ebbits

Enabling the business-based
Internet of Things and Services

(FP7 257852)

D5.1.1 Concepts and technologies in intelligent service structures 1

Published by the ebbits Consortium

Dissemination Level: Public

European Commission
Information Society and Media

SEVENTH FRAMEWORK PROGRAMME

Project co-funded by the European Commission within the 7th Framework Programme
Objective ICT-2009.1.3: Internet of Things and Enterprise environments

Document control page

Document file: D5.1 Concept and Technologies in Intelligent Service Structures v1.0_injet.doc
Document version: 1.0
Document owner: Fraunhofer FIT

Work package: WP5 – Centralized and Distributed Intelligence
Task: T5.1 Architectural Analysis and Description of the Centralized and Distributed Intelligent Service Structure
Deliverable type: R

Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Andreas Zimmermann (FIT), Ferry Pramudianto (FIT)	2010-09-23	Initial structure, Time plan, Tasks distribution
0.2	Ferry Pramudianto (FIT)	2010-11-01	Multi-sensor Fusion added.
0.3	Roberto Checcozzo (COMAU)	2010-11-01	Review of Manufacturing added.
0.4	Michael Jacobsen (TNM)	2010-11-01	Review of Food Production added.
0.5	Amro Al-Akkad (FIT)	2010-11-01	Mobile Sensing added.
0.6	Gonzalo Alcaraz (ISMB) Mirko Franceschinis (ISMB) Claudio Pastrone (ISMB) Maurizio Spirito (ISMB) Riccardo Tomasi (ISMB)	2010-11-01	Opportunistic P2P Communication and Messaging data format for low level device communication added
0.7	Antje Böhm-Peters (SAP)	2010-11-05	Consolidation of Distributed Information and Knowledge added. Technologies for Knowledge Derivation from Sensor Information added
0.8	Ferry Pramudianto, Amro Al-Akkad	2010-11-08	Resource and Control Management added
0.9	Ferry Pramudianto		Messaging data format for high level application communication
0.10	Matts Ahlsén (CNET) Stefan Asanin (CNET)	2010-11-08	Service-Oriented Architectures for Distributed Intelligent Service Structures Devices and Service Discovery in Intelligent Environments added
0.11	Jan Hreno (TUKE)	2010-11-10	Semantic & Context Modeling added
0.12	Ferry Pramudianto	2010-11-14	Conclusion added.
0.13	Andreas Zimmermann	2010-11-14	Executive Summary added.
0.14	Ferry Pramudianto, Amro Al-Akkad	2010-11-23	Document restructured, added more information to product life-cycle, executive summary and conclusion are improved with relationship among chapters, Terminology added.
0.15	Gonzalo Alcaraz Claudio Pastrone Maurizio Spirito	2010-11-24	Additional contributions on PROFINET and PROFIBUS International Application Profiles, LonMark functional profiles, KNX

1.0		2010-11-30	Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Summary of comments
Jesper Thestrup	2010-11-16	Comments on logical structure of the document and focus on ebbits relevant topics including balance and need for addressing the last part of the product life-cycle. Minor comments on formatting and references.
Maurizio Spirito	2010-11-20	Relationships between chapters are not explicitly stated, comment on the content and structure.

Table of Contents

1. Executive Summary

The ebbits project aims to develop architecture, technologies and processes, which allow businesses to semantically integrate the Internet of Things into mainstream enterprise systems and support interoperable real-world, on-line end-to-end business applications. It will provide semantic resolution to the Internet of Things and hence present a new bridge between backend enterprise applications, people, services and the physical world, using information generated by tags, sensors, and other devices and performing actions on the real-world. Ebbits opens possibilities to offer a wide range of new business services based on choreography of physical devices, software services, and people that we introduced as Internet of People, Thing, and Services (IoPTS).

Fulfilling this aim, work package 5 (WP5) will provide an intelligent service infrastructure that facilitates distributed intelligence collaborating in the IoPTS. It defines intelligent services as services that first support human and machine decision makers by providing the required information. Such services are aware of the context and its resources, are autonomously able to decide and reacting based on given information, and such services are able to collaboratively work together.

This deliverable gives common research directions to the consortium members in work package 5 by describing the state-of-the-art of the existing technology and standards that used in business domain. This knowledge is relevant for the construction of the building blocks of intelligence and interoperability required in ebbits. In this regard, intelligence relates to sensor and actuator networks, devices, operational plants, and also takes place on an enterprise information system level. The technology investigated and elaborated in this deliverable covers the basis for the development for the architecture of intelligence system (Task 5.1), the multi-sensor fusion framework (Task 5.2), the context modeling framework (Task 5.3) and the control management framework (Task 5.4).

This deliverable is firstly started with an initial glossary of terms (Chapter 3) and a deep review of the application domains car manufacturing as well as food production aiming at providing a common understanding of the background and the constraints of the work planned in work package 5. Subsequently, a technical part elaborates firstly, the state-of-the-art of sensing technology that includes the multi-sensor data fusion architectural models and techniques (Chapter 5), mobile sensing (Chapter 6), knowledge derivation from sensor information (Chapter 7). These technologies describe the foundation of intelligent multi-sensor fusion framework. Secondly, state-of-the-art related to control management is discussed in Chapter 8 that includes resource management in industrial environment, wireless sensor network as well as some examples of real world applications. These technologies which will be used as a reference for task 5.4. Furthermore, knowledge modeling, context modeling, distributed knowledge architecture, and handling inconsistence of distributed models that are needed as a foundation of task 5.3 and as a reference for sensing and control management is discussed in Chapter 10 and Chapter 9.

This deliverables discusses various technologies that are also important to ensure intelligent services being able to communicate and collaborate with each other (Chapter 10-13). The exploitation of the Hydra middleware for the Web Service-oriented discovery, messaging formats and interconnection of heterogeneous devices covers one major part of this description.

Conclusion of this deliverable is elaborated in Chapter 15 that can be summarized as the following: the JDL model is a promising architectural model covering the duality of sensing and control, however it has to be extended in order to illustrate the specific tasks needed in ebbits. Feedback Control theory that has been used extensively in industrial applications needs to be extended by the use of semantic and planning algorithm to achieve a more intelligent and autonomous system as envisioned in this project. Ebbits must define specific context and semantic models that are needed for sensing and control processes as well as for reflecting system performance to achieve a generic sensing and control framework. Collaboration among applications, devices and services will be facilitated by Hydra middleware. It will be extended using opportunistic communication paradigm to solve unreliability of the Internet. Moreover, ebbits will establish a data format standardization to ensure syntactic interoperability for new resource constrained devices created for ebbits.

2. Introduction

The ebbits project aims to develop architecture, technologies and processes, which allow businesses to semantically integrate the Internet of Things into mainstream enterprise systems and support interoperable real-world, on-line end-to-end business applications. It will provide semantic resolution to the Internet of Things and hence present a new bridge between backend enterprise applications, people, services and the physical world, using information generated by tags, sensors, and other devices and performing actions on the real-world. Ebbits opens possibilities to offer a wide range of new business services based on orchestration of physical devices, software services, and people that we introduced as Internet of People, Thing, and Services (IoPTS). Fulfilling this aim, work package five (WP5) is responsible to provide an intelligent service infrastructures. Intelligence comprises ability to learn, to interpret data, to be aware, to be autonomy, and to collaborate. Hence, we define intelligent services as services that first support human and machine decision makers by providing the required information, services that are aware of the context and its resources, services that may autonomously plan, decide and react to situation, services that could learn, and services that are able to collaboratively work together.

2.1 Purpose, context and scope of this deliverable

The purpose of this deliverable is to give common research directions to all consortium members by describing the state-of-the-art of the existing technology and existing standards that are relevant to constructing the building blocks of intelligence and interoperability required in ebbits including intelligence in sensor and actuator network, devices, operational plants, and enterprise information system level and discussing how they can be used and extended in work package 5. The technology elaborated in this deliverables covers the basis for the development for the architecture of intelligence system (Task 5.1), the multi-sensor fusion framework (Task 5.2), context modeling framework (Task 5.3) and the control management framework (Task 5.4).

2.2 Deliverable Organization

This deliverable is organized as the following sections:

- Section 1 summarizes this deliverables, the relation the chapters to each other as well as to the tasks of WP5.
- Section 3 elaborates terminology used in this deliverables that has been commonly used within the corresponding domains.
- Section 4 reviews the application domains chosen by the consortium to validate the ebbits platform. The application domains include food production chain that involves farms, slaughterhouses, retail stores, and car manufacturing specifically for body welding, and Powertrain manufacturing.
- Section 5 describes the modern models and approaches to fuse data input and low level information into high level information that support decision makers by reducing uncertainty.
- Section 6 elaborates sensing in mobile environment that is needed to satisfy several requirements of ebbits (e.g.: animal movement patterns in barn). This includes opportunistic and participatory sensing as well as real world application examples.
- Section 7 describes the transformation of sensor data and information into knowledge in enterprise systems that can be done through sensor languages, learning and planning strategies as well as rule engines.
- Section 8 describes control theory and practices in industry that is relevant for designing automation processes in ebbits. Moreover this chapter discusses control for resource management and network resource management in WSN.
- Section 9 describes the state-of-the-art approaches for consolidating and populating information and knowledge distributed within the enterprise, web and internet.

- Section 10 describes the state-of-the-art work in semantic data handling such as inconsistence as well as semantic context modeling that is used by Hydra middleware.
- Section 11 describes overview of service oriented architectures such as centralized, distributed, and hybrid architectures. In addition, several SOA implementations that were done in EU Projects were discussed including SOA4ALL, Socrades, and HYDRA Middleware that will be used as a based platform for developing ebbits. Furthermore, SOA standards in product lifecycle management are also discussed.
- Section 12 elaborates state-of-the-art of discovery for services and devices such as WS-Discovery, UPnP and 3-layer discovery in hydra that abstracts heterogeneous discovery protocols into UPnP and Semantic Discovery.
- Section 13 elaborates opportunistic communication that studies the message delivery in an unreliable network connection that will be used to extend the P2P network infrastructure of the Hydra middleware.
- Section 14 describes standards of messaging and data formats used for industrial devices, wireless sensor networks as well as enterprise applications that are needed to ensure syntactic interoperability in ebbits.

