as extending the functionality provided by SOAP to allow the querying of services and the describing of services. Within the model the business registry is logically centralised, but physically distributed with data replicated across nodes on a regular basis.
Essentially the UDDI XML Schema can be viewed as being comprised of three types of information:

- Business Information - (businessEntity within the model) Information about the business that is providing the service.
- Service Information - Technical and business descriptions of Web services that are provided by the business.
- Service specification information - This represents the additional information (tModel) that may be required to access the service (for example, what protocols are appropriate, what security is required, etc). These models are abstract enough that they can be used to point to on-line documents, links, etc

UDDI can be viewed as being a layer that resides above WSDL. Rather than focusing on the lower level binding side of a service, it focuses on providing information about the service itself - who provides the service, contact details, description of the service, references to remote documents etc. Descriptions tend to be plain text with detailed specification of a service or the processes involved referenced rather than directly specified. Like WSDL, UDDI also makes use of XML and so can also be extended to reflect additional properties. In terms of our assessment criteria:

| | |
|---|---|
| Service interface | UDDI is able to bind with different types of underlying service interface specification. |
| Behavioural specification | UDDI does not support the representation of service state information or internal workflow. |
| Service invariant | UDDI does not support the specification of service invariant information. |
| Exception specification | UDDI does not support the specification of failure nodes and exceptions, outside of simple natural language text descriptions. However, links could be provided to 3rd party specification documents if needs be. |
| QoS specification | UDDI does not directly support the specification of QoS properties. |
| Testing support | UDDI does not directly provide support for run-time monitoring, although (linking) 3rd party specifications could provide such information.<br>UDDI does not provide any addition support for testing a service aside from links to specifications (for example, WSDL). |

**Table 3 UDDI –Criterion Assessment**

More recently, REpresentational State Transfer (RESTful) web services have been regaining popularity, particularly with Internet companies. By using the PUT, GET and DELETE HTTP methods, alongside POST, these are often better integrated with HTTP and web browsers than SOAP-based services. They do not require XML messages or WSDL service-API definitions.
A highly dynamic and loosely coupled environment increases not only the probability of deviation situations that occur during the execution of composite services, but also the complexity in exception handling. Due to the distributed nature of SOA, loosely coupled feature of web services, the monitoring and exception handling issues about web services in SOA context is still an open research issue. When running composite web services, each sub service can be considered autonomous. The user has no control over these services. Also the web services themselves are not reliable; the service

provider may remove, change or update their services without giving notice to users. The reliability and fault tolerance is not well supported; faults may happen during the execution. Exception handling in the context of web services is still an open research issue.

#### 3.1.1.2 Mashup

***1. Definition:***

The term “mashup” [80] has its etymological origin in the music world, mainly in the hip-hop world, where a song is obtained by mixing and matching two or more existing pieces.

In the IT context, a mashup is a web application built from several existing applications in order to serve a user need, this need being not fulfilled by these applications separately. A mashup is a holistic concept: « the whole system is more than the sum of its parts » [Aristote]. In this sense, a mashup aggregates and mixes third-party data already available elsewhere and forms a new, composition and presentation of this data. Mashups are a mix of various protocols and technologies.

***2. Typology of mashups***

Currently, 2 main categories exist:

- Mashup v1.0: Only an aggregation of different sources, No interaction, Drag & drop of components or data sources in my page (Netvibes, iGoogle).
- Mashup v2.0: Composition: different sources of information become a single one. The mashup result is static. All mix is done through creation step (Microsoft Popfly, Google Mashup Editor,…).

**3. Architecture of mashups [1]**

The architecture of a mashup application is comprised of three components:

- API/content providers. These are the providers of the content being mashed. Content is often exposed through Web-protocols such as REST, Web Services / SOAP, and RSS/Atom. When APIs are not available (Wikipedia, TV guide), the content can be extracted by parsing the provider's Web pages, which were originally intended for human consumption.
- The mashup site. This is where the mashup application and its logic resides. The traditional approach is to implement a mashup as a server-side application with technologies such as Java Servlets, CGI scripts, PHP or ASP. Content can also be mashed directly within the user’s browser through client-side scripting (AJAX technologies), making a mashup a Rich Internet Application (RIA).. Finally, mashups can also use a combination of server and client-side logic to run the application.
- The client’s web browser. This is where the application is rendered graphically and where user interaction takes place.

#### 3.1.1.3 Widget

A widget is an interface element that a computer user interacts with, such as a window or a text box, to obtain some information or perform a simple action. It is a light-weight application or program, usually launched in small files that are running upon a widget engine (e.g. the Yahoo! Widgets engine). Their objectives are to give easy access to some frequent functions and visual information and to provide a single interaction point for the direct manipulation of a given kind of data. Widgets are visual basic building blocks which combined in an application hold all the data processed by the application and the available interactions on this data.

Widgets can be found for virtually all operating systems, as well as running inside all major web browsers. Standardisation efforts have been pushed by several industrial players such as Netvibes [146] . However, the main effort is being led by the World Wide Web Consortium (W3C) with its widget model [147] .

Three main widget types can be identified:

- **Desktop widgets** that are a small footprint applications, which reside on the user’s desktop. Desktop widgets enable the user to view on demand, encapsulated information from predetermined data source(s).
- **Mobile widgets** are like desktop widgets, but for a mobile phone. Several J2ME-based mobile widget engines exist for example WidSets. Fragmentation in the mobile operating system space makes mobile widgets largely incompatible cross-platform.

- **Web widgets** are components, built using web technologies (JavaScript, AJAX), that are embedded in web pages and which provide somewhat atomic functionalities, in line with the widget philosophy. Web widgets are usually limited by the browser that they run in, and are therefore unable to access device-specific information.

### 3.1.2 "Mobile" Technologies

#### 3.1.2.1 Mobile Ajax

Ajax [2] [3] is a browser technology that involves the use of existing web standards and technologies (XML/XHTML, DOM, CSS, JavaScript, XHR - XMLHttpRequest) to create more responsive web applications that reduce bandwidth usage by avoiding full page refreshes and providing a more 'desktop application-like' user experience.
Mobile Ajax is the extension of Ajax principles to the Mobile environment, which includes other constrained devices such as gaming consoles or set-top boxes featuring web browsers. While technologically the same thing, Mobile Ajax is looked at as a special case of Ajax in general, as it deals with problems specific to the mobile space including the areas of constrained devices and constrained web browsers in general.
Mobile Ajax uses Widgets for creating simple applications. When it comes to the mobile environment, we are seeing the same phenomenon i.e. we see Mobile Widgets powered by Mobile Ajax and also some fragmentation.

#### 3.1.2.2 Java ME

Java Platform, Micro Edition (Java ME) [4] provides a robust, flexible environment for applications running on mobile and other embedded devices. Java ME includes flexible user interfaces, robust security, built-in network protocols, and support for networked and offline applications that can be downloaded dynamically. Applications based on Java ME are portable across many devices, yet leverage each device's native capabilities.

#### 3.1.2.3 Java FX Mobile

JavaFX Mobile [148] is a complete, pre-integrated software system for advanced mobile devices designed to enable developers to author rich, high-impact content and network-based services. Built around open and standards-based technologies, JavaFX Mobile enables control and flexibility for the mobile ecosystem.
JavaFX Mobile streamlines the environment and reduces reliance on underlying technologies by providing a complete middleware and application stack implemented in Java.

#### 3.1.2.4 Flash Lite

Flash Lite [149] is a lightweight version of Adobe's Flash Player optimised for mobile devices. Flash Lite is a client-side user interface development technology comparable with Sun's Java FX and uses ActionScript as its scripting language.
The most notable drawback of this technology is that only a relatively small portion of mobile phones offer support for Flash Lite.

#### 3.1.2.5 Widset

WidSets [150] is a simple service that brings you information normally accessed via the Internet, directly into your mobile phone. Using mini-applications called widgets, it sends you updates made to your favourite websites. The system uses RSS feeds to push information from these websites directly into your mobile phone the minute they're updated. WidSets is created by a small group of Nokia employees.

## 3.2 Research platforms

### 3.2.1 Platform related to DiY

#### 3.2.1.1 MARMITE

MARMITE [24] is a framework which enables end-user to create their own mashup with an incremental execution; users can execute composite service step by step and see the intermediate results (see following figure). It is implemented as a Firefox plug-in too. Such as in Yahoo pipes, Marmite composite services are a set of boxes (called operators) chained with wires. Marmite displays intermediate results as a table (where each row is a structure that has many attributes displayed in different columns). However, some services should have alternative displays means such as a map,

and video player. In the example above, there are three chained services: find events service, filter events service, and yahoo map service. Users can link the output of the find event service (which are a set of events defined with attributes: event name, time, venue name, city, latitude, longitude) with the input of filter events service which will remove all events that satisfies a given condition (e.g. events happened before 2006-11). Thereafter, users link the output of the filter service to the inputs of yahoo Map service in order to display the places where the events happened. MARMITE authors have tested their framework on a sample of six persons, where two of them are experienced with programming, and two others are experienced with spreadsheet but not programming, and the remaining two others are not experienced with programming or spreadsheet. As a result, three out of six did not succeed to build a composite service and those who have succeeded are those who have knowledge in development and one of those who have knowledge in spreadsheet.

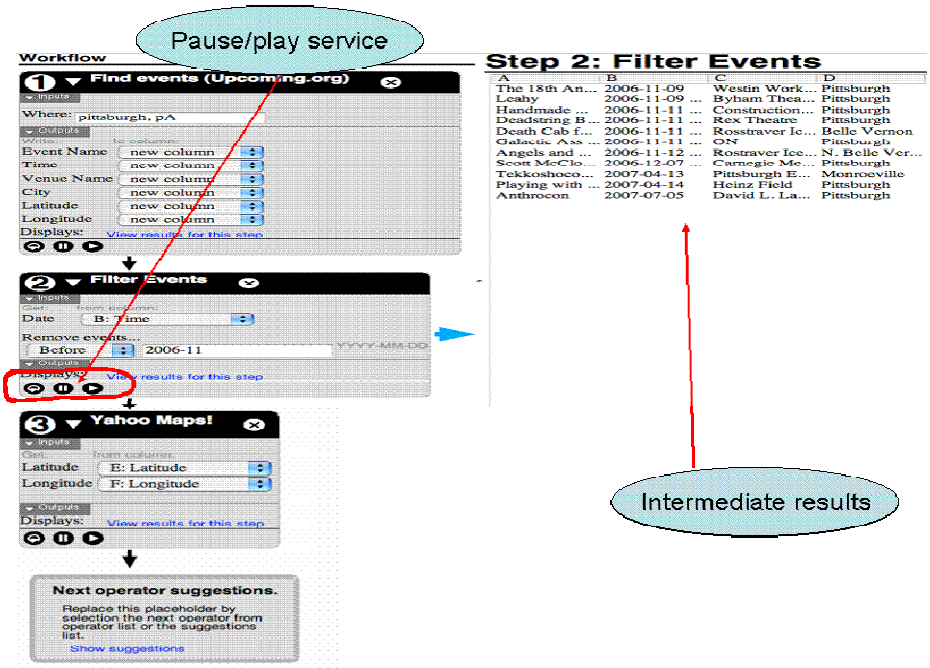



**Figure 18**: **MARMITE mashup creation tool**

### 3.2.1.2 IST OPUSE

IST OPUCE [26] offers the end-user the possibility to create his own services and share them among other users to harness the collective intelligence. The innovative elements of OPUCE are two-fold: The first one is that OPUSE Service Creation Environment (SCE) does not display the actual user interface of the service; instead, it displays a box which contains a representative picture of the service. The second innovation resides in that OPUCE SCE is also oriented to telecom services. It takes into account real-time events such as incoming-call, change in presence status, and end-call. It enables the end-user to implement services such as "when the call ends, send an MMS"

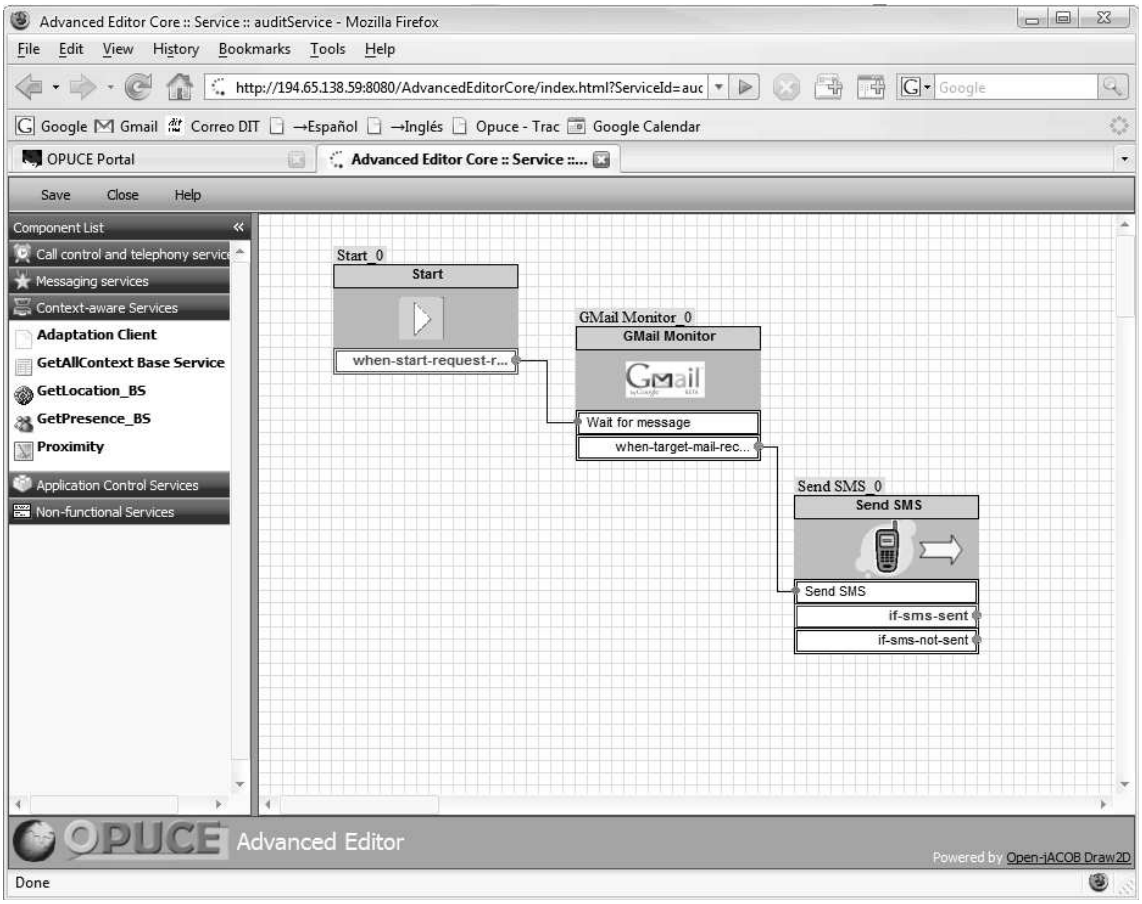



**Figure 19**: **Message Reachability composition using the Advanced Service Editor.**

#### 3.2.1.1 IST SPICE

The IST SPICE project [20] created a natural language composer tool (essentially developed by Alcatel Lucent partner) intended to facilitate the creation of simple personalized composite services.

The original user's request is analyzed, and the main concepts are extracted from it. These concepts are used by a discovery mechanism that searches into the service directories the right services matching with them. Then, a composition engine assembles the discovered services according to the logic of the request in order to build a composed service, which is deployed and executed in order to deliver the result to the user.

The figure below shows the various steps involved in the restatement of the natural request and its translation as an executable composite service.

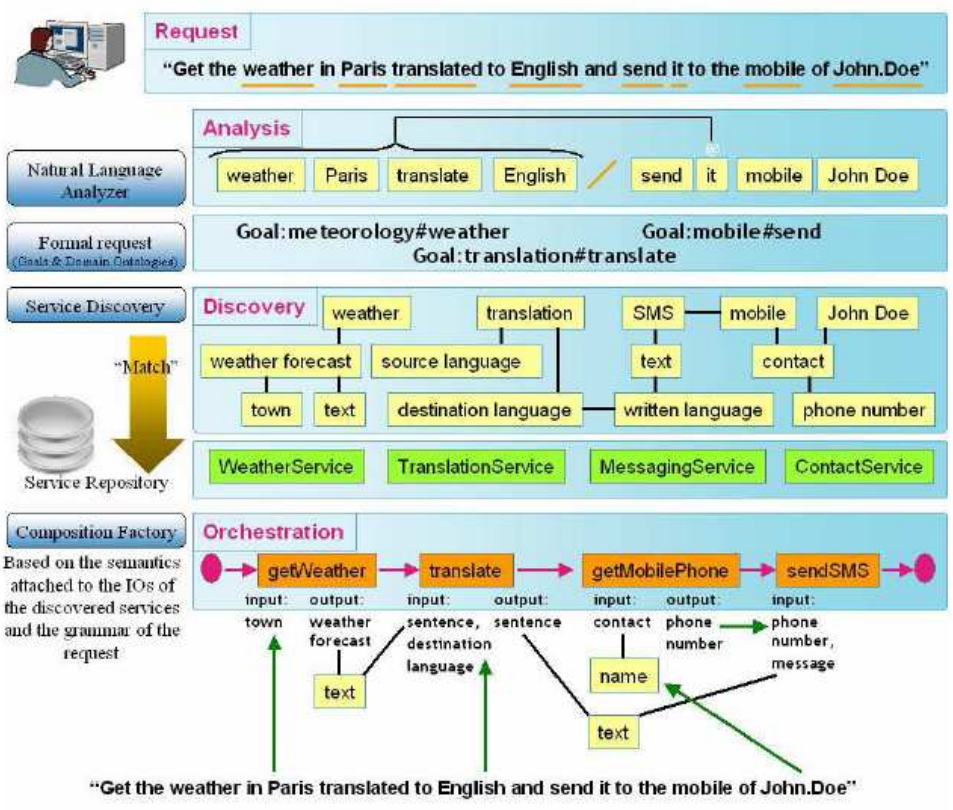



**Figure 20: An example of natural language based service composition**

### 3.2.1.2 SOA4ALL Platform

SOA4All is an Integrated Project funded by the European FP7 programme, under the Service and Software Architectures, Infrastructures and Engineering research area. The goal of the project is realizing a world where billions of parties are exposing and consuming services via advanced Web technology. It aims at providing a framework that integrates SOA and four pillars, namely, Web 2.0, Web principles, context, and semantic technology into a domain-independent service delivery platform.


**Figure 21: SOA4All platform**

The SOA4All architecture consists of the following components: SOA4All Studio, Distributed Service Bus, Platform Services and Business Services.

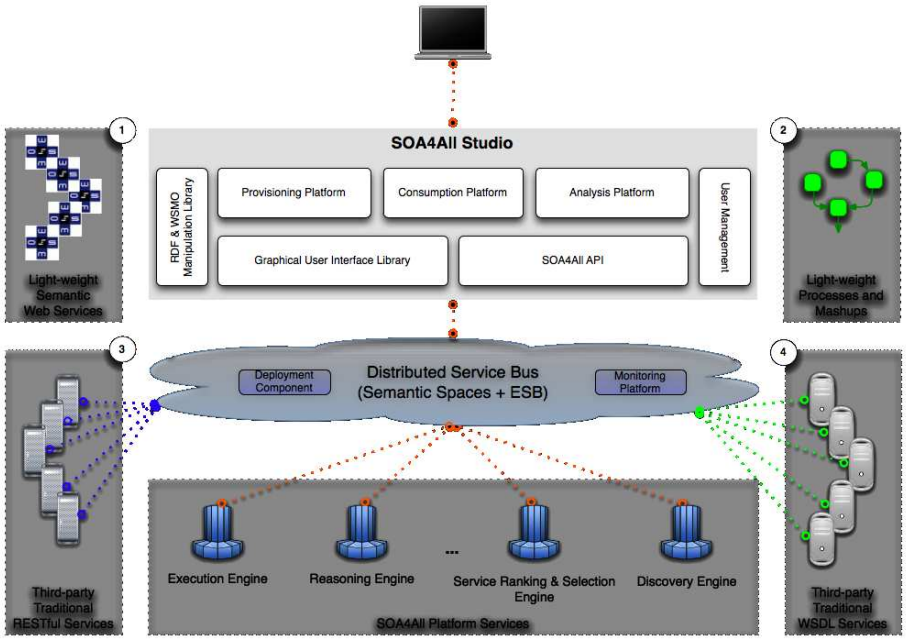



**Figure 22: SOA4All Architecture**

The DSB (Distributed Service Bus) is the core communication and integration element of the platform. It provides the traditional enterprise service bus (ESB) in a more scalable shape, where the service bus is distributed among several nodes. Hence, the single point-of-failure that is associated with ESBs is removed. The DSB component used in SOA4All is based on Petals, an open-source JBI compliant implementation.
Another element of the architecture that fuses with the DSB is the semantic spaces. These are a virtual layer on top of distributed semantic repositories. They are offered as a set of operations for publishing RDF triples or querying SPARQL endpoints. The implementation of these semantic spaces is based on P2P technology, and is deployed on the same Grid-aware middleware as the service bus nodes. The semantic spaces are conceived for large-scale data sharing. In the case of SOA4All, this data includes service annotations, goals, process descriptions, monitoring data or user profiles.
SOA4All Studio is the web-based user front-end.
Platform services are those that provide the core functionality of the SOA4All platform, including discovery, ranking, selection, and composition of services. All the components are exposed through the DSB, in such a way that maybe consume by any other service.
Finally, business services include 3rd party Web services, both WS-* and RESTful-based, but also lightweight processes.

**SOA4All approach for Semantics**

SOA4All makes use of WSMO-Lite , the lightweight web service modeling ontology based on WSMO. WSMO-Lite defines an RDF model which captures the service semantics, which is combined with domain-based ontologies to describe concrete services. The minimal service model has a concept ‘service’ and some ‘operations’ attached to it. WSMO-Lite defines four categories describing the semantics of a service:

- Functional semantics, the functionality offered by the service.
- Non-functional semantics, QOS parameters and restrictions, based on user constraints and preferences.
- Behavioral semantics, guide the client on the service invocation.
- Information model, describe the data exchanged between service and client.

WSMO-Lite is used to annotate existing service descriptions, both WS-* and RESTful. In this regard, it follows the bottom-up modeling approach, i.e. building an incremental layer of semantic descriptions on top of the syntactic layer. The SAWSDL W3C standard is the annotation mechanism proposed for

WSDL-based services. Likewise, MicroWSMO allows adding semantic annotations over the hRESTS microformat, which is a format that defines the service model structure of RESTful services.

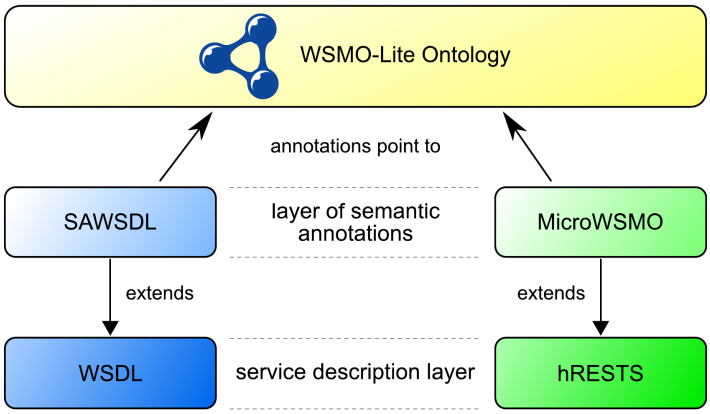



**Figure 23: WSMO-Lite Ontology**

The SOA4All initiative has developed the WSMO-Lite Editor tool for annotating WSDL descriptors using SAWSDL annotations as part of the SOA4All Studio. These semantics descriptions are the core enablers of the automation activities. They are stored in the service registry, one of the platform services, and they are used for reasoning.

### 3.2.2 Platform related to SE

#### 3.2.2.1 ITEA-AMEC

One of AMEC Framework's missions [81] is to define the standards of inter-component relations. In any AMEC System we can identify three types of components:

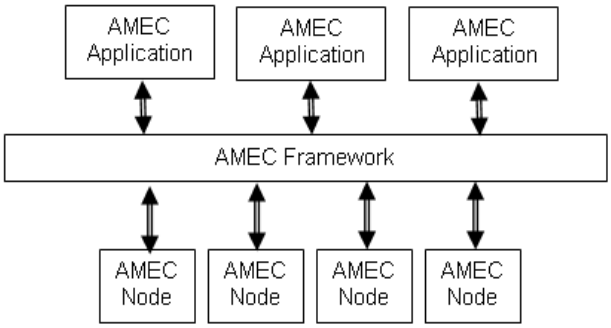



**Figure 24: AMEC System Overview**

**AMEC Framework (AF)**: The Central Computational Module, which controls AMEC Nodes in the local area. AMEC Applications can reach AMEC Framework from remote points. AF has a pluggable connector architecture which provides support for connectors to be implemented for new kind of connection methods. It includes basic set of services such as AN Manager, AA Manager, Message Manager, etc...

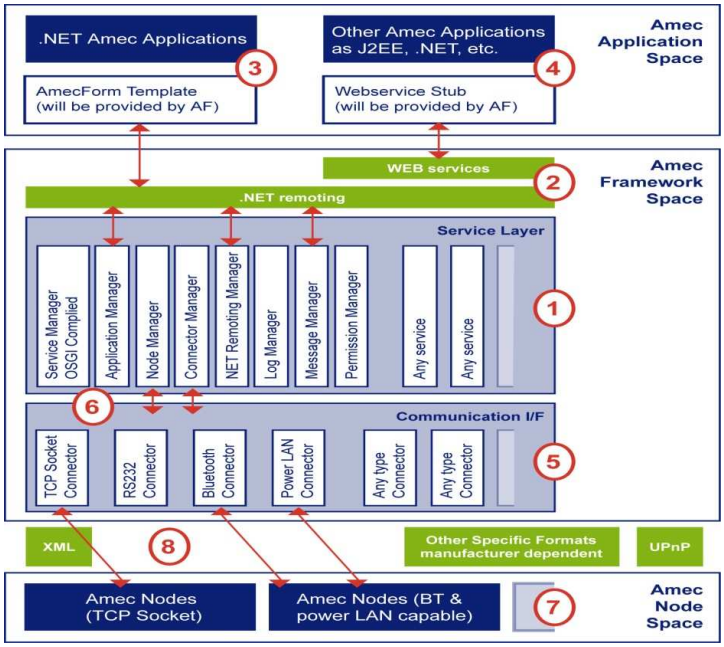



**Figure 25: AMEC System Overview**

**AMEC Node (AN)**: Simple or complex computational units which can connect to AMEC Frameworks by at least one interface and do some work. AN's should send their AN Descriptors, which is a XML message, to the connected AF, to be used by the connected AA's. AF will provide some helper program's for favourite AMEC Nodes.

**AMEC Application (AA):** The program that controls AMEC Nodes by connecting the required AMEC Framework. AF will provide a basic application to extend to simplify connection mechanisms for all of the development environments.

**AMEC Application and AMEC Node Example**


**Figure 26: AMEC Clock**

This is an AMEC Node sample, consisting of a simulation of an AMEC Clock. This node is controlled by "Application X" and its available commands are GET TIMEZONE and SET TIMEZONE. These commands are triggered from the AMEC Remote Application X as shown below. The execution is then done in AMEC Clock.

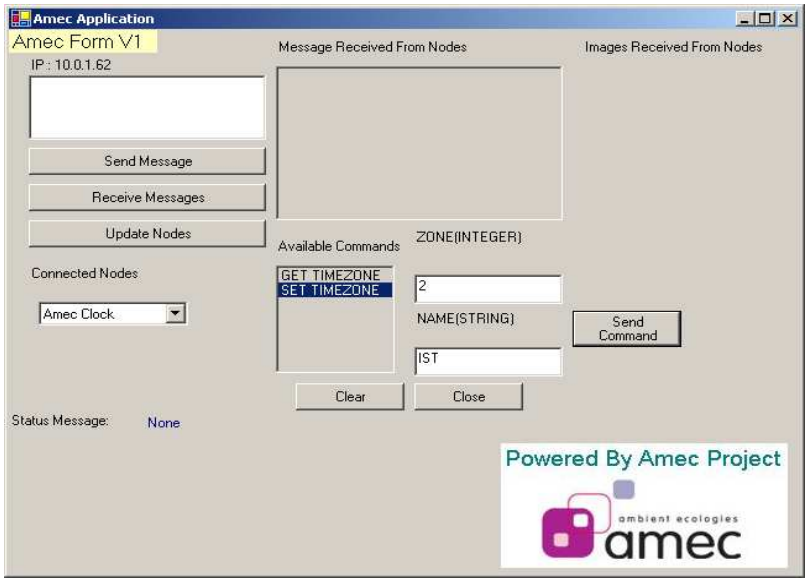



**Figure 27: AMEC Remote Application X controlling an AMEC Clock**

### 3.2.2.2 ITEA-AmIE

Europe is currently the world's major area with the highest proportion of elderly citizens and is projected to remain so for the next 50 years (source: United Nations). The main goal of AmIE project is to offer a comprehensive response to these concerns through the development and testing of a complete intelligent, distributed system for users assisted living at home (including sensors, actuators, distributed intelligence, databases, ubiquitous connectivity and friendly adaptive interfaces as its main ingredients). The overall objective is to improve the quality of life, providing customized support to all the people in need of assistance, according to their own specific situation, and in a non-intrusive and respectful way.

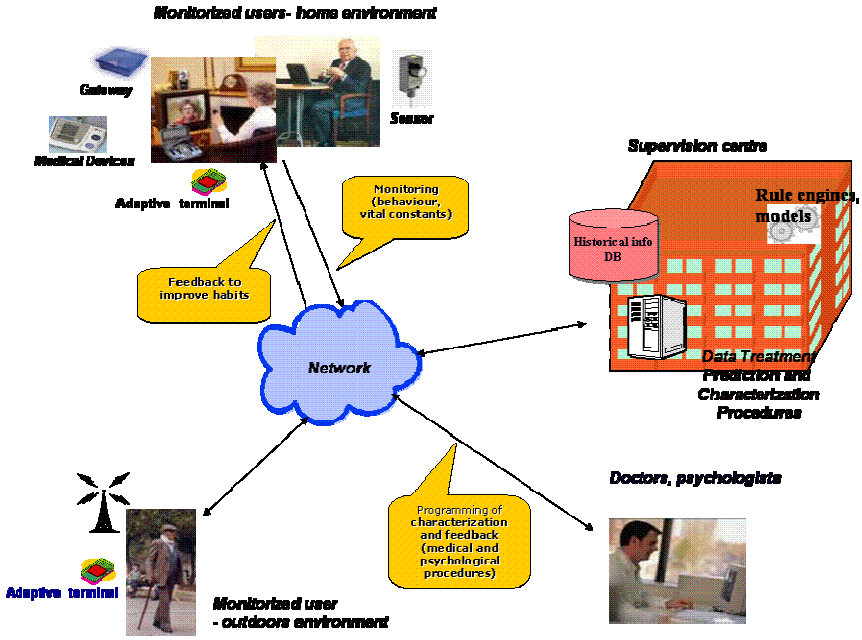



**Figure 28: *AmIE structure***

Several R&D challenges which need to be addressed to build such ambitious systems can be identified in what follows:

- User_centered design: Designing an architecture which allows for a user-centered interaction, requires a high degree of flexibility, adaptation and comfort. In user-centered paradigm it is the system that tries to meet the user personal interaction style and not vice-versa as it often happens.
- Context awareness: Pervasive computing environments are characterized by interactions among large numbers of small, heterogeneous, at least partially autonomous, embedded and mobile devices.
- Multimodal communication: The use of multi-modal interfaces aims at allowing the artificial system to engage in a human-like dialog with the user, with the objective of exploiting the richness, robustness, and flexibility of face-to-face conversation. Avatars (Virtual Characters) are employed in the home environment to act as a Virtual Assistant providing suggestions and feedback to users. However there is still no consensus on how to deal with distributed interfaces from various devices and I/O modalities.
- Reasoning and intelligence: In order to add intelligence to the system, technologies related with Artificial Intelligence are applied like rule based applications using new ways to establish facts and XML-based languages like RuleML or the Rule Interchange Format. With these techniques the overall system can learn form previous behaviours and adapt automatically to them. The techniques of Machine Learning, Semantic Engineering, Multi-agents system and Reactive Intelligence are some of the reasoning techniques that can be used to implement intelligence in the system.
- Living Ontologies: Ontologies contain the formal definitions of the key domain concepts and relations between those concepts that a community of practice has agreed upon. In the medical domain, ontology for example describes the main types of care processes, what steps these processes consist of, the actors involved in these processes, and so on. As ontological definitions are formalized, an ontology allows for the precise definition of and reasoning about the rules that prevail in a community. As such, they are indispensable elements of the AMiE rule definition module. Current research in ontologies still focuses very much on formal knowledge representation and analysis algorithms and techniques. How to ground ontology's in evolving, real-world communities, and how to ensure that the ontological definitions serve actual community information needs, such as sense data interpretation ("intelligent sensing") and service activation, is still an open question.

AmIE puts together research and industry to design a complete platform that will comply with the above mentioned requirements. Context awareness, including sensors and medical devices; communications; adaptive intelligence and characterization models; rule engines and ontologies; adaptive interfaces; actuators, domotics; applications for the doctors and psychologists to provide knowledge and supervise the evolution of the elderly people…. are all included in the scope of the project. More information on the AmIE project can be found on the website of the AmIE project [52] .

One of the concepts that was developed during the AmIE project is **Casensa**. This is a platform for creating, activating and deactivating ambient intelligence in the home of elderly people, via a magnetic NFC reader board and cards and a touch screen that the caregiver can use to control the creation environment. More information on this concept can be found on the wiki of Alcatel-Lucent R&I [52] and the website of In-HAM [52]

ITEA AmIE is going to end in December 2009.

#### 3.2.2.3 ITEA-Easy Interactions

The goal of the EASY Interactions project [52] is to assess and to propose enhanced and innovative approaches in the human-system interactions and environment for different application domains.
The project have focused on the way to introduce to the market new Human-System Interactions (HSI) in application domains that required enhanced HSIs like critical situation, mobility, enterprise business, home, industrial.

The EASY Interactions project doesn't aim to create new technologies for HSI. Innovation produced by the project is on the integration of the new HSI generation in smart systems able to cope with Context Awareness and Multimodality. Actually, the state-of-the art has shown that these new interfaces are promising but each interface used separately limits their capabilities to manage complex systems or critical situation.

This project particularly takes interest in a first category of technology which concerns the natural approach to control/command system. These technologies concern: Eyes and head tracking, Gesture Analysis and Speech-to-Text. The second category which concerns to EASY Interactions consists in the interface to return information to the user. It deals with: Speech Synthesis, Talking head, Spoken Dialog and Advanced Display Devices.

In order to probe all this concepts, a prototype had been built inside the EASY Interactions project. This prototype is a new multimodal interface to control a bridge crane [56]

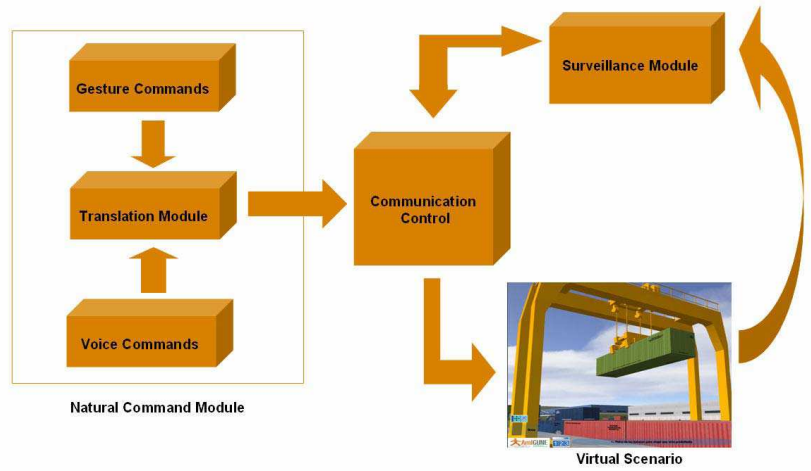



**Figure 29: Prototype Module Diagram**

The prototype is thought to be working in an industrial environment, receiving commands from the different users of the area. So, the following requirements need to be fulfilled:

1. The system has to understand natural language (gesture commands, specifically the commands given by body (position), hands and voice)
2. The system will work under a very variable illumination conditions and, generally, they are not optimised
3. There could be some interpretation errors among user given commands and commands understood by the system
4. The system needs some kind of "context awareness"

As have been pointed in the prototype of the EASY Interaction project, it is possible to introduce advanced interfaces in "*Industrial Environments"* using noise reduction methods and redundant command interpretation to increase the user security and the system robustness. This project is going to end at December 2009.

#### 3.2.2.4 Smart M3

Smart M3 software platform provides shared access to semantic information for agents in one device or in different devices [138] [139] Smart M3 is not bound to any specific operating system, programming language, or transport mechanism.

The platform philosophy is not to require tight coupling between interacting devices: Agents publish voluntarily information to shared semantic information brokers (SIBs) for other agents to use [140] The agents which are producing or consuming information to SIB are called knowledge processors (KPs). Agents communicate autonomously and anonymously via SIB.

Smart M3 has analogy to the Semantic Web, but is locally limited and without strict ties to standardized ontology. The availability of information between agents is based on common local ontology, data formats in each use case, and common information access.

One or more SIBs may be connected to each other form a smart M3 space (Figure 30). Each KP may access several spaces. A KP accessing a single SIB has available the union of information available

within the same space [138] At semantic level an application may consist of one or several KPs, running on same or different devices (Figure 31). A Smart M3 device contains a SIB, or one or more KPs, or both. To communicate with outside SIBs or KPs each device must have at least one associated connectivity method. SIBs are discovered using underlying discovery mechanics of service or transport layer.

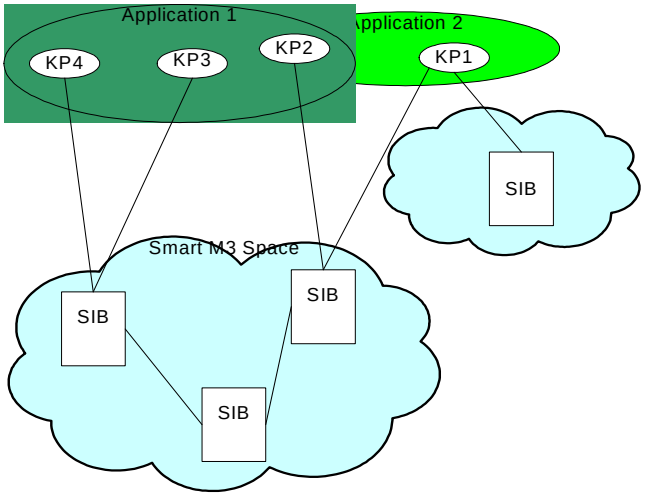



**Figure 30. Semantic overview of Smart M3 architecture**

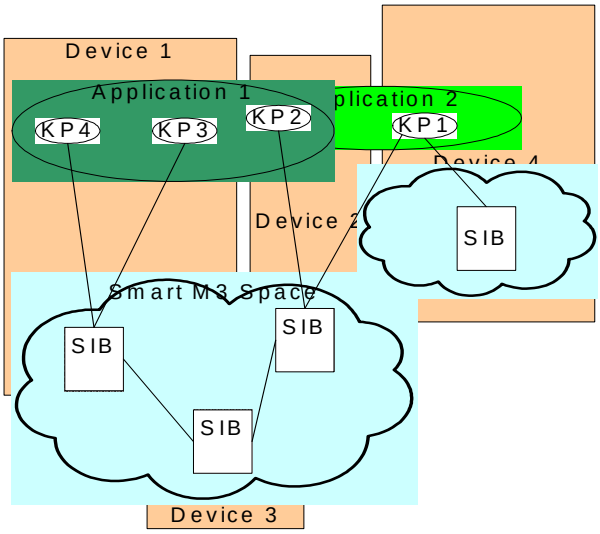



**Figure 31. KPs and SIBs distributed between different devices**

The messages between SIB and KPs follow the Smart Space Access Protocol (SSAP) [141] An operation consists of request and response messages. The SSAP operations for SIB are atomic or synchronized, i.e. only one operation can manipulate the smart space contents at a time. SSAP is transport independent, so the same SIB can serve KP over several transport technologies. Information in messages is in the form of serialized RDF graphs using either RDF/XML format, or RDF-M3 format. RDF-M3 encoding is a lightweight way to encode the object-subject-predicate RDF triples[141] .

The KPs may query SIB, or subscribe SIB for persistent queries: Whenever an insert, remove or update operation is made for the SIB the persistent queries are re-evaluated for changes and subscribers notified.

To authorize access to Smart Space by join, KPs and SIBs use implementation specific credentials.

The information in SIB may be inconsistent with the rules of ontology. As such they are not considered as error, if the KPs are expected to process the inconsistent information. The smart space may also have embedded reasoning mechanisms to fix such inconsistencies.

Smart-m3 is being developed at Artemis JU programme in Sofia (smart objects for intelligent applications) and Finnish national DIEM (Device interoperability ecosystem) projects. The Smart M3 open source code is available at SourceForge [142] .

The publicly available Smart M3 Linux implementation is shown in Figure 32. Linux device implementation uses internally D-Bus for inter process communication (IPC) between KPs and the SIB. For outside connectivity NoTA [144] and TCP connectivity exist. The development platform includes code generators for different languages to ease building interfaces for accessing SIB. They generate ontology description (OWL-DL [143] based APIs for SIB access and/or a template KP to access the SIB. The currently available implementation includes generators for C and Python. The Python generated KP is monolithic using TCP as transport.

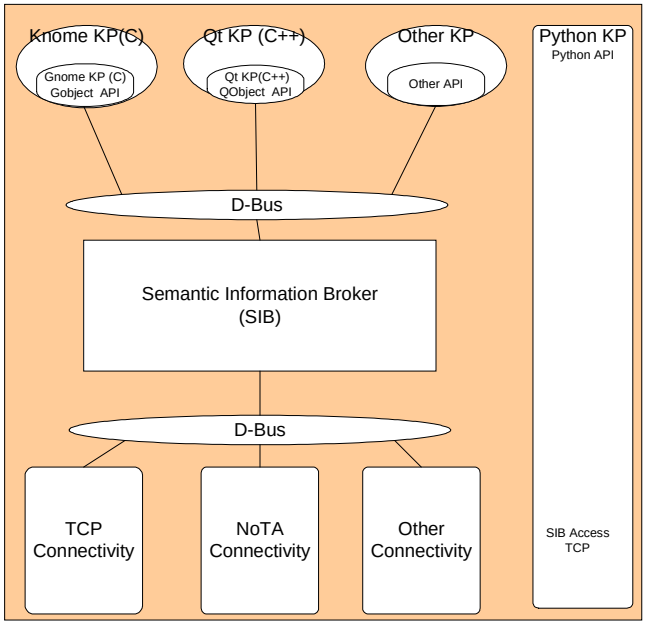



**Figure 32. Smart M3 implemention on Linux platform**

For further reading take a look at Smart-M3 wiki [145] .

## 3.3 Security aspects related to DiY and SE

We can define a smart environment as one that is able to acquire and apply knowledge about its inhabitants and their surroundings in order to adapt to the inhabitants and meet the goals of comfort and efficiency [208] . These goals are normally directed to adapt the environment to the user preferences, to increase the performance of the user in his day-to-day tasks, and to optimize the energy consumption of the systems involved.

From a functional point of view, security is intended to assess the risks present on a given environment and to develop safeguards and countermeasures to protect the environment and its users from those risks [209] . Though there is no universal agreement about security issues and terminology, it is widely accepted [210] that security intends to provide six security services: confidentiality, integrity, availability, authentication, non-repudiation and access control. In this paper we pay more attention in the security threat of the data and control information exchanged between the different agents.

### 3.3.1 Confidentiality.

Confidentiality is the protection of information in the environment from unauthorised access. In smart spaces, the term information acquires a unique perspective [211] . The computer systems involved are potentially capable, as a whole, of sensing nearly every aspect of interactions between users and the environment or among users themselves, and all sensed information has the potential of being stored, transmitted, queried and replayed. In smart offices, some of this information will need to be protected due to its business-related sensitivity, but a great deal of the information sensed by the system will be personal information about users. Therefore, apart from the confidentiality issues usually present in information systems, new confidentiality concerns raise regarding users privacy [212] . Even when the sensible information is protected using cryptographic mechanisms, in smart spaces there is the risk of traffic analysis to obtain some information, for example, when a device is being used. This risk is facilitated by the use of wireless technology.

### 3.3.2 Integrity.

Integrity guarantee that only authorised parties are able to modify information in the environment. As with confidentiality, protection of stored and transmitted information has been traditionally achieved through cryptographic means. Another threat consists on the repetition of a message without modifying it, or in some cases the delay or the alteration of the sequence of a set of messages. In deal with this threats, before applying the integrity mechanism, some uniqueness information must been appended to the message. Integrity is usually applied to the information, but is especially important to applying it to the mobile code in smart spaces, because of the risk to introduce malicious code in a system. Therefore, the same challenges raised by smart spaces regarding integrity and the same techniques proposed to address them are applicable here.

### 3.3.3 Availability.

Another security issue is to ensure all the services are available to authorized parties when needed. Traditional availability requirements of computers and networks remain valid for smart spaces, but there are specific threats to availability in smart spaces directly derived from the technology used. The nature of wireless communications makes them vulnerable to jamming. Furthermore, resource limitation creates new types of denial of service attacks, such as sleep deprivation [213] , where battery limitations are exploited to disable a mobile device by keeping it awake until battery runs out. Of special significance is the availability of the security services, as the failure of these services may compromise the entire system.

### 3.3.4 Non-repudiation

Non-repudiation intends to ensure that a transferred message has been set and received by parties claiming to have sent and received the message. This service guarantee that the sender of a message cannot later deny having sent the message and the recipient cannot deny having received the message. This is a capital service when the content of the message have a contractual implication. This situation is typical in business scenarios as e-commerce. This service is usually not necessary in the communication between the agents that belongs to the smart office. When needed, it is provided by using digital signature.

### 3.3.5 Authentication.

Authentication of users and devices within the smart space can take advantage of existing approaches for computer and network security. Due to the largely decentralized and dynamic nature of smart spaces, key management is the main problem we encounter when dealing with authentication in these environments. Solutions that rely on connectivity to an authentication or revocation server, from Kerberos to public-key certificates, can only be applied to smart spaces where we can assume a hierarchical arrangement of principals, and where addition and removals of principals in the system may be controlled. In [214] such a centralized approach is applied to a smart home. Related to authentication we have to consider the different roles users play in smart spaces. Solutions are proposed to model this situation, being the main one Role-Based access control [215] . Other works have extended this model to take into consideration the environmental information [216] . Delegation is related with acting in behalf of other users after former user permission. Often, traditional security is somewhat static and context insensitive. Smart spaces integrate context and situational information, transforming the environment into a conscious space. Authorization is not an exception. Authorization services make use of available environmental information and take it into account in order to take access control decisions. Security policies should be able to change dynamically to adapt the

permissions granted to the provided context information. [217] . The authentication service is concerned with ensuring that an interaction is authentic, that is, that all parties involved in the interaction are correctly identified, with an assurance that their identities are not false. A failure in authentication can usually lead to violations of confidentiality, integrity or availability, as these services rely on the validity of the identities of the actors involved. Thus authentication is a key issue for security, and so it is for security in smart spaces. Authentication of devices within the smart space can take advantage of existing approaches for computer and network security. Public or private key cryptography can be used to authenticate information interchanges between devices, taking into account the considerations about resource limitations stated above. However, due to the largely decentralised and dynamic nature of smart spaces, key management is the main problem we encounter when dealing with authentication in these environments. Solutions that rely on connectivity to an authentication or revocation server, from Kerberos to public-key certificates, can only be applied to smart spaces where we can assume a hierarchical arrangement of principals, and where addition and removals of principals in the system may be controlled. In such a centralised approach is applied to a smart home, and we will show later its applicability to smart offices as well. In smart spaces where devices communicate through ad-hoc networks and where devices need to be added and removed easily secure transient associations.

Closely related with the authentication issues are the authorization mechanisms. Once the system has identified the user and when some action is requested, the system has to decide whether the user is or nor granted to take. Regarding to authorization issues, it is necessary to deal with trust management systems. These systems have to cope with policy addition and removal and with the automatization of validity checking tasks. The latter takes special relevance when dynamic policy systems are considered. Solutions are proposed to model this situation, being the main one Role-Based access control. Other works have extended this model to take into consideration the environmental information. Delegation is related with acting in behalf of other users after former user permission.

### 3.3.6 Key management

The cryptographic procedures used to offer security services require distributing and sharing keys. Asymmetric cryptography is usually used to obtain scalable security architecture, due to the easiness of the key management. But symmetric cryptography is necessary to obtain a better performance in a smart space with reduced resources in some devices. So, it is an important challenge to decide the better method to protect every message using symmetric cryptography or asymmetric cryptography.

### 3.3.7 General security considerations

The security services have been traditionally achieved using cryptographic means, and the algorithms and techniques available are well-known and mature. However, smart spaces introduce additional challenges to the use of cryptographic solutions. Looking for ubiquity, the tendency is to use mobile devices that are resource limited in terms of bandwidth, computing capability and battery power. Due to these constraints, traditional cryptographic approaches may not be applicable to certain scenarios. To address this problem, [218] proposes a lightweight security solution based on components that can adapt to environments with scare resources. The security solutions based on cryptography provide different degrees of trade-off between resource consumption and security strength. Resource constraints and security requirements must be evaluated for each environment to decide the approach to take. The security solution must be transparent to the user as much as possible; in order to fulfil the focal point of pervasive computing. This means that the user must to operate the minimum possible with the computers systems, to guarantee the security. Finally, the security solution needs to be able to scale when the smart space topology and the number of users increases.

# 4 Toolsets

An approach towards the definition of different/heterogeneous toolsets from the point of view of the user centric design.

## 4.1 Mobile devices

### 4.1.1 Entire platform

#### 4.1.1.1 Apple iPhone

iPhone comprises device, Operating System (OS) and software applications developed for this device by Apple Inc. Last iPhone generation is iPhone 3GS released in June, 2009. iPhone combines a smartphone, communications interfaces and hardware human interfaces in a 4.5 in. x 2.4 in. and 135gr. device. iPhone 3GS CPU is a Samsung 32-bit RISC processor at 600 MHz with permanente storage on flash memory up to 32 Gbytes and 256Mbytes RAM memory.

iPhone hardware human interfaces are display, speakers, microphone and video camera. Display iPhone is a 3.5 inch touchscreen LCD witch multitouch sensing and four sensors: proximity, ambient light, compass and accelerometer. The most interesting sensor is an 3-axis accelerometer witch senses phone orientation, changes orientation screen and can also be used to control third party applications. iPhone includes stereo speakers , microphone and connector for headphones. The iPhone 3GS feature a built in autofocus 3.2 megapixel camera with auto macro. It can also record VGA video at 30 frames per second.

iPhone 3GS supports multiple connection capabilities:

- USB 2.0 (up to 480 Mbps).
- WiFi 802.11b/g(2.4 GHz band up to 54Mbps)
- Bluetooth 2.1+EDR (theoretical data transfer speeds of up to 3 Mbps)
- GSM QuadBand
- GPRS/EDGE
- TriBand UMTS/HSDPA
- 7.2Mbps HSDPA.
- A-GPS.

**SOFTWARE**

The iPhone runs an operating system know as iPhone OS. It is based on Mac OS X, including Core Animation software and Open GL API components wich is responsible for the interface motion Graphics. iPhone supports Apple applications, as well as from third-party developers written and compiled specifically for iPhone OS.

Apple controls software and hardware aspects of the iPhone. All iPhones must be activated (assigned a carrier and a telephone number) before most features become available. From software point of view, is mandatory to use AppStore in order to distribute any kind of software developed for iPhone. However, exists a proccess called "Jailbreaking". This proccess allows users to install apps not available on the AppStore or modify iPhone basic functionality.

iPhone user interface is based on the concept of direct manipulation using multitouch screen and iPhone sensors such as accelerometer. Home screen contains icons and a dock at the bottom of the screen, showing the most accessed applications. Nowadays, multitasking is limited to Apple applications. Nevertheless applications can be designed to work together, allowing to share or cross-propagation data from one to another.

3GS iPhone version includes several default applications such as: Messages, Calendar, Photos, Camera, Maps, Clock, Calculator, Voice, Memos, Notes, iTunes , AppStore, Phone, Mail and Safari Navigator.

AppleStore is default place to download native iPhone applications. These applications are created by Apple or third-party developers. Moreover, iPhone OS supports third-party web applications using Safari web browser. These application must be created using web Technologies such as AJAX.

**iPHONE SOFTWARE DEVELOPMENT ENVIRONMENT.**

A Software Development Kit (SDK) is available to third-party developers and allows to make and to test iPhone applications. These SDK is free only for University environments and needs a Mac OS X computer running the Mac Xcode IDE. For commercial environments SDK price is from 99$ to 299$.

SDK is broken down into the following framework sets:

- Cocoa Touch. Includes multi-touch events and controls, accelerometer, localization an camera support.
- Media. Open AL, Open GL, audio/video/image controls, Core Animation and Quartz
- Core Services. Networking, Core Location, Threads and embebed SQLite BBDD.
- OS X Kernel. Supports TCP/IP, sockets, file system, security and Power Management.

The focus of application development is the Xcode application. Xcode is an integrated development environment (IDE) that provides support for project management, code editing, building executables, source-level debugging, source-code repository Management and performance tuning. Main Xcode Windows, is shown in next Figure and provides quick access to all of the key elements of application.

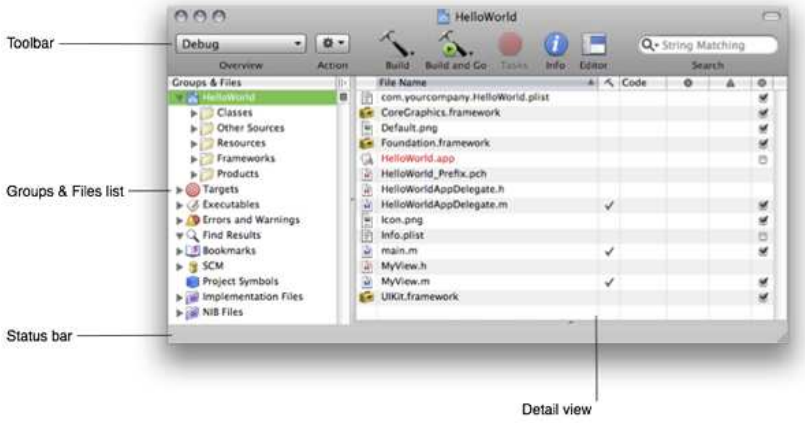



**Figure 33: Interface Builder.**

Interface Builder is the tool used to assemble application's user interface visually. Using Interface Builder, we can assemble application's window by dragging and dropping preconfigured components onto it. The components include standard system controls such as switches, text fields, and buttons. After to place the components on the window's surface, we can configure their attributes using the inspector, and establish the relationships between those objects and code. Interface Builder eliminates the custom code needed to create, configure, and position the objects that make up interface.

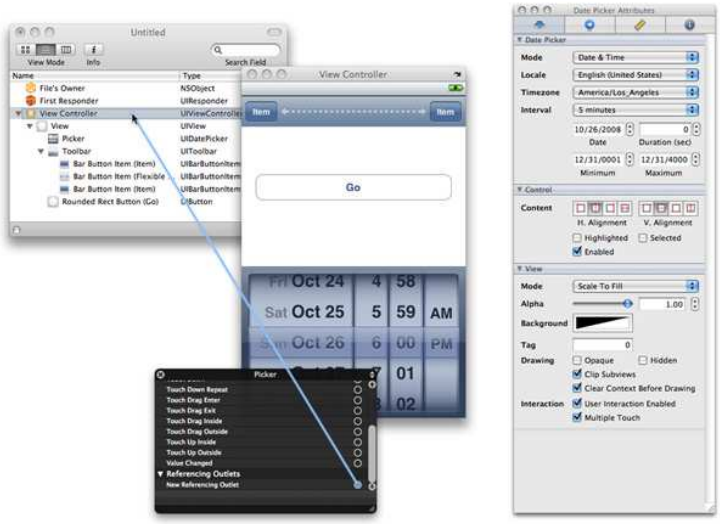



**Figure 34: iPhone application development example.**

#### 4.1.1.2 Archos Tablets

Archos has been pioneer in the field of portable connected tablets. Archos range comprises two product lines, one dedicated to true Wintel platforms including the ARCHOS 9 remarked at CES 2010 and the other product line is dedicated to ARM processors based platforms. Archos is recognized worldwide for offering the most opened machines that are able to play most of digital multimedia standards and based on open Operating Systems like Linux and Android. Archos is about to launch new products featuring both Android and Linux (Angstrom distribution) operating systems. These new "dual OS" ultra-open platforms will offer incredible opportunities for customizing high range multimedia portable products.

**ARCHOS 9: A PC DESIGNED FOR PORTABILITY**

Not so long ago, Netbooks were handy machines, but as a matter of fact, they were mostly scaled-down laptops.
The ARCHOS 9 PCtablet is an interesting evolution because it changes the rules of form and design.


**Figure 35: Archos 9**

**A NEW REFERENCE IN THE FIELD OF PC AT CES 2010**

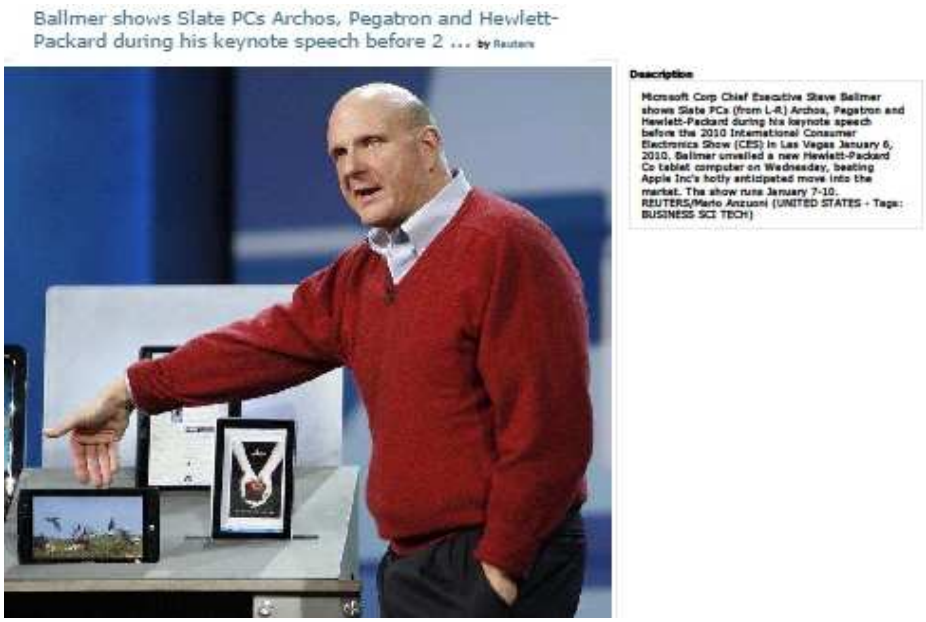



**Figure 36: Archos at CES 2010**

Software: Nothing special as it is a true standard PC running on Windows 7 (or any other compatible OS like Windows XP or Linux)

## ARCHOS 5 INTERNET TABLET: A DEVICE THAT FITS IN A POCKET


**Figure 37: Archos 5**

**Tech Specs**

- High resolution touch screen, 800x480 pixels, 4.8'' TFT LCD, 16 million colors User interface Touch screen, ON/OFF, vol+ and vol- buttons, retractable virtual keyboard Processor(s)
- Central Unit:
  - Main processor: ARM Cortex™-A8, 32 bit, In-order, dual-issue, superscalar core, 800 MHz
  - Additional processor: 32 bit DSP, 430 MHz
- RAM memory 256 MB* (Low-Power Double Data Rate SDRAM) Mass storage memory • Flash Memory: 8 to 32 GB* + Micro SD Slot (SDHC compatible)
- Hard Drive: 160 to 500 GB*
- Operating system Android “Donut” - enabling application installation or development
- Program language C - requires special developer-edition firmware that embed a C compiler
- Connectivity On board WiFi (802.11 b/g/n), USB 2.0 host with optional accessories PC accessories Possible attachment of numerous standard USB PC accessories (keyboard, mouse, memory key, memory cards reader, camera and other future computer accessories)
- Others applications Webbrowser, Email, Contacts, DroidIn™, Ebuddy™IM, Twidroid™, Deezer™, Dailymotion™, QuickPedia, ThinkFree Mobile, Moov, CraigsPhone, Pages jaunes™ (only for France), High Paying Jobs (only for US & Canada), Alarm clock, Calculator...

**Interfaces**

- B 2.0: Media Transport Protocol (MTP)
- USB 2.0 Host: Mass Storage Class (MSC) and Picture Transfer Protocol (PTP) with optional Mini Dock, Battery Dock and DVR station
- Micro SD (SDHC compatible) on ARCHOS 5 Internet Tablet 8 to 32 GB
- Add-on connectors to connect to the DVR Station and other ARCHOS add-ons

**Communication**

- WiFi (802.11 b/g/n)
- Bluetooth 2.0 A2DP, EDR, ARCP, HID, Dial Networking
- FM transmitter (requires the Car-mount that features the antenna)
- FM receiver (RDS)
- High speed connection to cellular network through tethering in Bluetooth with most of 3G phones (e.g. Nokia)

**Video Codecs**

- MPEG-4 HD (up to 720p)
- MPEG-48 (ASP@L5 AVI, up to DVD resolution)
- H.264 HD (up to 720p)
- WMV (MP@ML, up to DVD resolution) including WMV protected files
- MKV (up to 720p, 2500kbs, 23fps0
- M-JPEG (Motion JPEG Video) in QVGA resolution
- With optional plug-in (downloadable from your tablet or on www.archos.com):• Cinema: MPEG-2, WMV HD (720p), VOB

**Audio codecs**

- Stereo MP3 decoding @ 30-320 Kbits/s CBR & VBR,
- WMA, Protected WMA, WAV (PCM/ADPCM), AAC (except protected content),
- AAC+ stereo audio files
- Ogg Vorbis up to 320kbs @44hz
- Flac up to 1000kbs @44hz.
- With optional software plug-ins (downloadable from your tablet or on www.archos.com): AC3 stereo audio and 5.1 sound files (via SPDIF output of DVR Station)

**Photo viewer**

JPEG, BMP, PNG, GIF Subtitles Support subtitles files with .srt, .ssa, .smi, .sub extensions

**Video recording**

- Via the optional DVR Station or DVR Snap-on. Records NTSC/PAL/SECAM in MPEG-4 AVI format with stereo sound, VGA resolution (640 x 480) @ 30 or 25 f/s

- Via the optional TV Snap-on (only available for Europe): Records TV in MPEG-2 TS format. Audio recording
- Via the optional DVR Station or DVR Snap-on: Stereo line-in, WAV (IMA ADPCM or PCM) format
- Voice recording with the built-in microphone Miscellaneous
- Built-in speaker"• Built-in leg stand• Built-in Microphone Power source
- Internal: Lithium Polymer battery
- Device charges via computer USB port
- External: Power adapter/charger (via optional add-on)

**Battery life**
- Music playback time: up to 22 hours
- Video playback time: up to 7 hours

**Scalability**
- Device automatically downloads latest firmware updates when the WiFi connection is activated
- • Updates can also be downloaded at www.archos.com

**Dimensions & weight**
- • ARCHOS 5 Internet Tablet Flash series: 143,2x78,8x10,4 mm - 182g
- • ARCHOS 5 Internet Tablet Hard Drive series: 143,2x78,8x20 mm - 286g

**Linux software development environment (needs the special developer edition firmware):**
(More information on Angström distribution site: http://www.angstrom-distribution.org/ )

**Android software development environment:**
(More information on Lars Vogel site: http://www.vogella.de/articles/Android/article.html )

### 4.1.2 OS platform

#### 4.1.2.1 Symbian S60

Symbian is an operating system designed for Mobile devices and smartphones developed by Symbian Ltd. In 2008, Symbian Ltd. was acquired by Nokia. This operating system includes libraries, an user interface, and a development environment with common tools for creating open source applications.
The Manufacturers now include this operating system are: Ericsson, Fujitsu, Mitsubishi, Motorola, Nokia, Samsung, Sendo, Sharp, Siemens and Sony Ericcson. Nowadays, it is the world's most popular mobile operating system, accounting for 50% of smartphone sales.

**CURRENT FEATURES:**
- o Phone and Messaging.
- o Applications.
- o Internet.
- o Camera and photos.
- o Music and radio.
- o GPS and maps.
- o Banking and Security.
- o Video and TV.

Is expected in the near future to have:
- o Location alerts.
- o Touch connection.
- o Parallel calendars.

The current Symbian open version (Symbian^2) includes: Web browser, email support, productivity and personal information management applications, Web runtime for creation of Web applications, an EKA2 real-time microkernel, symmetric multiprocessing support, SQLite integration, full multi-tasking and multi-threading, a rich middleware including location, multimedia and seamless multibearer communications framework, LBS & GPS support, remote management support, advanced audio/video capture and playback, home screen support for embedded widgets, flexible support for different form factors and input methods, and an improved location event framework.

**APPLICATIONS DOWNLOAD:**
Applications can be downloaded from Symbian Horizont. These applications can be developed in:

- Java ME.
- Flash Lite.
- Web Technologies.
- Python.
- Ruby.
- .NET.
- Symbian C++.

Nowadays exists more than 140 package applications divided into:

- Location
- OS base Services.
- Multimedia
- Productivity
- Security
- Personal Communications
- User Interface
- Data Communications
- Device Connectivity
- Device Management
- Runtimes
- Tools

Each language needs a different development tool and not all of these tools are open licensed.

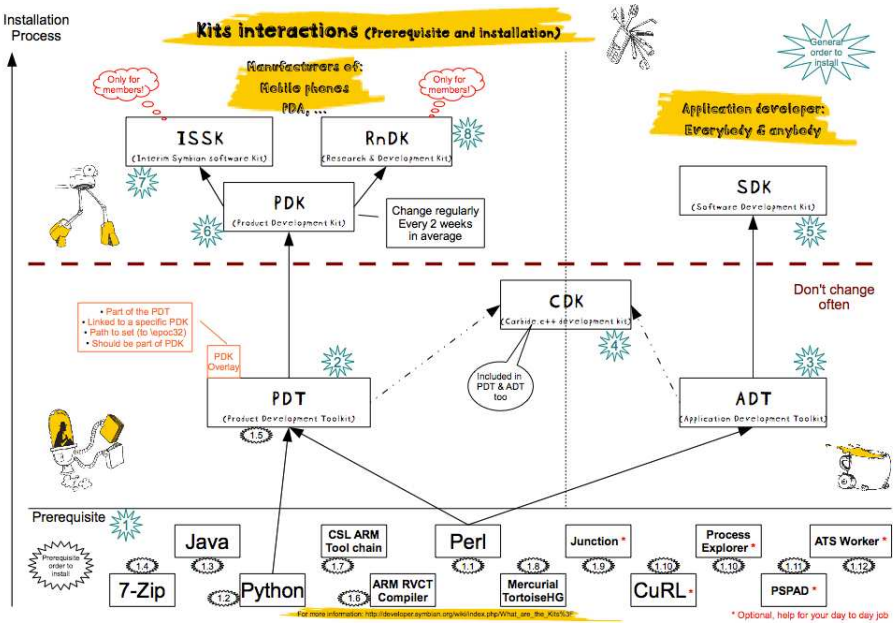



**Figure 38: SYMBIAN modules schema.**

**C++ developers**
For C++ developers is necessary is necessary to have ADT and SDK packages.

*ADT (Application Development Toolkit) 1.0*
ADT is used with the SDK and contains all the tools necessary for application development. The Carbide.c++ IDE is the principle tool used for C++ development and is built on the Eclipse tool framework. Carbide plug-ins includes build, debug, static analysis, dynamic analysis, and a variety of other specialized tools and utilities. Carbide.c++ supports Symbian Build System v1 and v2 (Raptor). Carbide.c++ allows the developer to edit, build, and debug application on the emulator and on phone

targets. The ADT also includes Carbide.ui Theme Edition. Carbide.ui allows the developer to create themes for devices and provides access to the more than 1,000 customizable theme elements in the UI.

**Application Development SDK- (from Nokia)**
The Application Development SDK available is the Edition SDK from Nokia This will change with future Application Development SDKs, but until the Symbian Foundation creates its own SDK, the S60 5th Edition SDK from Nokia is being used as the SDK for Symbian^1. This SDK enables application development in Symbian C++, Open C/C++, Java and Web technologies. The SDK includes the all key resources needed for application development (documentation, API reference, examples, and an emulator), and is used with the Application Development Toolkit (ADT) for IDE support.

*For Java developers:*
Symbian Foundation does not have the full Java package for the time being because of licensing issues, only platform APIs and stub implementations for them are provided. The full implementation is available directly from Nokia for those S60 licensees that have relevant licenses from the appropriate 3rd party Java technology vendors.

*Flash Lite:*
Adobe supplies the industry-standard Flash Professional CS4 authoring tool for use in creating Flash/Flash Lite content, allowing designers to prepare anything from simple animations through to complex multi-component ActionScript-driven applications.

*WEB TECHNOLOGIES.*
Web Runtime consists of several components that allow installation and execution of Widgets. Widgets are essentially applications written in HTML/JavaScript/CSS, with JavaScript based access to some device APIs. Widgets execute inside the web browser and are very similar to web/AJAX applications we see on the web.

Widget development is supported by an growing set of plugins for major development environments such as Eclipse-based Aptana Studio, Microsoft Visual Studio and Adobe Dreamweaver.

*PYTHON*
Python applications ("scripts") are simply text files containing code written in the Python programming language. The scripts can be written in any text editor and can be run either from within the Python Interactive Shell application on a device or the emulator, or as standalone applications on mobile device. The development environment consists of the Python reference documentation, the Application packager for making your scripts into stand-alone applications and the phone SIS files. Both are provided for Windows and Linux/Mac.

*RUBY*
Ruby is a dynamic programming language with a complex but expressive grammar and a core class library with a rich and powerful API. Symbian Ruby brings the Ruby programming language to the Symbian platform under the terms of the Ruby GPL License. It is a port of the official Ruby code base and should run most of the existing Ruby scripts. Symbian Ruby source code is hosted at the official Ruby central repository, so it is always in synch with the latest Ruby Core releases.

*.NET*
The .NET Compact Framework is a subset of the full .NET platform designed by Microsoft for mobile applications on Windows Mobile"Red Five Labs" provides Net60, the .NET Compact Framework on Symbian. Microsoft originally designed and developed the .NET platform with support for multiple languages and operating systems. Red Five Labs implemented the Common Language Infrastructure (commonly referred to as .NET) on the Symbian platform with support for C++, Java and .Net.