3. Definition of Terminology

3.1 Food Lifecycle

Pig, hog or swine, is the species as a whole, or any member of it. The singular of "swine" is the same as the plural.

Shoat, piglet, pig, is unweaned young pig, or any immature pig.

Sucker is a pig between birth and weaning.

Runt is an unusually small and weak piglet, often one in a litter.

Boar or hog is male pig of breeding age.

Sow is a breeding female, or female after first or second litter.

Suckling pig is a piglet slaughtered for its tender meat.

Farrowing is giving birth.

Carcass is the dead body of an animal, esp. one that has been slaughtered for food, with the head, limbs, and entrails removed

Dehairing is a process of removal of hairs and bristles of pigs by a scraper after scalding in a tank or a steam cabinet

3.2 Automotive Industry

Body Welding, The process of manufacturing a body of a car by welding the joints together.

Powertrain the process of manufacturing parts of car that generate power and deliver it to a road surface. The parts normally include at least Engine and Transmission.

Body in White or **BIW** refers to the stage or progress automobile manufacturing in which the car body sheet metal (including doors, hoods, and deck lids) has been assembled or designed but before the components (chassis, motor) and trim (windshields, seats, upholstery, electronics, etc.) have been added

Bake is a process of drying body paint of a car by putting the car inside a huge oven.

3.3 Technology

Data is a collection of discrete objects, facts or events out of context.

Information, in general terms, is data plus conceptual commitments and interpretations

Knowledge is the result of learning. Knowledge is the internalization of information, data, and experience.

Intelligence is the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal.

Service in SOA is unassociated, loosely coupled units of functionality that have no calls to each other embedded in them.

Ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts

Concept is an abstract idea or a mental symbol sometimes defined as a unit of knowledge.

4. Review of the Application Domains

This Chapter provides an inventory of existing devices, services and technologies exploited by each application domain of ebbits.

4.1 Car Manufacturing

4.1.1 Automotive Plant production process

The automatic plant to produce car are generally divided in area dedicated to specific activities. Generally each car-maker retains in house part of the production process, as the complete manufacturing of a car requires a large investment, areas and manpower quantities.

In particular, main parts of the car-makers concentrate their effort to perform by itself the following part of the manufacturing process:

- Powertrain (PWT) plant: where engines are machined and assembled;
- Body in white shop: where the body of the car is assembled and welded;
- Painting shop: where the body in white is prepared to be painted and finally painted and baked;
- Final Assembly: where the painted body is assembled with the engine, the suspensions and all the other particulars.

As COMAU is particularly skilled on PWT and Bodywelding (BW) processes, these will be considered in the ebbits project.

4.1.2 Body in white

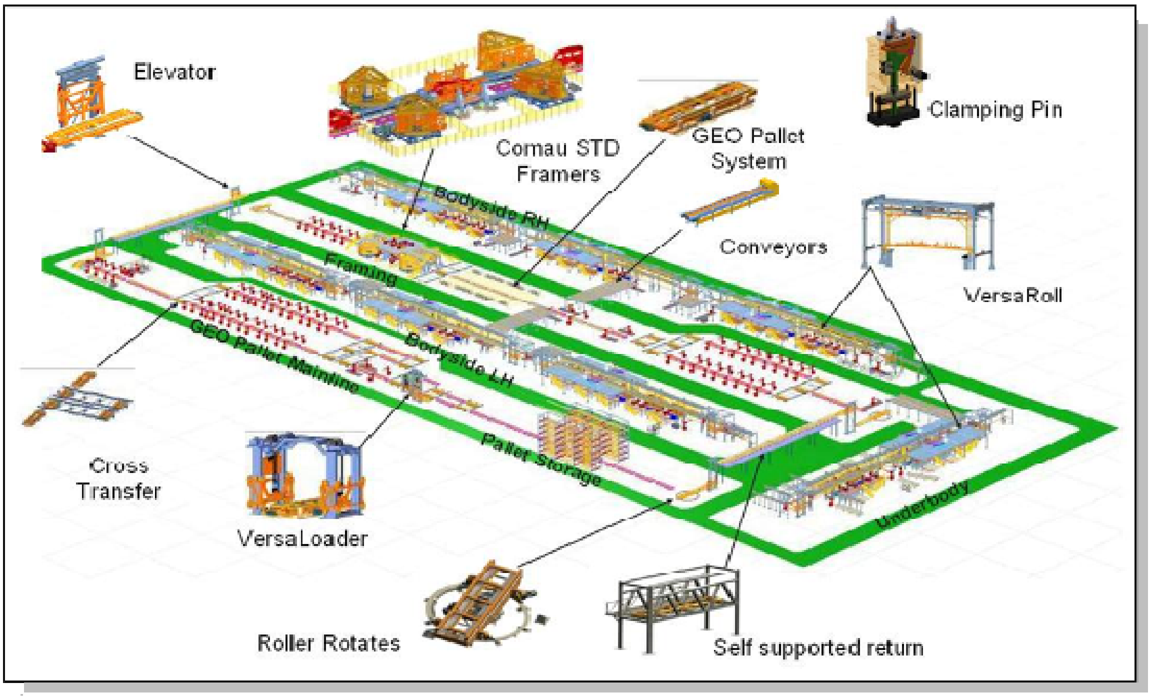


Fig. 4-1. Example of a body in white manufacturing plant layout

As shown in Fig. 4-1, generally, the floor panel is the largest body component to which a multitude of panels and braces will subsequently be either welded or bolted. As it moves down the assembly line, held in place by clamping fixtures, the shell of the vehicle is built. First, the left and right quarter panels are robotically disengaged from pre-staged shipping containers and placed onto the floor pan, where they are stabilized with geometric fixtures and welded.

The front and rear door pillars, roof, and body side panels are assembled in the same fashion. The shell of the automobile assembled in this section of the process lends itself to the use of robots because articulating arms can easily introduce various component braces and panels to the floor pan and perform a high number of weld operations in a time frame and with a degree of accuracy no human workers could ever approach. Robots can pick and load roof panels and place them precisely in the proper weld position with the assistance of vision devices.

The body is built up on a separate assembly line from the chassis. Robots once again perform most of the welding on the various panels, but human workers are necessary to bolt the parts together. During welding, component pieces are held securely in a jig while welding operations are performed.

As the body moves from the isolated weld area of the assembly line, subsequent body components including fully assembled doors, deck lids, hood panel, fenders, trunk lid, and bumper reinforcements are installed. Although robots help workers place these components onto the body shell, the workers provide the proper fit for most of the bolt-on functional parts using pneumatically assisted tools.

The control architecture used in a body in white are continuously changing and evolving. In the close future the architecture will be as depicted in Fig. 4-2.

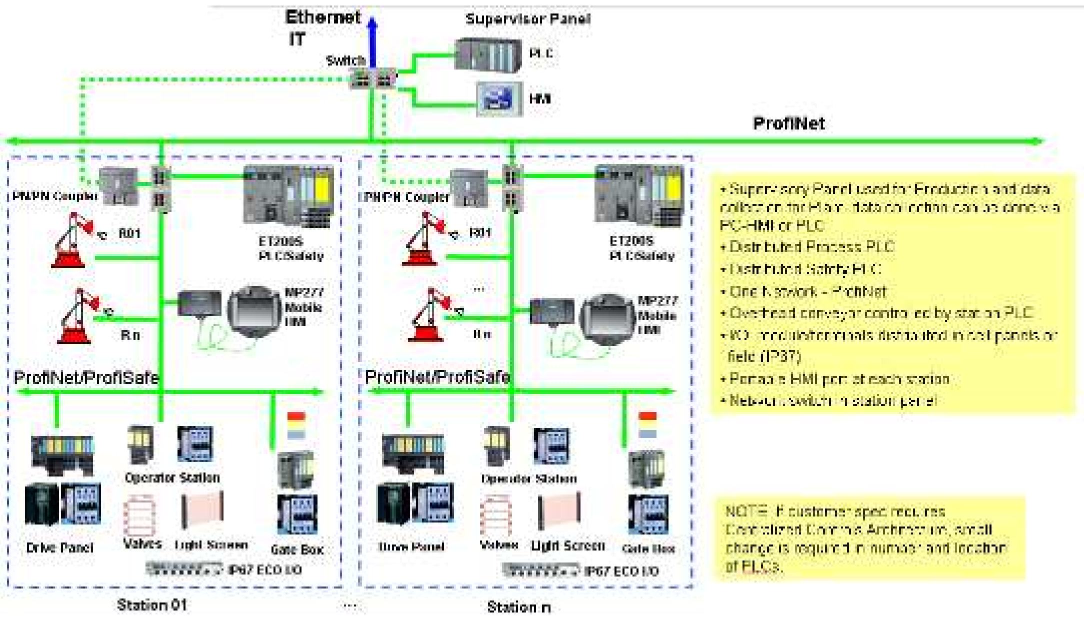


Fig. 4-2. Distributed control architecture in auto industry.

Powertrain and engine assembly

Powertrain machining activities differ depending from the different kind of engine has to be produced. The process for an aluminums basement for 16V engine includes the following operations:

- cubing machining with MC (Machine Centre)
- machining with MC
- Intermediate Washing Machine
- Intermediate Leak Test Machine
- Guides and Seats Press Machine
- machining with MC
- Intermediate Washing Machine
- Caps Assembly machine
- machining with MC
- Final Washing Machine
- Press plugs Machine
- Final Leak Test Machine
- visual check

The process for a cast iron basement for 4 cylinders engine includes the following operations:

- cubing machining with MC
- machining with MC
- Intermediate Washing Machine
- Intermediate Leak Test Machine
- Bedplate assembly machine
- machining with MC
- Cylinder bore and crankbore Honing machine
- Final Washing Machine
- Press plugs Machine
- Final Leak Test Machine
- visual check
- Cylinder bore and crankbore Measuring machine

4.2 COMAU business and expertise

Comau is organized into 3 Business Units (Fig. 4-3): 1. Body Welding & Assembly, 2. Powertrain Machining & Assembly, and 3. Robotics & Service.

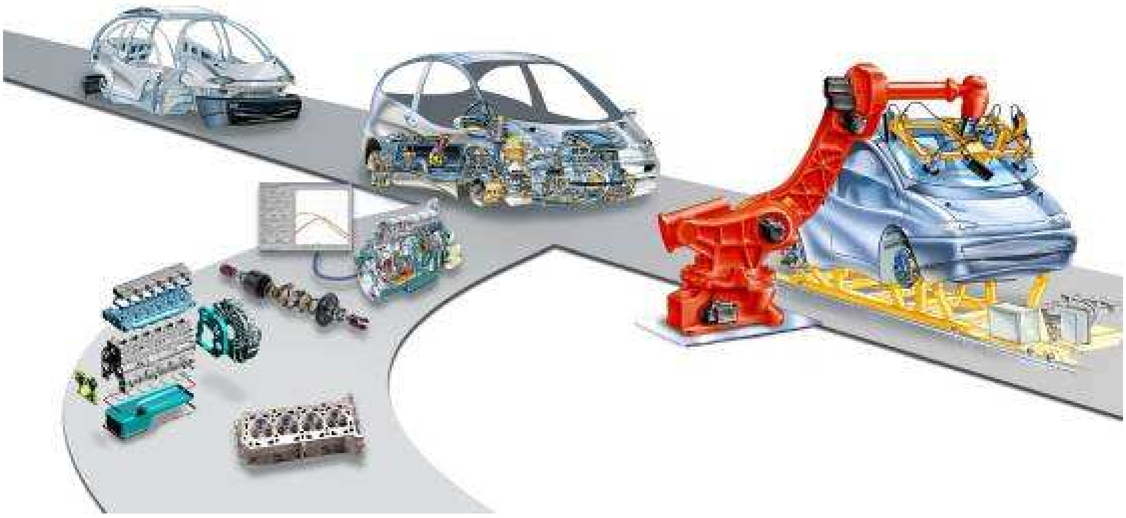


Fig. 4-3. Comau Business Units expertises

The offering of full services, from product engineering to maintenance activities, and the integrated global organization allow Comau to compete in the continuously evolving automotive global market, in the areas of: Sheet Metal Dies; Powertrain Machining & Assembly; Body Welding & Assembly Systems; Robotics; Aerospace Production Systems; Maintenance & Engineering Services.

4.2.1 Body Welding and Assembly

Comau Body Welding & Assembly is a global leader in the supply of advanced production systems for vehicle full body, body components manufacturing, and complete turnkey body shops worldwide. Comau provides the most cost-effective low, medium and high volume body welding and assembly systems in the world.
BWA solutions are oriented to the maximum flexibility of the production plant. Recently the new COMAU product line for the body in white and final assembly processes, the Versa series (Fig. 4-4), was introduced on the market. Distinctive characteristic of these product are:

- Self contained Large Sub-assembly manufacturing machine
- High Speed Part Transfer system
- Selectable and flexible station process
- Capable of 4+1 model random or infinite model batch production

This solution permit to reach excellent performances for the customer of the automatic plant:

- High flexibility, reached by the modularity of all the device and structure;
- Low time to market, thanks to the simple and rapid assembly operations;
- Better process quality, through the innovative position of the robots that allow to easily reach all the points on the working part;
- Decreased cycle time and more part per hour produced, by the new and high speed transportation system;
- Lower running costs, with the possibility to easily move under the structure to perform maintenance operations.

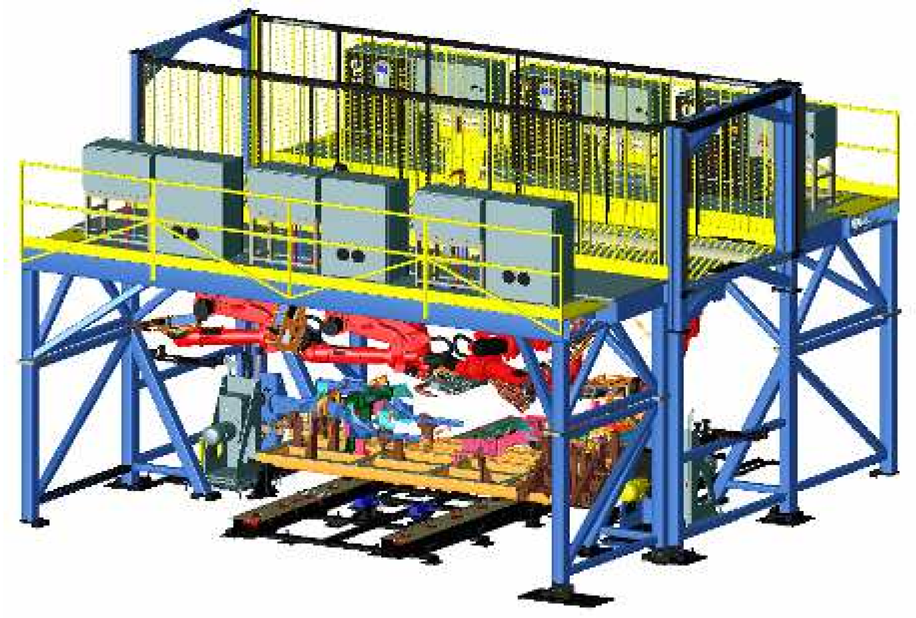


Fig. 4-4. Module implementing Versa-Series solutions

The new versa solutions are characterized by the modularity of the structure to be assembled on the customers' production plant that requires less area to perform the same operation. Moreover, the modular design of the production station can be easily exploited to rapidly design the production flow of the plant, without the need of any particular custom design.

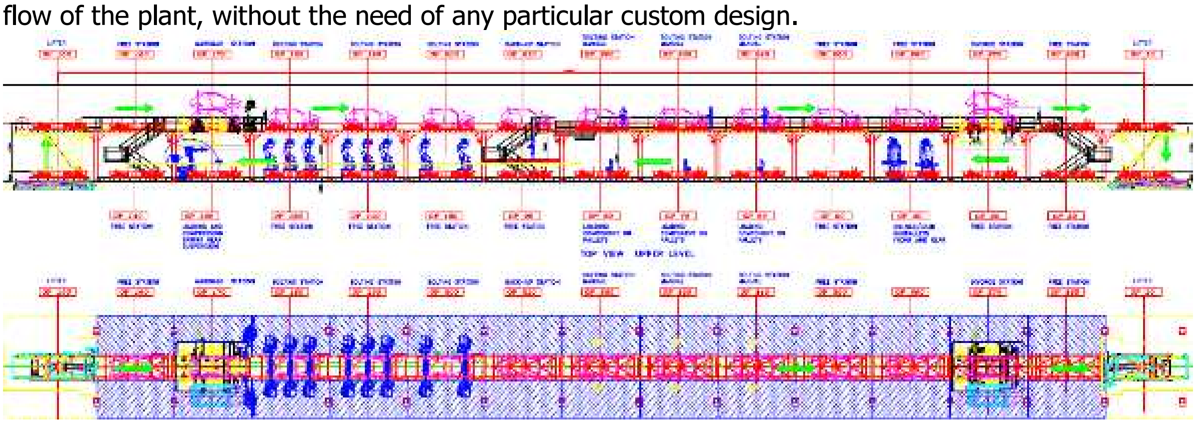


Fig. 4-5. Modular installation of Versa-Series

4.2.2 Power train machining and assembly

Power train Machining & Assembly concentrates its business on the design, manufacturing and delivery of machining lines and assembly systems. PWT machines and systems are specialized for the production of Cylinder Heads, Blocks and Transmissions, though they have also found effective application in other fields, as the aerospace or the power generation industry.

Machining

Comau's machine portfolio is designed to cover the whole range of our Customers products in type (Heads, Blocks, Crankshafts, Transmissions) size and material (Aluminium, cast iron, CGi, etc). Through the ability to engineer and deliver architectures and processes our solutions can meet the demanding requirements in flexibility and production scalability that the Customers require. The competence to develop transfer lines or to integrate them into hybrid solutions completes the technological offer.

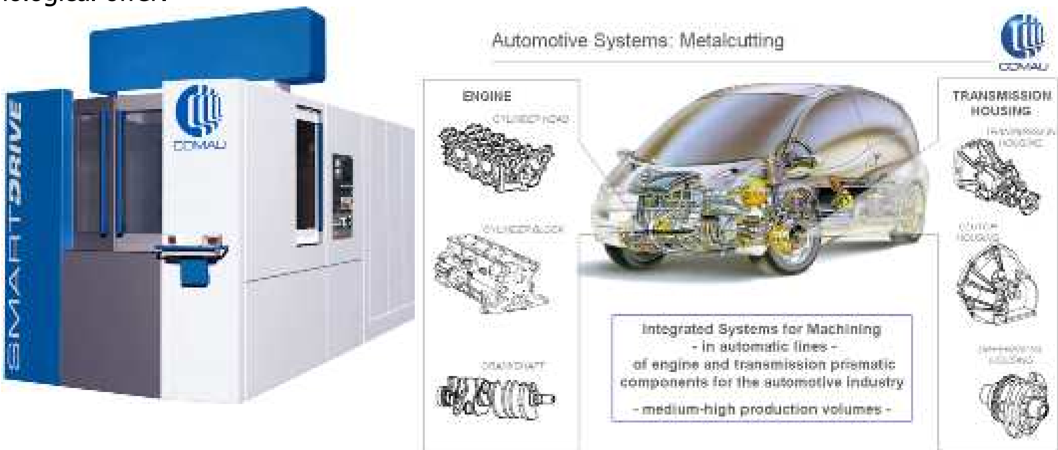


Fig. 4-6. SmartDriveComau and Metal Cutting Comau expertise

SmartDrive Comau

Comau condensates its long experience in the machining world in the new SmartDrive module. This is the latest family of CNC Machining Centers offering modular configuration to cope with different market requirements in terms of flexibility and production scalability, assuring high productivity, best performances and low life-cycle costs.

Comau offers its skills and ability to provide its customers complete engine production plant solutions, by supply modular machining centers and transportation systems.