#### 4.1.2.2 Maemo

MAEMO is an open source development platform for smartphones and Internet Tables. This platform comprises Maemo Operating System and Maemo SDK and is based on Debian Linux Distribution. Maemo is a registered trademark of Nokia Corporation.
Internet Tablet devices are a set of mobile Internet appliances, made by Nokia, focused on Internet and media features. These devices are between a Personal Digital Assistant (PDA) and a UltraMobile

PC (UMPC). The Maemo devices on market include the Nokia N800, N810 and N900. The latest version of Maemo is Maemo 5 and is only available for Nokia N900. These Nokia devices support the same web technologies on PCs: HTML, JavaScript and CSS.

**Figure 39: Nokia N900 including Maemo 5**

The Maemo 5 operation system comes preloaded with applications such as:

- Web browser (with Adobe Flash).
- Phone Application.
- VoIP: SIP and Skype.
- Conversations: IM chat and SMS.
- Media: Camera, Photos and Media Player.
- Production : Email, Calendar, PDF reader, Contact.
- Ovi Maps (Find position on a map using the GPS, Search an address or location and plan routes).
- Utilities : Clock, Notes, Calculator, Sketch.
- System Tools : File manager, Application manager for downloads, Widgets
- Games.

Maemo has its own integrated development environment (IDE) and also provides support for Eclipse IDE and Qt framework. Currently C is the only official programming language for Maemo, but thanks to the community, Python scripting language and C++ are available also for Eclipse IDE.

Maemo has its own website (at maemo.org), which includes:

- Links to mailing lists (and their archives).
- Link to the defect tracking system (bugzilla) at bugs.maemo.org.
- White papers and tutorials.
- Trademark usage guidelines.
- Development news.
- Download information.
- API references library versions used in maemo SDK.
- The maemo wiki and other community supported resources.

Next image shows Maemo 5 software architecture. Only components in red colour are proprietary:

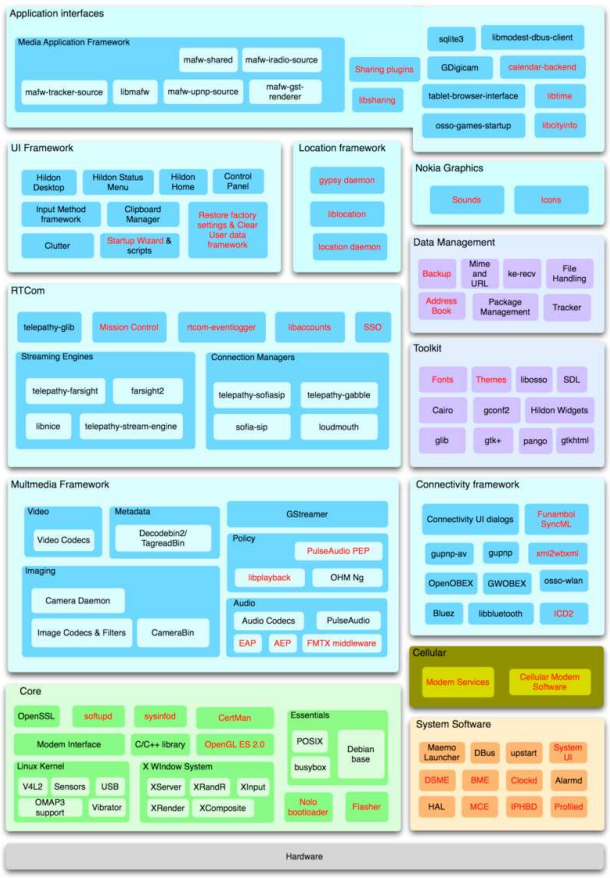



**Figure 40: Maemo Architecture.**

- Core Domain contains bootloader and Calibration Area, Linux kernel, X windowing system (X11), OpenGL ES, and (some) essential user space components.
- 
- System Software domain is responsible for user modes, device states including start-up/shutdown, display and battery management.
- Connectivity domain provides the APIs, user interfaces and kernel components for Wireless and Bluetooth.
- RTCom domain provides both services, and applications, for realtime Internet communications (Voice Over IP calls and Instant Messaging) on Maemo platform.
- Multimedia Framework domain contains the multimedia APIs and is also responsible for policy, audio-routing, and event-sounds.
- Location domain implements a framework that applications can use to become location aware. Location implements also an API that provides a set of location related utilities like geocoding, reverse geocoding, show on map, select from map, get map data and POI (Point Of Interest) repository. It also includes everything related to GPS and SUPL.
- Toolkit domain contains the graphical UI libraries to draw content on the device's screen and used by the applications to create user graphical interfaces.
- Desktop domain contains Window Manager, Status Menu and their plugins, Control Panel and its applets, and application services such as Startup.
- Data Management domain consists of application and user data related subsystems. Applications domain provides end user applications like Browser, Email, Calendar, Games and public interfaces for 3rd party applications.
- Graphics domain provides sounds, icons and pre-installed content for the device.

These components make it possible to develop applications mixing following components:

- graphical user interface: windows, dialogs, menus, toolbars and data selection.
- Application Development: Desktop Widgets (small utilities or extension of an existing application) and control panel applets.
- D-Bus: is a service daemon that allows related processes to pass events to each other in a consistent mannet without having to define a custom IPC (Interprocess Communications) protocol. The D-Bus daemon also passes important events from the core system to applications.
- Multimedia Components: include video and audio support programming, APIs for Browsing Multimedia Content, video and audio codecs, dedicated DSP (Digital Signal Processor) and libraries for making 2D vector Graphics.
- Connectivity components: Scanning of available WLAN networks for Internet connectivity, VoIP, Instant Messaging and Presence, a high level API for Bluetooth.
- Generic Platform Components.
- Data Sharing.
- Packaging, Deploying and Distributing: Adding software to Maemo is done through "packages". Packages are discreet archives of applications and additional information that make installing software easier than compiling the software from scratch. The package management system makes use of package repositories, which are essentially web or FTP sites that contain packages and make them available for download.
- Kernel and debugging.

#### 4.1.2.3 Google Android

Android is an open source and customizable mobile platform. Android offers a software stack for Mobile devices that includes an operating sytem based on Linux kernel, middleware, a set of APIs that allows to develop applications and key applications developed by Open Handset Alliance, integrated by Google, HTC, Motorola, Intel, Nvidia and other companies.
The major manufacturers with phones with Android pre-installed are Samsung , Motorola, HTC Corporation and Dell.
Android provides a SDK with tools and APIs necessary to begin developing applications on Android platform using Java language. Android SDK is available for Windows, Mac OS X and Linux i386.

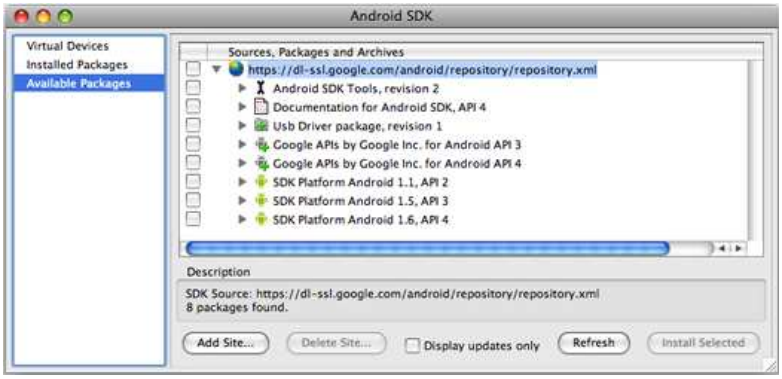



**Figure 41: Android SDK**

Android SDK is supported in following development environments: Eclipse IDE, JDK 5 or 6 and Apache Ant. The recommended way to develop an Android application is to use Eclipse with the ADT plugin. This plugin provides editing, building, debugging functionalities, including a device emulator.

The main features of toolsets provided by Android SDK are the following:

- Handset layouts: The platform is adaptable to larger, VGA, 2D graphics library, 3D graphics library based on OpenGL ES 1.0 specifications, and traditional Smartphone layouts.
- Storage: The Database Software SQLite is used for data storage purposes

- Connectivity: Android supports connectivity technologies including GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, and Wi-Fi.
- Messaging: SMS and MMS are available forms of messaging including threaded text messaging.
- Web browser: The web browser available in Android is based on the open-source WebKit application framework.
- Media support: Android supports the following audio/video/still media formats: H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP.
- Additional hardware support: Android can utilize video/still cameras, touchscreens, GPS, accelerometers, magnetometers, accelerated 2D bitblits (with hardware orientation, scaling, pixel format conversion) and accelerated 3D graphics.
- Multi-touch: Android has native support for multi-touch which is available in newer handsets such as the HTC Hero. The feature was initially disabled at the kernel level (possibly to avoid infringing Apple patents on touch-screen technology).
- Maps: in order to add mapping capabilities, the Google APIs add-on includes a Maps external library. The classes of the Maps library offer built-in downloading, rendering, and caching of Maps tiles, as well as a variety of display options and controls.

Like many phone-based application stores, the Android Market is a catalog of applications that can be downloaded and installed to target hardware over-the-air, without the use of a PC. Android Market includes over 10,000 freeware and paid-for applications.

### 4.1.3 Sensor devices

Several sensor platforms are available for solutions architects and developers in order to design original, adaptive and application aware wireless sensor networks (WSNs). The hardware of such platforms consists generally of different sensor boards (see Figure 39) combined together and with a wireless communication interface to gather specific information about the surrounding environment of the sensors (house, office, factory, field, etc.) in the perspective of providing feedback services about this location. The leading corporations in the development of wireless sensor networks toolkits provide different packages whether for academic or manufacturing purpose.

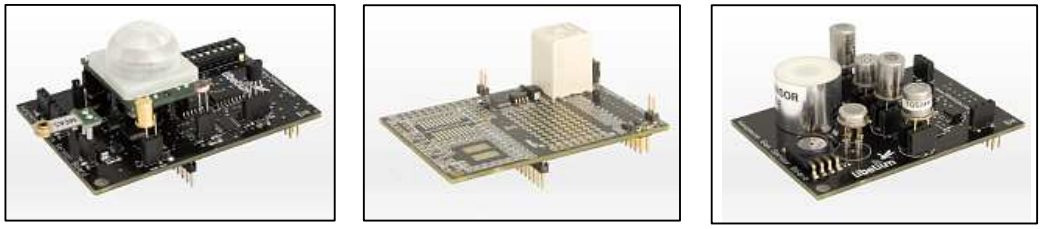


**Figure 42: sensor boards**

For example, Crossbow Technology (www.xbow.com), Sun (www.sunspotworld.com) and Libelium (www.libelium.com) have done a pioneer work in the design of both hardware and software for WSNs. They propose several development kits for academics, professionals and OEMs based on various hardware (iMotes, MicaZ, SunSPOT, WaspMotes, etc.) supplied with open or proprietary software platforms (TinyOS, Java VM, .NET micro framework). These toolkits can be integrated DIY environment since they are compliant with standardized PHY, MAC or NWK protocols (802.15.4, 802.11, ZigBee, etc.)

## 4.2 Hardware platform

### 4.2.1 Arduino

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Arduino is intended for create interactive objects and environments, such as
The Arduino hardware consists of a preassembled or built by hand board. This board can sense the environment by receiving input from a variety of sensors and can affect its surrounding by controlling different actuators and communicating with software running on a computer.

Several versions of Arduino boards can be found here: http://arduino.cc/en/Main/Boards
The main classifications are made attending to PC communication interface. This interface can be Serial (RS232), USB, Bluetooth, Ethernet, XBee or ArCan.

Next image shows an USB Arduino board example.

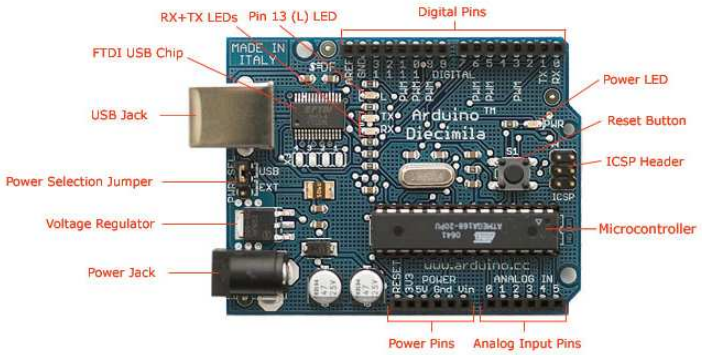



**Figure 43: Arduino board example.**

The Arduino software consists of a development environment (IDE) and core libraries distributed under GNU General Public License and Creative Commons License. The IDE is written in Java and core libraries are written in C and C++. Arduino boards can be programmed using Arduino language or straight C/C+. Arduino language is a set of C and C++ functions that can be called from code.
Arduino community provides simple programs about basic Arduino behaviour. http://arduino.cc/en/Tutorial/HomePage. These basic examples controls:
Digital and Analog I/O: turn on and off LEDs, read pushbuttons, play speakers, MIDI interfaces, analog sensors...

- Sensors: accelerometers, knock detection devices, ultrasonic object detection.
- Display control: matrix of LEDs and Liquid Cristal Displays.
- EEPROM control: clear, read and stores values in an EEPROM.
- Motors: Stepper Motor control and servos.
- Communications between Arduino board and PC: send data to the computer in order to process them and receive orders through keyboard, mouse or process running on computer.

Next image shows Arduino development environment with functions basic library.

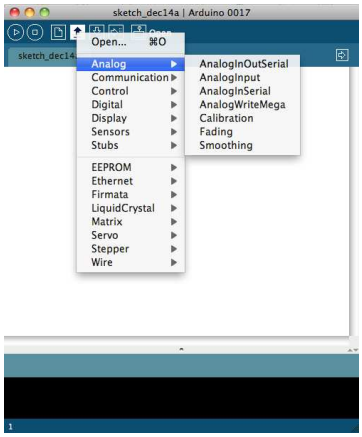



**Figure 44: Arduino development environment.**

### 4.2.2 BeagleBoard

The BeagleBoard is a single-board computer designed around the TI OMAP3530 (Open Multimedia Application Platform) processor, the highest end processor of the OMAP35xx product line. The OMAP3530 processor is a powerful hardware platform for mobile multimedia applications.
The BeagleBoard was designed by a small team of TI's engineers under the firms "small dreams" program. The design goal was to make it as simple and cheap as possible, and simultaneously

keeping the ability to connect a wide array of add-ons using cheap external components. This was coupled with an open source dissemination strategy, where technical and design data of the hardware and software were released to the public for free. This follows the basic idea of making it as simple, cheap and easy as possible for small developers and hobbyists to use and extend the capabilities of the board. The BeagleBoard has been very successful in this respect, as can be attested by the large number of copies and derivatives from it that exist today

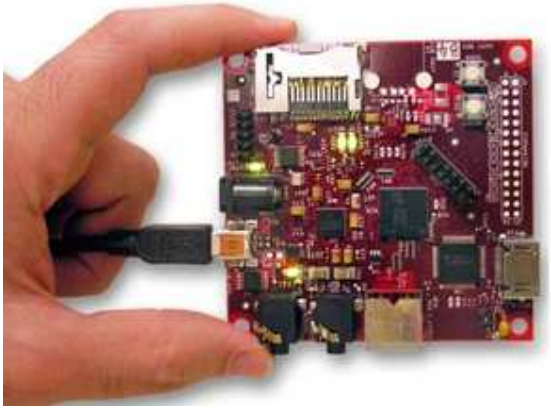


**Figure 45 - The BeagleBoard**

The BeagleBoard is based on a PoP (package-on-package) sandwich combining TI's OMAP3530 processor running at 720 MHz with 256MB of 166MHz DDR RAM and 256MB of NAND Flash (rev. C4 board). No more memory can be added to the board, expanded storage is obtainable via a SD/MMC+ slot onboard.

**Figure 46 - Package-on-package, with processor, flash and RAM**

The board can drive a LCD panel equipped with a DVI-D digital input. It is also equipped with a DVI-D interface that uses an HDMI connector, allowing output to a standard LCD monitor (with appropriate cables). There is an additional S-Video output which can carry different video output data from what is found on the DVI-D output if the software is configured to do it. It will support NTSC or PAL format output to a standard TV.

Onboard I/O interfaces include:

- USB OTG (high-speed On-The-Go)
- USB 2.0 host port
- 9-pin RS-232 header
- Audio I/O with 1/8-inch stereo mini-jacks
- 14-pin JTAG
- Power LED, three software-controlled LEDs
- Reset and user-programmable buttons

The board allows further expansion through a 28-pin connector near the board's edge. Carrying I2C, I2S, SPI, and MMC/SD signals, it can be used with top- and bottom-stacked daughtercards, as well as with offset daughtercards and ribbon cable connectors.

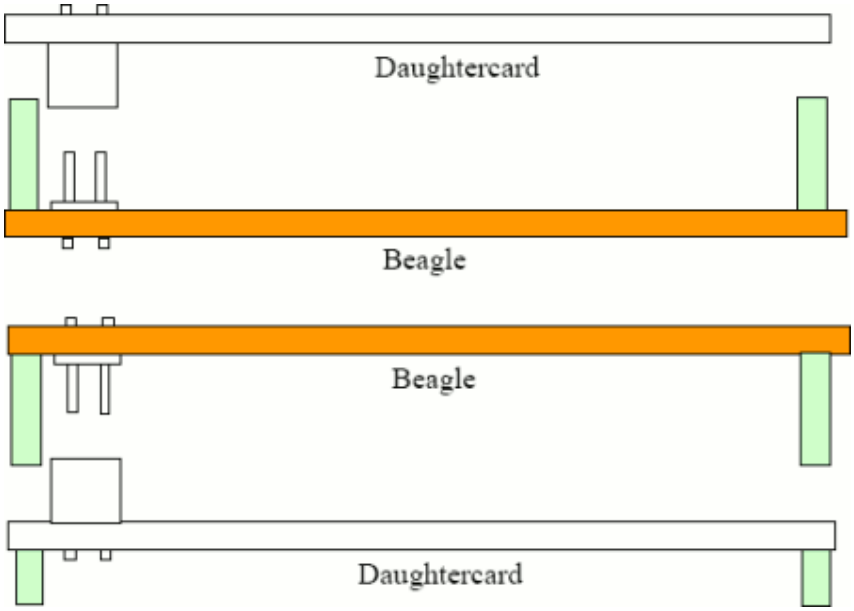



**Figure 47 - BeagleBoard stacking**
**Top- and bottom-stacked daughterboards.**
**Offset boards and ribbon cables are also possible**

The BeagleBoard measures approximately 76 by 76 mm. The board uses up to 2W of power and it can be powered over USB by a development host, and also features a 2.1mm input for an optional 5V, 500mA external power supply.

The BeagleBoard usually runs Linux, most commonly the Ångström distribution. There is a large array of open source development tools, and an active, distributed development community, with most relevant information hosted on a Wiki system. Several robotic, multimedia and "hardware mashup" projects are based on the BeagleBoard.

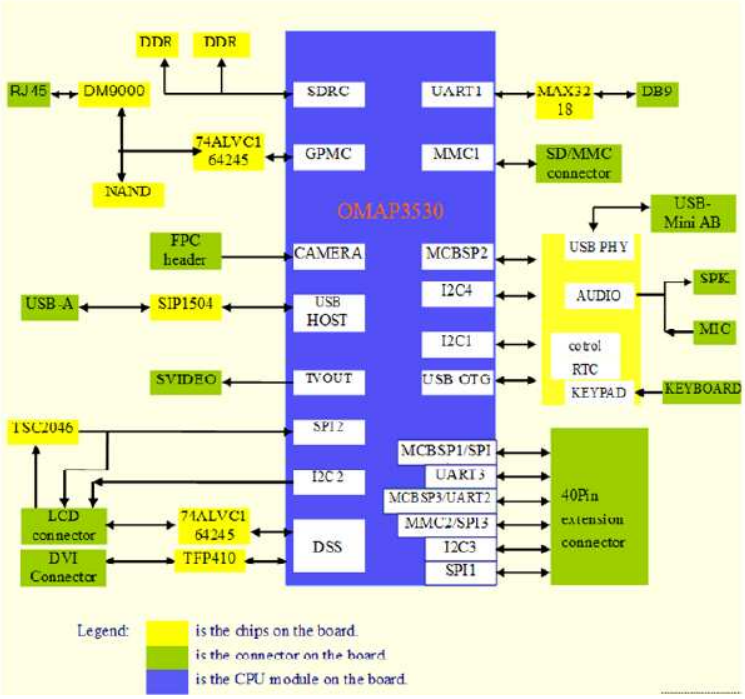



**Figure 48: Architecture diagram of the Beagle board**

### 4.2.3 Fox Board

The FOX Board G20 is a Linux embedded single board computer. The heart of this single-board computer is the Netus G20, a 40×40 mm CPU module that integrates an Atmel ARM9 , 64MByte of

SRAM and 8MByte dataflash memory. The FOX Board G20 provides in a compact footprint the physical and electrical interfacing needed to exploit the features of the Netus G20 CPU module.

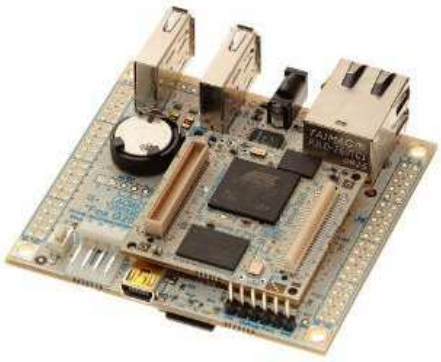


**Figure 49: Fox board G20**

The FOX Board G20 also supports the NetusPS1 power supply module. This module is necessary because the AT91SAM9G20 CPU used on Netus G20 module requires a well-defined power-up and power-down sequence to guarantee reliable operations. The AT91SAM9G20 (Atmel Smart ARM-based Microcontroller) CPU is based on the ARM926EJ-S processor (ARMv5), with a clock speed of 400MHz (achieving about 1.1 DMIPS/MHz). It's nearly a full computer on a chip, with built-in controllers for SDRAM and static memories (including NAND Flash and CompactFlash) and also containing support for a wide range of peripherals, including USB Full Speed Host and Device interfaces, a 10/100 Base T Ethernet MAC, Image Sensor Interface, Multimedia Card Interface (MCI), Synchronous Serial Controllers (SSC), USARTs, Master/Slave Serial Peripheral Interfaces (SPI), two three-channel 16-bit Timer Counters (TC), a Two Wire Interface (TWI) and four-channel 10-bit ADC.
The AT91SAM9G20 is targeted at power-constrained applications that also need high performance and it supports both Linux and Windows CE. One of the most important advantages of the AT91SAM9G20 is its low power dissipation. It consumes only 80 mW in full-power mode with all peripherals turned on.
The FOX Board G20 runs Linux, and is delivered with the Gentoo distribution. It can also run the OpenEmbedded or the Ångström distribution. Some add-on boards are available from the manufacturer.

### 4.2.4 SiMiUS

The SiMiUS system has been developed by the department of Electronic and Microelectronics of University of Mons (UMONS - Micro) in Belgium (formerly known as "Faculté Polytechnique de Mons - FPMs"). SiMiUS is a single-board computer based on a PIC microcontroller (16F876 family).
This board follows a similar approach as Arduino, with the goals of being evolutionary and educational, since it is used by students for practical projects while following a course on microprocessors and embedded systems. It is easy to program, with several supported languages (C, Basic, Pascal, assembly), and can be configured with a large number of peripherals via its SPI port.
A broad range of sensors are available for the SiMiUS board. For example, a distance sensor, a temperature sensor and an accelerometer can be connected to the I2C port. A GPS or a Bluetooth module can be connected to the SPI port. It is planned to add even more features to the SiMiUS board, such as a MMC/SD socket and a Zigbee module.

The SiMiUS board (version 1.0) currently offers:

- une double alimentation (5V et 3,3V)
- a PIC microcontroller (16F876 family)
- a RS232 interface
- a LCD display
- an I2C and a SPI bus


**Figure 50: SiMiUS board**

In addition to the SiMiUS, UMons has also developed another board, Mini SiMiUS. This board has a smaller footprint than the SiMiUS board, and a reduced power consumption. It is targeted to embedded applications, with the capability to connect to up to 7 sensors/actuators. The Mini SiMiUS board contains also an I2C port, where it is possible to connect an LCD display, a keyboard, etc.

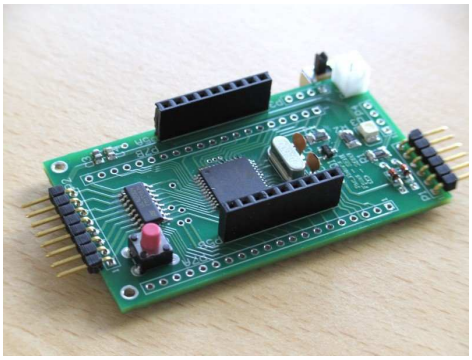


**Figure 51: Mini SiMiUS board**

### 4.2.5 Bug Labs BUG 2.0 platform

Bug Labs offers the BUG system, a fully integrated device development platform, oriented towards open source hardware and software components. It is built around a hardware platform called BUGbase and a modular collection of snap together components called BUGmodules These modular electronic blocks allow the user to assemble a new mobile device quickly and without the complexity of hardware design. The intended application is to allow developers to rapidly prototype new mobile devices. The BUG system is akin to a mashup framework for both hardware and software.

The BUGbase is the heart of BUG applications. It's a powerful palm-sized computer with storage, WiFi, Bluetooth, rechargeable battery, mini LCD, USB and serial I/O, and ports for BUGmodules. It can be connected to any combination of up to four BUGmodules, allowing it to interact with the physical world. For example, the BUGvonHippel is an I/O module giving access to virtually any input or output, be it a servo motor, heart rate monitor, card reader, etc. The BUG system is entirely built of open sourced hardware and software, running Linux and programmed in Java.

**Figure 52: BUG 2.0**

The second generation of the BUGBase product is based on the OMAP3 platform from Texas Instruments (TI) and is software compatible with the BeagleBoard , giving it access to all applications written for that device. BUG 2.0 also adds support for Android, allowing developers to port current Android applications to the BUG as well as create new Android applications that use BUG's modules to create new Android-based hardware applications.

### 4.2.6 DIY Robots: Tux Droid, Lego Mindstroms

**TUX DROID**

Tux Droid is a smart companion or consumer device whose main function is to easily access computer services and Internet without having direct computer contact.

Tux Droid is a product of Kysoh which is currently the main private contributor of this project. It settles the project with a working open source hardware, firmware and software basis.

Tux Droid is a Linux and Windows (XP and Vista) wireless Tux mascot, driven by Atmel AVR RISC microcontrollers, with a programmable interface, allowing it to announce events by its gestures and sounds. The events are detected by specific gadgets, which are handled by the Tux Gadget Manager. The Tux Droid supports Linux kernel 2.4 or later and needs a 800 MHz CPU and 128 MB RAM. For media detection it needs an Internet connection.

Tux stablishes a 2.4 GHz wireless full-duplex digital link with the USB dongle. The dongle is used to communicate between Tux and the computer.

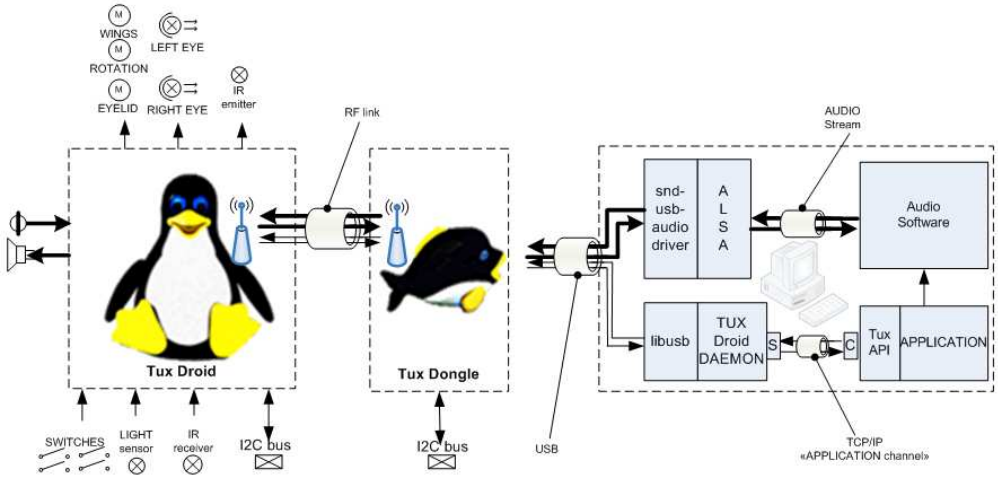



**Figure 53: Tux Droid Architecture.**

Detailed features of Tux Droid are: push button on the top of the head, moving eyes, LED's in the eyes, microphone in the beak, switch on each wing, flapping wings, spin left and right, sound controller, plug audio IN/OUT and speaker. User can controls Tux droid behaviour from PC or the included remote control.

Tux Droid gadgets are mini-applications residing on the computer desktop and deliver a variety of personalized information such as e-mail, weather forecast, news, podcasts, etc... Most gadgets will need to be configured first to make any use of them: to determine start and stop event, to display help for the Gadget, etc. In order to add gadget animations, Tux Droid incorporates Attitunes. Attitunes are robot animations combining movements, sounds, Text-To-Speech, lights and fun.

In order to create smart companion features for this robot exists an Open Source Project called Tux Droid Suite Development. Contributions developed with this software can be about various topics: core software uit, APIs in various programming languages (C++, Pascal, Java, JavaScript, Phyton and Ruby), gadgets or packages build for Linux.

A public SVN repository has been setup at http://svn.tuxisalive.com/ and contains all Tux software: from developed tools and libraries to gadgets and APIs.
From the PC side Tux Droid is a USB device and can be accessed through USB drivers. Using these drivers and Tux software layers is possible to build a smart-companion robot. The bottom-up description of its four layers follows:

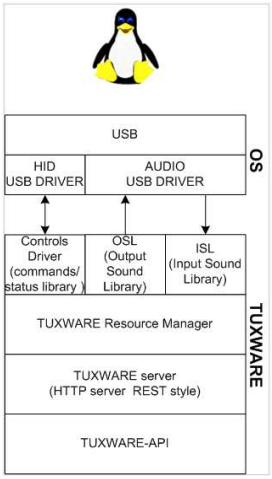



**Figure 54: Tux Droid Software Layer.**

A software package for Tux droid can be developed combining several of following elements:

- Tux Droid can be considered as a Server and supports http requests: http://wiki.tuxisalive.com/index.php/How_to_http_requests
- using Javacript API and HTLM for controlling robot: http://wiki.tuxisalive.com/index.php/Javascript_API_and_HTML_pages
- Tux Droid Gadgets, compiling Java applications with Maven software project management, for example: http://wiki.tuxisalive.com/index.php/Compiling_Java_applications_with_maven
- Gadgets can be also developed at different software programming languages such as perl, java or python. These gadgets are managed by a Framework which acts as an editor and gadget container: http://artisan.karma-lab.net/node/1567
- Using Google gadgets.

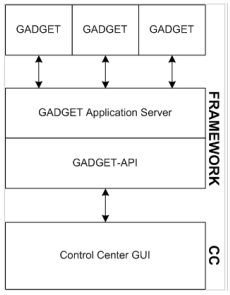



**Figure 55: Tux Dorid Framework Architecture.**

**LEGO MINDSTORM.**

LEGO MINDSTORMS NXT is the second generation of robotics products from The LEGO Group. Lego robots are available in a basic kit or in separate components.


**Figure 56: Lego Mins Storm central unit.**

Sensors ican be light, touch, ultrasonic, compass, gyroscope, infrared , RF ID or sound sensors. Also, Servo Motor and Bluetooth communications modules are available.

Lego provides, LEGO MINDSTORMS NXT software, in order to program NXT robots and download programs to the NXT via USB or Bluetooth connectivity. LEGO MindStorm NT Software is powered by LabView from National Instruments. LabView users can also program MINDSTORMS NXT robots in LabVIEW. System Requirements are a PC with Windows (XP) or MAC with Mac OS 10.5, with USB port and 512Mb RAM minimum (Bluetooth port is optional).

Lego Mindstorm Software is an intuitive, icon-based drag-and-drop programming language designed for an easy introduction to programming for all users. By choosing program blocks that work with the motors and make the sensors react to inputs, is simply to build up a program block by block, and create programs that range from simple to complex.

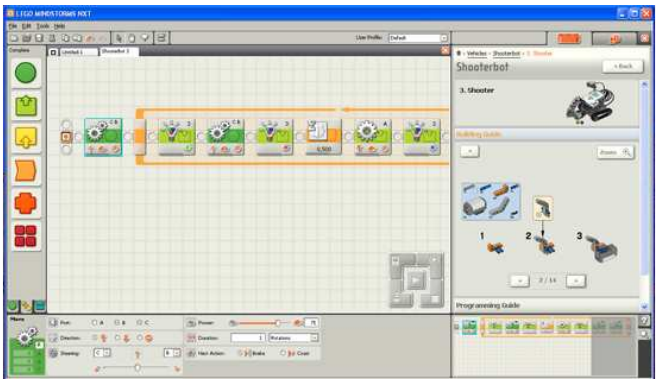



**Figure 57: Lego MindStorm Software.**

The LEGO MINDSTORMS NXT toolkit comes with programming examples and a user-friendly walk-through introduction to the different programming blocks.

Users can drag and drop blocks from the left side of the screen on to the diagram. Each block performs a unique function such as moving the motors, displaying a message, detecting a sound, or measuring a distance. By combining a series of blocks we can program a complete robot.


**Figure 58: Lego MindStorm: programming robot example .**

The compiled program is sent to the NXT using the wireless Bluetooth link or the USB cable. The NXT then executes the code that it receives from computer.

## 4.3 Software platform

### 4.3.1 Mashup

It is interesting to see that Web Mashups can be realized at three levels:

- Data And Service Mashup,
- Functional Mashup,
- Visual Mashup.

The result of the Mashup can be a composition of one (Horizontal Mashup) or more levels (Vertical Mashup).

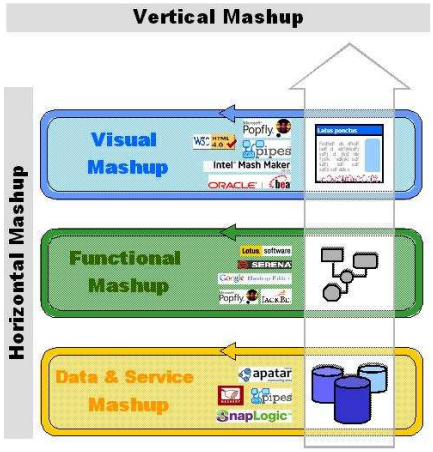



**Figure 59: horizontal and vertical Mashup**

| | Data | Function | Visual | Enterprise | Mass Market |
|---|---|---|---|---|---|
| **Yahoo! Pipes** | + | | + | - | + |
| **Microsoft Popfly** | + | + | + | | + |
| **Intel Mash Maker** | + | + | + | | + |
| **Lotus Mashup** | + | + | + | + | - |
| **Google Mashup Editor** | + | + | + | | + |
| **JackBe** | + | + | + | + | |
| **Apatar** | + | + | | + | |
| **Serena** | + | + | + | + | - |
| **Mashery** | + | | | + | |

**Table 4: Mashups Type Table**

#### 4.3.1.1 WSMX Engine

**Introduction**

A Web service is a computational entity which is able to achieve a users goal by invocation. A service, in contrast, is the actual value provided by this invocation (Bellwood, 2002; Preist, 2004).

In contrast to Services, Web Service due tackle implementation details being linked to concrete specifications and protocols: WSDL (interface described in a machine-processable format, how can a system interact with Web Service) , SOAP (using SOAP messages), conveyed using HTTP with an XML
serialization in conjunction with other Web-related standards.

Web Services are currently the preferred standards based way to realize SOA [6]. Other alternatives such as REST [194] are available for describing and implementing services.