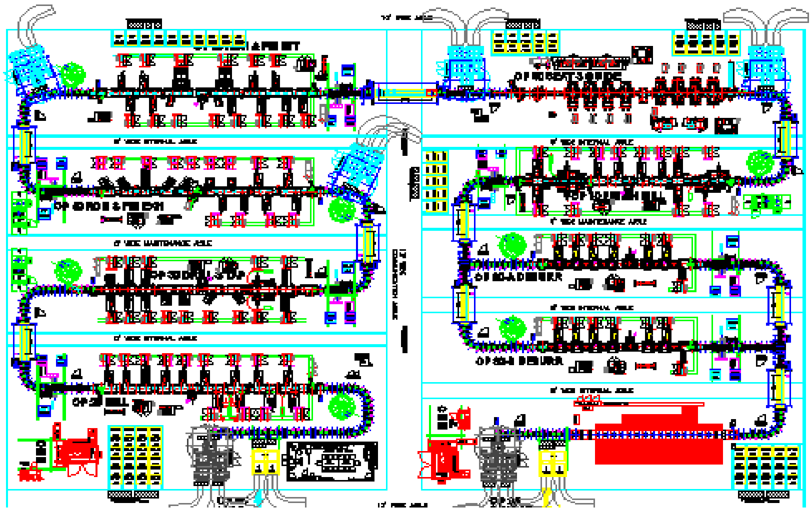


Fig. 4-7 Layout of a Powertrain plant

Assembly Standard modular solutions

Comau engineer, manufacture and install systems for the assembly of Engines (cylinder head, short block, long block), transmissions (automatic, manual, CVT) and components, as well as of ancillaries – valve seats and guides, plugs, caps within the metal cutting lines.

The supplied assembly systems are based on standard modular solutions, synthesis of the knowledge Comau has gained in the assembly processes oriented especially to the automotive market but not limited to.
In order to reduce development times and installation costs, Comau can provide standard assembly modules based on serial process or parallel operations.

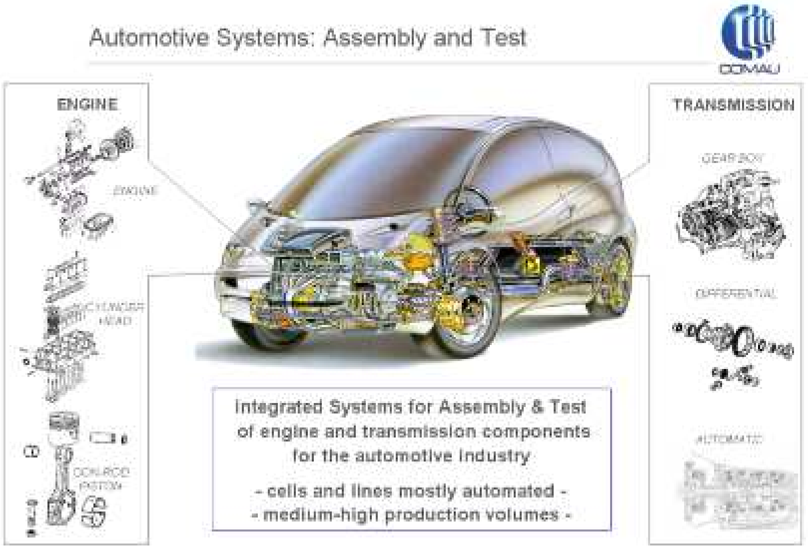


Fig. 4-8 Comau powertrain assembly expertise

SmartCell Comau

An assembly module designed for multiple assembly operations through interchangeable tooling and automatic tool change.

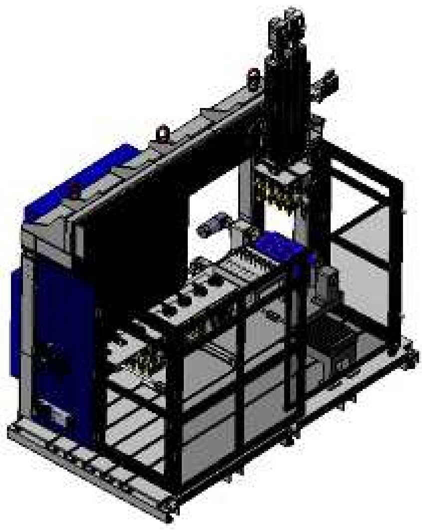


Fig. 4-9 SmartCellComau – parallel production flow assembly module

SmartBoxComau

A cell based on a standard cartesian structure able to manage all assembly operations, with different specific tools for specific tasks in cylinder head and engine assembly.

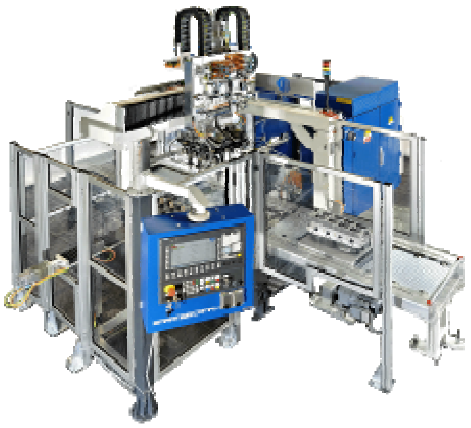


Fig. 4-10 SmartBoxComau – serial production flow standardized assembly module

Comau can provide all the complete production line for any particular component in its market area, integrating efficient transportation system dedicated to the customer's production location.

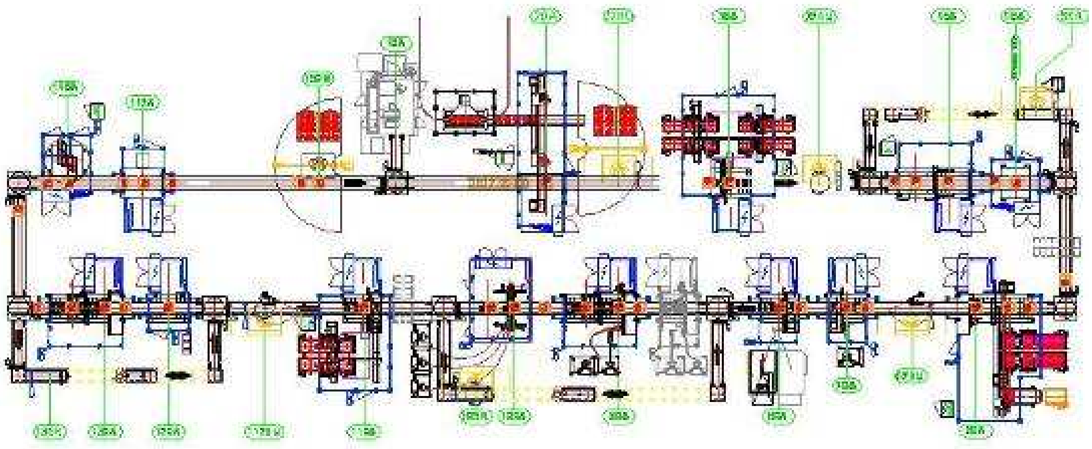


Fig. 4-11 layout of engine assembly automatic line

4.2.3 eCOMAU sustainability products

The industrial sector uses more energy than any other end-use sector, currently consuming about one-half of the world's total delivered energy. Energy is consumed in the industrial sector by a diverse group of industries and for a wide range of activities, such as processing and assembly, space conditioning, and lighting.

The benefits based on enhancing the efficient use of energy will drive to significant economical benefits and will permit to To achieve sustainability severe targets and improve them continuously.

To provide services in the sustainability area, Comau created in the 2009 a Task Force to launch new energy efficiency solutions, products and to perform development dedicated to the reduction of energy consumption in the production process.

eCOMAU is structure to provide to industrial customers specialized Consulting through highly specialized managers:

- Industrial Energy Managers
- Product / Process Experts focused on Design for Sustainability, Production Energy Efficiency, Sustainability Policy & Life Cycle Cost Analysis

Service activities through the Energy Efficiency Service Team:

- Team dedicated on achieving energy targets/objects on shop floor
- Skilled on Energy Efficiency Parameters, Method and Action Plan execution

And specialized Eco Products & Technologies by Comau's " Plug & Save Solutions"

- Comau Robotics: New Robot generation

- Comau Powertrain: Smart Drive
- GreenFit Solutions (Machining, Painting, Automation, Robotics)

eCOMAU is able to follow all the customer process in their reduction of energy requirement and CO2 emission up to the final certification of the low-emission production plant:

- Assessment
  - Machine architecture, layout and drawings analysis
  - Actual installed power and consumption measurement and analysis
  - Cycle diagram analysis
  - Energy vectors, worked yearly hours, maintenance costs data collect
- Intervention
  - Intervention definition (payback analysis based)
  - Engineering activity
  - Customer site activity
- Monitoring
  - New consumption measurements
  - Collected data analysis and saving certification

4.3 TNM Business and Expertise in Meat Production Lifecycle

TNM A/S is an IT company focusing on the agricultural sector in Denmark. Through close cooperation with other companies, trade associations and universities TNM A/S delivers state-of-the-art farming IT infrastructures and data analytics based on statistic and mathematical models. TNM collaborates closely with companies related food production chains such as feed producers, logistics, slaughter houses, and retail stores thus providing ebbits with an overview of technology in the food production lifecycles.

According to TNM experiences, In order to get a pork chop on the plate, a number of steps have to be taken. The major steps are visualized in Fig. 4-12 .

Fig. 4-12. meat production chains from feed, farms to retail stores.

We go through the entire chain of steps with a strong focus on the processes on the farm.

4.3.1 Field

Most fields can be harvested once a year. The typical process used to obtain crops is

- The field is ploughed and harrowed in order to prepare the soil for seeding. Ploughing turns over the upper layer of the soil which brings fresh nutrients to the surface. Furthermore, weeds and the remains of previous crops are buried, allowing them to break down. It also aerates the soil, and allows it to hold moisture better.
- The field is seeded with the desired type of crops.
- The field is fertilized. Both synthetic and organic fertilizers are used.
- The field is sprayed with pesticides and herbicides.
- The field may be watered.
- The crops are harvested.

All of these steps (except for the watering) are executed with the help of machines such as tractors and combine harvesters. These machines are often fitted with GPS devices which help optimize the use fertilizer, pesticides and herbicides.

The type of crop on a particular field is rotated in order to avoid the buildup of pathogens and balancing of nutrients over the years.