UDDI, WSDL, and SOAP are important steps in the direction of a web populated by services. However, they only address part of the overall stack that needs to achieve scalable web service

discovery, selection, mediation and composition [188] . Semantic technologies are necessary for efficient dealing with large numbers of available services.

The research effort of WSMO working group is focussed in Semantic Web Services, organized in three aims: providing a standard for describing semantic web services (**WSMO**) [189] , defined by a formal language (**WSML**) [190] , and reference implementation (**WSMX**) [191] :

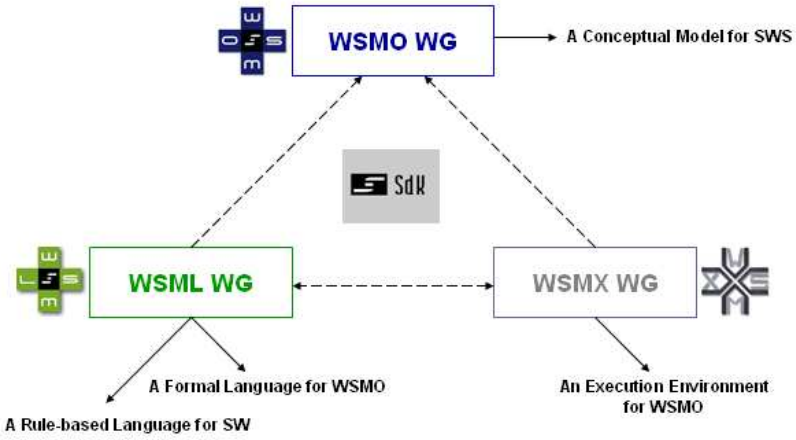



**Figure 60: WSMO working group**

**WSMO (Web Service Modelling Ontology)**
The main drawback of traditional Web Services is their lack of proper support for machine processable semantics. This lack makes necessary human intervention to actually discover, combine, and execute Services. The goal is to minimize any human intervention, so the integration of business logics can be done in a task-driven way and with the least support from the user side. [192]

So, why we need semantic in Web Services? Semantically technologies to automation of the Web Service usage process (Publication, Discovery, Selection, Composition, Mediation, and Execution). And, why we need Ontology in Semantic Web Services? Data model to allow machine data interpretation.

According to the www.w3c.org WSMO is defined like “The potential to achieve dynamic, scalable and cost-effective infrastructure for electronic transactions in business and public administration has driven recent research efforts towards so-called Semantic Web services, that is enriching Web services with machine-processable semantics. Supporting this goal, the Web Service Modeling Ontology (WSMO) provides a conceptual framework and a formal language for semantically describing all relevant aspects of Web services in order to facilitate the automation of discovering, combining and invoking electronic services over the Web”.

WSMO provides means to describe Web services that provide access (searching, buying, etc.) to services. WSMO is designed as a means to describe the former and not to replace the functionality of the latter [188] .

WSMO has composed of the following top elements:

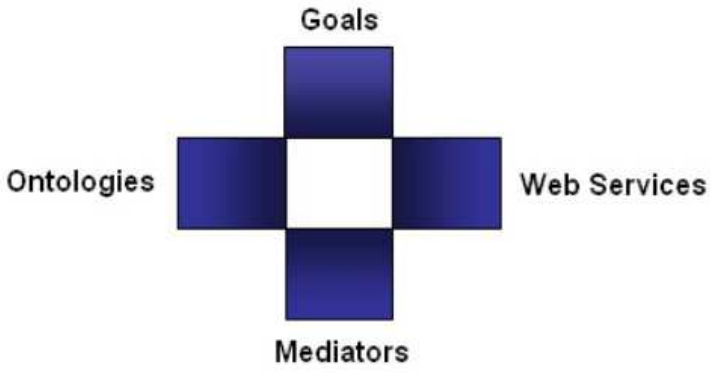



**Figure 61: WSMO elements**

- *Goals* - The client's objectives when consulting a Web Service.
- *Ontologies* - A formal Semantic description of the information used by all other components.
- *Mediators* - Connectors between components with mediation facilities. Provides interoperability between different ontologies.
- *WebServices* - Semantic description of Web Services. May include functional (Capability) and usage (Interface) descriptions.

Three layers of WSMO:

- WSMO Lite (restricted version of WSMO Standard, which might disappear)
- WSMO Standard (which is the main version)
- WSMO Full (which contains extensions according to business needs)

**WSML (Web Service Modelling Language)**

WSML is a language for the specification of ontologies and different aspects of Web services. In this respect WSML provides a syntax and semantics for the WSMO. WSML uses well-known logical formalisms in order to enable the description of various aspects related to Semantic Web Services.

WSML has some variants based on different levels of logical expressiveness and the use of different languages a paradigm:

- WSML-Core, corresponds with the intersection of Description Logic and Horn Logic (without function symbols and without equality), extended with datatype. It has the least expressive power of all the languages of the WSML family and therefore has the most preferable computational characteristics. WSML-Core provides support for datatypes and datatype predicates
- WSML-DL, extends WSML-Core to an expressive Description Logic, namely, *SHIQ*, thereby covering that part of OWL which is efficiently implementable.
- WSML-Flight, extends WSML-Core in the direction of Logic Programming. Modeling primitives, and rule language extended with inequality and (locally) stratified negation
- WSML-Rule, extends WSML-Flight to a complete Logic Programming language, including function symbols. WSML-Rule no longer restricts the use of variables in logical expressions.
- WSML-Full, unifies the Description Logic (WSML-DL), and Logic Programming paradigms (WSML-Rule) under a paradigm First-Order with non-monotonic extensions

**WSMX (Web Service Modelling eXecution environment)**

WSMX [9] is the reference implementation of WSMO. It is an execution environment for business application integration where enhanced web services are integrated for various business applications. The aim is to increase business processes automation in a very flexible manner while providing scalable integration solutions.

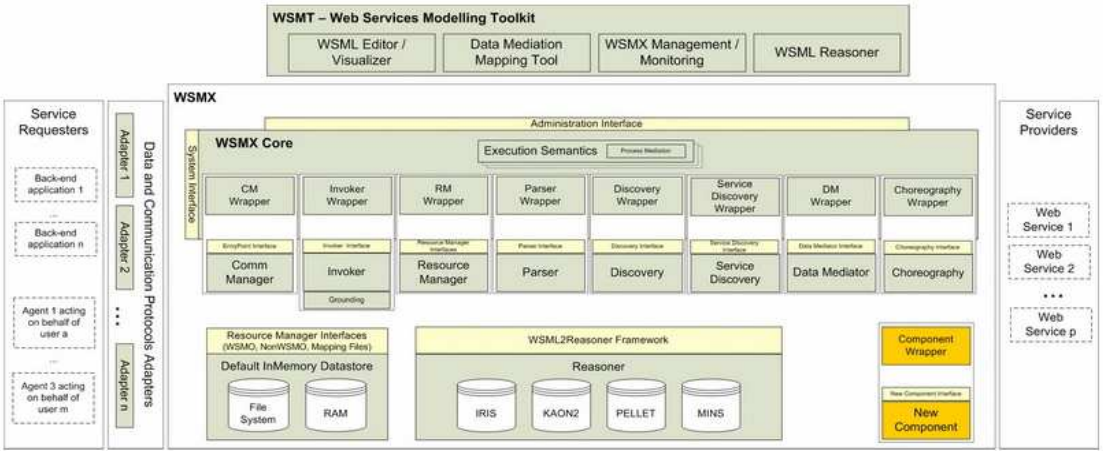



**Figure 62: WSMX implementation**

WSMX is an execution environment that enables discovery, selection, mediation, invocation and interoperation of SWSs. The development process for WSMX includes establishing a conceptual model, defining its execution semantics, developing the architecture of the system, designing the software and building a working implementation of the system. The research results for WSMX provide guidelines and justification for a general SWS architecture.

The following main components have already been designed and implemented in WSMX:

- **Adapters** aim to solve data representation mismatches on the communication layer.Transforms the format of a received message into WSML compliant format with mapping rules.
- **Parser** to validates, compiles and storage persistently description file into internal memory model.
- **Invoker**, WSMX offers invocation of selected Semantic Web services while obeying their interfaces. WSMO service descriptions are grounded to WSDL. An adapter has developed to enable any underlying protocol of service providers (e.g. SOAP or proprietary protocols).
- **Choreography**. Choreography takes up how the user interacts with the services, to make use of its functionality, in other words “the published communication pattern of a web services”. The role of the Choreography Engine is to make sure that the requestor's communication pattern matches the provider's communication pattern through the next step: Prepares the available data, The Process Mediator filters, changed or even replaced data, and Data to be finally sent to the communication partner.
- **Process Mediator** provides the means for runtime analyses of two choreography instances and uses mediators to compensate possible mismatches
- **Discovery**. Discovery is the action for finding a Web Services to achieve the desiderate goal. Discovery in Semantic Web Service is extend discovery component is based on simple matching and play significant part semantic discovery in prototypical stage
- **Mediation**. It’s the most important component in the sense of there is not any component related with each other without a mediator in-between. The reason why they might not be able to communicate directly is because they use a different syntax or semantics of data (in this case data mediation being needed), or that their communication patterns mismatch. WSMX offers distinct implementations for data and process mediation.

The next figure provides a WSMO Mediator Structure:


**Figure 63: WSMO Mediator Structure**

Data mediation is based on paradigms of ontology, ontology mapping, and ontology management. Data from different sources are mediated based on their semantic similarities as expressed by their reconciled conceptualizations.

Process mediation is the adjustment of the different patterns in order to make them match. The process mediator provides the functionality for a runtime analysis of two given patterns, and compensates for the possible mismatches that may appear, by, for instance, generating dummy acknowledgement messages, grouping several messages into a single one, changing their order or even removing some of the messages in order to facilitate the communication between the two parties.

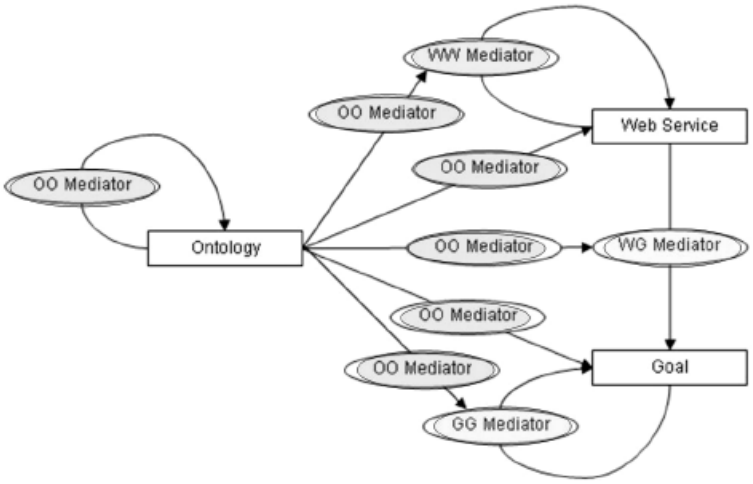



**Figure 64: Process Mediation**

According to the entities they connect, mediators can be classified as (above figure) [197] :

- *OO Mediators (*data mediation)*:* importing ontologies with heterogeneity resolution
- *GG Mediator (*data mediation)*:* goal definition by reusing an already existing goal allows definition of Goal Ontologies
- WG Mediator *(*data mediation, process mediation for communication): connects web services and goals, which means that the web service/s is/are used to achieve the goal
- WW Mediator *(*data mediation, process mediation for communication and for coordination): connect two web services.
- Resource Manager, The Resource Manager stores internal memory model to a data store of any WSMO entity (Web services, goals, ontologies and mediators) and non-WSMO related objects in WSMX. Depending on the scope of data stored in the underlying registries, WSMX distinguishes registries to be local or global. Data stored in a local registry is relevant for the

operation of WSMX in a particular context. In some cases individual functional components may also require local storage. On the other hand, a global registry can be shared across several domains (e.g. registries of SWS descriptions).

Currently WSMX provides five repositories: Web Service Repository, Goal Repository, Ontology Repository, Mediator Repository and Data Repository.

Reasoning, although reasoner is not part of development process, we have considered giving a brief description because is part a important component of the whole WSMX framework. WSML-compliant reasoner provide reasoning services for the mapping process of mediation, as well as functionality relating to validation of a possible composition of services, or determination if a composed services in a process is executable in a given context. Also, this component will be used for finding capabilities that exactly match the requester's goal, as well as capabilities subsuming this goal.

#### 4.3.1.2 Lotus Mashups

Under the code name QEDWiki, IBM provided a Mashup creator [5] for the professional world, unlike Yahoo or Microsoft, which target the mass market (even if the user must have a powered profile for creating a Mashup). QEDWiki has been replaced by IBM Lotus Mashups, which provides a lightweight mashup environment for assembling personal, enterprise, and Web content into simple, flexible, and dynamic applications.


**Figure 65: Lotus Mashup**

#### 4.3.1.3 Yahoo! Pipes

Yahoo defines Pipes [6] as a composition tool to aggregate, manipulate, and mashup content from around the web.
It enables to:

- combine many feeds into one, then sort, filter and translate it.
- geocode feeds and browse the items on an interactive map.
- power widgets/badges on your web site.
- grab the output of any Pipes as RSS, JSON, KML, and other formats.

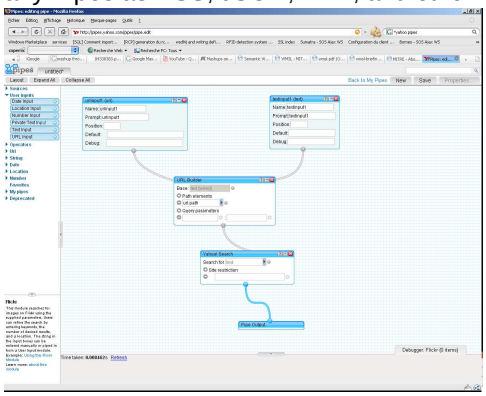


**Figure 66: Yahoo pipes screen shot**

### 4.3.1.4 Microsoft Popfly

Popfly [21] offers a component based, visual environment for developing mashups. It presents an easy and interactive way to build and share mashups, gadgets, games, Web pages and applications. The user interface contains a list of blocks like Display (Bar Graph, Moving slideshow, carousel etc), Maps (Geonames, Virtual Earth, Yahoo Traffic), News and RSS (Digital Podcast, MSN News Feeds, Yahoo answers), Social Networks (Twitter, Facebook, Technorati), Tools (Filter, Merge, Sort, RegExp )etc. Popfly requires no download of any sort, in order to use it, but the installation of a plug-in, Microsoft Silverlight may be required. Microsoft Silverlight is a cross browser, cross-platform plug-in, designed to deliver the next generation of .NET based media experiences, and rich internet applications for the Web.

The following figures illustrate a mashup in which geotagged photos from Flickr, for a given input string are mapped onto the Virtual Earth application. We have chosen to get photos of 'mountains' that are in Flickr; their number can also be specified. Then these photos are mapped by Virtual Map and the result is shown in Fig 66. Photos present in Flickr directory are already geotagged; that is they provide information about the coordinates (latitude, longitude) to place on the map. However, when we use other applications, we might need to put an additional block, *Geonames*, linking that application to Virtual Earth.

Popfly presents an interactive user environment for the creation of mashups, games and web pages and gadgets, and has a collection of more than 50,000 mashups in its library. Unfortunately, it was recently announced that Popfly shutting down forever, towards the end of August 2009, due to some inherent issues.

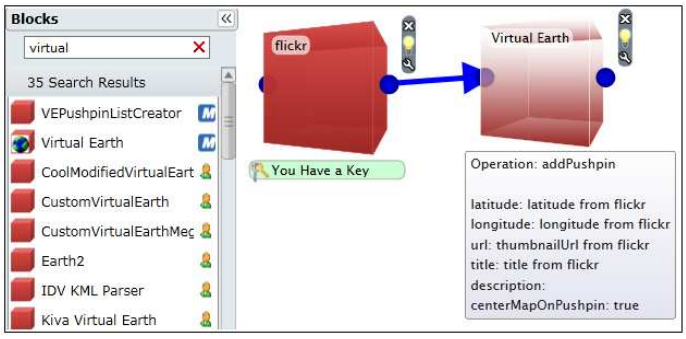



**Figure 67: A mashup for searching geotagged photos from Flickr and mapping them in Virtual Earth**


**Figure 68: The output of the above mashup**

### 4.3.1.5 Intel MashMaker and MARGMASH

Intel [7] [22] [23] provides a browser extension for Firefox and Internet Explorer. The web users can dynamically modify web pages in the browser and mash them with data from different online sources or other Mashups.

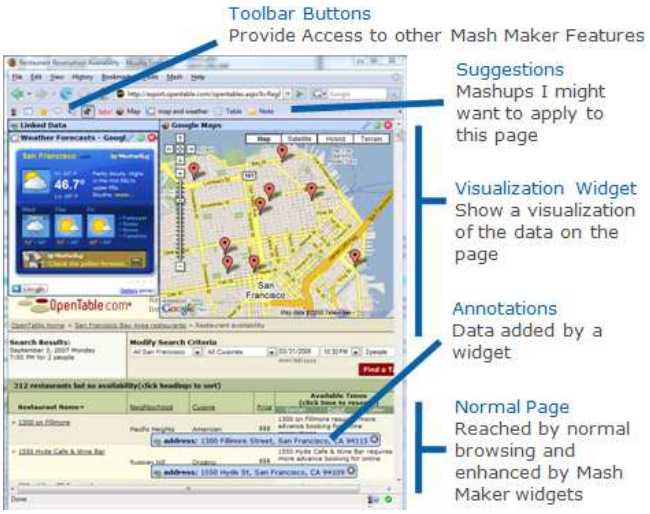



**Figure 69: Intel MashMaker**

MashMaker and MARGMASH are two other mechanisms which enable end users to create their own mashup from existing web sites. MashMaker is a Firefox plugin which enables the end user to create his own mashup from existing web sites. The most important innovation here is data extraction from web pages which contain unstructured data. Figure 68displays a "Yellowpages" web page in which mashmaker component extracts automatically all addresses, phones and names. Thereafter, if the user wants to display these addresses in a Map, he has just to load a Map service such as (Yahoo Map or Google Map).

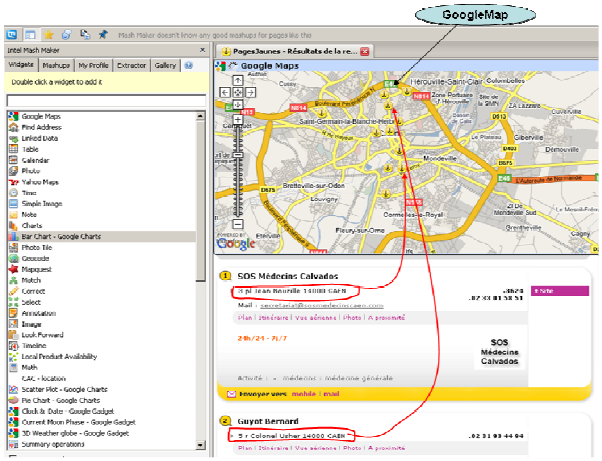



**Figure 70: MashMaker example**

### 4.3.1.6 EZWEB

EZWEB [25] is a widget container web application. In this framework each resource (service or data) is identified with an URI and has an internal representation (XML) and possibly a graphical interface representation (the widget). A widget may be a composition of many resources which is usually defined by a developer. The particularity of this framework is that it enables the end-user to chain widgets between each others by mapping compatible inputs and outputs. This enables the end-user to add functionalities of a widget to another. Following figure illustrates the widget container and the widget chaining tool of the framework.

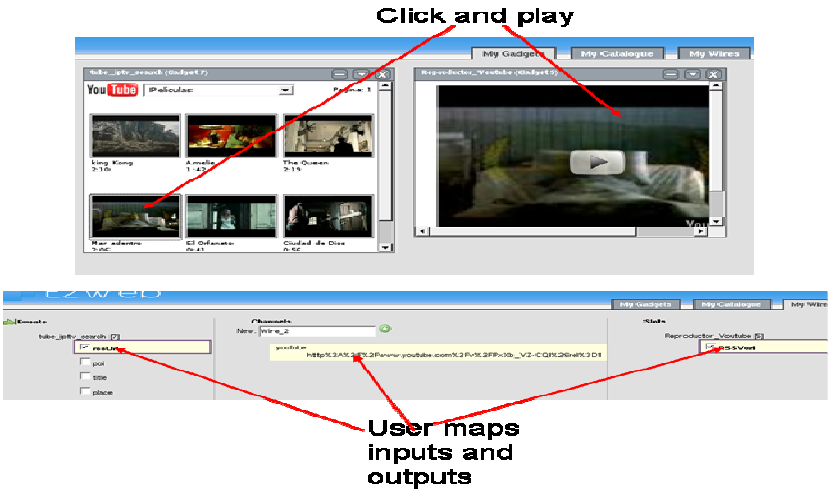



**Figure 71: EzWeb end-user environment**

### 4.3.1.7 Apple Automator

Automator is a desktop application designed to create workflows that perform repetitive tasks. It is a proprietary software program bundled in Mac OS X (starting from 10.4). It can be considered a sort of higher-level scripts.

Those workflows can be created by any user. The process is visually driven: the user connects individual actions that the computer will do and then just run the workflow. About 250 actions are included by default, and the user might download or program new ones. As these pieces have turned out to be really handy, some developers sell Automator actions (for example, Automated Workflows [46] ) and people without programming skills can create workflows with capabilities that were not available with the actions that came preinstalled.

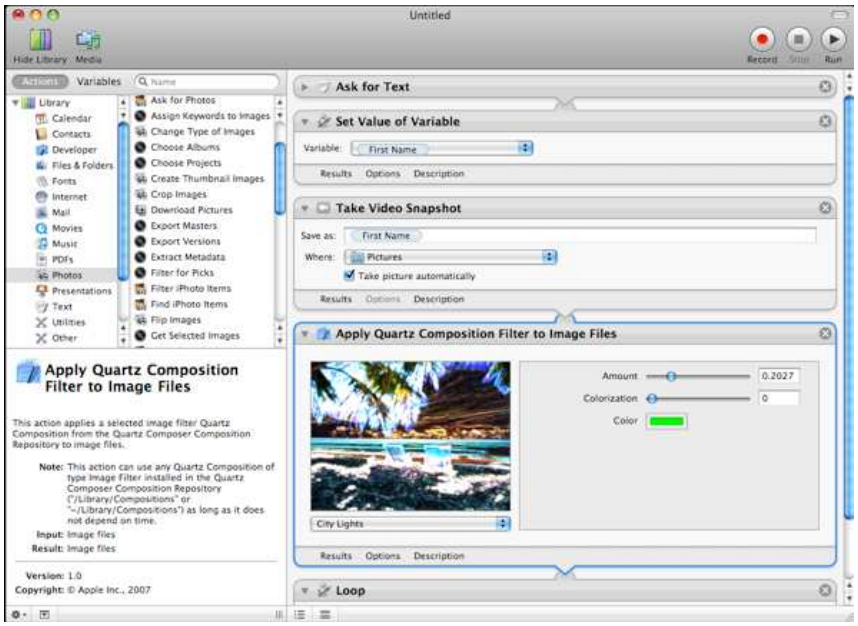



**Figure 72: Screenshot of Apple Automator. Courtesy of automator.us.**

The main features of Automator are extensibility and connectivity. Key applications allow moving data from one to another, and application developers can add support for their applications [47] , further increasing the actions that can be performed.

But besides other desktop applications, workflows can interact with web services (using SOAP or XML-RPC) [48] [49] . For example, a user can create a workflow that every evening checks the weather for the following day and sends a message suggesting the user to use a raincoat, shorts or something appropriate for the expected weather.

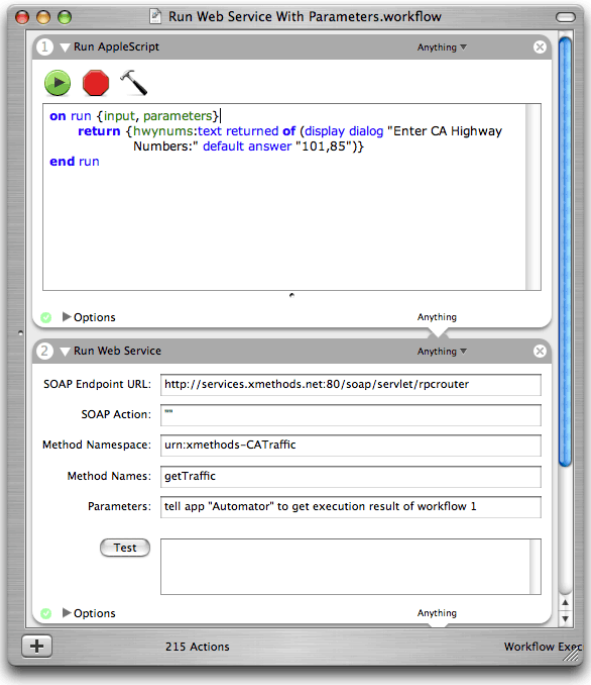



**Figure 73: Automator workflow that calls a web service. Courtesy of Jonathan Rentzsch.**

#### 4.3.1.8 HyperCard

HyperCard was designed as a software construction kit. It integrates a software development and run-time environment that resembles Rapid Application Development packages like Visual Basic. The concept behind HyperCard is the *stack* [50] . Each stack is composed of cards, which in turn hold several HyperCard objects (graphics, buttons…). Buttons are used to go from a card to another, just like when you press a button in a web site to go from a page to another. In fact, HyperCard was one of the first implementations of hypermedia (limited by the inability to link to stacks on the network). As the author of HyperCard, Bill Atkinson, expressed back in 2002, had HyperCard hard networking capabilities; it would have been the first web browser [51] .

HyperCard can be used to create a contact list, a calendar, slideshows and other things you may come up with. And objects from someone else's stack can be copied and reused just as people do with plain text.

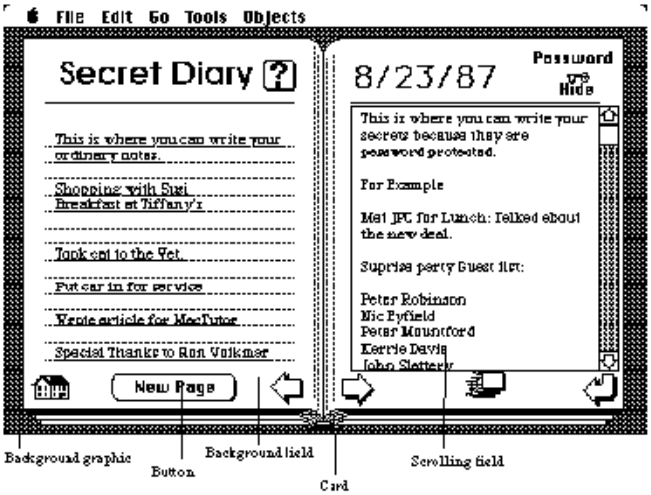



**Figure 74: A HyperCard stack. Courtesy of MacTech.**

The objects contain a script, in HyperTalk language (an extremely simple language similar to Pascal). Functionality is extended creating new objects with scripts that accomplish the desired behaviour,

based on an event-driven model, and external code can be referenced from the scripts (such as a C or Pascal function).

#### 4.3.1.9 OpenKapow

Openkapow [82] is an open service platform, where both experimented developers and non skilled users can build their own services (called robots) and run them from openkapow.com. These robots accesses web sites and allows to use data, functionality and even the user interface of other web sites in an graphical way.

A robot in openkapow is a small program that automates what a person can do in a browser. This includes navigating web sites by clicking on links and submitting forms, extracting data from a site and much more. Robots are created in the development environment RoboMaker without any programming and robots are then hosted and run on openkapow's servers. The behaviour of a robot can be affected by input values (for example the username and password to use to log in to a password protected site) and the robot produces an output (for example the current rate of a specific stock).

An openkapow robot is an XML file that has been defined using the RoboMaker development environment. This XML file is then uploaded to the openkapow servers and can there be referenced with a unique URL. When this URL is called the XML file is interpreted by the openkapow Mashup Server that then performs the actions of the robot

OpenKapow Mashup Server toolkit provides a graphical development interface for different types of mashups. There are available different tool editions which have their own functionalities supporting specific needs:

- Data Collection Edition: This edition is oriented to the creation of enterprise mahups. It provides access to different data sources (structured or not) both in Web format or in file format. This edition provides a tool for developers to combine data form different sources in different formats, and especially for mashups that have a very light representation layer or even for those which do not have any.
- Web 2.0 Edition: This edition optimise the easy and fast mashup creation from components form any Web making possible the reuse of both internal data and public Webs. Once the applications are created, they can be published in way of RSS/Atom "feeds" or REST services.

Some of the Openkapow toolkit advantages are:

- Graphical development environment with an easy to use, intuitive user interface (see figure 2).
- Non-skilled user oriented. It is not necessary to have programming knowledge. Mashups can be created selecting "elements" form other applications, in a totally graphical environment.
- A free complete version of the toolkit is available with the functionality to develop and deploy REST services, RSSs and WebClips.

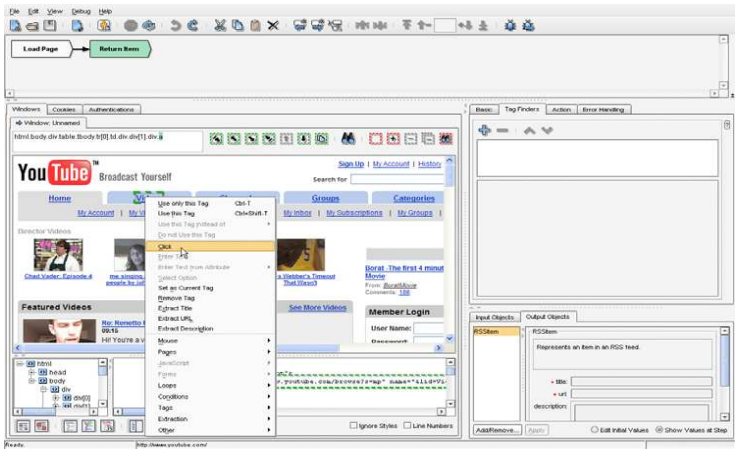



**Figure 75. Openkapow graphical development environment**

Concluding, OpenKapow is a very complete toolkit, with an adapted graphical interface that allows both experimented developers and non skilled users to use it and create mashups with more or less complexity, integrating information from different sources.

### 4.3.2 Video editing tools

#### 4.3.2.1 PC-based Video editing tools

The following tools allow to compose yourself your video by cutting, modifying and composing new videos.

| Video editing and composer tools | Descriptions |
|---|---|
| Cinelerra | **Cinelerra** [36] is an advanced non-linear video editor and compositor for Linux. It has many features for uncompressed content, high resolution processing, and compositing. |
| Kdenlive | **Kdenlive** [37] is a modern filmakers need to mix different kinds of media, including video, audio and images. |
| Blender | **Blender** [38] is the free open source 3D content creation suite, available for all major operating systems under the GNU General Public License. |
| Pitivi | **PiTiVi** [39] is an open source video editor, written in Python and based on GStreamer and GTK+. |
| Avidemux | **Avidemux** [40] is a free open-source program designed for multi-purpose video editing and processing |
| Jahshaka | **Jahshaka** [41] is an opensource, free, video editing, effects, and compositing suite. |
| Kino | With **Keno** [42] , you can load multiple video clips, cut and paste portions of video/audio, and save it to an edit decision list (SMIL XML format). |

#### 4.3.2.2 Web 2.0 video editing tools

By uploading media files, the following tools allow composing, online, new videos using shots from other videos, transition images, audio files, …

| Online video editing and composer tools | Descriptions |
|---|---|
| Cuts | **Cuts** [43] provide online interface upload your own video, or grab one from YouTube for example, add some pre-recorded sound clips. The creations can be shared with friend. |
| Jaycut | **JayCut** [44] mixer allows you to make creative master pieces, or mixes for short, out of your video clips and photos. The created video can be exported in facebook or myspace. |
| MovieMasher | **MovieMasher** [45] is a set of Adobe Flash™ applets that provide front-end tools |

| | |
|---|---|
| | for common video editing tasks. The major functions found in movie masher are : trim, composite and timeshift video, mix and fade multiple audio tracks, add effects, transitions and titling. |

# 5 State of the art relating to DiY Smart Experiences applications

## 5.1 DiY Applications

Do-it-Yourself application creation (DiYApp) allows users to create applications without any coding, interface design or previous programming experience.
Do-it-Yourself application creation should be an intuitive user interface and easiest-to-use tool for creating applications. Users can build sophisticated custom applications by simply answering a series of straightforward questions. By using DiYApp, we can quickly deploy applications that are much less expensive and easier to maintain than traditional softwares.

### 5.1.1 Web-based DiY Applications

Zoho Creator [8] is a online service to create custom online applications easily and quickly. Zoho Creator offers intuitive drag-and-drop user interface to create customized online database applications and collaborate with friends, team members, clients or with the general public. You can create custom online application by importing data from your spreadsheet (.xls .csv .tsv) or create one from scratch using easy drag-and-drop web form builder. Zoho Creator helps user to analyze his data on the fly by searching, sorting and filtering. Google Sites [9] , makes creating a team site as easy as editing a document. Google Sites allows to centralize all types of information from videos to presentations and share your site with just a few people, your entire organization, or the world. Google Sites allow user to have a private area where other users could contribute without having to give them full administration privileges. Sampa [10] is an online private place where you can share stories and pictures, plan events and communicate with your close circle of friends. It allows also end user to create a website, plan the next family reunion and tell everyone about your last vacation abroad. All are done in a secure and private place. The DreamFace [11] , called Outsider, allows people to create, personalize and share their own web applications through a unique concept called Web Channels. Web Channels are mini dynamic applications designed for change. They incorporate DataWidgets; widgets which use diverse data sources coming from both corporate data and the internet, to display information and interact with other DataWidgets. JackBe [12] has developed and enterprise mashup platform called Presto. The Presto Platform empowers application developers and users to create, customize and share enterprise Mashups. It provides a dynamic integration approach to building enterprise mashup applications faster, leveraging internal and external data. Coghead [13] is a web based platform which software developers, IT professionals and tech-savvy business people to develop applications that automate business operations. It combines several capabilities:

- An Adobe Flex-based drag-and-drop editing,
- Powerful workflow and logic capabilities which give user the ability to model and automate any unique business process.
- Integration tools**,** Coglet, API allow you to weave your applications into the existing data infrastructure and build dynamic web capabilities for the business.
- scalable, secure and robust platform.

OblinQ [13] is the place where you can start showcasing stuff you found online. With OblinQ you can create your "Showcase" and collect showcase things. Then as you shop the web, do research, find interesting websites, products, videos, you simply add them to your OblinQ Showcase with a click.

### 5.1.2 NFC based applications (SmartTouch)

Near Field Communication (NFC) is short-range wireless connectivity technology that evolved from a combination of existing contact-less identification and interconnection technologies. Products with built-in NFC will simplify the way consumer devices interact with one another, helping people speed connections, receive and share information and make secure payments [13] .

ITEA SmartTouch utilises the combination of the capabilities of mobile devices, i.e. the virtually ubiquitous cellular phones, and new short-range communication (e.g. NFC) technologies together with

the existing backend systems. The goal of project is to develop versatile and easy-to-use mobile services based on NFC technology. The following list summerizes NFC applications developed by the ITEA SmartTouch project, but only those that provide DiY aspects, like installation, creation or adaptation of an application.

**NFC for Payment and Ticketing:**

- **"RMV-HandyTicket für NFC-Handys"**. Basic elements of this NFC ticketing approach are the mounted radio chips at the stops and stations called ConTags. They are the interface between customers and RMV information and ticketing services. The round blue ConTags enable an automatic starting of the ticket program on the mobile phone.

**NFC for special people:**

- The aim of **BlindNFC** is to overcome the obstacles that visually impaired people (not only but especially blind) have at home to identify particular items and their content. BlindNFC allows for labelling of music, the clothes, food or medicines with the simple help of a SmartPhone or PDA and some tags. The concept avoids the use of dedicated devices, used up to now, giving an extra functionality to a device that blind people use in their daily life.
- **NFC for children.**
  - The idea of **NFC Attendance Supervision for primary school** is to give parents real-time information about their child's arrivals and departures. Children starting first grade 'log in' to school by touching, with an NFC 'card', an active reader device that records the card ID (the child's name), the 'direction' (arrival in school), and a time stamp in the backend system. Parents receive information about their child's arrival in and departure from school either through the Citizens' Portal, or, should they so choose, as a text message to their mobile phone.
  - And **"Amazing NFC"** is a City orienteering for pedagogical purposes and it has two routes, a survival track and a culture/historical track. For example: it was also possible for the students to create a citizen's initiative by touching a tag on the bus. In the Zoological Museum, students made use of their NFC phones by downloading content from the museum's tags, which were part of the Oulu University NFC project.

**NFC for home**

- **Auto-adaptive remote controller**. The typical scenario would be a user wanting to manage a domestic appliance at home. He will only have to touch one appliance tag with the NFC-reader enabled mobile device and the touch-sensitive interface of the remote controller will be auto-configured. Then he will be able to interact with the selected appliance.
- **User identification in Comfort application.** In particular, at first step the user preferences for comfort in the home need to be defined (each user could change the configuration or profiles of each zone at any time.), but the system easily guides you to assign all the comfort magnitudes (i.e. temperature, humidity). Then, just by touching an NFC reader located near the front door (or in any other place/ room), the system automatically adapts the comfort conditions to the new context.

More information on the ITEA SmartTouch project can be found on the SmartTouch website [105] .

### 5.1.3 DIY geospatial web applications

#### 5.1.3.1 Introduction

Since the advent of Google Maps[105] , web mapping or web cartography applications have evolved from static, read only maps to highly interactive, customizable and essentially desktop-like GIS applications. Internet GIS has also dramatically increased the number of map producers and consumers and their qualifications. The barrier towards consulting and even creating online maps has been lowered in such a way that people outside of the profession and even without any geographical knowledge whatsoever have entered the field.

Obviously, many categorizations of geospatial applications and frameworks are possible depending on the viewpoint one wishes to choose. One way to subdivide the geospatial landscape would be by looking at the overall purpose of the application or framework: merely publishing geographical data or actually gathering, editing or even analyzing and processing it. Alternatively, one might choose to describe the complete spectrum of architectures involved: standalone desktop, 2 tier, 3 tier, cloud computing, etc. In the context of the DIYSE project, it is the mixture of both DIY content (who can use

it ?) and power/expressiveness (what can you do with it ?) which is probably the most interesting to ponder upon.

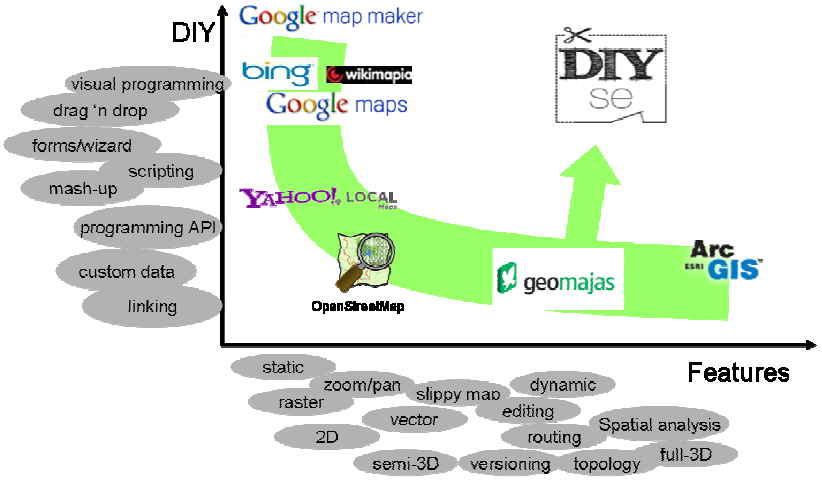



**Figure 76: 2D categorization of geospatial applications/frameworks**

Figure 76 positions a number of popular geospatial applications/frameworks along two axes: the horizontal axis provides a measure of the complexity or feature richness of the application while the vertical axis indicates the level of DIY accessibility. It is clear that most of the application space is either closely situated by the horizontal axis, indicating a significant barrier for access by inexperienced users or closely situated by the vertical axis, indicating a significant lack of features. In this project, first steps will be taken towards filling this apparent void.

### 5.1.3.2 Google Maps

One of the earliest areas where mashup technology was adopted has traditionally been the world of web mapping or web GIS (for the subtleties in terminology, see). It could be argued that Google Maps [105] is one of the key applications, if not the most prominent application to have brought mashups to the general public. [110]
The key feature of Google Maps was the introduction of advanced Web 2.0 interactive technology (AJAX/SVG[102] ) which leads to so-called asynchronous or "slippy" map experience and editing capabilities that were previously reserved to desktop GIS. By combining this with a powerful Javascript API, a new type of highly customized web maps became available for integration in external web sites. Since the introduction of Google Maps, other companies like Yahoo and Microsoft have followed suit with similar products and some open source clones have followed.

Google Maps[105] offers an API and some related products like My Maps, Map Maker or Google Earth API. By adding a couple of lines of JavaScript to their web page, web developers are able to create an inline custom map based on Google background raster data (road, satellite, terrain or hybrid) and some overlay data which could be inlined in the page or retrieved from KML [103] or GeoRSS [104] data sources. Version 2 of the API [106] also allows more advanced features like (reverse) geocoding (translating addresses to coordinates and vice versa), showing panoramic views, looking up directions, showing data on Google Earth and so on. The API itself is targeted to advanced end users like web developers, but the related applications like My Maps and Map Maker are targeted to inexperienced end users.

My Maps is a visual mapping environment based on Google Maps that allows end users to create personalized, annotated, customized maps.

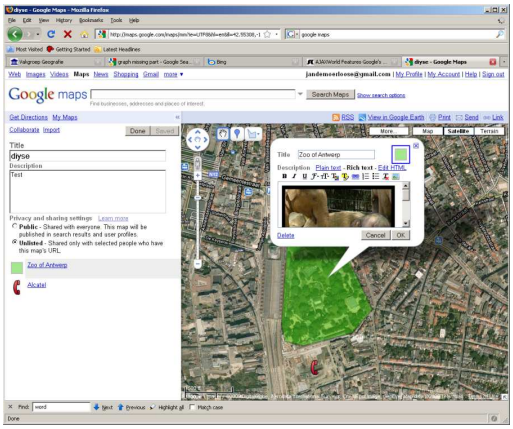



**Figure 77: Google MyMaps screenshot showing a shape and a placemark**

My Maps supports the same type of objects that are available through the JavaScript API: placemarks (points), lines and shapes (polygons). Object information is shown in a callout which can contain rich text and HTML. No HTML knowledge is required for adding objects to the map, but HTML for inserting pictures or videos has to be coded manually. Geographical object editing or digitization is supported through AJAX-style vector graphics handling (SVG/VML).

### 5.1.3.3 Bing Maps

Bing Maps (previously called Virtual Earth) is Microsoft's contender for Google Maps and offers an API that is very much comparable in terms of features. Bing Maps includes a feature called "My Places" editor, which offers functionality similar to Google "My Maps". The API offered by Bing Maps is not limited to standard AJAX but includes plugin-technologies like Silverlight.

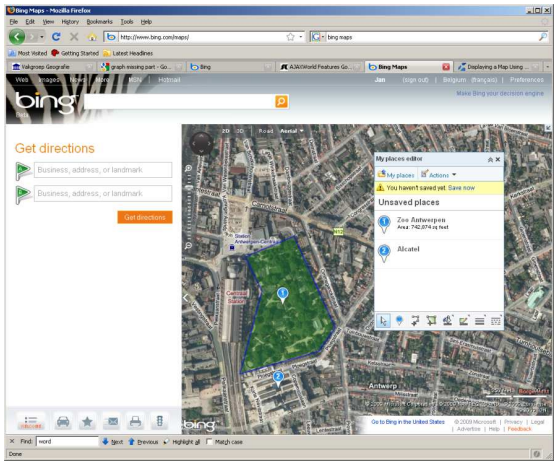



**Figure 78: Bing Maps screenshot with "My places editor" showing a pushpin and an area**

### 5.1.3.1 Yahoo Maps

Yahoo! Maps is a similar mapping infrastructure as Google's and Microsoft's and also offers a similar API. As opposed to My Maps and Bing Maps, there is no end user level tool for creating personal maps.

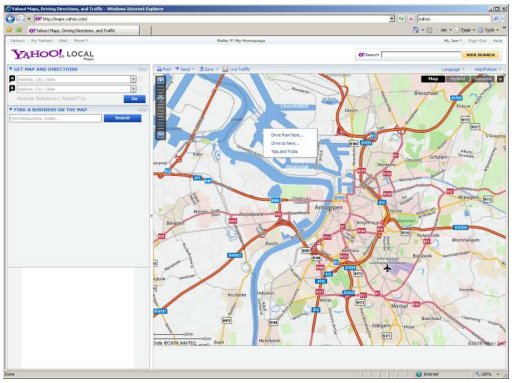


**Figure 79 Yahoo! Maps screen shot**

### 5.1.3.2 OpenStreetMap

OpenStreetMap is a free editable map of the world created by a huge number of voluntary collaborators all over the world. It is arguable one of the first examples of crowd-sourcing in the geospatial area and has since been copied by commercial companies like TomTom [113] .The OpenStreetMap data can be visualized and freely accessed by a number of web services. Changes to the data can be made through provided Flex and Java-based editors.

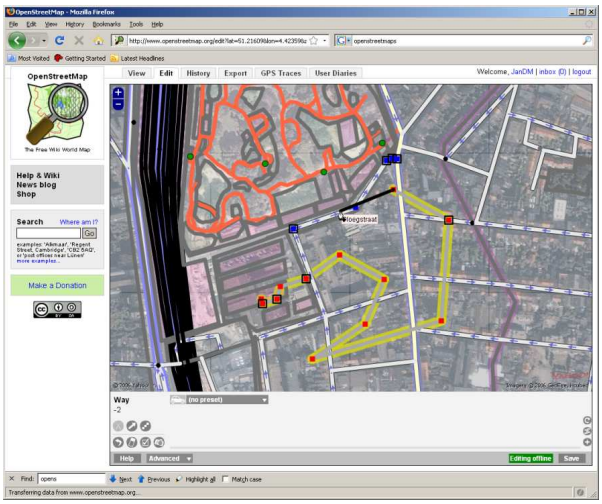


**Figure 80 OpenStreetMap Flex editor (PotLatch)**

The OpenStreetMap datamodel distinguishes itself from classical GIS data by the fact that it does not have a predefined metadata system but instead allows end users to add new types of geographical objects by tagging them in a specific way.

### 5.1.3.3 Geomajas

Geomajas[111] is an extensible open source web mapping framework, enabling integrated GIS solutions for businesses and government. Geomajas has an open architecture that enables easy sharing, integrating, and updating of GIS data. Geomajas is a component framework for building rich Internet applications (RIA) with sophisticated capabilities for the display, analysis, and management of geographic information. It is a building block that allows developers to add maps and other geographic data capabilities to their Web applications.

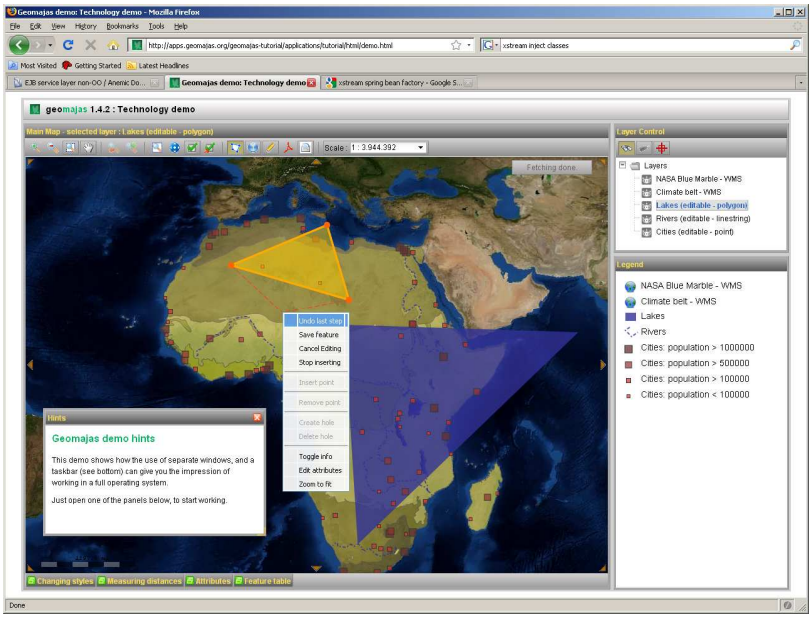



**Figure 81: Geomajas technology demo screenshot**

### 5.1.4 Media/Content applications

#### 5.1.4.1 Automatic video editing

Automatic video editing refers to a process which automatically selects suitable or desirable segments from an original video source and aligns them with a given piece of incidental music to create an edited video segment to a desired length [27] [28] .

Hua et al. [27] , [28] develop an approach for extracting temporal structure and determining the importance of a video segment in order to facilitate the selection of highlight segments. They also extract temporal structure, beats and tempos from the incidental music. In order to create more professional-looking results, the selected highlight segments satisfy a set of editing rules and are matched to the content of the incidental music.

Their system is composed on three stages (Figure 82) :

**Content analysis** : consisting of video temporal structure parsing, attention detection, sentence detection in the audio track of the original video, and beat/ tempo detection in the music.

**Content selection** : which selects a particular set of “important” and informative video segments that match motion with tempo, as well as shot boundaries with sentences in the audio track and music beats. The total length of the selected video segments may be determined either by the duration of the incidental music, or another desired value. This central stage is the primary and the most challenging one.

**Composition** : Which renders selected video segments with music by adding appropriate transitions between the selected video segments.

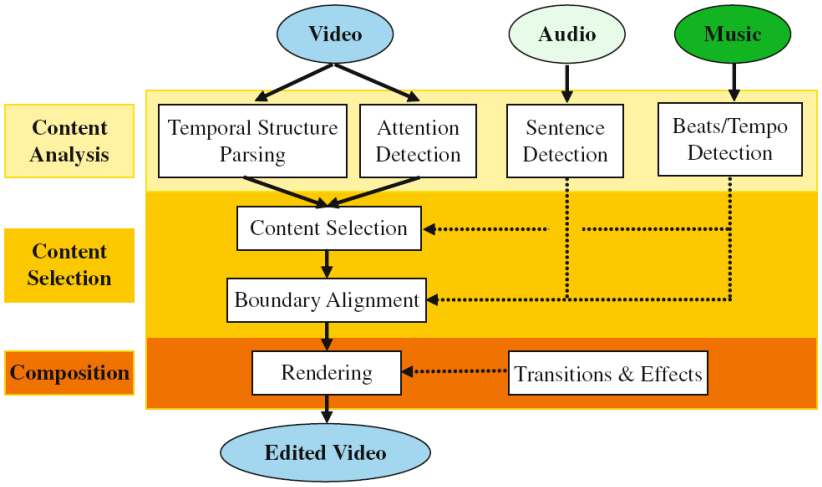



**Figure 82 : System overview [27]**

There are three "input" data sequences, namely, music, audio, and video, in the authors editing system (Figure 83).

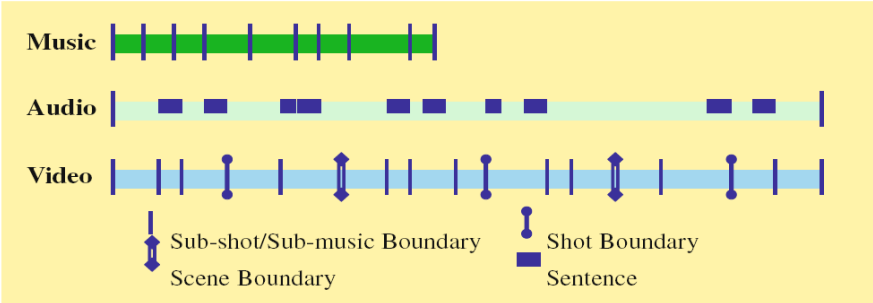



**Figure 83 : Three input data sequences [27]**

The strategy of Hua et al. [27] is to parse the video sequence into hierarchical structures consisting of scenes, shots, and sub-shots. For music, they segment it into clips by strong beats, and for each clip, tempo is estimated, which indicates the speed of the music sub-clips.

A video consists of a series of scenes:

In their system, the video segment selection is also based the work proposed by Ma et al. [29] . Hua et al. [27] , [28] refine the method by adding an "attention fusion" function, which generates improved results (Figure 84)

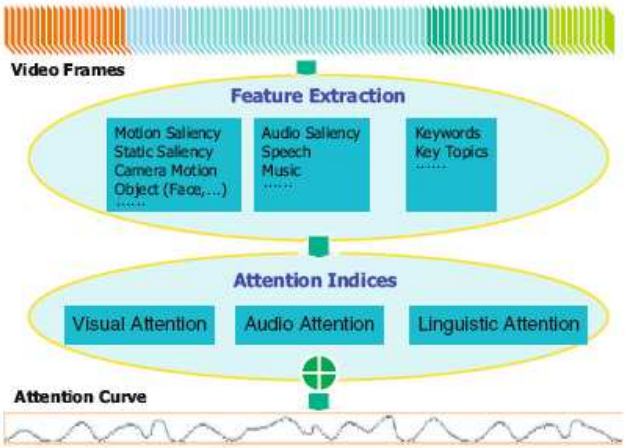



**Figure 84 : Architecture of attention model [28]**

Müller Arisona et al. [30] present Odessa framework, which automates the preparation of footage during composition and the editing process. Real-time audio analysis extracts music feature vectors, which are mapped to editing parameters of a video montage engine. The engine can be controlled through high-level editing parameters, such as looping speed or cutting rate. The authors apply a reverse editing of a video. The original video is analysed and re-organised in terms of individual scenes, groups, and shots (Figure 85). Odessa implements methods such as audio beat tracking or music similarity analysis to operate in real-time.

Figure 86 shows the process of automated video editing: audio analysis is applied to determine beat and part boundaries. Odessa's editing component assigns musical parts to available slots for clips, which are filled with pre-processed scenes of a given video clip. For each slot, shots of the assigned scene are looped according to given Non-Linear Editing (NLE) curves with respect to extracted beat boundaries (Figure 86: four shots of scene 1, one shot of scene 2, and two shots of scene 3). Automated editing as shown in this example releases the performing artist to deal with repeating time structures at the beat level.

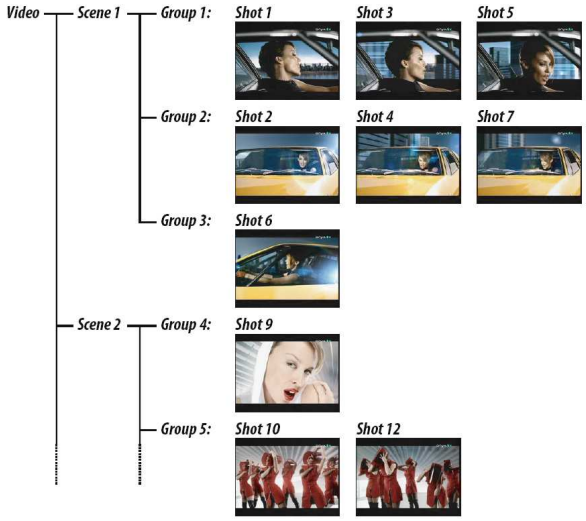



**Figure 85 : Reverse editing [30]**

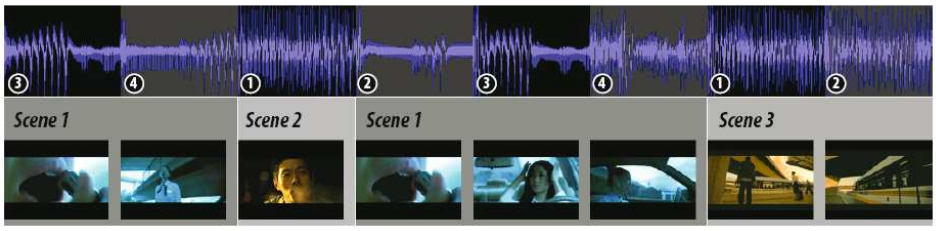



**Figure 86 : Automated non-linear [30]**

Takemae et al. [31] [32] propose an automatic video editing system using stereo-based head tracking in multiparty conversations for conveying the contents of the conversations to the viewers. Their system can automatically detect participants' head 3D position and orientation during a conversation. Based on the detection results, their system selects the shot of the participant that most participants' heads are facing. This approach exploits participants gaze behaviour to select the most effective shots of participants.

Mudhwuchutyula et al. [33] propose an automatic mechanism for XML based video metadata editing, in tandem with video editing operations. An implementation framework for editing metadata in accordance with the video editing operations is demonstrated. The video metadata editing mechanism has been implemented in the context of Digital Video Album (DVA) system for editing metadata in accordance with presentation/ summarization operations performed on video. Digital Video Album (DVA) focuses on techniques to index, retrieve, summarize and access digital video from home video and digital TV.

Foote et al. [34] have presented a music video creation system that can automatically select and align video segments to music. Authors have analyzed both audio and video sequences in order to automatically create music videos for a given sound track.

Ahanger et al. [35] have proposed composition techniques in a specific domain of video-based news. The structure of a new Item is described on Table 5. A new item composed on headline (segment type headline), an Introduction (segment type introduction), a Body (many segments types) and an enclose (segment enclose). The types of segments are divided on two categories :

**Single-presentation type** *(Ssp):* The segment types that allow only a single segment of its kind to be included in a composition. This includes segments of type Headline, Introduction, and Enclose.

**Multiple-presentation type (Smp):** The segment types that allow multiple segments of its kind to be included in a composition. This type includes segments that can belong to a body

| | | |
|---|---|---|
| **Headline** | | |
| **Introduction** | | |
| Body (current) | **Comment** | |
| | **Wild Scene** | |
| | **Interview** | **Question & Answer (QA)** |
| | **Speech** | |
| | **Enactment** | |
| **Enclose** | | |

**Table 5 : Structure of new Item [33]**

They propose two type of composition :

**Creation Instance**: First single presentation, the selection can be *interest-based for* This is achieved by selecting the segment with the highest selection interest. A segment s can be selected with a

uniform probability if it have the same interest with other. For Multiple presentation composition might be desired based on a preference for a particular type or ordering (e.g.,Wild Scene before Speech).

**Creation Period** :

If composition is period based or thematic ordering is required then, in addition to the rules. The single-presentation-type segments are selected according to the rules listed on Table 6.

| | Rules | Explanation |
|---|---|---|
| 1 | $s_k \mid \forall s \in S_h : b_k \leq b \wedge b = b_k$ | To build a news item in chronological order we select a segment belonging to the **Headline** set with the earliest time and date. |
| 2 | $s_k \mid \forall s \in S_{in} : b_k \leq b \wedge b = b_k$ | Similarly, we select a segment from the **Introduction** set with the earliest time and date. |
| 3 | $s_k \mid \forall s \in S_e : b_k \geq b \wedge b = b_k$ | We select a segment from the **Enclose** set that has the latest time and date. |
| 4 | $s_k \mid \exists m : (\forall s \in S_{sp} : m \geq l(s) \wedge m = l(s_k))$ | If more than one segment is available for a particular date, then we use the segment $s_k$ with the highest interest. |

**Table 6 : Creation rules compositions rules [33]**

## 5.2 Ambient experience applications

### 5.2.1 Ambient-oriented programming

The introduction of ever-smaller mobile devices has led to a tremendous increase in research into *mobile ad hoc networks*. Such networks have two discriminating properties, which clearly set them apart from traditional, fixed computer networks: applications are deployed on *mobile* devices connected by *wireless* communication links with a limited communication range. Such networks exhibit two phenomena that are rare in their fixed counterparts [14] :

- **Volatile Connections.** Disconnections become so omnipresent that they should be considered the rule, rather than an exceptional case.
- **Zero Infrastructure.** Services must be discovered on proximate devices, possibly without the help of shared infrastructure.

The ambient-oriented programming (AmOP) paradigm facilitates development of applications for mobile ad hoc networks by dealing with the above phenomena. In order to accomplish this, AmOP languages adhere to a set of well-defined characteristics [17] :

- **Non-blocking Communication.** All distributed communication is *non-blocking*, i.e., asynchronous, in order to ensure loose coupling between communicating parties. It is this loose coupling which significantly reduces the impact of volatile connections on a distributed application. With respect to communication, two degrees of coupling between communicating parties can be distinguished [18] :
    - **Decoupling in Time.** *The communicating parties do not need to be online at the same time.*
    - **Synchronisation Decoupling.** *The control flow of communicating parties is not blocked upon sending or receiving.*
- **Ambient Acquaintance Management.** An AmOP language should have built-in support for the discovery and management of proximate devices and their hosted services. However, the way in which communicating parties can discover one another reveals yet another degree of coupling with important repercussions in mobile ad hoc networks:
    - **Decoupling in Space.** *The communicating parties do not need to know each other beforehand [18]* .

### 5.2.2 AmbientTalk

AmbientTalk is a programming language developed specifically for writing programs to be deployed on mobile ad hoc networks. From a language-engineering standpoint, AmbientTalk is [19] :

- Dynamically typed, which is not the same as being untyped: AmbientTalk values are typed, but variables are not.

- Object-oriented, but prototype-based: AmbientTalk provides full support for objects, delegation (a form of inheritance) and polymorphic message dispatch. Objects are not instantiated from classes, as in Smalltalk, but rather cloned from existing objects or created ex-nihilo, as in Self.
- Flexible: like Scheme, Smalltalk, Self and many other languages, AmbientTalk embraces the use of elegant and expressive block closures to achieve a level of reusability far exceeding that of Java or similar languages lacking true closures.
- Event-driven: AmbientTalk has built-in support for actor-based concurrency, which operates based on entirely asynchronous and event-driven communication. There are no threads, no locks, no deadlocks and no data-level race conditions.
- Distributed: AmbientTalk has built-in language constructs to make objects discover and talk to one another in a peer-to-peer manner across a TCP/IP network.
- Symbiotic: built on top of the Java Virtual Machine, AmbientTalk exploits the Java reflection API to enable AmbientTalk objects to collaborate with Java objects. This enables reuse of existing Java libraries while not abandoning the dynamic programming style of AmbientTalk.

# 6 State of the art of DIY concepts and scenario

## 6.1 Concepts

### 6.1.1 Concepts related to service compositions

A service as defined by OASIS[1] is *a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description*. One of the ways of implementing a service is provided by the Web Service. Web Services may in turn be divided in WS-* based and RESTful-based services.

WS-* services are defined by a standard interface described by the Web Services Description Language (WSDL). These web services interact with each other by means of SOAP messages serialized in XML. SOAP is a message layout specification that defines a uniform way of passing XML-enconded data between endpoints. Finally, the other element on the WS-* stack, UDDI provides a registry that act as a mechanism for clients in order to find web services. On the other hand, RESTful web services are based on the REST[2] architecture and define their interfaces through the so-called RESTful APIs.

Traditional WS-* standards, provide the foundations for a web of services but there are other elements required to achieve automatic web service discovery, selection and composition. In order to achieve this level of automation semantics applied to services (semantic web services) are proposed as a solution.

A Semantic Web service is defined as an extension of Web service description through the Semantic Web annotations, created in order to facilitate the automation of service interactions (McIlraith et al. 2001). Semantic annotations help to specify the meaning of the data exchange by the service, as well as its capabilities. Two or more services annotated with the same concept of an ontology will unequivocally interpreted in the same way. Furthermore, the use of semantic annotations provides the infrastructure to perform reasoning.

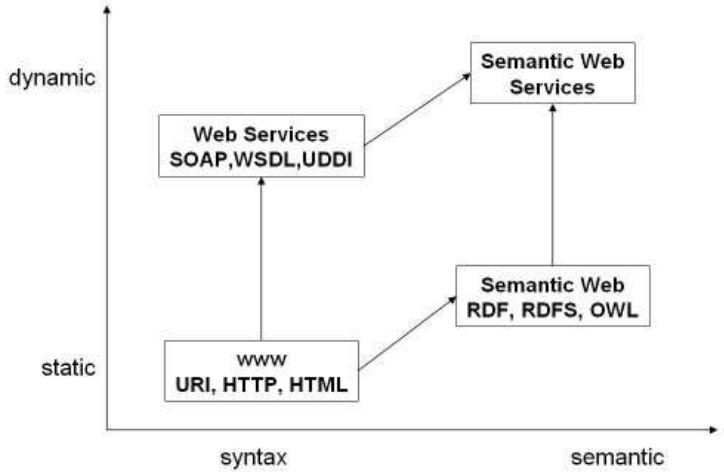



**Figure 87: Semantic Web Service**

In order to describe web services from a semantic perspective, there are two prominent approaches, WSMO and OWL-S. WSMO-lite, a much lightweight approach based on WSMO has been recently proposed as an alternative. Some previous European projects that use one of the previous approaches are:

- Infrawebs[3], an FP6 project that provides a toolset for creating, maintaining and executing Semantic Web Services annotated with the Web Services Modeling Ontology (WSMO).

[1] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
[2] http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
[3] http://www.foibg.com/ijita/vol13/ijita13-2-p11.pdf

- SOA4All, an FP7 project that makes use of WSMO-Lite as the semantic service modeling framework.

One of the benefits of atomic web services is the possibility of combining them together to create a composite of services that delivers some higher functionality that satisfies the user goals.
Service composition can be defined as creating a composite service, by combining other component services [188] . It is mandatory when a request can only be satisfied not by a single service, but for a combination of them. Service composition has the potential to reduce development time and effort for new applications by reusing existing services. Historically, manual service composition has been recognized as a prone-error task, hence, the need for an automated way of composing them. To carry out the composition some requirements must be stated in terms of goals or preferences.

### 6.1.2 Concepts related to semantics

#### 6.1.2.1 Introduction

Semantics is science about the study of meaning, or interpretation of meaning of an element, symbol, word, expression, o formal representation.It can refer to meaning of words in the linguistic fields, meaning of designed structures in the fields of software design, and meaning of assigned symbols in the research field of semiotics and so forth.

The main problem which we encounter is information without semantic. Putting humans as example, we have an unlimited information that is at our disposal which language we are able to understand, but we have a limit knowledge (implicit conceptual maps domain), so we are able to know the meaning of a small percentage of information and proceed in an intelligent way in a few of domains.

If we extrapolate the real world to the web, and human to machines, we can observe the same problem in a different scale, and we'll notice the great limitation of machines to understand or provide meaning to any information available in web. In the current Web, the data (facts obtained from events) or information (data are processed, organized, structured or presented in a given context so as to make them useful) are available without any semantic information (conceptual maps) associated that can make the role of the human brain. Giving semantic to information to resolve this limitation and allow interoperability between systems is the lead goal of semantic approach of the DiY-SE.

Whereas Wigg (1999) views knowledge as fundamentally different from data and information since knowledge consist of truths and beliefs, perspectives and concepts, judgments and expectation, methodologies and know-how and is possessed by human, agents, or others active entities and is used to receive information and to recognize and identify; analyze, interpret and evaluate; synthesize and decide; plan, implement, monitor and adapt to act more or less intelligently. In other words, knowledge is used to determine what a specific situation means and how to handle it.

The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. [198]

#### 6.1.2.2 Semantic describe by Ontologies

To support the sharing and reuse of formally represented knowledge among AI systems, it is useful to define the common vocabulary in which shared knowledge is represented. A specification of a representational vocabulary for a shared domain of discourse -- definitions of classes, relations, functions, and other objects -- is called ontology.

There exist many definitions of ontology, amongst which the definition given by Tom Gruber is a widely accepted one. An ontology is "an explicit specification of a conceptualization …When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. Those sets of objects, and the describable relationship among them, are reflected in the representational vocabulary with which a knowledge-base program represents knowledge…" [175] .

Although this isn't the only way to specify a conceptualization, it has some nice properties for knowledge sharing among AI software (e.g., semantics independent of reader and context). Practically, an ontological commitment is an agreement to use a vocabulary (i.e., ask queries and

make assertions) in a way that is consistent (but not complete) with respect to the theory specified by an ontology. We build agents that commit to ontologies. We design ontologies so we can share knowledge with and among these agents

The research fields in Ontology Engineering (OE) range *from* ontology creation methodologies, with which we capture semantics from domain experts in a domain, domain ontology modeling approaches, with which we design, model and store domain ontologies, *to* ontology applications, the knowledge base of which is the created domain ontologies. The basic readings on OE in general are referred to [177] [178] [179]

A fundamental feature of OE is the *interoperability*. The IEEE [181] defined interoperability as: "the ability of two or more systems or components to exchange information and to use the information that has been exchanged." In other words, interoperability allows different organisations (both private and public) to exchange information and knowledge to reach a common goal. The European Union has founded the IDABC programme F, which developed the European Interoperability Framework F (EIF). This framework describes policies and standards to be agreed by all organisations involved to promote interoperability. EIF have identified four types of interoperability enabling the transfer data across systems that have some relevance to DIY-SE; namely semantic, technical, organisational, and legal interoperability. The different types of interoperability are explained below:

- Semantic interoperability enables different systems to understand the intended meaning of the data being exchanged. Suppose two systems are exchanging data about a common user, then these systems should be able to recognise the user even if their internal representation is different.
- Technical interoperability considers technical issues related to linking computer systems and services. It includes key aspects such as open interfaces, interconnection services, data integration and middleware, data presentation and exchange, accessibility and security services. This type of interoperability results in more reliable exchange and reduces the amount of maintenance compared to ad-hoc solutions.
- Organisational interoperability is concerned with defining business goals and modelling business processes for the interaction between organisations. Furthermore, this type of interoperability aims at addressing the requirements of the user community by making services available, easily identifiable, accessible and user-oriented. Semantic markups can describe the relationships between activities and the role of their participants (e.g. actors, data) in a meaningful manner.
- Legal (and political) interoperability addresses the legal and political constraints on how the information is exchanged and used by the different organisations. These constraints include laws related to copyright, privacy, freedom of information, telecommunication regulation, and trade policies. For example, an organisation (e.g. health centre) attempting to access the medical record of a patient located in another country would have to abide to the regulations in its own country as well as those of the patient's.

In DIY-SE, we tend to tackle the semantic interoperability problem using ontologies.