4.3.2 Feed Production

The harvested crops are transported to a feed production facility. We will ignore the crops that are processed for human consumption, such as the wheat processed into flour.

At the feed production plant crops are mixed with various other components to produce feed with specific nutritional values. The other components can be

- Residual parts from other food processing such as barley residue from beer brewing.
    - Fishmeal
    - Minerals

4.3.3 The Farm

Farms with animals come in different shapes. We will focus on the pig producing farm and briefly cover the farms producing dairy and poultry.

Pig Production

Pig production can be divided into two areas of expertise:

- Production of piglets
- Fattening of piglets

The fattening is sometimes divided into two sub-categories: Fatting from 7 kg to 30 kg and fattening from 30 to 100 kg. However, we will consider the fattening as one category.

Production of Piglets

In order to produce piglets the farmer needs to work intensely with his sows. The Fig. below shows the essential events in the life cycle of a sow on a farm.

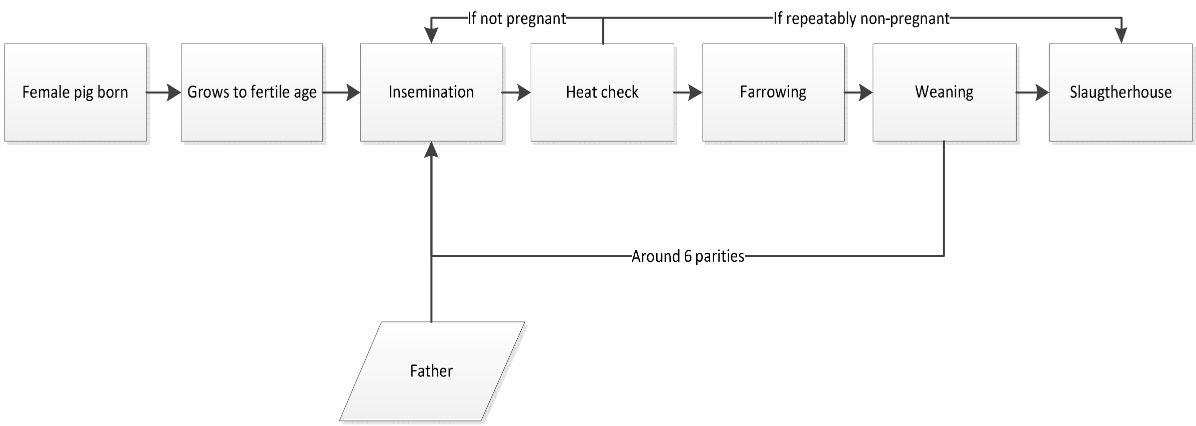


Fig. 4-13. Life cycle of a sow

When a female pig is born and it is expected to be a good breeding sow it is allowed to grow to its fertile age. Some farms are specialized in breeding good female pigs which they sell to other farms to use. When a sow is fertile it is inseminated with semen (either naturally or artificial) from a hog which typically is of another race as it yields the best meat. After a number of days the sow is checked to determine whether the insemination turned out to be a success. Then after about 116 days of pregnancy the sow farrows and gives birth to a number of piglets, typically between 10 and

15. The piglets stay with their mother until they age of 4 to 5 weeks where they are weaned from their mother. The piglets now go to the fattening stage. Fig. 4-13 has omitted the feeding and medication parts. The sow is fed according to a number of parameters

- Age and/or weight. If the farmer notices a sow that is thin he can increase the amount of feed given to the sow. conversely, he can decrease the feed portion if the sow becomes too fat.
- Day since insemination. As the embryos grow within the sow it needs more feed. Also it has been shown that decreasing the feed ration before farrowing gives a smoother birth.
- Number of piglets. When the sow produces milk for its piglets it needs extra feed if it has more piglets than usual.

As the sow gets older it starts to perform worse, that is, it gets fewer piglets. At some point the farmer chooses to introduce a new sow into production and send the old sow to the slaughterhouse.

Fatting of Pigs

Fattening of pigs is basically a matter of feeding the pigs optimally. However, as Fig. 4-14 shows we can also have factors such as medicine and climate

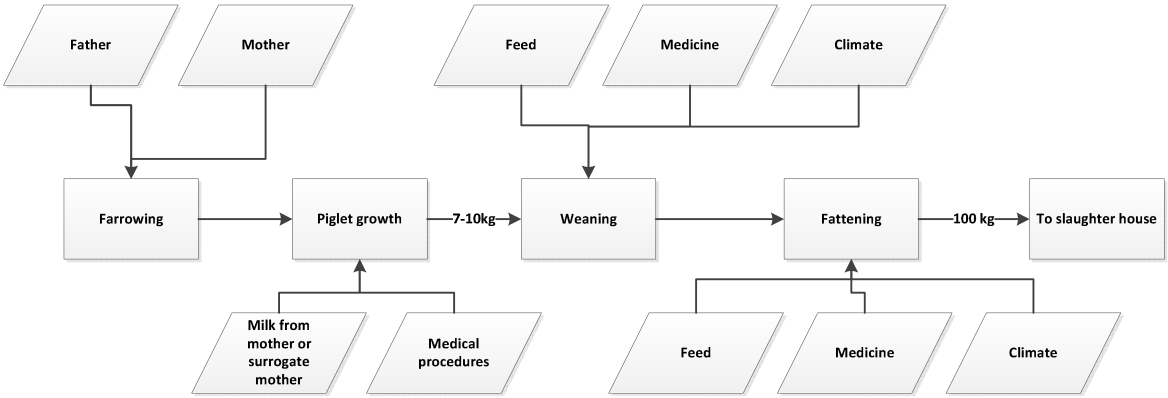


Fig. 4-14. Elements of fattening a pig.

The piglet is born when the sow farrows. By drinking milk from its mother it grows to about 10 kg until it is weaned. After that it is fattened until it reaches a weight of 100 kg where it is send to the slaughterhouse. The small piglets are sensitive to their climate (temperature, humidity etc.). A bad climate reduces growth, increases the risk of illness and, in the worst case scenario, death.
The piglets also undergo a number of medical treatments. Fore mostly, they are vaccinated against various diseases. The piglets are also treated with medicine if they catch some disease. Finally, male piglets are often castrated as their testicles produce a hormone which sometimes results in "boar taint" which is a very offensive odor.
When the pig reaches a weight of about 100 kg it is transported to the slaughterhouse.

Dairy

In order to produce milk the dairy farmer upholds a stock of cows. In order to get the cow to produce milk it must be bred and have a calf. A female calf may be raised to go into the dairy production while male calves are fattened in order to produce beef (often by another farmer). The calves are separated from their mother soon after birth and are given controlled amounts of milk/milk-replacement.
After giving birth to a calf the cow is in its lactation period (i.e. it produces milk) for approximately 300 days. It is important to milk the cow twice every day. Otherwise it ends its lactation period and needs to give birth again.
On a modern dairy farm the cows are milked by an automatic milking robot. Each cow is trained to go to the milking machine when its udder is filled with milk. The automatic milking robot is also 'instructed' in where the teats are placed in each of the cows found on the farm. That is, each cow can be recognized by the milking robot from the RFID in its ear tag.

Poultry

The poultry production comes in three categories

- Egg producing farms that produce eggs for human consumption.
- Breeders that produce fertilized eggs and hatch them.
- Broilers that fattens the chickens produced by the breeders.

4.3.4 Electronic Sensors and Devices on the Farm

The farm has a number of stationary electronic devices. The main uses can be summarized in four categories

- Management
- Feeding
- Climate
- Milking
- Other machineries such as trucks

Besides the 4 main categories we have other small devices for relatively specialized use. The four main categories are described with a section for each. After that we list a number of the other devices we know of.

We conclude with an overview of how it is possible to access the information stored and produced by the systems and sensors. Observe that none of the sensors are directly available. They are all placed behind some kind of interface.

In addition to the stationary equipment the farmer also uses mobile equipment. Tractors are often fitted with GPS equipment. However, these parts are not discussed here.

Management

Management is performed on a standard PC. By management we mean

- Economical entry, reporting and forecasting
- Management of the animals at the farm. That is, information about
  - Basic information such date of birth and parents
  - Performance information such as number of piglets of each farrowing or amount and quality of milk from a milking.
  - Location of the animals
- Economical information
  - Costs with respect to new animals, feed, electricity etc.
  - Income with respect to sold animals and produced milk.

The management system stores the information that it handles in a database of some sort (most likely a SQL database).

The management system helps the farmer improve productivity. It can tell him whether an animal is performing worse than average and should be replaced. It can also help him structure his every day work by generating lists of tasks based on the status of the farms animals.

Finally, the management systems assist the farmer in doing various reports requested by the public.

Vendors

- DLBR IT (DK) http://it.dlbr.dk/Forside.htm
- AgroSoft (DK) http://agrosoft.dk/
- AgroCom (DE) http://www.agrocom.com/

- AgroVision (NL) http://www.agrovision.com/

Feeding

A farm may have one or more automatic feeding systems. They can be put into two categories

- Valve feeding systems, which essentially do not know which specific animals they are feeding.
- Individual feeding systems, which knows what animal it is about to feed.

Furthermore, the animals may get feed from grassing on fields. There is currently no tracking of this kind of feed available.
The feed system is obviously very important to farmers that produce animals. Each animal needs to get at the amount of feed that gives the best growth rate. Too little feed and the animal grows more slowly, too much and the feed is wasted.

Valve feeding system

A valve feeding system mixes feed and outputs feed at a number of valves. A valve is typically placed such that the feed is put into a feed trough for the pigs to eat. A valve is typically configured with information about

- Number of animals that have access to the feed from the valve.
- The (average) weight of the animals
- The (average) day since last insemination of the animals
- An identifier that specifies a feed curve. The feed curve maps weight of day since last insemination to an amount of feed per animal.
- An identifier specifying the feed mix to be used. The feed curve may also incorporate information with respect to what feed mix to use.
- The time at which to mix the feed and send it to the valves.

Individual feeding systems

An electronic feeding system is build such that only one animal can access the feed valve. When an animal comes to the valve the animal is recognized using the RFID tag in its ear tag. The animal is now given feed according to parameters such as:

- Age of the animal
- The day since last insemination
- An identifier specifying a feed curve.
- An identifier specifying a feed mix.
- Number of portions to split the daily ration into.