Ontologies are increasingly being applied in complex applications, e.g. for Knowledge Management, E-Commerce, eLearning, or information integration. In such systems ontologies serve various needs, like storage or exchange of data corresponding to an ontology, ontology-based reasoning or ontology-based navigation [176]

Several ontology languages have been developed during the last few years, Some of them are based on XML syntax, such as Ontology Exchange Language (**XOL**), **SHOE4** (which was previously based on HTML), and Ontology Markup Language (**OML**), whereas Resource Description Framework (**RDF**), and **RDF Schema** are languages created by World Wide Web Consortium (W3C). Two additional languages are being built on top of **RDF(S)**—the union of RDF and RDF Schema—to improve its features: Ontology Inference Layer (**OIL**) and **DAML+OIL9**. Finally the Web Ontology Language (**OWL**) is designed for use by applications that need to process the content of information instead of just presenting information to humans. Although it is based on XML, RDF, and RDF Schema (RDF-S) , OWL facilitates greater machine interpretability of Web content than that supported by them providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite (a small subset, easier for frame-based tools to transition to, easier

Reasoning), OWL Full (maximum expressiveness and the syntactic freedom of RDF with no computational guarantees), and OWL DL based on Description Logic: [200]

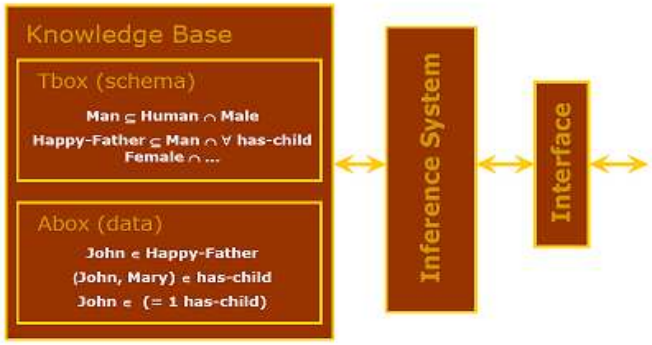



**Figure 88: DL Architecture**

### 6.1.2.3 Semantic apply to DIY-SE

Making technology (sensors, processors, actuators, information terminals, and other devices interconnected through a network) will become invisible, embedded in our natural surroundings, present whenever we need it, enabled by simple and effortless interactions, attuned to all our senses, adaptive to users and context and autonomously acting to enhance the services that can provide to humans. In order to fulfil all these expectations, must amalgamate results from several research areas, on of them is semantics to providing:

- Ontology-based search

In DIY-SE project, ontology is required to enable information exchange across (human and non-human) agents within a community in a scalable and reusable manner. It facilitates the collaboration between the community members to share their achievements and software/hardware components through DIY activities.

For example in the save-the-world scenario, a person might be interesting in browsing or searching existing solutions before he makes his own solution. Compared to keyword based search, ontology-based search as the following advantages [180] .

- Users will make fewer mistakes.
- Users will be able to complete information-finding tasks in less time.
- It is easy for less experienced users.
- It is easy for the users who lack a detailed knowledge of the underlying technical system implementation.
- The effort for developing and maintaining the ontology will not significantly exceed the effort to develop and maintaining the free text approach.

Figure 89 shows an ontology-based content management, which is a use case of ontology-based search. Firstly, an ontology is used to annotate the content that is of interest to the user, which might either mean that ontology-based descriptions are embedded in the documents themselves or references to the documents are added to the ontology.

Secondly, a Knowledge Worker (KW) finds some information, forms a semantic query in terms of the same ontology and submits it to the Content Management System (CMS) for processing. In its reply, the system returns (references to) resources and helps the KW to refine his query, e.g. by suggesting generalizations, more specialized terms or related terms.

The ontology-based search components in the DIY-SE project can make use of this idea to iteratively refine user's query until he finds the existing solutions he wants. If he finds existing solutions, then he can reuse them or build his own solution based on them. If he does not find any solutions, then he needs to create a new solution, and push this new solution to the community website. When the ontology engineer sees this new solution, he will annotate it automatically (or semi-automatically) with the domain ontology. If the ontology engineer sees new concepts contained by this new solution, then

he can enrich the ontology by introducing these new concepts. We refer to the community-based ontology creation methodology for the details (see the next subsection).

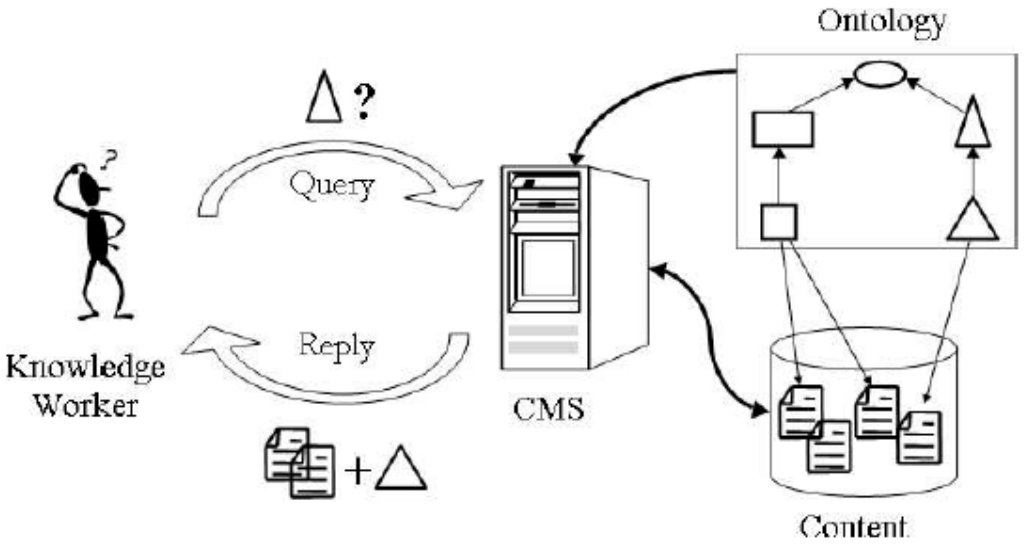



**Figure 89: ontology-based content management system (Mika et al., 2003)**

Another application of ontology-based search is the search products from Google: the Search Options and Rich Snippets.

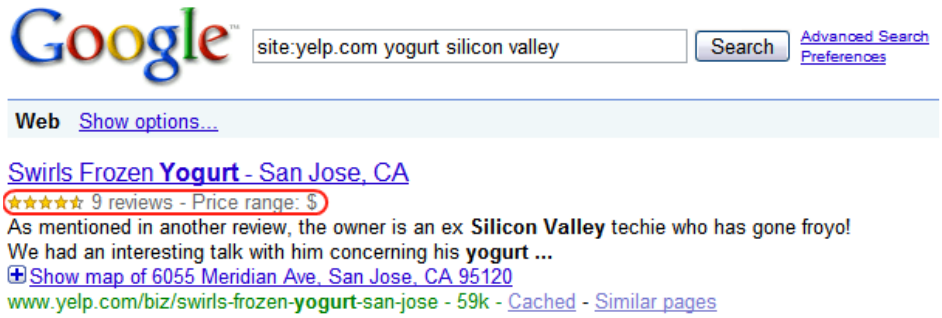



**Figure 90: Google Search Options and Rich Snippets**[4]

The two features notably extended Google's core search product and the 'rich snippets' part in particular was based around structured content. Rich snippets extract and show more useful information from web pages. Google is using structured data open standards such as microformats and RDFa to power the rich snippets feature. It is inviting publishers to mark up their HTML. By doing so, web information is semantically annotated and becomes meaningful.

For instance in Figure 91, reviews and price range is shown under the site http://www.yelp.com/biz/swirls-frozen-yogurt-san-jose when “yogurt” and “silicon valley” are typed. Google search engine finds that this search result is from a website containing the information of a restaurant. Therefore, the reviews and price range are shown.

These two features are built based on Google’s previous products, such as SearchWiki, which enables users to add, annotate, and remove your search results. Note that SearchWiki is not a wiki system.

The Google Search Options and Rich Snippets support real-time information, adding more meaning to the data and filtering results. The new features show that Google is adapting to the Semantic Web technology.

[4] http://www.readwriteweb.com/archives/top_10_semantic_web_products_of_2009.php

In DIY-SE project, we can adapt several perspectives from the Google semantic search products.

1. The web is getting more complex, users are getting more sophisticated. Therefore, Google is evolving accordingly. Nevertheless, Google is hiding the complexity from the average users. This is an important issue especially in the DIY-SE theme. Our ontology-based search needs to serve for two kinds of people for DIY: 1) experienced users (geeks and nerds), 2) non-experienced users. For the non-experienced users, to hide system complexity is necessary.
2. Semantic search is made as an option for Google. End users can do a lot of things with Google search but they don't have to. In DIY-SE, we shall make ontology-based search as suggestions, instead of mandatory components.

- Integration and sharing knowledge: User (profiles, preferences ...), devices, task, interface, and services models.

The numerous, diverse, and distributed in users' environment devices (Internet-of-thing), user (Internet of user), and services (Internet of services), along with the diversity of data, information, and knowledge provided by services, makes impossible for them to work together, as each one of them may be designed independently for particular applications in some domain in which that particular resource might be used. In order to solve this, it's necessary to define a vocabulary that defines the shared concept, known and shared by the community that uses it, and that is the base to develop some DIY-SE application.

In the context of knowledge sharing, I use the term ontology to mean a *specification of a conceptualization*. That is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents to refer the same concepts in some domain.

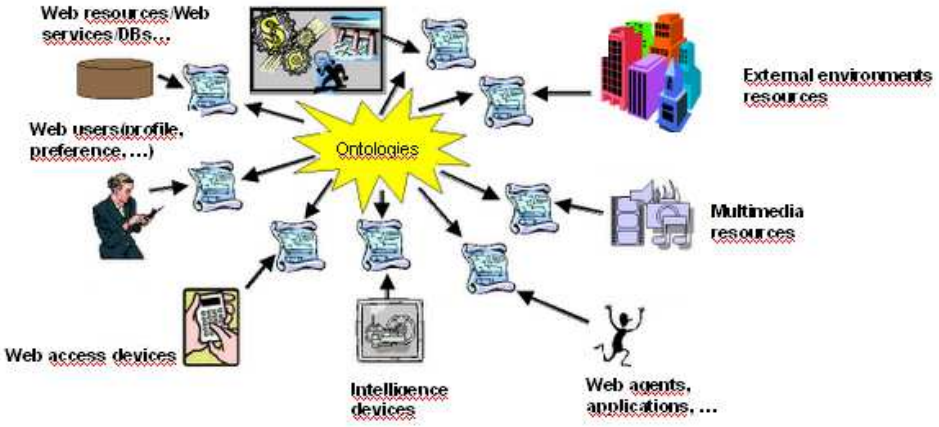



**Figure 91: Shared information**

- Understanding and interoperability among machines

The increasing volume of data, description and information of devices and application services in modern times has called for more automated. This, added to requirements to make transparently physical (devices) to user and to extend syntactic interoperability between services (giving by SOA) to of semantic interoperability, this constrain a formal and precisely knowledge representation to allow to machine processes and understand information and how to act with it.

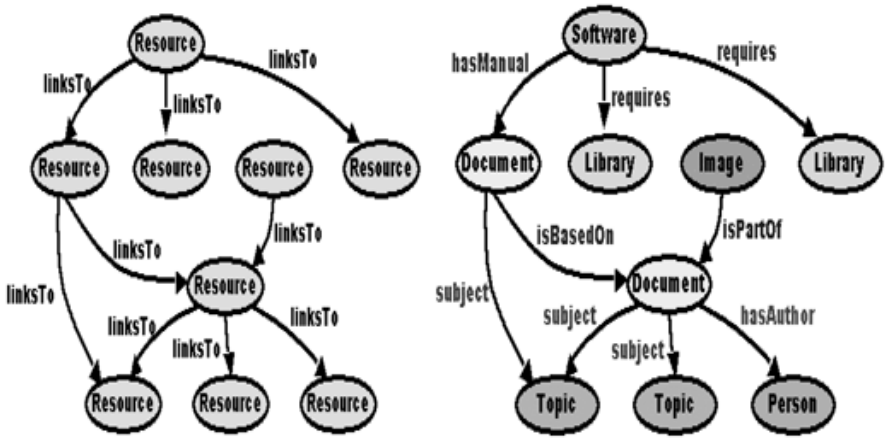



**Figure 92: From syntactic to semantic**

We aim a semantic layer providing intelligence task like: the discovery of services (goal and context oriented), invocation of services (context based parametrisation and data transformation), creating, composition and orchestration of services (mediation of their functional as well as non-functional characteristics), and allowing the user (technical and not) to work only with a highest level, and functional layer.

- Implicit knowledge: capability to reasoning

It's impossible to gather all the information about domain, and even more make whole and explicit knowledge specification it.

Reasoning is a cognitive process that yields conclusions from given premises. It occurs whenever human beings make implicit information explicit. One form of reasoning is deduction. By definition, in deductive reasoning, the truth of the premises ensures the truth of the conclusion (In contrast to inductive reasoning, in which the truth of the premises does not warrant the truth of conclusion).

And we need to get the maximum expressiveness to retaining computational completeness and be able to apply a reasoner. They typically experience reasoning as seeing the information from the premises to find new, not explicitly given information.

- Intelligence ambient: context-awareness models

To support intelligence applications, the model for context-awareness functionality are also provided. The models are created as a combination of OWL ontologies and SWRL [Horrocks, 2004] rules and serve as the basic mechanism supporting the self-* properties [Zhang, 2008]. At present, the context-awareness models are used in the self-management and self-diagnosis task. The ontologies contain the models of devices and the state machines representing the actual status of devices. The ontology containing possible malfunctions, which may occur on devices is also used. The context is modelled using the SWRL rules, which can be defined on the device level (monitoring and reacting to the state of the single device) and the system/application level (monitoring and reacting to the context created as a combination of multiple devices states). The continual execution of rules may lead to generation of several malfunctions containing the description of error and the related remedies, which are provided to application users [201] .

These are possible applications of ontology in the DIY-SE project. It can also be extended into other possible applications, such as ontology-based data matching, ontology-based data clustering, ontology-based information retrieval and ontology-based information grouping. These topics can go very deep and far. We believe that some of these topics will be addressed in the future in DIY-SE.

### 6.1.2.4 State of the art

According to web ontology language, actually the owl working group have developed a new version of OWL 2 (27 October 2009), which adds new functionality that we resume [202] :

- keys;
- property chains;
- richer datatypes, data ranges;
- qualified cardinality restrictions;
- asymmetric, reflexive, and disjoint properties; and
- enhanced annotation capabilities

Nowadays the goals of semantic technologies are more approached to automation issues and introducing semantic to concepts close to web 2.0:

- *Semantic Web Services*, Increasing of automation in web services processes.
- The area of *semantic rules* is perhaps the most important frontier today for the Semantic Web's core technology and standards, and includes a number of exciting research issues. Rules extend databases and ontologies with more powerful, flexible, and active forms of "structured" knowledge (as opposed to "unstructured" knowledge such as text), and have a number of close relationships to other aspects of the overall Semantic Web such as ontologies, query and search, wikis, policies and trust, e-science and e-commerce, and services.
- *Linked Data*, Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. More specifically, Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF."
- *Multimedia Semantic,* applications can benefit from semantic metadata for creating, searching and presenting multimedia content.

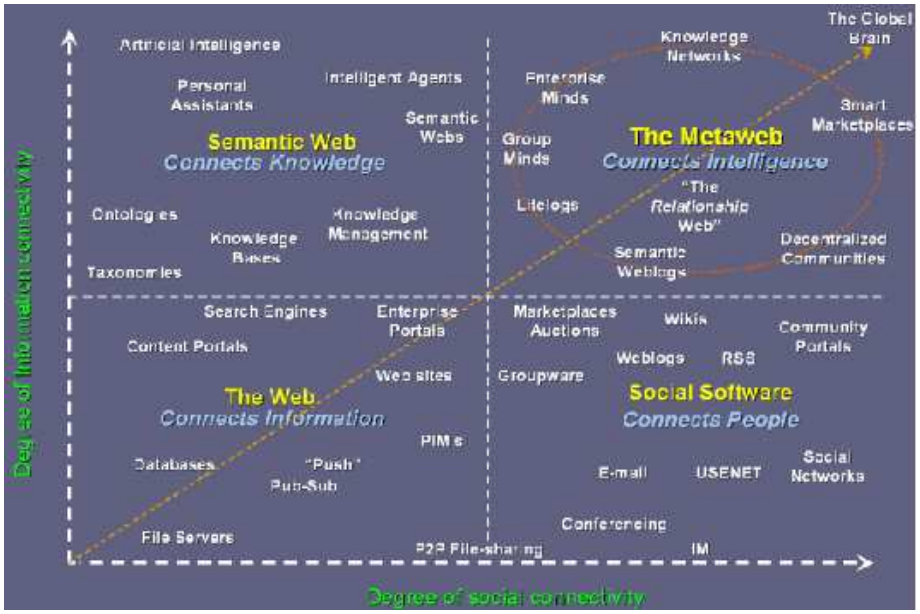



**Figure 93: Connectivity**

An overview of research and development ontology related to ambient intelligence:

- iHAP Ontology. The iHAP Ontology has been made from reusing some FIPA ontologies and defines new terms to build a vocabulary that represents knowledge in smart environments.[203]
- GUMO (general user model ontology) is a user model ontology, which models many static as well as dynamic dimensions of a user,
- ConOnto (includes location, time, activities, software and hardware profiles, also includes meta-information that describes negotiation and fuzzy ontologies to be used in systems that will negotiate and infer about context information, WSMO (to models semantic web service)

- The NRL ontology is a set of various security related models covering the representation of credentials, algorithms, assurances, but also the service security aspects directly supporting the SOA approach

To end up, some projects dealing with smart ambient, that we used as basis due to their relationship with smart ambient and semantic technologies, are:

1. *SmartMuseum Project*, The overall objective of the project is to develop a platform for innovative services enhancing on-site personalised access to digital cultural heritage through adaptive and privacy preserving user profiling. The most important result of us is a common ontology for content, context and user preference presentation [204] .
2. The *SM4ALL* project will investigate an innovative middleware platform for inter-working of smart embedded services in immersive and person-centric environments, through the use of composability and semantic techniques for dynamic service reconfiguration [205]

### 6.1.3 Concepts related to semantics engineering

#### 6.1.3.1 DOGMA

In the later 90's of the last century, Prof. Robert Meersman from VUB STARLab brought forward the idea of applying the principles of database design methodology to ontology engineering – Developing Ontology Grounded Methods and Applications (DOGMA) paradigm in ontology engineering is introduced in [182] [183] .

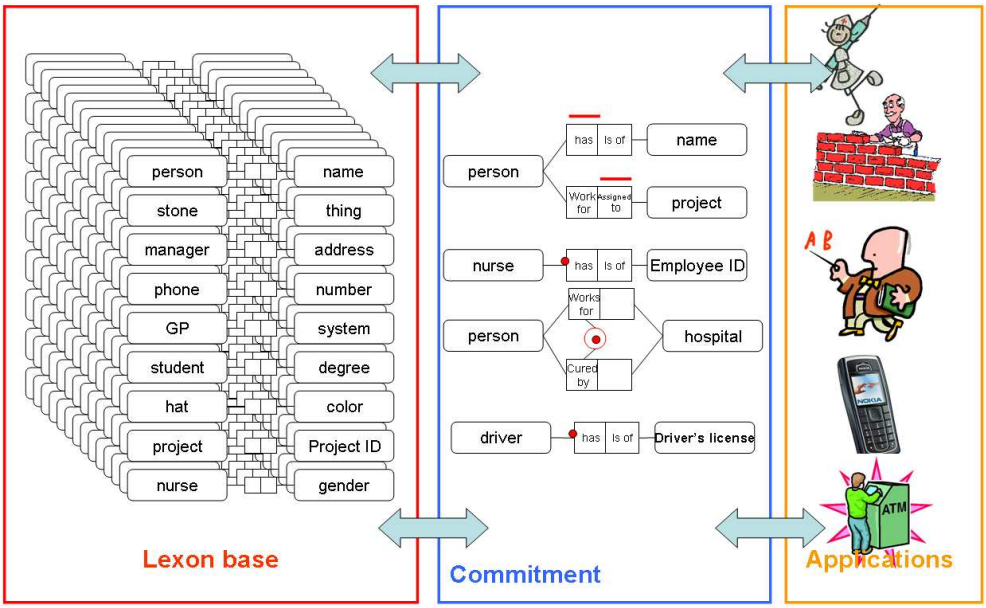



**Figure 94: DOGMA framework**

In DOGMA, ontology is decomposed into two layers (Figure 94): 1) lexon base layer that contains a vocabulary of simple binary facts called lexons and 2) commitment layer that formally defines rules and constraints by which a set of applications may make use of these lexons.

**1) Lexon.** A lexon, which represents a simple fact, is a quintuple < γ, $t_1$, $r_1$, $r_2$, $t_2$>, where $t_1$ and $t_2$ are terms that represent two concepts in some language (e.g. English). $r_1$, $r_2$ are roles ($r_1$ corresponds to "role" and $r_2$ is "co-role") referring to the relationships that the concepts share with respect to one another. γ is a context identifier which serves to disambiguate the terms $t_1$, $t_2$ into the intended concepts, and in which the roles $r_1$, $r_2$ become "meaningful".

For example, we have a lexon <γ, *employee, work for, is the working place of, enterprise*> indicating a fact – (an) employee works for (an) enterprise.

**2) Commitment.** The ontological *commitment* layer formally defines rules and constraints by which an application (or "agent") may make use of lexons. A commitment uses application specific instantiation, integrity constraints and logical connections. It bridges the ontology base and the applications, aiming

to find a balance between monolithic representation of generic and specific, declarative and operational semantics of the application domains.

A commitment needs to be expressed in a commitment language. A commitment is expressed as a *constraint* (or rule) applied on the semantic path. Suppose we have a lexon <γ, employee, work for, is the working place of, enterprise> with a constraint - "*one* employee works for *at most one* order manager". We apply the uniqueness constraint *UNIQ* on the lexon *p1* written as: *p1 = [employee, work for, is the working place of, enterprise]: UNIQ (p1).*

### 6.1.3.2 DOGMA-MESS

DOGMA-MESS (DOGMA – Meaning Evolution Support System) is a methodology to support interorganizational ontology engineering [184] and to assist a community of domain experts to gradually enrich ontologies.

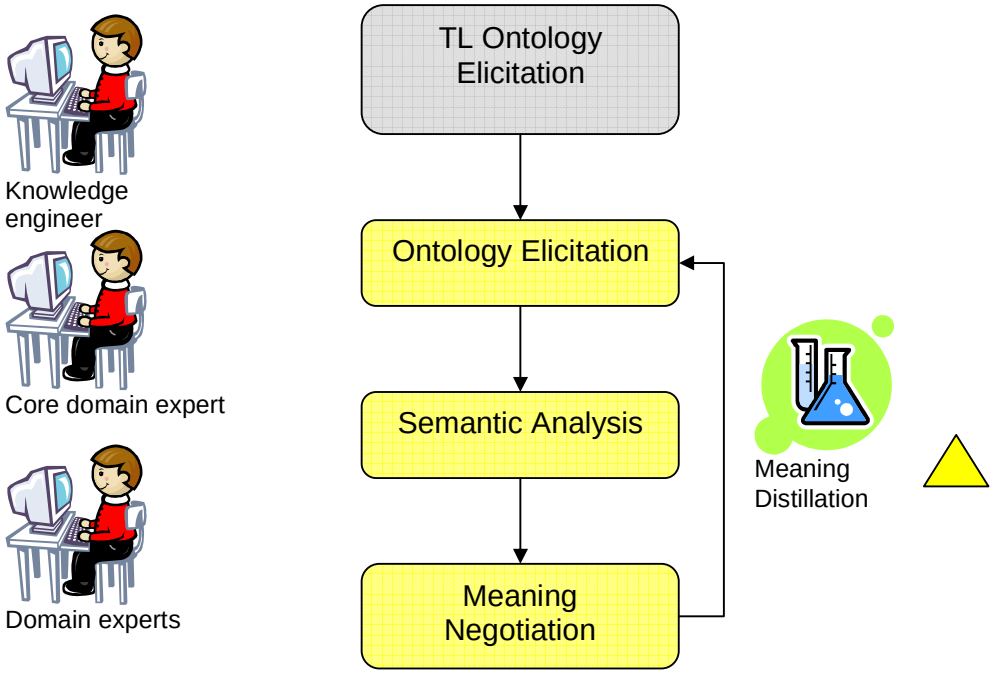



**Figure 95: DOGMA-MESS iterative process**

Figure 95 shows how a domain ontology is created and evolves among a group of domain experts. It supports users playing three main roles: the domain experts, the core domain experts and the knowledge engineer.

The domain experts are professionals in a particular domain. The core domain experts have thorough expertise in the cross interests between different enterprises. The knowledge engineer, who has excellent expertise in representing and analyzing formal semantics, is responsible to assist the domain experts and core domain experts in the processes of ontology creation, validation and evolution.

The top level ontology is the ontology containing knowledge of the philosophical foundation of ontology creation. During the common meaning distillation iterations, validated versions of domain ontology are created.

The main focus in DOGMA-MESS is on how to capture relevant commonalities and differences in meaning of concepts. It is a community grounded methodology to address the issues of relevance and efficiency.

The major task of the ontology engineering track is to capture the ontologies that really matter the overlapping interests from different organizations or enterprises. The domain experts need to specify the domain templates that reflect the organizational perspective in their Organizational Ontologies.

The domain experts are shielded from complexity issues by performing specific tasks in the elicitation process.

For instance, a template of “Competence” defines that a competence needs to have “Competence level”, “Action” and “Actor”. The “Action” has a certain “Quality”. When a domain expert gets the template, he may specify the “Competence” as “Speech clarity”, the “Competence level” as “Very good”, the “Action” as “Speak” and “Understand”, and the “Actor” as “Person actor”.

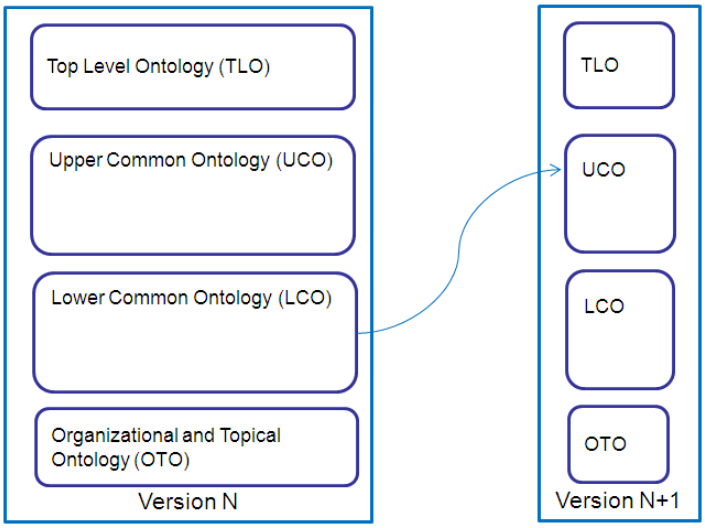



**Figure 96: DOGMA-MESS evolutionary process**

Figure 96 shows an overview of how an ontology evolves.

Within each version, four levels of ontologies need to be distinguished:

- *Organizational Ontology* and *Topical Ontology* (OTO) seeks to systematically represent the knowledge structure the individual domain experts have on given *themes* (or *tasks*). A Topical Ontology lays foundation for application (or task) specific ontologies and conceptual models. Its semantic space covers multiple subjects and dynamic evolution of the core concepts within a topic. Concepts within a topic represent terminology of an application’s structure, assumption and framework. Within a version, every domain expert (or every enterprise-wise stakeholder group) is responsible to build his own OTO based on the ontology models in the UCO. For example, we have a Conceptual Graph model that describes the definition of “Teacher” and “Course” at the UCO level. Based on it, a domain expert may introduce a new relevant concept “Patience” by adding a conceptual relation “has skill” to “Teacher”. Similarly, a new relevant concept “Oral comprehension” can be introduced at the OTO level.
- *Lower Common Ontology* (LCO) constitutes the most important and complex layer for DOGMA-MESS. The concept definitions at this level are the organizational and topical specializations, which are created at the Organizational Ontology or Topical Ontology (OTO) level after they are aligned[i] and merged[ii]. This process happens within a version. The candidate concepts are analyzed by an SDT, which contains the decisions of whether a concept can be lifted (or merged) to the LCO or not. The concepts, which are not lifted, will keep in the OTO and wait for the next iteration of versioning process. In the next section, a concrete SDT example with a lifting rule will be demonstrated.
- Each domain has its own *Upper Common Ontology* (UCO); the *Upper Common Concept Type Hierarchy* organizes the (evolving) concept types that are common to the domain. For example, ‘Actor’ at the TLO level can be translated into the concepts ‘employer’ and ‘employee’ at the UCO level of the Human Resource Management (HRM) domain. Domain experts define domain canonical relations in terms of the domain. For example, the domain canonical relation ‘hire’ can be applied between ‘employer’ and ‘employee’. When an ontology evolves, all the concepts in the UCO of *version N* are lifted to the concepts in the UCO of *version N+1*. The *core domain experts* are responsible to standardize the concept definitions at this level.

- *Top Level Ontology* (TLO) defines the abstract concept types, such as 'Actor', 'Object', 'Process' and 'Quality'. Conceptualization at this level is not allowed to be changed. The relations between these concept types fall into two categories: 1) the hierarchical relations, which are also called *subsumption* ontological relations. 2) Other Core Canonical Relations, such as "part-of" *merelogical* relation, "property-of" relation and "equivalent" relation.

### 6.1.3.3 Semantic Decision Table

A decision table is defined as a "tabular method of showing the relationship between a series of conditions and the resultant actions to be executed" [185] . More specifically, Wets et al. [186] define a decision table as "a tabular representation used to describe and analyze procedural decision situations, where the state of a number of conditions determines the execution of a set of actions".

Researchers have been investigating the study of decision tables for more than fifty years. As an important tool to support Information System Management, decision tables have many outstanding advantages, i.e. they are easily learned, readable and understandable by non-technical people. Seeing the advantages, the interest of decision tables has been rising steadily. However, often the definition of concepts, variables and hidden (or meta-) decision rules that underlie remain implicit. When decision tables get larger, ambiguities, content inconsistencies and conceptual reasoning difficulties arise. The situation gets naturally worse when a group of decision makers need to build decision tables in a collaborative environment. Thus, the concept of Semantic Decision Table (SDT) was proposed [187] .

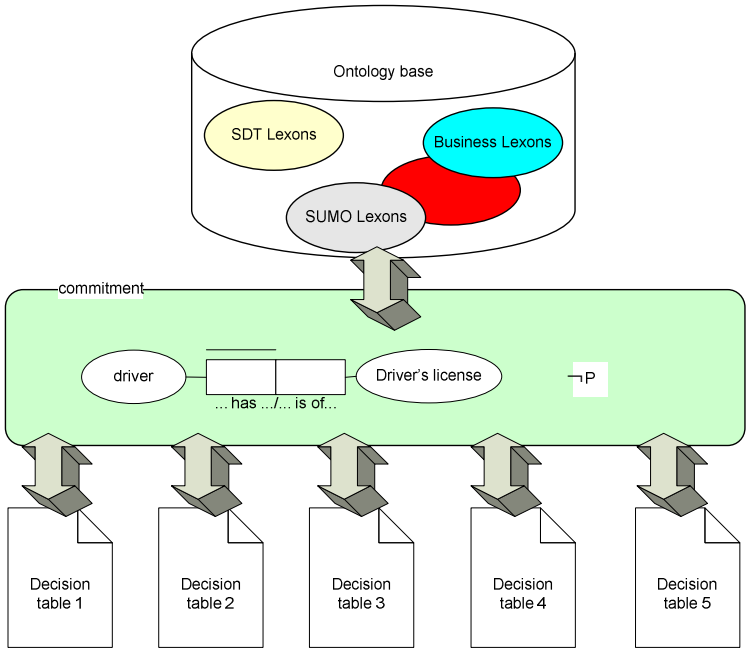



**Figure 97: Semantic Decision Table model**

SDT provides a means to capture and examine decision makers' concepts, as well as a tool for refining their knowledge and facilitating knowledge sharing in a scalable manner. An SDT is the result of annotating a (set of) decision table(s) (or any well structured decision resources) with (domain) ontologies. It is modeled based on the framework of Developing Ontology-Grounded Methods and Applications (DOGMA, section 6.1.3.1) (Figure 97).

We have designed a methodology to assist a decision group to create SDTs. The methodology is illustrated inFigure 98. As the first step, which is in the phase of *preparation*, users need to study the decision maker candidates individually in order to establish a proper decision group with the required abilities. The decision problems and the environment inputs are also defined in this phase. We take a case in the human resource management (HRM) domain as an example to scope the decision problems. We chose a relevant HRM domain ontology as one of the environment inputs. The group

candidates are required to know the company policies and the problem of hiring new employees and training current employees. Below are the sub-activities involved in this phase:

- *Study decision maker individuals*. One needs to study the relevant background information of every decision maker, mainly based on the requirement of the decision making problems. This sub-activity is to form a proper group for SDT construction.
- Define environment input. The input of the environment are the physical or nonphysical factors that interact with decision makers and the decision making problems, i.e., the decision making software (nonphysical) that will be used to assist the decision making, the input and the output of the software, the network of PC's (physical), and the type of domain ontology.
- Scope decision problems. One should scale down the problem domain in order to reach the final goal more easily. This activity directly influences the quality of the final result as it defines the context and the scope of the decision making. The decision problem needs to be described and clarified in a document.

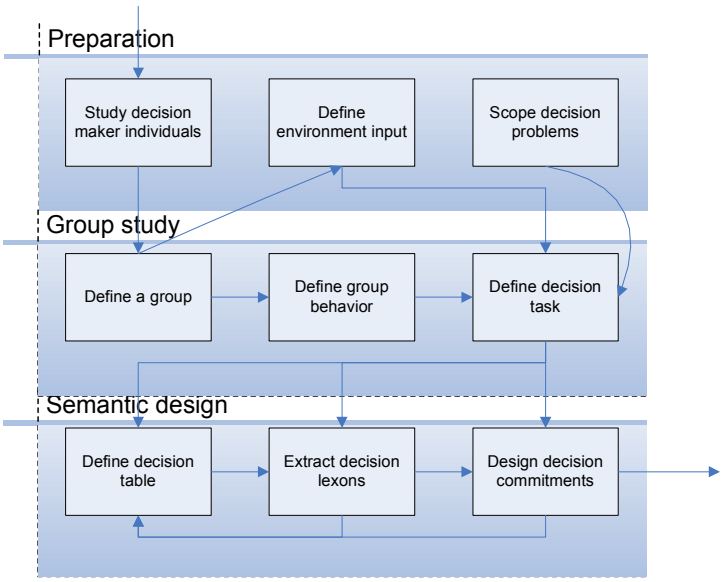



**Figure 98: SDT construction methodology**

In the phase of group study, the decision problems are scoped (divided) into several more detailed and more focused decision tasks. A decision task specifies condition stubs and action stubs of a decision table. Below are the sub-activities that support this process.

- Define a group. This sub-activity is to define a standing group. a standing decision group is composed of a set of decision maker's profiles, each of which corresponds to a decision maker. It is an activity of selecting/filtering group members. If the profiles of decision makers overlap others', it is recommended to choose the most suitable ones amongst them. The static structure of decision maker's profiles is also defined during the execution of this activity.
- Define thresholds for group behavior. The group behavior is defined in a structured manner. We often use voting as the group behavior setting for the final decision making. Common voting types are as follows.
    - Single vote. The voter can select one candidate. It precludes him voting for anyone else.
    - Multiple votes. The voter can vote for several candidates.
    - Ranked vote. The voter has to rank the alternatives in order of preference.
    - Scored vote. The voter gives each alternative a number within a certain range.
- Define decision task. A decision problem can be further divided into several tasks that need to be accomplished within a predefined period of time. They are executed sequentially or in parallel, depending on the situation. The result of this activity leads to a context identifier, a set of condition stubs and a set of action stubs of the final semantic decision table.

With regard to the technical issues of SDT, Semantic Decision Rule Language (SDRule-L/SDRule-ML) and Decision Commitment Language (DECOL) are designed and implemented to model, store, reason and publish SDT rules.

The activities of extracting SDT lexons and SDT commitments are actually the activities of creating application ontology, the methodology of which we refer to section 6.1.3.2.

### 6.1.3.4 Challenges of OE in DIY-SE

A requirement from the project call is described as below:

> "What is called Web 2.0 or social software is in reality a wave of individualization and democratization of the Internet, which promotes user participation. Good examples are the so-called social networks, such as myspace.com and friendster.com, but also personal Web blogs and the contribution to common content management platforms such as Wikis. In all cases the social phenomenon is based on increased and simplified user participation. In social networks it is easy to discover new friends and easy to express characteristics about oneself or opinions about others.
>
> Again, problems of scale occur which require the use of additional intelligence to be solved.
>
> One of the gravitational principles of Web 2.0 is that we can see the Web as a platform. Going from the 'traditional' Web to the current Web 2.0 has changed the way we perceive and interact with the Web and its content. With the advent of Web 2.0, the style of communication has changed completely and there now exists a broad 2-way channel. Moreover, each resource in the Web is a vehicle for community interaction."

The DIY-SE project is not limited to web services. We propose a component concept with layered functionality (and templates) supporting interactivity, we consider multi-modal input and output, and don't restrict ourselves to the traditional multi-media use-cases. Indeed, the DIY-SE project also addresses sensors, actuators, smart objects, media devices at home (TV, ambient lighting), public devices (public signage etc.). It combines all these into a DIY application creation environment for non-technical users.

Ontologies can be served as the knowledge repository to be used by DIY-SE sensors, actuators, smart objects, media devices and so forth. A generic ontology for ambient home does not exist, which becomes the **first challenge** of Ontology Engineering (OE) in this project.

DOGMA-MESS has been used in Co-Drive project[5]. Interesting results have been accumulated. Current DOGMA-MESS system involves domain experts and knowledge engineers. It is interesting to involve the end users of ontology-based applications because they can be considered as the concept providers. And, these concepts would be very interesting and valuable for enriching the DIY-SE ontologies, How to systematically collect and filter their concepts in DOGMA-MESS is the **second challenge** of OE in the DIY-SE project.

With regard to the DIY-SE mash-up applications, ontology-based search will play an important role for finding useful application components. How to design and implement such a DIY-SE ontology-based search engine is the **third challenge**. In particular, below issues concerning ontology-based search need to be addressed in the DIY-SE project.

- The ontology-based search engine needs to support users to *iteratively* refine user's query until he finds the existing solutions he wants.
- The ontology-based search engine needs to serve for two kinds of people for DIY: 1) experienced users (geeks and nerds), 2) non-experienced users. For the non-experienced users, *to hide system complexity* is necessary.
- The ontology-based search engine should not be made as a mandatory component. Users are the ones who *independently* decide to use it or not.

In DIY-SE environment, the smart objects need to communicate with each other. To have domain ontology is not enough. It is also necessary to have rules based on the ontology, such as "If a person

[5] http://www.codrive.org/

is crying, then he must be hungry. Therefore, the alert light for nurses must be turned on". As discussed, semantic decision tables (SDT) are a user friendly manner to create decision rules based on domain ontologies. The **last challenge** could be to how to use semantic decision tables for non-technical end users to create rules, store them and share them.

### 6.1.4 Concepts related to context awareness

"Being Human: Human Computer Interaction in the year 2020" [83] is a report that tries to capture the dynamic interaction between users and computers focusing on a 10 year lead. However, one year after its publication, it appears that the authors may have been too conservative on the development of this evolution towards the richness of the context interactions. Context can be defined as any information that can be used to characterise the situation of an entity at a given time. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

In these rich interaction environments, services can no longer be predefined in detail at design time, but need to be seamlessly linked to user and execution environment context, in order to provide service centric application able to behave accordingly either reactively (i.e. a context change has been detected and the service adapts accordingly its behaviour) or pro-actively (i.e. the service detects in advance something the user is not aware of and proposes to adapt its behaviour accordingly). The requirements for future architectures that can support these environments were labelled by research bureau Gartner as the "**Context Driven Architecture**" (CoDA) [84] . This CoDA approach can be seen as a required next step on the SOA evolution path.

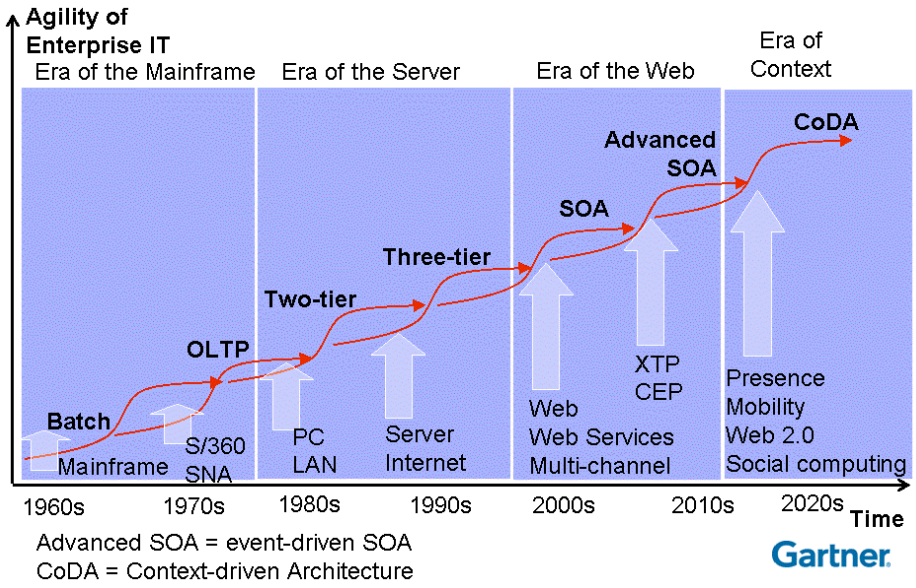



**Figure 99: Software Architecture in the Context of History (Gartner ©)**

The CoDa objective is the support a continuously changing ecosystem of evolving processes in rich dynamic contexts. However, the current specification, initiatives, standards and tools are not well suited to support such environments [85] .
In order to accommodate to this SOA evolution, context-awareness and user-awareness [86] stand out as revolutionary concepts in Future Internet, which is more user oriented rather than system oriented.

**Context-awareness** is fundamental for future service centric applications in order to provide rich and consistent experiences for the users, taking into account the information extracted from the context in each moment. Context-awareness can de defined as the ability of capturing and processing context information.

**User-awareness** can be described as the ability to address users no matter the terminal, communication devices or location, opening up a vast new field of possibilities, where devices are able to extract user and context required information (like, for example, the identity or location). Here, we

can talk about profiling, which consists of a first stage of collecting user information everywhere and anytime. The collection data and their nature are highly dependent on the type of user and its expectative and necessities.

So, Context-Aware Computing can be understood as a subset of Ubiquitous Computing which aims to make surrounding computing systems aware of the users' current situation. Its main goal is to feed surrounding reactive system with users' and devices' context, so that they can use such information as clues to infer suitable reactions and adaptations of the environment and, hence, for optimizing the use of resources and also the execution of composition. For doing so, systems need to have the intelligence enough to understand these context and user information combined with interfaces with learning capabilities and adaptation based on the context and situation.

The following **challenges** and research priorities are identified in this domain:

- Develop techniques based on collecting interesting information from users, for example, developing a profiling component that will collect the characterization of the user's personality, patterns and specific behaviour. This profile will be crucial in order to adapt the environment and applications to users' requirements and expectations.
- Develop techniques of auto-configuration and adaptation via composition depending on circumstances and user profiles and requirements, providing enough transparency for users to be aware only on the user-part of the services they require. Also exploitation of workflow-related technologies for automating composition decisions and management.
- Create a flexible context-modelling framework with efficient means of presenting, maintaining, sharing, protecting, reasoning and querying device, user and network context information.
- Provide service centric applications easily accessed and used by the technology-agnostic user.

The context management is decomposes in two main steps. First of all, the characteristics of real world are abstracted conceptually, and then those concepts, which define the contextual information, are represented by a context model. The existence and usage of an efficient context model and user profiling component is a key factor in designing context aware services.
The techniques and mechanisms required to perform the context abstraction and representation are the following ones:

- **Context discovery**: Locates and accesses contextual sources.
- **Context acquisition**: Enables the acquisition of contextual information from diverse context sources.
- **Context modelling**: Abstraction and representation of contextual information for further processing.
- **Context aggregation**: Includes the combination and aggregation of different context information to derive new knowledge
- **Context reasoning**: Elaboration of context with reasoning mechanisms.

#### 6.1.4.1 Current initiatives, projects and results

The approaches to be used to abstract the characteristics of real world conceptually can be classified into two [87] :

- **Context theoretic modelling**: This approach aims at modelling the contextual information primarily as situations; and the contextual information changes as actions.
- **Context contextual modelling**: This approach aims at describing context as concepts and the relations among such concepts.

After abstracting the context, this data has to be implemented and represented by a context model as structured information. So, there are also a set of approaches that define how to implement a context model [88] [89] :

- **Key-value models** are the most simple data structures associating context characteristics (defined as a key) with specific values of contextual information. The disadvantage of this model is the lack of efficiency for sophisticated structuring purposes.
- **Graphical models** represent contextual entities and their relationships graphically. This model is particularly useful for structuring. Some examples are Unified Modeling Language (UML) [90] and Object-Role Modeling (ORM) [91] .

- **Mark-up scheme models** integrate the model schema and values using mark-up tags to define hierarchical data structures. This approach is often used to represent entities profiles using standards such as Composite Capabilities / Preferences Profile (CC/PP) [92] and User Agent Profile (UAProf) [93] to cover the high dynamics of contextual information. Another example is SGML [94] .
- **Object-oriented-models** consist of encapsulating contextual information into objects, emphasizing reusability and controlled access to contextual information. The information can only be accessed through well defined interfaces and is therefore hidden to from other objects. Examples of this model can be found in Cues (TEA project) [95] and Active Object Model (GUIDE project) [96] .
- **Logic-based models** represent a highly formal modeling approach where context is defined as facts, expressions and rules. One of the first logic based context modeling approaches was published as *Formalizing Context* in early 1993 by McCarthy and his group at Stanford [97] . Another early representative of this kind of approach is the *Extended Situation Theory* by Akman and Surav [98] .
- **Ontology-based models** use ontologies, which are used to represent concepts and relations between concepts. It is high and formal expressiveness and possibility for applying ontology reasoning techniques, enabling contextual knowledge sharing and reuse. Some examples are the Context Ontology Language (CoOL) [99] and CONtext ONtology (CONON) [100] . A promising emerging context modeling approach based on ontologies is the CoBrA system [101] .This system provides a set of ontological concepts to characterize entities such as persons, places or several other kinds of objects within their contexts.

#### 6.1.4.2 Conclusion

In [92] the authors evaluate the appropriateness of the different context modeling approaches for ubiquitous computing and the identified particular requirements. The conclusion is that the most promising approach for context modeling can be found in the ontology-based models category. However, it is important to remark that this conclusion does not mean that the other approaches are unsuitable for ubiquitous computing environments.

### 6.1.5 Concepts related to multi-modal interaction

Multimodal interfaces allow users to interact with computers using multiple different modes of interaction (e.g. voice interaction, gestural interaction or/and tactile interaction). The aim of the multimodal interfaces is to enable more natural and effective human-computer interaction by allowing users to interact using whichever mode, or combination of modes, are most appropriate given the situation and their preferences and abilities (e.g. it is natural to ask questions orally, while writing them it isn't).

This implies that the multimodal interfaces have the benefit of the use easiness. But there is another benefit that the multimodal interfaces imply: system robustness. Due the use of different modes of interaction, it is possible to have redundant commands. With them, if the preferred mode of interaction fails in a certain context, the system may switch to a more appropriate mode or it can combine modalities in order to obtain lost information.

Multimodal interfaces have been shown to have many advantages [57] : they prevent errors, bring robustness to the interface, help the user to correct errors or recover from them more easily, bring more bandwidth to the communication, and add alternative communication methods to different situations and environments.

Different modes are best suited for different kinds of input or outputs. For example, it is easier for a user to select among millions of names in a directory by saying the name they are interested in rather than searching in a huge menu, but if the user only has to choose among say three or four options, it is easier to click a button in a graphical user interface than to use speech.

It should be noted, however, that multiple modalities alone do not bring benefits to the interface: the use of multiple modalities may be ineffective or even disadvantageous. In order to accomplish a human-like multimodal analysis of multiple input signals acquired by different sensors, the signals cannot be always considered mutually independently and those input signals might not be combined

in a context-free manner at the end of the analysis. So, the final target of a multimodal interface have to be the fusion of the different information obtained from the different interaction modes in order to create the best response to the user.

Many fusion approaches have been developed. There are different approaches which depends on the moment of the signal integration:

- Integration at the *feature level* (early fusion)
- Integration at a *higher semantic level* (late fusion)
- *Something in between* (intermediate fusion)

Early multimodal interfaces were based on a specific control structure for multimodal fusion. For example, Bolt's "Put-That-There" system [58] combined pointing and speech inputs and searched for a synchronized gestural act that designates the spoken referent.

To support more functional systems, general processing architectures have been developed which handle a variety of multimodal integration patterns and support joint processing of modalities. The most common infrastructure that has been adopted by the multimodal research community involves multi-agent architectures such as the *Open Agent Architecture [59]* and *Adaptive Agent Architecture [60]* [61] . Multi-agent architectures provide essential infrastructure for coordinating the many complex modules needed to implement multimodal system processing and permit this to be done in a distributed manner. In a multi-agent architecture, the components needed to support the multi-modal system (e.g., speech recognition, gesture recognition, natural language processing, multimodal integration) may be written in different programming languages, on different machines, and with different operating systems. Agent communication languages are being developed that handle asynchronous delivery, triggered responses, multi-casting, and other concepts from distributed systems.

When using a multi-agent architecture, for example, speech and gestures can arrive in parallel or asynchronously via individual modality agents, with the results passed to a *facilitator*. Next, sets of meaning fragments derived from the speech, or other modality, arrive at the multimodal *integrator* which decides whether and how long to wait for recognition results from other modalities, based on the system's temporal thresholds. It fuses the meaning fragments into a semantically and temporally compatible whole interpretation before passing the results back to the facilitator. At this point, the system's final multimodal interpretation is confirmed by the interface, delivered as multimedia feedback to the user, and executed by the relevant application.

#### 6.1.5.1 State of the art

In the last two decades there has been a huge research activity within multimodal user interfaces.
In 1980 Bolt [58] presented the "Put That There" concept demonstrator, which processed speech in parallel with manual pointing during object manipulation. Since then major advances have been made in speech recognition algorithms and natural language processing, in handwriting and gesture recognition, as well as in speed, processing power and memory capacity of the computers. Today's multimodal systems are capable of recognizing and combining a wide variety of signals such as speech, touch, manual gestures, gaze, head and body movements.
These advanced systems need various sensors and cameras and a lot of processing power and memory. They are therefore best suited for kiosk applications. Following, there are an overview of various multimodal systems.

Zhang Q. et al. propose a multimodal pointing technique combining eye gaze and speech inputs [62] . The technique was tested in a user study on pointing at multiple targets. The powerful of the combination eye gaze – speech recognition and particularly the use of gaze as a disambiguation channel in speech was highlighted also by Zhang Q., Imamiya A., Go K. and Gao X. ("Overriding errors in speech and gaze multimodal architecture", Proceedings of the 9th International Conference on Intelligent User Interfaces – 2004).

Other project dealing with multimodal interaction is "eyeCOOK", which is presented as a multimodal attentive cookbook to help a non-expert computer user cook a meal [63] .The user communicates using eye-gaze and speech commands, and eyeCOOK responds visually and/or verbally, promoting communication through *natural* human input channels without physically encumbering the user.

EyeCOOK was designed specifically to use *natural* input modalities, in order to reduce the need for users to provide explicit input, or change their behaviour to accommodate interface constraints. So, this approach improves the learnability and intuitiveness of interfaces designed for novice users.

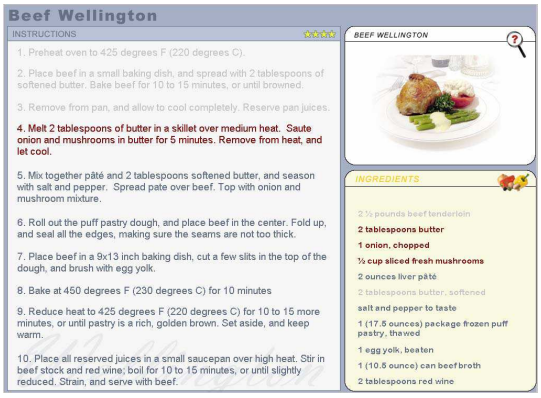



**Figure 100: eyeCOOK visual interface**

As voice commands, eyeCOOK uses context-sensitive, localized grammars, which allows more synonyms for a given speech recognition command, reducing the chance of misinterpreting a word. The multimodality appears when the user is in range of the eye tracker. In this moment eyeCOOK substitutes the object of the user's gaze for the word 'this' in a speech command. For example, 'Define this' will trigger the define operation on the current eye gaze target. Finally, this system had several Temperature sensors. Those were used to keep track of the status of the oven and the elements of the stove, so the system can increase it's ability to guide the user's cooking experience.

There are other areas in which the multimodal interaction is useful, for example in the video area, as MVIEWS shows [64] . It is a system for annotating, indexing, extracting, and disseminating information from video streams for surveillance and intelligence applications. An analyst watching one or more live video feeds is able to use pen and voice to annotate the events taking place. The annotation streams are indexed by speech and gesture recognition technologies for later retrieval, and can be quickly scanned using a timeline interface, then played back during review of the film. Pen and speech can also be used to command various aspects of the system, including image processing functions, with multimodal utterances such as "Track this" or "If any object enters this area, notify me immediately."

A multimodal input approach can be used in a smart home environment, that is expected to help people with physical disabilities to control their living environment [65] . The system is based on EIB home automation network, and as a human-computer interface a genuine Communication and Control Device for Disabled is used (CCDD). This device enables a disable user to control living environment and communicates with others using two kind of interfaces: On/Off-like and keyboard. The first kind are the on/off devices which any switch type device. For the presentation purposes, the CCDD has 2 channels. First one is a 4 line LCD with backlight and the second one is a simple sound system.

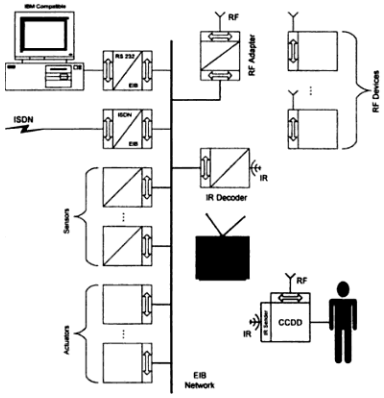



**Figure 101: Smart Home proposed configuration**

D. Chen et al. worked in a project related to multimodal system for detecting human activity and interaction patterns is proposed [66] . Activities of groups of people are firstly treated as interaction patterns between any pair of partners and are then further broken into individual activities and behaviour events using a multi-level context hierarchy graph. They detect candidate interactions among a group of people by fusing audio and video channels. A camera network installed records both video and audio.

**Figure 102: Examples of social interactions in video**

For the fusion of the information, they propose a system which detects interactions in the video stream and in the audio stream. Those detected interactions have a confidence degree, so the interaction in one moment is considered when the linear combination of those two confidence degrees is more than 0,5. With this detection, the next step is to fine detection of the interaction. In this process the kind of interaction is detected using a multi-level context hierarchy.

Another use of the multimodal inputs is an intelligent design studio [67] . As the authors say, the absence of software support for early-stage design suggests that conventional graphical user interfaces are inadequate for this task. But natural multimodal interaction demands a better understanding of how designers use sketching in combination with other modalities to communicate ideas about design.

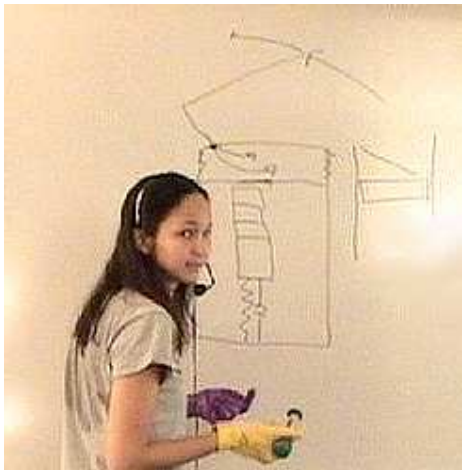


**Figure 103: Proposed design method using drawing, speech and gestures**

Their exploratory studies had shown that speech, sketching and gestures are the main modalities of interacting in the designing process, and they use them in this way.

Finally, there is a huge application scenario in the mobile environment. For example, in [68] , the authors present two techniques to control a mobile device. The first is a 3D audio radial pie menu that

uses head gestures for selecting items. An evaluation of a range of different audio designs showed that egocentric sounds reduced task completion time, perceived annoyance, and allowed users to walk closer to their preferred walking speed. The second is a sonically enhanced 2D gesture recognition system for use on a belt-mounted PDA. An evaluation of the system with and without audio feedback showed users' gestures were more accurate when dynamically guided by audio-feedback.

### 6.1.5.2 Standarization

As the application fields can be enormous such as ports and airports, tourism, police, office, security, banks, factory plants… it could be really interesting to standardize some kind of interface to make it easier the information exchange and collaboration. Nowadays all the systems are connected to a network, and generally to the Internet, so that standardization have to take into account this vision.

This whish pushed to the World Wide Web Consortium (W3C) to create a Multimodal Interaction Working Group as part of its Multimodal Interaction Activity. The mission of this group is to develop open standards that enable the following vision:

- Extending the Web to allow multiple modes of interaction:
  - GUI, Speech, Vision, Pen, Gestures, Haptic interfaces, ...
- Anyone, Anywhere, Any device, Any time
  - Accessible through the user's preferred modes of interaction with services that adapt to the device, user and environmental conditions

The primary goal of this group is to develop W3C Recommendations that enable multimodal interaction with various devices including desktop PCs, mobile phones and less traditional platforms such as cars and intelligent home environments. For rapid adoption on a global scale, it should be possible to add simple multimodal capabilities to existing markup languages in a way that is backwards compatible with widely deployed devices, and which builds upon widespread familiarity with existing Web technologies. The standards should be scalable to enable richer capabilities for subsequent generations of multimodal devices.

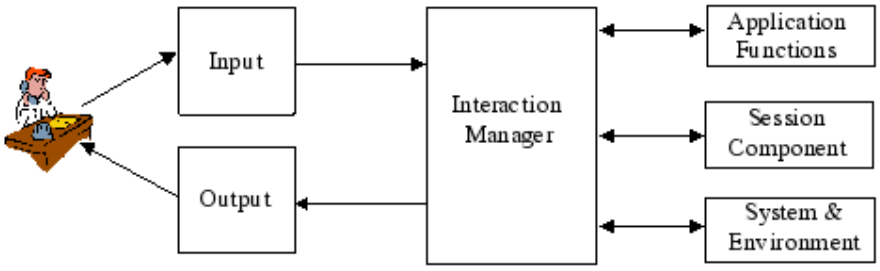



**Figure 104: The W3C basic multimodal interaction framework (source: [69] )**

This Working Group had developed the W3C interaction framework in order to define the multimodal interaction architecture.

- *Human user — A user* who enters input into the system and observes and hears information presented by the system.
- *Input* — An interactive multimodal implementation will use multiple input modes such as audio, speech, handwriting, and keyboarding, and other input modes.
- *Output* — An interactive multimodal implementation will use one or more modes of output, such as speech, text, graphics, audio files, and animation.
- *Interaction manager* — The interaction manager is the logical component that coordinates data and manages execution flow from various input and output modality component interface objects.
- *Session component* — The Session component provides an interface to the interaction manager to support state management, and temporary and persistent sessions for multimodal applications.

- *System and Environment component* — This component enables the interaction manager to find out about and respond to changes in device capabilities, user preferences and environmental conditions.

This framework also identifies three important blocks inside the inputs, which allows the correct function of the multimodal interaction. In the following, those blocks are described.

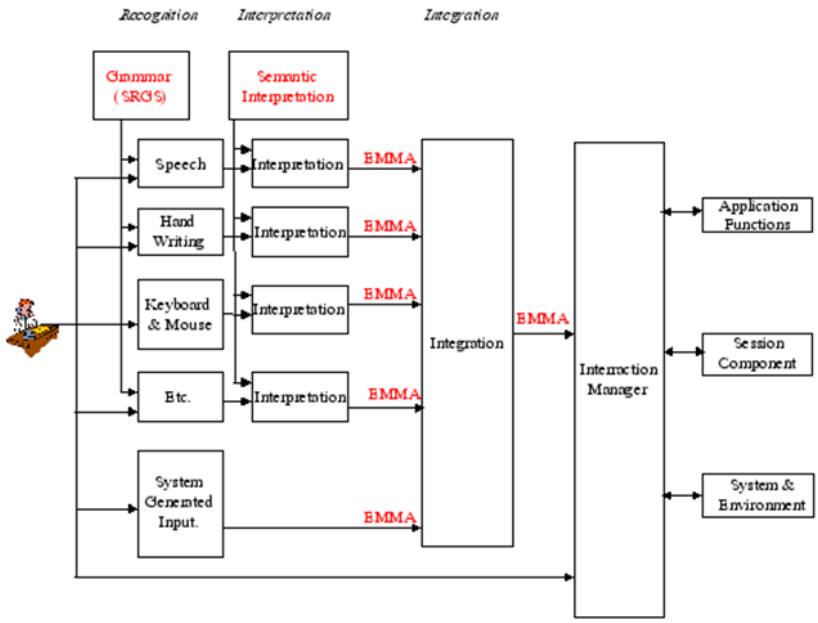



**Figure 105: The W3C Multimodal interaction framework's input components (source: [69] )**

- *Recognition component* — Captures natural input from the user and translates the input into a form useful for later processing. For example, some of these recognition components can be: Speech recognition, handwriting recognition, gesture recognition.

- *Interpretation component* — This component process the results of the recognition components and identifies the "meaning" or "semantics" intended by the user. For example, many words that users utter such as "yes," "affirmative," "sure," and "I agree," could be represented as "yes."
- *Integration component* — Combines the output from several interpretation components.

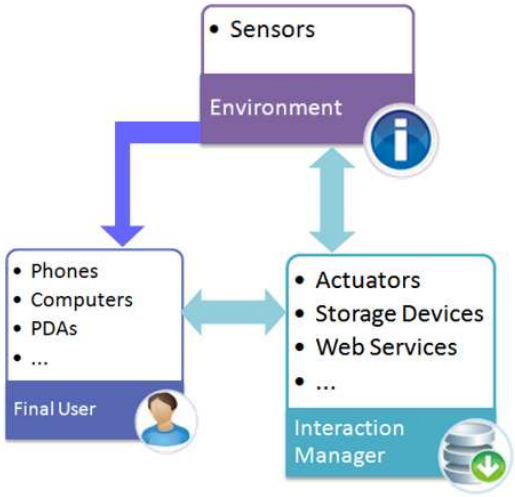
The future milestones of the Multimodal Interaction Working Group includes the definition of the following W3C Recommendations:

- Multimodal Architecture and Interfaces (MMI Architecture)
- Extensible Multi-Modal Annotations (EMMA)
- InkML - an XML language for ink traces
- EmotionML

### 6.1.6 Concepts related to the interaction with the environment

As shown throughout this document, the recent advances in computer science and electrical engineering are radically changing the way we interact with our physical environment, whether we are talking about classic computers (PCs, Macs, phones, PDAs, etc.) or about the new paradigm that is the ubiquitous computing, where devices are not only computers but also objects from our daily lives (TV, domestic equipments, sensors, actuators, RFIDs, etc.). These objects are now able to automatically produce localized and relevant data in a specific area and will increasingly expand users' ability to remotely interact with the real world wherever they are.

| | Past | Present | Future |
|---|---|---|---|
| Size | Large shoe box | Pack of cards | Dust particle |




| | | | |
|---|---|---|---|
| Weight | Kilograms | Grams | Negligible |
| Applications | Military | Some industries & Academic testing | Everywhere |
| Power | Large batteries | AA / Lithium-ion Batteries | Hybrid (Solar + AA/Li-ion) |

**Table 7 : Evolution of Sensor Devices**

The table above shows the technical advances of distributed sensing devices during the twenty last years. It is clear that the next substantial evolution of the human-machine interactions will arise from this incursion of the technology everywhere in our daily lives. ITS (Intelligent Transportation Systems), HANs (Home Area Networks), BANs (Body Area Networks) or IoT (Internet of Things) are few illustrations of the current trends in the information and communication technologies (ICTs).

**Environment-awareness** is the key new feature that will distinguish the future of ICTs from the actual terminal-dependent approach. Indeed, this environment "consciousness" will allow users better and more efficient interactions with the machines via different interface technologies: RFID, Sensors, Actuators, etc.

Note that environmental awareness cannot be achieved without **interface technologies** which link the environment to clients (users or devices). These interfaces are sensors and actuators; they are the only devices that can gather information about the physical conditions of the environment. Depending on the application, sensors can be connected either with wired or wireless communication mediums. They can also form a hybrid network with actuators combining wired and wireless interconnections.

Environment interactions can be categorized in two classes:

1) **Synchronous interactions:** In synchronous systems, the user gets regularly information about his surrounding environment via either sensors or actuators that transmit their sensed information to the client each explicit period of time.

2) **Asynchronous interactions**

   a) **On-Demand interactions:** An on-demand interaction mode implies that the user interrogates the system each time he is looking for information about his environment. RFID technologies are good example of this approach.

**b) Event-Centric interactions:** In event-centric systems, the environment alerts the user each time an event occurs. This concept is already used in Emergency Monitoring Systems for example, and can be extended to different new types of services.

**c) Publish-Subscribe interactions:** Publish/Subscribe systems are quite similar to the event-centric interaction method, except that in this case users can subscribe to different sorts of events and get updates each time an event that interests the user occurs. This kind of interactions can unveil several application possibilities using sensors, actuators and other smart objects.

To achieve this interaction with the environment, sensors and actuators go through different steps:

- **Data acquisition**: Sensors gather information about the environment
- **Data processing**: Sensors may operate data processing (compression, aggregation, coding, etc.) to the collected data.
- **Data Transmission**: Depending on the class of interaction, sensors send the collected data to the destination (user, interaction manager, etc.)

# 7 State of the Art of DIY ecosystem and business model

## 7.1 Business model

A business model is the method of doing business and generates revenue. Classically, on a micro-level,a business model spells-out how a company makes money by specifying where it is positioned in the value chain. More recently, howver, literature extends the notion of BM to the whole value network (on meso level), where the focus shifts from single firms to networks of firms ([220]

Some models are quite simple, a company produces a good or service and sells it to customers. Other models are more complicated, television programming has been broadcasted over the airwaves free to anyone with a receiver for much of the past century. The broadcaster is part of a complex network of distributors, content creators, advertisers (and their agencies), and listeners or viewers. Who makes money and how much is not always clear at the outset.

Internet commerce has created new kinds of business models. But the web is also reinventing old models as Auctions, one of the oldest forms of brokering; the Web has popularized the auction model within a wide array of goods and services (eBay).
Internet Business models have been defined and categorized in many different ways and they continue to evolve, so the following table coming from **Michael Rappa**, director of the Institute for Advanced Analytics at North Carolina State University, can vary through the next years.

### 7.1.1 Brokerage Model

Brokers are market-makers: they bring buyers and sellers together and facilitate transactions. Brokers play a frequent role in business-to-business (B2B), business-to-consumer (B2C), or consumer-to-consumer (C2C) markets. Usually a broker charges a fee or commission for each transaction it enables. The formula for fees can vary. Brokerage models include:

- **Marketplace Exchange** -- offers a full range of services covering the transaction process, from market assessment to negotiation and fulfillment. Exchanges operate independently or are backed by an industry consortium. [Orbitz, ChemConnect]
- **Buy/Sell Fulfillment** -- takes customer orders to buy or sell a product or service, including terms like price and delivery. [CarsDirect, Respond.com]
- **Demand Collection System** -- the patented "name-your-price" model pioneered by Priceline.com. Prospective buyer makes a final (binding) bid for a specified good or service, and the broker arranges fulfillment. [Priceline.com]
- **Auction Broker** -- conducts auctions for sellers (individuals or merchants). Broker charges the seller a listing fee and commission scaled with the value of the transaction. Auctions vary widely in terms of the offering and bidding rules. [eBay]
- **Transaction Broker** -- provides a third-party payment mechanism for buyers and sellers to settle a transaction. [PayPal, Escrow.com]
- **Distributor** -- is a catalog operation that connects a large number of product manufacturers with volume and retail buyers. Broker facilitates business transactions between franchised distributors and their trading partners.
- **Search Agent** -- a software agent or "robot" used to search-out the price and availability for a good or service specified by the buyer, or to locate hard to find information.

**Virtual Marketplace** -- a hosting service for online merchants that charges setup, monthly listing, and/or transaction fees. May also provide automated transaction and relationship marketing services. [zShops and Merchant Services at Amazon**.com**]

### 7.1.2 Advertising Model

The web advertising model is an extension of the traditional media broadcast model. The broadcaster, in this case, a web site, provides content (usually, but not necessarily, for free) and services (like email, IM, blogs) mixed with advertising messages in the form of banner ads. The banner ads may be the major or sole source of revenue for the broadcaster. The broadcaster may be a content creator or

a distributor of content created elsewhere. The advertising model works best when the volume of viewer traffic is large or highly specialized.

- **Portal** -- usually a search engine that may include varied content or services. A high volume of user traffic makes advertising profitable and permits further diversification of site services. A personalized portal allows customization of the interface and content to the user. A niche portal cultivates a well-defined user demographic. [Yahoo!]
- **Classifieds** -- list items for sale or wanted for purchase. Listing fees are common, but there also may be a membership fee. [Monster.com, Craigslist]
- **User Registration** -- content-based sites that are free to access but require users to register and provide demographic data. Registration allows inter-session tracking of user surfing habits and thereby generates data of potential value in targeted advertising campaigns. [NYTimes]
- **Query-based Paid Placement** -- sells favorable link positioning (i.e., sponsored links) or advertising keyed to particular search terms in a user query, such as Overture's trademark "pay-for-performance" model. [Google, Overture]
- **Contextual Advertising / Behavioral Marketing** -- freeware developers who bundle adware with their product. For example, a browser extension that automates authentication and form fill-ins, also delivers advertising links or pop-ups as the user surfs the web. Contextual advertisers can sell targeted advertising based on an individual user's surfing activity.
- **Content-Targeted Advertising** -- pioneered by Google, it extends the precision of search advertising to the rest of the web. Google identifies the meaning of a web page and then automatically delivers relevant ads when a user visits that page. [Google]
- **Intromercials** -- animated full-screen ads placed at the entry of a site before a user reaches the intended content. [CBS MarketWatch]

**Ultramercials** -- interactive online ads that require the user to respond intermittently in order to wade through the message before reaching the intended content. [Salon in cooperation with Mercedes-Benz]

### 7.1.3 Infomediary Model

Data about consumers and their consumption habits are valuable, especially when that information is carefully analyzed and used to target marketing campaigns. Independently collected data about producers and their products are useful to consumers when considering a purchase. Some firms function as infomediaries (information intermediaries) assisting buyers and/or sellers understand a given market.

- **Advertising Networks** -- feed banner ads to a network of member sites, thereby enabling advertisers to deploy large marketing campaigns. Ad networks collect data about web users that can be used to analyze marketing effectiveness. [DoubleClick]
- **Audience Measurement Services** -- online audience market research agencies. [Nielsen//Netratings]
- **Incentive Marketing** -- customer loyalty program that provides incentives to customers such as redeemable points or coupons for making purchases from associated retailers. Data collected about users is sold for targeted advertising. [Coolsavings]
- **Metamediary** -- facilitates transactions between buyer and sellers by providing comprehensive information and ancillary services, without being involved in the actual exchange of goods or services between the parties. [Edmunds]

### 7.1.4 Merchant Model

Wholesalers and retailers of goods and services. Sales may be made based on list prices or through auction.