The individual feeding systems give various possibilities to track the feeding of each animal.

Vendors

- Big Dutchman (DE) http://www.bigdutchman.de/ (pig)
- Skiold (DK) http://skiold.dk/ (pig)
- Nedap (NL) http://www.nedap.com/ (cow and pig)
- Schauer (AT) http://www.schauer.co.at/index.php?L=0 (cow and pig)
- Lely (NL) http://www.lely.com (cow)

Climate

The climate control of a house uses a number of sensors. Examples of sensors used by a climate controller are

- Indoor temperature sensor(s)
- Indoor humidity sensor(s)
- Outdoor temperature sensor
- CO2 sensors
- Airflow meters

The job of the climate control is to ensure an indoor climate (temperature, humidity being the most important parameters) that is as close to optimal for the growth of the animals in the house as possible. Many animals are not capable of sweating which means that a hot environment can be very stressful to the animals. If its too cold the animals need to use energy to keep warm. In both cases it limits the growth of the animal compared to the optimal.
In order to enforce some control on the indoor climate the climate control utilizes one or more of the following techniques:

- Natural ventilation where air ducts are opened letting warm air out.
- Active ventilation using fans.
- Evaporation of water to cool the air. The water is sprayed into the air. The technique is also used to reduce the amount of dust in the air.
- Heating elements to increase temperature.

It should be possible to retrieve details on most of these operations, e.g., the energy used in the heating elements.

Vendors

- Big Dutchman (DE) http://www.bigducthman.de
- Skiold (DK) http://www.skiold.dk
- Fancom (NL) http://www.fancom.com

Milking

A milking robot handles the milking of the cows. It identifies the cow via an RFID tag embedded in the ear tag of the cow. The milking robot frees the farmer from milking his cows twice every day.
The milking robot is configured with parameters such as

- The position of the teats for each cow

The milking robot is most likely combined with some sensors that analyze various parameters with respect to the milk. Examples of parameters

- Amount of milk /milk yield.
- Flow of milk
- Conductivity
- Blood
- Cell count

These parameters are usually stored in a database.

Vendors

- Delaval (SE) http://www.delaval.com
- Lely (NE) http://www.lely.com
- GEA Farm Technologies (DE) http://www.gea-farmtechnologies.com/hq/en/

Other Machinery and Electronic Equipment

Movement sensors

Records and detects movements that can be sign of illness or that the animal is in heat. The movement sensors can be attached to the animal using a necklace or by being attached to the leg.
Examples

- Nedap (NL) http://www.nedap-lactivator.com/

Water consumption sensors

Records water consumption which can be used to detect abnormal behavior and in turn illness.
Examples

- Skov (DK) http://www.skov.com

Weighing devices

A farm may have a number of weighing devices. For example, weighing devices may be used for measuring the weight of

- Feed delivered by truck
- Pigs send to slaughterhouse
- Crops harvested on the field

Manure Treatment

Manure treatment systems use a range of different methods to improve value of manure and/or reduce problems caused by the manure such as smell and pollution.
Examples

- Infarm (DK) http://www.infarm.dk

Crops treatment

Harvested crops may need drying before being stored more permanently.

- Kongskilde (DK) http://www.kongskilde.com

4.3.5 The Slaughterhouse

When the animals arrive at the slaughterhouse they are killed and cut into 'manageable' but still relatively large pieces. Animals are inspected and samples are taken in order to detect signs of diseases and infections.

The slaughterhouse may also record the quality (e.g., meat percentage) of the delivered animals which is used to compute the price of the animal. This information is reported back to the farmer. The process of pig slaughtering is slightly different from company to company as well as from country to country that is affected by government regulations and culture / religious believes. An example of pig slaughtering process flow in China is depicted in Fig. 4-15.

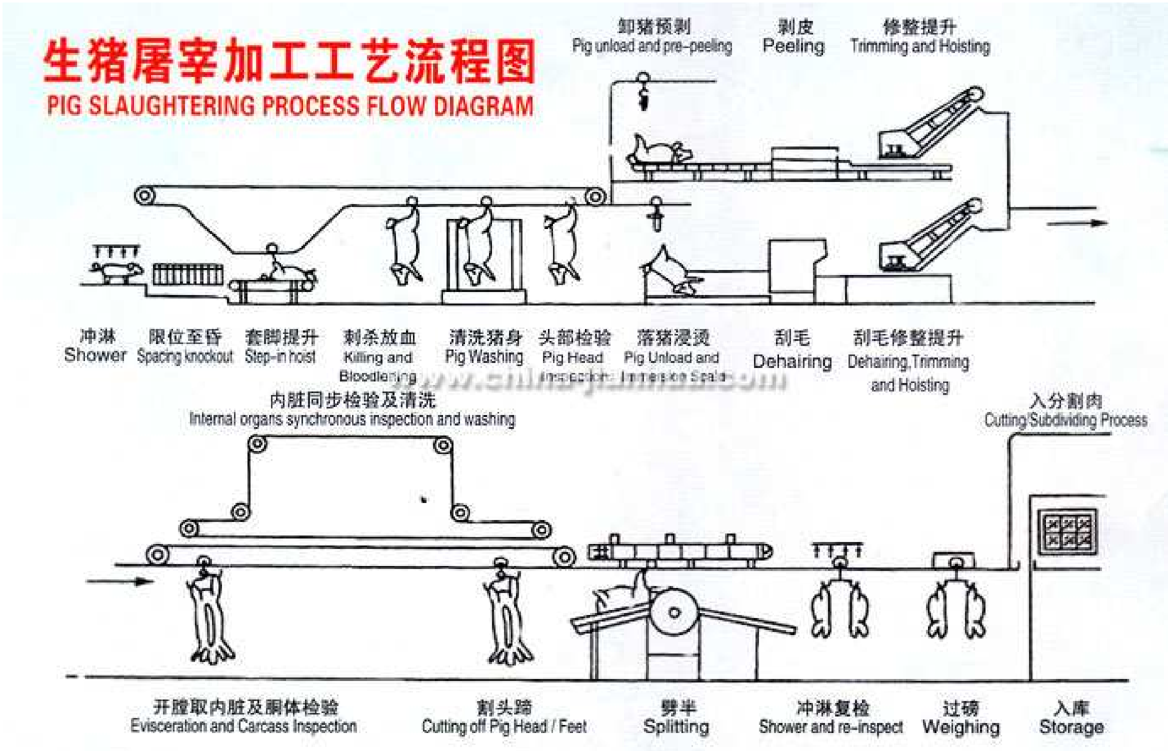


Fig. 4-15 Pig Meat Slaughtering Process Flow[1]

The major processes can be summarized as follow[2]:

Receiving

Pigs are usually slaughtered after 4-7 months. Pigs intended for pork are usually slaughtered 1-2 months younger than pigs for bacon. The pigs are transported with trucks that have compartments with an individual capacity of 12-15 pigs. On arrival, they are unloaded and driven in large pens having a capacity equivalent to a truck compartment. The pigs are held there for 24 hours to recover from fatigue and stress; and they are provided with enough water to flush out intestinal pathogenic bacteria. Moreover, health inspections can be held during that holding period. The live animals are weighed prior to processing so that yield can be accurately determined.

Stunning

Before slaughtering, pigs undergo electrical or carbon dioxide stunning. In the first case, they are stunned using high frequency (50 Hz), low voltage electric current applied by means of two electrodes, which are placed on either side of the brine using tongs. The current induces a state of immediate epilepsy in the brain during which time the animal is unconscious. In the later case, the pigs are passed through a well with a CO2 and air atmosphere. Legally a minimum of a 70 % concentration of CO2 by volume is required, but a 90 % concentration is recommended. The pigs are again rendered unconscious due to the acidification of the cerebrospinal fluid upon inhalation of the CO2. With the CO2 method “blood splashing” is eliminated, and it also removes the human element required in the electrical stunning. During their state of unconsciousness, the pigs are hoisted onto an overhead rail for slaughtering.

Sticking & bleeding

In a state of surgical anesthesia, the pigs are shackled and hoisted for exsanguinations. The stunned animals undergo exsanguinations (sticking) with blood collected through a special floor drain or

[1] Retrieved from http://www.china-jianhua.com/Upload/PicFiles/2009.4.28_3.23.49_7790.jpg on Sept. 17th 2010.

[2] Retrieved from http://www.hyfoma.com/en/content/food-branches-processing-manufacturing/meat-fish-shrimps/pig-slaughtering/ on Sept, 17th 2010

collected in large funneled vats or barrels and sent to a rendering facility for further processing. The carotid artery and jugular vein are cut to drain out blood and to get the muscles relaxed for easy dehairing. Pigs should be allowed to bleed for about 5 minutes.

Scalding & Dehairing

Pig carcasses are not skinned after exsanguinations. Instead, the carcasses are dropped into scalding water which loosens the hair for subsequent removal. The carcasses should be kept under water and continually moved and turned for uniform scalding. In large plants, carcasses enter the scalding tub and are carried through the tub by a conveyer moving at the proper speed to allow the proper scalding time. During the hard-hair season (September-November), the water temperature should be 59° to 60°C and the immersion period 4 to 4,5 minutes, while in the easy-hair season (February-March), a temperature of 58°C for 4 minutes is preferable. In small plants without automation, hair condition is checked periodically during the scalding period. The dehairing process is begun with a dehairing machine, which uses one or more cylinders with metal tipped rubber beaters to scour the outside of the carcasses. Hot water (60°C) is sprayed on the carcasses as they pass through the dehairer moving toward the discharge end. The carcasses are removed from this machine, hand scraped, then hoisted again, hind quarters up. The carcasses are hand-scraped again from the top (hind quarters) down. Any remaining hairs can be removed by singeing with a propane or similar torch. Once the remaining hairs have been singed, the carcasses are scraped a final time and washed thoroughly from the hind feet to the head. Some plants pass the carcasses through a singeing through gas flames.