- **Virtual Merchant** --or e-tailer, is a retail merchant that operates solely over the web. [Amazon.com]

- **Catalog Merchant** -- mail-order business with a web-based catalog. Combines mail, telephone and online ordering. [Lands' End]
- **Click and Mortar** -- traditional brick-and-mortar retail establishment with web storefront. [Barnes & Noble]

**Bit Vendor** -- a merchant that deals strictly in digital products and services and, in its purest form, conducts both sales and distribution over the web. [Apple iTunes Music Store]

### 7.1.5 Manufacturer (Direct) Model

The manufacturer or "direct model", it is predicated on the power of the web to allow a manufacturer (i.e., a company that creates a product or service) to reach buyers directly and thereby compress the distribution channel. The model is linked to the notion of “desintermediation", referring to “cutting out the middleman". The manufacturer model can be based on efficiency, improved customer service, and a better understanding of customer preferences. [Dell Computer].

- **Purchase** -- the sale of a product in which the right of ownership is transferred to the buyer.
- **Lease** -- in exchange for a rental fee, the buyer receives the right to use the product under a “terms of use” agreement. The product is returned to the seller upon expiration or default of the lease agreement. One type of agreement may include a right of purchase upon expiration of the lease.
- **License** -- the sale of a product that involves only the transfer of usage rights to the buyer, in accordance with a “terms of use” agreement. Ownership rights remain with the manufacturer (e.g., with software licensing).

**Brand Integrated Content** -- in contrast to the sponsored-content approach (i.e., the advertising model), brand-integrated content is created by the manufacturer itself for the sole basis of product placement.

### 7.1.6 Affiliate Model

In contrast to the generalized portal, which seeks to drive a high volume of traffic to one site, the affiliate model, provides purchase opportunities wherever people may be surfing. It does this by offering financial incentives (in the form of a percentage of revenue) to affiliated partner sites. The affiliates provide purchase-point click-through to the merchant. It is a pay-for-performance model -- if an affiliate does not generate sales, it represents no cost to the merchant. The affiliate model is inherently well-suited to the web, which explains its popularity. Variations include, banner exchange, pay-per-click, and revenue sharing programs. [Barnes & Noble, Amazon.com]

- **Banner Exchange** -- trades banner placement among a network of affiliated sites.
- **Pay-per-click** -- site that pays affiliates for a user click-through.
- **Revenue Sharing -- offers a percent-of-sale commission based on a user click-through in which the user subsequently purchases a product.**

### 7.1.7 Community Model

The viability of the community model is based on user loyalty. Users have a high investment in both time and emotion. Revenue can be based on the sale of ancillary products and services or voluntary contributions; or revenue may be tied to contextual advertising and subscriptions for premium services. The Internet is inherently suited to community business models and today this is one of the more fertile areas of development, as seen in rise of social networking.

- **Open Source** -- software developed collaboratively by a global community of programmers who share code openly. Instead of licensing code for a fee, open source relies on revenue generated from related services like systems integration, product support, tutorials and user documentation. [Red Hat]

- **Open Content** -- openly accessible content developed collaboratively by a global community of contributors who work voluntarily. [Wikipedia]
- **Public Broadcasting** -- user-supported model used by not-for-profit radio and television broadcasting extended to the web. A community of users support the site through voluntary donations. [The Classical Station (WCPE.org)]

**Social Networking Services** -- sites that provide individuals with the ability to connect to other individuals along a defined common interest (professional, hobby, romance). Social networking services can provide opportunities for contextual advertising and subscriptions for premium services. [Flickr, Friendster, Orkut]

### 7.1.8 Subscription Model

Users are charged a periodic -- daily, monthly or annual -- fee to subscribe to a service. It is not uncommon for sites to combine free content with "premium" (i.e., subscriber- or member-only) content. Subscription fees are incurred irrespective of actual usage rates. Subscription and advertising models are frequently combined.

- **Content Services --** provide text, audio, or video content to users who subscribe for a fee to gain access to the service. [Listen.com, Netflix]
- **Person-to-Person Networking Services --** are conduits for the distribution of user-submitted information, such as individuals searching for former schoolmates. [Classmates]
- **Trust Services --** come in the form of membership associations that abide by an explicit code of conduct, and in which members pay a subscription fee. [Truste]
- **Internet Services Providers --** offer network connectivity and related services on a monthly subscription. [America Online]

### 7.1.9 Utility Model

The utility or "on-demand" model is based on metering usage, or a "pay as you go" approach. Unlike subscriber services, metered services are based on actual usage rates. Traditionally, metering has been used for essential services (e.g., electricity water, long-distance telephone services). Internet service providers (ISPs) in some parts of the world operate as utilities, charging customers for connection minutes, as opposed to the subscriber model common in the U.S.

- **Metered Usage** -- measures and bills users based on actual usage of a service.
- **Metered Subscriptions --** allows subscribers to purchase access to content in metered portions (e.g., numbers of pages viewed). [Slashdot]

# 8 References


[1] [http://www.ibm.com/developerworks/xml/library/x-mashups.html]

[2] The Open Ajax Alliance, http://www.openajax.org/index.php

[3] http://www.pavingways.com/products/mobile-ajax/mobile-ajax-faq

[4] http://java.sun.com/javame/index.jsp

[5] http://www-01.ibm.com/software/info/mashup-center/

[6] http://pipes.yahoo.com/pipes/pipe.edit

[7] http://mashmaker.intel.com/web/

[8] http://creator.zoho.com/

[9] http://www.google.com/sites/

[10] http://www.sampa.com/

[11] http://dreamface-interactive.com

[12] http://www.jackbe.com/

[13] http://www.oblinq.com

[14] Source NFC Forum: http://www.nfc-forum.org

[15] http://ttuki.vtt.fi/smarttouch/www/?info=intro

[16] T. Van Cutsem, S. Mostinckx, E. Gonzalez Boix, J. Dedecker, and W. De Meuter. AmbientTalk: Object-oriented event-driven programming in mobile ad hoc networks. In *Proceedings of the 26th International Conference of the Chilean Computer Science Society (SCCC 2007)*, November 2007, Iquique, Chile.

[17] J. Dedecker, T. Van Cutsem, S. Mostinckx, T. D'Hondt, and W. De Meuter. Ambient-oriented programming in Ambienttalk. In Proceedings of the 20th European Conference on Object-oriented Programming (ECOOP 2006), volume 4067 of Lecture Notes in Computer Science, pages 230–254. Springer, 2006.

[18] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. ACM Computing Surveys, 35(2):114–131, 2003.

[19] Vrije Universiteit Brussel, Programming Technology Lab. What is AmbientTalk? http://prog.vub.ac.be/amop/at/introduction

[20] SPICE project, http://www.ist-spice.org/

[21] Microsoft popfly, http://www.popfly.com/

[22] R.Ennals and D.Gay, User-Friendly Functional Programming for Web Mashups, ICFP'07, Freiburg, Germany. Oct. 2007

[23] Díaz, S. Pérez,and I. Paz. 2007. Providing Personalized Mashups Within the Context of Existing Web Applications. Lecture Notes in Computer Science, Volume 4831/2007, ISBN 978-3-540-76992-7

[24] J. Wong, J. I. Hong. Making Mashups with Marmite: Towards end-user programming for the web. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York: NY, pp 1435-1444.

[25] J.Soriano, etal, Fostering Innovation in a Mashup-oriented Enterprise 2.0 Collaboration Environment.

[26] Juan C. Yelmo, A USER-CENTRIC SERVICE CREATION APPROACH FOR NEXT GENERATION NETWORKS.

[27] X.-S. Hua, H.-J. Zhang, Automatic Home Video Editing, Signals and Communication Technology, Springer, 2009, pp. 353-386.

[28] [X.-S. Hua, H.-J. Zhang,, AVE - Automated Home Video Editing, ACM MM, 2003.

[29] Ma, Y. F., Lu, L., Zhang, H. J., Li, M. J. A User Attention Model for Video Summarization. *ACM MM 2002*, 533–542.

[30] S. Müller Arisona, P. Müller, S. Schubiger-Banz, M. Specht2, Computer-Assisted Content Editing Techniques for Live Multimedia Performance, R. Adams, S. Gibson, and S. Müller Arisona (Eds.): DAW/IF 2006/2007, CCIS 7, pp. 199–212, 2008.

[31] Y. Takemae, K. Otsuka, J. Yamato, automatic Video Editing System Using Stereo-Based Head Tracking for Multiparty Conversation, *CHI 2005*, 2005, pp. 1817-1820.

[32] Y. Takemae, K. Otsuka, J. Yamato, Development of Automatic Video Editing System Based on Stereo-Based Head Tracking for Multiparty Conversations, IEEE, 2005

[33] *C. L. Mudhwuchutyula*, M.* S. *Kankunhalli, P. Mulhem*, Content Based Editing of Semantic Video Metadata, IEEE International Conference on Multimedia and Expo, 2004

[34] Foote, J., Cooper, M., and Girgensohn, A. Creating Music Videos Using Automatic Media Analysis. ACM MM 2002.

[35] G. Ahanger, T. D.C. Little, Automatic Composition Techniques, IEEE TRANSACTIONS ON KNEWLEDGE AND DATA ENGINEERING, VOL. 10, NO. 6,1998

[36] Cinelerra, http://cinelerra.org/

[37] Kdenlive, http://kdenlive.org/

[38] Blender, http://www.blender.org/

[39] Pitivi, http://www.pitivi.org/

[40] Avidemux, http://www.avidemux.org/

[41] Jahshaka, http://jahshaka.org/

[42] Kino, http://kinodv.org/

[43] http://www.rifftrax.com/cuts

[44] Jaycut, http://jaycut.com/

[45] http://www.moviemasher.com/

[46] Automated Workflows. http://www.automatedworkflows.com/

[47] Automator Programming Guide, Apple Developer Connection. http://developer.apple.com/mac/library/documentation/AppleApplications/Conceptual/AutomatorConcepts/Automator.html

[48] Automator in Leopard. http://automator.us/leopard/index.html

[49] Automator "Run Web Service" Parameters, Jonathan Rentzsch. http://rentzsch.com/macosx/automatorRunWebServiceParameters

[50] HyperCard Programming, Fred Stauder. MacTech volume 3, issue 10. Available on http://www.mactech.com/articles/mactech/Vol.03/03.10/HyperCardProgramming/index.html

[51] HyperCard: What Could Have Been, Wired. http://www.wired.com/gadgets/mac/commentary/cultofmac/2002/08/54370

[52] www.amieproject.com

[53] www.ctolab.alcatel.be:7070/display/AmbientMedia/Assisted+Living

[54] www.in-ham.be/index.cfm?n01=news&newsid=68

[55] Easy Interactions project: http://www.itea2-easy-interactions.org/

[56] A. Isasi, T. Bartolomé, A. Picón. "Sistema De Interacción avanzada Utilizando Gestos y Voz Para El Manejo De Un Puente Grúa", Ibero American Conference (CIAWI), Lisbon (Portugal), 2008.

[57] P.R. Cohen and D.R. McGee, "Tangible multimodal interfaces for safety-critical applications," *Com-munications of the ACM*, 47(1):41-46, 2004.

[58] R. Bolt, "Put-That-There: Voice and gesture at the graphics interface," *Computer Graphics,* 14(3):262-270, 1980.

[59] D. Martin, A. Cheyer, and D. Moran, "The open agent architecture: A framework for building distributed software systems," *Applied Artificial Intelligence,* 13:91-128, 1999.

[60] S. Kumar and P. Cohen, "Towards a fault-tolerant multi-agent system architecture," *Int. Conf. on Autonomous Agents,* 2000.

[61] I. Cohen, N. Sebe, F. Cozman, M. Cirelo, and T.S. Huang, "Semi-supervised learning of classifiers: Theory, algorithms, and their applications to human-computer interaction," *IEEE Trans. on PAMI*, 22(12):1553-1567, 2004.

[62] Miniotas, Spakov, Tugoy and Mac Kenzie (2006 : "Speech augmented eye gaze interaction with small closely spaced targets" – Proceedings of the 2006 Symposium on Eye Tracking research & applications).

[63] J.S. Bradbury, J.S. Shell, and C.B. Knowles, "Hands on cooking: Towards an attentive kitchen", *ACM Conf. Human Factors in Computing Systems (CHI)*, 2003.

[64] A. Cheyer and L. Julia, "MVIEWS: Multimodal tools for the video analyst," *Conf. on Intelligent User Interfaces (IUI),* 1998.

[65] V. Lauruska and P. Serafinavicius, "Smart home system for physically disabled persons with verbal communication difficulties," *Assistive Technology Research Series (AAATE)*, pp.579-583, 2003.

[66] D. Chen, R. Malkin, and J. Yang, "Multimodal detection of human interaction events in a nursing home environment," *Conf. on Multimodal Interfaces (ICMI)*, 2004.

[67] A. Adler, J. Eisenstein, M. Oltmans, L. Guttentag, and R. Davis, "Building the design studio of the future," AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural, 2004.

[68] S.A. Brewster, J. Lumsden, M. Bell, M. Hall, and S. Tasker, "Multimodal 'Eyes-Free' interaction techniques for wearable devices," ACM Conf. Human Factors in Computer Systems (CHI,) 2003.

[69] W3C Multimodal Interaction Framework website, http://www.w3.org/TR/mmi-framework/

[70] D. Tapscott, A. D. Williams, Wikinomics: How Mass Collaboration changes everything. Penguin 2006.

[71] Y. Benkler,The Wealth of Networks How Social Production Transforms Markets and Freedom. Harvard 2006

[72] http://www.teced.com/services_design_participatory.html

[73] Beyer, H. & Holtzblatt, K. (1998). Contextual Design: Defining Customer-Centered Systems. San Francisco: Morgan Kaufmann. ISBN: 1-55860-411-1

[74] http://www.iso.org/iso/iso_catalogue

[75] http://www.iso.org/iso/catalogue_detail.htm?csnumber=21197

[76] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=31176

[77] http://en.wikipedia.org/wiki/Web_service

[78] SeCSE A1.D1a State of the art – Service Specification

[79] http://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383494

[80] http://en.wikipedia.org/wiki/Mashup_(music)

[81] http://www.hitech-projects.com/euprojects/amec/

[82] http://openkapow.com/

[83] R. Harper, T. Rodden, Y. Rogers., A. Sellen: "Being Human: Human Computer Interaction in 2020", paperback, 2 April 2008.

[84] Y. V. Natis, W. Clark and R. Valdes: "Context Delivery Architecture: Putting SOA in Context ", 23 October 2007

[85] W.M.P. van der Aalst, M. Adams, A.H.M. ter Hofstede, M. Pesic, and H. Schonenberg: "Flexibility as a Service", BPM Center Report BPM-08-09, BPMcenter.org, 2008

[86] eMobility Mobile and Wireless Communications Technology Platform: "eMobility Strategic Research Agenda", version 7, Ed. R. Tafazolli and V.V. Mattila, December 2008

[87] C. B. Anagnostopoulos, A. Tsounis, and S. Hadjiefthymiades: "Context awareness in mobile computing environments", Wirel. Pers. Commun., 42(3):445–464, 2007

[88] T. Strang, and C. Linnhoff-Popien, "A Context Modeling Survey", Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 – The Sixth International Conference on Ubiquitous Computing, Nottingham, UK, September 2004.

[89] M. Khouja, C. Juiz, I. Lera, R. Puigjaner, and F. Kamoun: "An ontology-based model for a context-aware service oriented architecture", in SERP 07: Proceedings of the 2007 International Conference on Software Engineering Research and Practice, 2007.

[90] Unified modeling language. http://www.uml.org, October 2007

[91] Object role modeling. http://www.orm.net, October 2007

[92] Composite capabilities / preferences profile. http://www.w3.org/Mobile/CCPP/, October 2007.

[93] User agent profile. http://www.wapforum.org, October 2007

[94] Standard generalized markup language (iso 8879:1986). http://www.iso.org, October 2007

[95] A. Schmidt, M. Beigl and H.W. Gellersen: “There is more to context than location”, Computers and Graphics 23, 6 (1999), 893–901].

[96] K. Cheverst, K. Mitchell and N. Davies: “Design of an object model for a context sensitive tourist GUIDE”, Computers and Graphics 23, 6 (1999), 883–891

[97] J. Mccarthy: “Notes on formalizing contexts”, in Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (San Mateo, California, 1993), R. Bajcsy, Ed., Morgan Kaufmann, pp. 555–560

[98] V. Akman and M. Surav: “The use of situation theory in context modeling”, Computational Intelligence 13, 3 (1997), 427–438

[99] T. Strang, C. Linnhoff-Popien, and K. Frank, "CoOL: A Context Ontology Language to enable Contextual Interoperability", 4th IFIP International Conference on Distributed Applications and Interoperable Systems, (DAIS 2003) Paris, France, November 2003. (LNCS, vol. 2893, pp. 236-247).

[100] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology Based Context Modeling and Reasoning using OWL", 2nd IEEE Int. Conf. on Pervasive Computing and Communications 2004 (PerCom 2004), Workshop on Context Modeling and Reasoning, Orlando, FL, USA, March 2004.

[101] H. Chen, T. Finin and A. Joshi: “An ontology for context-aware pervasive computing environments.” Knowledge and Engineering Review, Special Issue on Ontologies for Distributed Systems, 2003

[102] SVG standard, http://www.w3.org/Graphics/SVG/

[103] KML standard, http://www.opengeospatial.org/standards/kml/

[104] GeoRSS upcoming standard, http://www.ogcnetwork.net/gml-georss

[105] Google Maps, http://maps.google.com/

[106] Google Maps API, http://code.google.com/apis/maps/documentation/

[107] Yahoo maps, http://maps.yahoo.com/

[108] Bing Maps, http://www.bing.com/maps/

[109] OpenStreetMap, http://www.openstreetmap.org/

[110] http://news.cnet.com/Mapping-a-revolution-with-mashups/2009-1025_3-5944608.htmlOpenLayers, http://openlayers.org/

[111] Geomajas, http://www.geomajas.org/

[112] OpenStreetMap API, http://wiki.openstreetmap.org/wiki/API_v0.6

[113] TomTom Map Share, http://www.tomtom.com/page/mapshare

[114] Powell, H. (2009). Time, Television, and the Decline of DIY. *Home Cultures, 6*(1), 89-108.

[115] Hoftijzer, J. W. (2008). Co-creation: het nieuwe Doe-Het-Zelf? *Tijdschrift voor Industriele Productontwikkeling, 16*(5), 12-14.

[116] Mutaanen, U.M. (2005). The crafter's manifesto. http://ullamaaria.typepad.com/hobbyprincess/2005/03/draft_craft_man.html; accessed October 2009

[117] Poulter, S. (2006, December 1). DIY Gives Way to DFY (Done For You) Thanks to Immigrant Labour. from http://www.dailymail.co.uk/news/article-419617/DIY-gives-way-DFY-thanks-immigrant-labour.html

[118] Watson, I. (2000). Professionals and Neophytes: Contrasting Approaches to the Web. *Business Information Review, 17*(3), 125-129.

[119] Horovitz, B. (2004, April 26). It's a Do-It-Yourself World. 2009, from http://www.usatoday.com/money/2004-04-26-self-serve_x.htm

[120] http://www.instructables.com

[121] http://www.pokono.com

[122] ITU. (2005). ITU Internet Reports 2005: The Internet of Things.

[123] Aarts, E., & de Ruyter, B. (2009). New Research Perspectives on Ambient Intelligence. *Journal of Ambient Intelligence and Smart Environments, 1*(1), 5-14.

[124] Dodson, S. (2008). Forward: A Tale of Two Cities *The Internet of Things: A Critique of Ambient Technology and the All-Seeing Network of RFID*. Amsterdam: Institute of Network Cultures

[125] European Commission. (2009). Internet of Things — An action plan for Europe

[126] Jeppesen, L. B., & Frederiksen, L. (2006). Why Do Users Contribute to Firm-Hosted User Communities? The Case of Computer-Controlled Music Instruments. Organization Science, 17(1), 45-63.

[127] Harhoff, D., Henkel, J., & von Hippel, E. (2003). Profiting From Voluntary Information Spillovers: How Users Benefit by Freely Revealing Their Innovations. *Research Policy, 32*(10), 1753-1769.

[128] Franke, N., & Shah, S. (2003). How Communities Support Innovative Activities: An Exploration of Assistance and Sharing Among End-users. Research Policy, 32(1), 157-178.

[129] Rosner, D., & Bean, J. (2009). Learning from IKEA hacking: I'm Not One to Decoupage a Tabletop and Call It a Day. Paper presented at the 27th International Conference on Human Factors in Computing Systems, Boston, MA.

[130] Leadbeater, C., & Miller, P. (2004). The Pro-Am Revolution: How Enthusiasts are Changing our Economy and Society. London: Demos.

[131] von Hippel, E. (2005). *Democratizing Innovation*. The MIT Press: Cambridge, Massachusetts.

[132] Frissen, V., & Slot, M. (2009). The Return of the Bricoleur: Redefining Media Business. In B. Sapio, L. Haddon, E. Mante-Meijer, L. Fortunati, T. Turk & E. Loos (Eds.), *The Good, the Bad and the Challenging: The User and the Future of Information and Communication Technologies: A Transdisciplinary Conference* (Vol. 1, pp. 88-99). Koper: ABS-Center.

[133] Bakardjieva, M. (2001, June 12-14). *Becoming a Domestic Internet User.* Paper presented at the 3rd International Conference on Uses and Services in Telecommunications, Paris.

[134] Bakardjieva, M. (2005). Internet Society: The Internet in Everyday Life. London: SAGE Publications.

[135] Bakardjieva, M. (2006). The Consumption Junction Revisited: Network and Contexts. In R. Kraut, B. Malcolm & S. Kiesler (Eds.), *Computers, Phones, and the Internet: Domesticating Information Technology* (pp. 97-108). New York: Oxford University Press.

[136] Flowers, S. (2008). Harnessing the hackers: The emergence and exploitation of Outlaw Innovation. *Research Policy, 37*(2), 177-193.

[137] Wyatt, S. (2003). Non-Users Also Matter: The Construction of Users and Non-Users of the Internet. In N. Oudshoorn & T. Pinch (Eds.), *How Users Matter: The Co-Construction of Users and Technology* (pp. 67-79). Cambridge, Massachusetts: The MIT Press.

[138] Jukka Honkola, Hannu Laine, Ronald Brown, Ian Oliver: Cross-Domain Interoperability: A Case Study. NEW2AN 2009: 22-31

[139] Soininen Juha-Pekka, Lappeteläinen Antti (2009). "M3 Smart Environment Infrastructure". NoTA 2009 Conference. Available at http://www.vtt.fi/files/projects/nota2009/juhapekkasoininen.pdf.

[140] Ian Oliver, Jukka Honkola (2008) Personal Semantic Web Through A Space Based Computing Environment. Middleware for Semantic Web 08 at ICSC'08, Santa Clara, CA, USA

[141] SIB Service Interface v. 0.9, Nokia Corporation, 2009. Available at http://sourceforge.net/projects/smart-m3/

[142] Smart M3 open source code. Available at http://sourceforge.net/projects/smart-m3/. Accessed 16.12.2009.

[143] OWL Web Ontology Language Quide. Available at . http://www.w3.org/TR/owl-guide/

[144] NoTA World, Open Architecture Initiative. Available at http://www.notaworld.org/

[145] Smart M3 Wiki. Available at http://en.wikipedia.org/wiki/Smart-M3.

[146] Netvibes Universal Widget API Specification, http://dev.netvibes.com/doc/uwa_specification

[147] W3C widget specification, http://www.w3.org/TR/widgets/

[148] Java FX Mobile, http://java.sun.com/javafx/

[149] Adobe Flash Lite, http://www.adobe.com/products/flashlite/

[150] WidSets, http://www.widsets.com

[151] ACM. 2003. ACM Code of Ethics and Professional Conduct.

[152] Arhippainen, L. and Tähti, M. (2003) Empirical Evaluation of User Experience in Two Adaptive Mobile Application Prototypes, in Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia (MUM 2003), Norrköping, Sweden, pp. 27- 34

[153] European Commission: Design for All (DfA). http://ec.europa.eu/information_society/activities/einclusion/policy/accessibility/dfa/index_en.htm

[154] Hassenzahl, M. (2003) The Thing and I: Understanding the Relationship between User and Product. In: (Blythe, M.A., Overbeeke, K., Monk, A.F. and Wright, P.C. Eds.) Funology: From Usability to Enjoyment. Dordrecht, the Netherlands: Kluwer Academic Publishers, pp. 31-42.

[155] Hassenzahl, M. and Tractinsky, N. (2006) User Experience – A Research Agenda. Behaviour and Information Technology, Vol. 25, No. 2, pp. 91-97.

[156] Kaasinen, E. and Norros, L. (eds.) (2007) Älykkäiden ympäristöjen suunnittelu: kohti ekologista systeemiajattelua (Design of smart environments: towards ecological systemic thinking). Teknologiateollisuus, Tampere, Finland.

[157] Kankainen, A. (2003) UCPCD: User-Centered Product Concept Design, in Proceedings of the 2003 conference on Designing for user experiences, ACM Press, pp. 1-13.

[158] Wright, P., McCarthy, J., and Meekison, L. (2003) Making Sense of Experience. In: (Blythe, M.A., Overbeeke, K., Monk, A.F. and Wright, P.C. Eds.) Funology: From Usability to Enjoyment. Dordrecht, the Netherlands: Kluwer Academic Publishers, pp. 43–53.

[159] Baty, S. Decontructing Analysis Techniques, 2009 in Johnny Holland magazine http://johnnyholland.org/2009/02/17/deconstructing-analysis-techniques/

[160] Bhat, S. Touchtone, TU Delft, 2009 http://www.soumitrabhat.com/projectdetails.php?id=6

[161] Designing Games, Persona, Storytelling, Use cases, Storyboard, Blueprint, Wizard-of-Oz, Experience prototyping, System map, Touchpoint matrix, Ecology map definitions from Tools section www.servicedesigntools.org

[162] Dishman,E. Designing for the New Old Now: Asking, Observing and Performing Future Elders and

[163] Shedroff, N. Research methods for design Effective Experiences from Design Research edited by Brenda Laurel; 2003, MIT Press

[164] Funamizu, M. Mac's petit inventions: Gadgets for the physically challenged, 2009, Johnny Holland magazine http://johnnyholland.org/2009/02/23/gadgets-for-the-physically-challenged/

[165] Goodman-Deane, J. Clarkson, J. Langdon, P. Waller; S. Tools for Supporting Inclusive Design; Engineering Design Centre, University of Cambridge http://www-edc.eng.cam.ac.uk/~jag76/hci_workshop08/goodman.pdf

[166] Introduction to ISO 13407, 1999 http://www.ucc.ie/hfrg/emmus/methods/iso.html

[167] Laurea, DIYSE funding application

[168] Mattelmäki, T. Design Probes, 2006, Published by University of Art and Design Helsinki https://www.taik.fi/kirjakauppa/images/6bbfa3805609ef167fbff8d7a186c86e.pdf

[169] Modeling, http://www.cooper.com/about/process/modeling.html

[170] Moggridge, B. Designing Interactions, 2007, MIT Press

[171] Rohrer, C. 2008, http://www.useit.com/alertbox/user-research-methods.html

[172] Sanders and Chan; Emerging trends in Design research, 2007 and

[173] Sanders; From User-centered to Participatory DesignApproaches, 2002 from http://maketools.com/

[174] Wikipedia article on User-centered design, 2004. http://en.wikipedia.org/wiki/User-centered_design

[175] Thomas R. Gruber (1993). Toward principles for the design of ontologies used for knowledge sharing. Originally in N. Guarino and R. Poli, (Eds.), International Workshop on Formal Ontology, Padova, Italy. Revised August 1993. Published in International Journal of Human-Computer Studies, Volume 43 , Issue 5-6 Nov./Dec. 1995, Pages: 907-928, special issue on the role of formal ontology in the information technology

[176] Oberle, D., Volz, R., Staab, S., and Motik, B.(2004): An extensible ontology software environment, handbook on ontologies) Staab, S., and Studer, R., Eds. Springer, Verlag, Berlin, Germany, 311—333

[177] Colomb, R.M (2007): Ontology and the Semantic Web: Volume 156 Frontiers in Artificial Intelligence and Applications (Frontier in Artificial Intelligence and Applications), IOS Press (April 15, 2007), ISBN-10: 1586037293, ISBN-13: 978-1586037291

[178] Staab, S, and Studer, R. (2004): handbook on ontologies, Springer-Verlaag, ISBN 3-540-4-834-7,

[179] Rittgen, P. (2008): handbook of ontologies for business interaction, published in the USA by Information Science Reference (IGI global), ISBN-13: 978-1-59904-660-0, 2007023438

[180] Mika, P., Iosif, V., and Akkermans H. (2003): ontology-based content management in a virtual organization, in Steffen Staab and Rudi Studer (Eds.), handbook on ontologies, pp. 455-476, chapter 23, Springer

[181] Institute of Electrical and Electronics Engineers (1990). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York.

[182] Meersman, R. (2001): Ontologies and Databases: More than a Fleeting Resemblance, in Proceedings of OES/SEO 2001 Rome Workshop, In, d'Atri, A. & Missikoff, M.,(eds.), Luiss Publications, 2001.

[183] Spyns, P., Tang, Y., and Meersman, R. (2007): A model theory inspired collaborative ontology engineering methodology, Journal of Applied Ontology, special issue on "Ontological Foundations for Conceptual Modeling", Giancarlo Guizzardi and Terry Halpin (eds.), Volume 4 (2007) 1, IOS Press, 1570-5838/07, pp. 1~23

[184] de Moor, A., De Leenheer, P., Meersman, R., DOGMA-MESS: A Meaning Evolution Support System for Interorganizational Ontology Engineering, Proc. of 14th ICCS conference, Springer-Verlag, V. 4068, p.189-203, Aalborg, Denmark, 2006

[185] CSA, (1970): Z243.1-1970 for Decision Tables, Canadian Standards Association, 1970

[186] Wets, G., Vanthienen, J., Mues, C., and Timmermans, H. (1998): Extracting complete and consistent knowledge patterns from data, in Proc. Of Sixth International Conference on Principles of Knowledge Representation and Reasoning: V&V Workshop, ISSN 1613-0073, editor: F. van Harmelen, Trento, Italy, 1998

[187] Tang, Y., and Meersman, R. (2007): Towards building semantic decision tables with domain ontologies, In Proceedings of International Conference of information Technology and Management (ICITM2007), 3~5 January, 2007, in, C. Man-chung, J.N.K. Liu, R. Cheung, J. Zhou, (eds.) ISM Press, ISBN 988-97311-5-0, pp. 14~21

[188] "The Web Service Modeling Framework WSMF" available at: http://www.wsmo.org/papers/publications/wsmf.paper.pdf

[189] "WSMO" available at: http://www.wsmo.org/ , "Web Service Modeling Ontology (WSMO)" available at "http://www.w3.org/Submission/WSMO/"

[190] "Web Services Modeling Language" available at: http://www.wsmo.org/TR/d16/d16.1/v0.21/#part:variants

[191] "Web Services execution environment" available at: http://www.wsmx.org/

[192] "State of the art –SWS Infrastructure, Annotation, LCMS" available at: "http://luisa.atosorigin.es/www/images/comission_deliverables/d3.1_stateoftheart_v1.0_final.pdf

[193] Mahmoud, Q. H., Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI), April 2005, http://java.sun.com/developer/technicalArticles/WebServices/soa/index.html

[194] Costello R.L., Building Web Services the REST way.http://www.xfront.com/REST-Web-Services.html

[195] http://wsmo4j.sourceforge.net/

[196] http://www.w3.org/Submission/WSMX/#L26

[197] "Semantic Web Services, State of the art ", http://www.webentity.de/Works/A1-_Semantic_Web_Services_-_State_of_the_art.pdf

[198] "State of the art analysis including assessment of system architectures for ambient intelligence" available at: http://www.hitech-projects.com/euprojects/amigo/deliverables/Amigo_D2_2_WP2_final.pdf

[199] [2] "Ontology" by Tom Gruber, available at: http://tomgruber.org/writing/ontology-definition-2007.htm

[200] "OWL Web Ontology Language" available at: http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1

[201] Hydra Project: "Applications of Semantic Technologies in AmI" available at http://www.hydramiddleware.eu/hydra_papers/Applications_of_Semantic_Technologies_in_AmI.pdf

[202] "International Semantic Web Conference (ISWC 2009)" http://iswc2009.semanticweb.org/

[203] "A contextual ontology to provide location-aware services and interfaces in smart environments" available at: http://dspace.uah.es/jspui/bitstream/10017/1263/1/iadis05-miriam.pdf

[204] "SMARTMUSEUM Report of User Profile Formal Representation and Metadata Keyword Extension" available at: http://www.smartmuseum.eu/del/D2.1_v1.1_SM.pdf

[205] "State of the art" available at: http://www.sm4all-project.eu/images/stories/deliv/d12.pdf

[206] Tribe M., Jana R., "New Media Art" open soruce wikibook available at https://wiki.brown.edu/confluence/display/MarkTribe/New+Media+Art

[207] Huybrechts L., (2008), Cross-over, Lannoo Belgium, 224pp

[208] Cook, D.J., Youngblood, M. Berkshire Encyclopedia of Human-Computer Interaction. In: Smart Homes. Berkshire Publishing Group (2004)

[209] Nixon, P.A., Wagealla, W., English, C., Terzis, S.: 11. In: Security, Privacy and Trust Issues in Smart Environments. John Wiley & Sons (2005) 249–270

[210] Stallings, W.: Criptography and network security: principles and practice. Second edition edn. Prentice Hall (1999)

[211] Langeheinrich, M.: Privacy by design: Principles of privacy aware ubiquitous systems. In: UBICOMP 2001, Lecture Notes in Computer Science. Volume 2201. (2001) 273–291

[212] Campbell, R., Al-Muhtadi, J., Naldurg, P., Sampemane, G., Mickunas, M.D.: Towards security and privacy for pervasive computing. In: Theories and Systems,Mext-NSF-JSPS International Symposium, ISSS 2002. Lecture Notes in Computer Science, Tokyo, Japan (2002) 1–15

[213] Stajano, F., Anderson, R.: The resurrecting duckling: security issues for ubiquitous computing. IEEE Computer 35 (2002) 22–26

[214] Al-Muhtadi, J., Anand, M., Mickunas, M., Campbell, R.: Secure smart homes using jini and uiuc sesame. In: Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference. (2000) 77–85

[215] Devdatta Kulkarni , Anand Tripathi, Context-aware role-based access control in pervasive computing systems, Proceedings of the 13th ACM symposium on Access control models and technologies, June 11-13, 2008, Estes Park, CO, USA

[216] Roles in Information Systems: A Survey, Haibin Zhu; MengChu Zhou, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. On page(s): 377-396, Volume: 38, Issue: 3, May 2008.

[217] A. K. Dey, Providing architectural support for building context-aware applications, Ph.D. dissertation, Georgia Institute of Technology, Nov. 2000. http://www.cc.gatech.edu/fce/ctk/pubs/deythesis.pdf

[218] Al-Muhtadi, J., Ranganathan, A., Campbell, R., Mickunas, M.: Cerberus: a context-aware security scheme for smart spaces. In: Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on.

[219] Al-Muhtadi, J., Mickunas, D., and Campbell, R. 2001 Tiny SESAME on the HP Jornada. Technical Report. UMI Order Number: UIUCDCS-R-2001-2198., University of Illinois at Urbana-Champaign.

[220] Ballon, Pieter (2007) 'Business modelling revisited: the configuration of control and value', in Info, 9 (5), 6-19.

[221] Sennett, Richard (2008) The craftsman. London: Allen Lane

[222] Punie, Yves (2003) EMTEL - A social and technological view of Ambient Intelligence in Everyday Life - What bends the trend? (EUR 20975 EN). Brussels: IPTS European Commission - Joint Research Centre, 68.

[223] Bakardjieva, Maria (2005) Internet society: the Internet in everyday life. London: SAGE, 220.

[224] Stewart, James (2007) 'Local experts in the domestication of information and communication technologies', in Information, Communication and Society, 10 (4), 547-569.