Evisceration

After scalding and dehairing, singeing, or skinning, the head is severed from the backbone at the atlas joint, and the cut is continued through the windpipe and esophagus. The head is inspected, the tongue is dropped, and the head is removed from the carcass. The head is cleaned, washed, and an inspection stamp is applied. Following heading, the carcass is eviscerated. The hams are separated, the sternum is split, the ventral side is opened down the entire length of the carcass, and the abdominal organs are removed. These viscera are received in a moving gut pan to segregate edible (heart, liver) and non edible offal. Intestines are cleaned for sausage casings. The thoracic organs are then freed. Non edible offal is discarded into a barrel to be shipped to the rendering plant.

Splitting

The carcass is cut into two halves. The meat is controlled.

Washing

The carcass is then washed from the top down to remove any bone dust, blood, or bacterial contamination. A mild salt solution (0.1 M KCl) weakens bacterial attachment to the carcass and makes the bacteria more susceptible to the sanitization procedure, especially if the sanitizing solution is applied promptly. Dilute organic acids (2 percent lactic acid and 3 percent acetic acid) are good sanitizers. In large operations, carcass washing is automated. As the carcass passes through booths on the slaughter line, the proper solutions are applied at the most effective pressure.

Cooling

Cutting and deboning are easier to carry out at lower temperature. Therefore, the carcasses are transferred to chill tunnels and chill rooms to cool them down to 0-1°C with air velocity typically 5 to 15 mph, equating to –5°C wind chill, for a 24-hour chill period. For thorough chilling, the inside temperature of the ham should reach at least 3°C. With accelerated (hot) processing, the carcass may be held (tempered) at an intermediate temperature of 16°C for several hours, or be boned immediately. When large numbers of warm carcasses are handled, the chill room is normally pre-cooled to a temperature several degrees below freezing –3°C, bringing the wind chill to –9°C to compensate for the heat from the carcasses

Cutting into smaller pieces

The carcasses are processed into 3 cuts of meat (fore-end, middle and hind leg). During further cutting into smaller pieces, the slaughters are assisted in their work by automated transport trays and conveyors. They help in cutting and sorting meat and bone. The products are finally efficiently packaged and stored at low temperature prior to further processing.

4.3.6 Retail Store / Food Production

The retail store received a batch of carcasses from a slaughterhouse transported by a logistic firm. Each batch is normally given id using RFID or Barcode. The store registered this delivery id and the id of the batch in its ERP system.
It further processes the meat into consumer sized packages which goes into the store. The half pig cut from slaughterhouse is firstly cut to primal cuts then again into retail cuts. There exits different ways for cutting pork into primal cuts e.g. British, American, and French Cut. The primal cuts determine the quality and price of the meat sold to end consumers. Pieces of the meat that cannot be sold as cut to the end consumers are normally processed to other kind of food such as sausages.

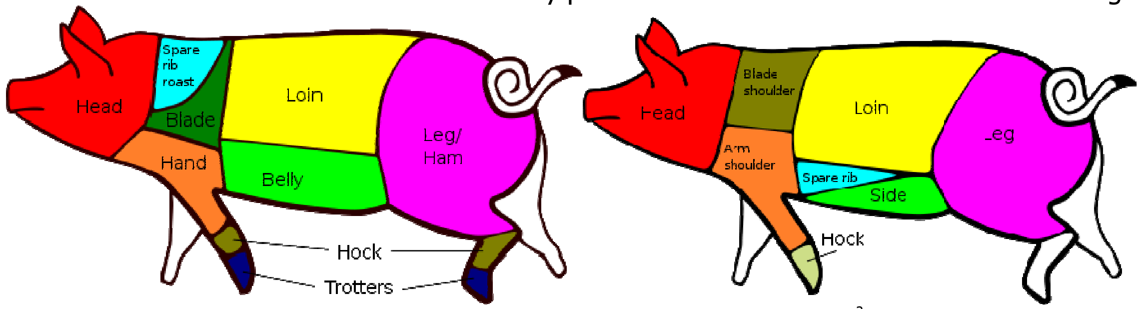


Fig. 4-16 Primal Cuts of Pork (British left and American Right)[3]

Making the rest meat into sausages involved a complex food production processes such as mixing with other substances to increase the taste and consistence, preserving the food, packaging the pieces into retail size (depicted in Fig. 4-17). Once the pieces are packed in into a packaging, each packaging is labeled using Barcode or RFID and registered into company's ERP system.

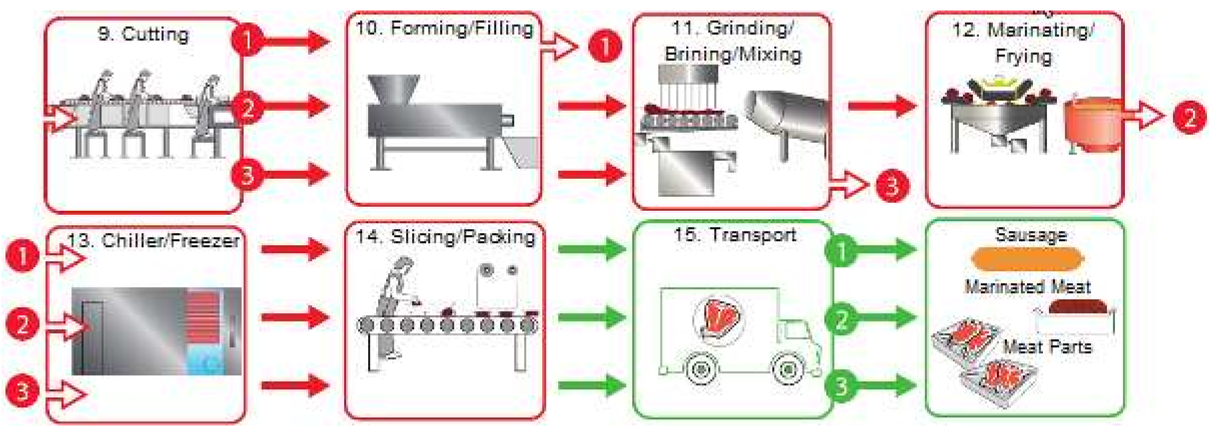


Fig. 4-17. Example of Food Production of Pork Meat

[3] Image retrieved from http://en.wikipedia.org/wiki/Pork on Sept. 17th 2010.

5. Multi-sensor Data Fusion

5.1 Definitions

The concept of Multi-sensor data fusion concept is born from humans and animals behavior as they learn using multiple senses for inferring situations. Over the past few decades, researchers have tried to mimic this behavior to sense situation within military and non-military applications. In the past, there have been many ambiguous terms and definition caused by the broad range application of multi-sensor fusion. In order to prevent further confusions, The Joint Directors of Laboratories (JDL) tried to standardize Multi-sensor fusion definition. They suggested multi-sensor fusion as (White 1991): "Multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from multiple sources." This definition was revised by Hall as "the combination of data from multiple sensors, and related information provided by associated databases, to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor alone." (Hall and Llinas 1997). Wald argued that the previous definitions focus only to methods, means and sensors, he defines Multi-sensor fusion as "data fusion is a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application." (Wald 1999).

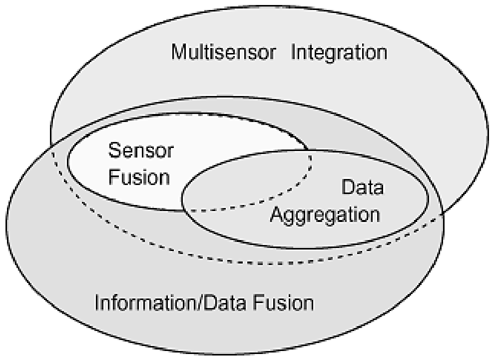


Fig. 5-1. The relationship among the fusion terms: multisensor/sensor fusion, multisensor integration, data (Nakamura, Loureiro et al. 2007)

Multi-sensor data fusion has also been named differently in many previous works for instance in robotics, computer vision (Luo and Kay 1995), and industrial automation (Brokmann, March et al. 2001), it is also known as multisensory integration. Lou defines multisensory integration as "the synergistic use of information provided by multiple sensory devices to assist in the accomplishment of a task by a system; and multi-sensor fusion deals with the combination of different sources of sensory information into one representational format during any stage in the integration process." (Luo, Chih-Chen et al. 2002).

5.2 Models and Classifications

Data fusion model describes various tasks and processes that need to be considered when fusing data and information. There have been various models defined to get a common understanding of sensor fusion.

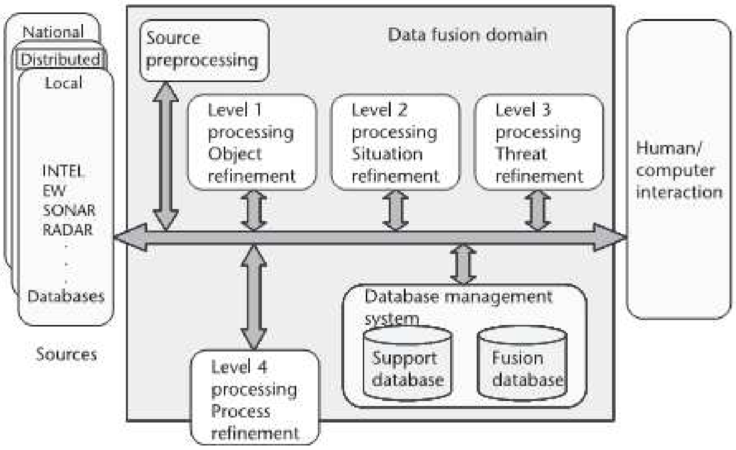


Fig. 5-2. The JDL Model (Liggins, Hall et al. 2009)

5.2.1 Information Based

Information based models focuses on abstracting the information handled in the tasks. This model represents the first generation of multi-sensor fusion. The first generic model was introduced by a data fusion working group of Joint Directors of Laboratories (JDL), a joint effort within the U.S. department of defense. However it was intended for decision support system in defense systems. JDL model introduced several components that cover sources, 5 processing levels, a database, and a human computer interaction. The sources receive input from various sensors, a priori knowledge, and human. The database is responsible to maintain the data needed by processes. HCI allows human operators to provide query, input knowledge etc.

The processes consist of 5 levels:

- Level 0 / *pre-processing*, also known as *process alignment or sub-object data association* aims at obtaining initial information about target's characteristics by combining the raw data (e.g.: signal, pixel).
- Level 1 processing /*object refinement* focuses on combining sensor data to estimate position, velocity, and attributes of the observed objects for supporting prediction to future values.
- Level 2 processing / *situation refinement* tries to describe relationship among the current entities, events and their environment which also includes clustering and relation analysis.
- Level 3 processing / *significance estimation* tries to project the current situation to the future to draw inference about possible threats, friend or foe vulnerabilities, and opportunities.
- Level 4 processing, / *cognitive refinement* is a monitoring process of the data fusion performance. This process also tries improving the performance. This is a part of resource management.

Nakamura summarized information fusion based on several aspects such as relationship among input data, the abstraction level of the manipulated data during fusion process, and the abstraction level of the input and output of a fusion process

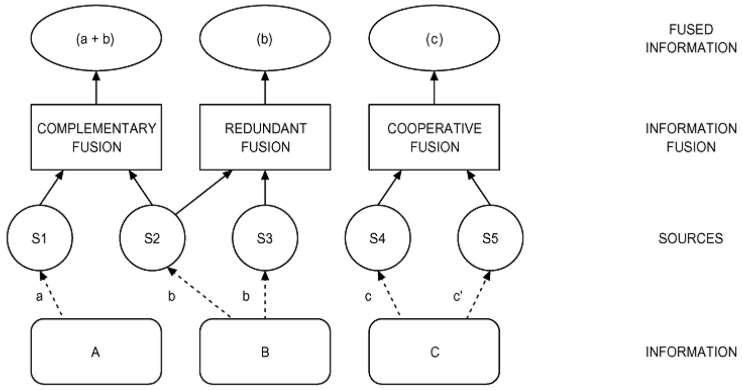
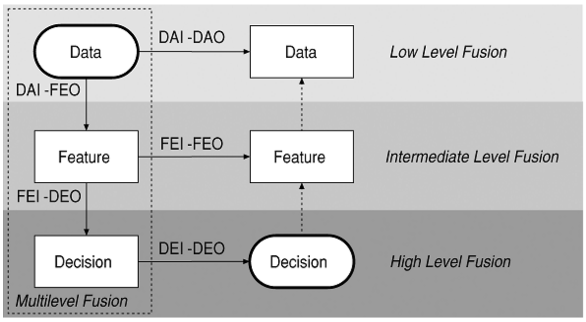


Fig. 5-3. types of information fusion based on the relationship among the sources(Elmenreich 2002)

Fig. 5-3 depicts, the relation among sources classify fusion to complementary when the information from different sources represents different pieces of a broader scene for instance several temperature sensors which are deployed to cover a wide area, redundant when different sources present the same pieces of information which is used to increase confidence e.g.: sensors, which are deployed with an overlapping range, and cooperative if the pieces of information can derive a new more-complex information e.g.: several type of sensors can infer condition such as storm, which none of the atomic sensor can provide.

Luo proposed four level of information fusion abstraction: signal fusion that deals with n-dimensional input data of sensors, pixel fusion that deals with images, feature fusion which extracts attribute from data, and symbol fusion that take decision of object recognition based on information (Luo, Chih-Chen et al. 2002). This classification raises ambiguity since one could also treat pixels in image as a 2-dimensional signal. Dasarathy proposed a Data-Feature-Decision model, which consists of 5 categories based on input and output of the fusion (Dasarathy 1997): Data In–Data Out (DAI-DAO), Data In–Feature Out (DAI-FEO), Feature In–Feature Out (FEI-FEO), Feature In–Decision Out (FEI-DEO), Decision In–Decision Out (DEI-DEO). Its contribution is that the input and output are abstracted clearly. Moreover, system performance is always tuned by decision block by examining the feedback from other blocks. However it is not clear how the processes in other blocks are improved. Iyengar proposed a 3-level abstraction that extends the previous classification: Low level / signal measurement level fusion, Medium level / Attributes or features fusion, High level / Symbol or decision level fusion (Iyengar, Chakrabarty et al. 2001). However, based on Dasarathy's work (Dasarathy 1997), Nakamura added one more level that is able to process data from different level Multi level when data from different level are fused and can results in any level of abstraction. (Nakamura, Loureiro et al. 2007).



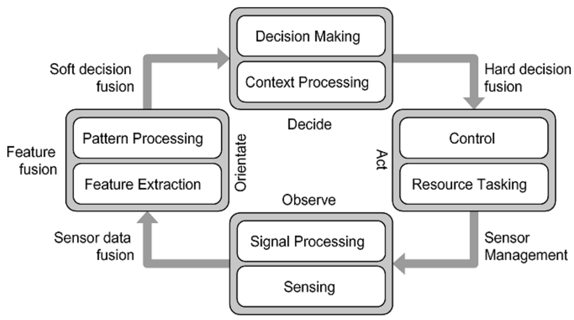


Fig. 5-4. The Dasarathy Model/DFD (left) (Nakamura, Loureiro et al. 2007) and The Omnibus model (Right) (Bedworth and O'Brien 2000)

5.2.2 Activity Based

This model concentrates on the abstraction of activities and their sequences of execution. Boyd introduced a control loop cycle, which is known as OODA cycle (observe-orient-decide-act). Information is first gathered in observe phase, then it is fused in the orient phase to get an interpretation of situation, action plan is defined in decide phase, and it is executed in act phase.

Mascolo and Musolesi implemented this model in SCAR algorithm to do routing in wireless sensor network (Mascolo and Musolesi 2006).

The U.K. Intelligence community introduced a process of developing a raw data into intelligence to support decision making. The activities consist of Collection that gather raw data from environment, Collation that compares, analyses, and correlates the data, Evaluation that fuses data and information, and Dissemination that deliver the result to users who produce decisions.

The disadvantage of the two models is that they only describes the main tasks and does not represent specific tasks of information fusion. Comparing to the JDL model, *Observe* in OODA and *Collection* in Intelligence model correspond to level 0. *Orient* covers level 1, 2, and 3 whereas *Collation* only covers level 1 and *Evaluation* covers level 2 and 3. *Decide* in OODA and *Dissemination* in *Intelligence* match up the level 4. *Act* is not covered by JDL model.

The omnibus model describes the specific tasks of the fusion processes explicitly in a cyclic sequence. The model initially gives an overall perception of the system, which then on the next iterations it goes into more detail subtasks. The omnibus model is enhances of OODA cycle by specifying detail activities representing the fusion tasks that have to be executed in each stage. These activities are similar to Waterfall model.

5.2.3 Role Based

This model shifts the focus on how information fusion can be modeled. In contrast to previous models, role based models do not specify specific task and activities of information fusion explicitly. On the other hand they specify the roles of components and their relationships to each other. Object oriented model, introduced by Kokar defines four roles: *Actor* who is responsible to interact with real world in terms of collecting information and acting, *Perceiver* assesses information and provides contextualized analysis to the director, *Director* builds an action plan specifying the system goals, and *Manager* controls the actor to execute the plans.

Frankel discussed architecture of human fusion consisting of local and global processes. The local estimator manages the execution of activities based on goals and timetables provided by global process. The global process refines the goals and timetables based on the feedback provided by the local process. Together with Bedworth, he modified the model for machine fusion. The improved model separates control-estimation and goal setting-goal achieving behavior as depicted in Fig. 5-5. Right. The local process act as an estimator trying to achieve goals by going executing the following processes: *sense, perceive, direct, manage, effect*. *Sense* gathers raw data from various sources and sends stimuli to the next process. *Perceive* handles stimuli according their relevance (focus) and informs controller about the stimuli used. *Direct* alerts controller with feedback and receives new goals and timetables (desires). Based on the objectives, *Manage* activates the controller to define what is practical, and then it sends its feedback as expectations about provided decision to the controller. *Effect* executes selected decisions.

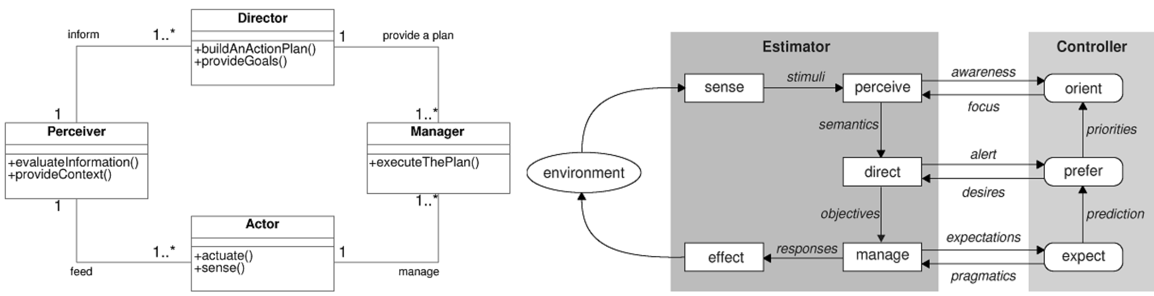


Fig. 5-5. Object oriented model (Left) (Kokar, Bedworth et al. 2000) and (Frankel and Bedworth 2000) (Right)

Global process controls the execution of local process. It has three tasks: *Orient* configures the importance of stimuli. *Prefer* prioritizes the most relevant aspects to reach the goals. *Expect* predicts and filters intentional objectives to determine what is practical to the estimator. In practice, the global process will likely be done through human query and definition of operation guidelines representing priorities, desires and pragmatics. Implementation examples of this model are Directed Diffusion (Intanagonwiwat, Govindan et al. 2000) and TinyDB (Madden, Franklin et al. 2005).

The role based model contribution is that it defines the actors and their role in information fusion tasks. However this model does not specially address the attributes of service oriented architecture, wireless sensor network, and internet scale distributed system.

5.2.4 WSN Model

Sensor fusion in WSN is closely related to wireless communication due to limited resources characteristic of WSN. In-Network aggregation aims to reduce the resource consumptions by applying local processing and fusing data while it is being routed to the sink node. In WSN model, the network topology influences how the data fusion is applied. For instance, in multi hop network The Directed Diffusion takes advantage of publish-subscribe events. The nodes in paths subscribe to the event that a sink is interested in and they implement filters that decide whether the data should be fused, forwarded or dropped. In hierarchical network, LEACH (Heinzelman, Chandrakasan et al. 2000) does the data fusion in each of the cluster head. Since being a cluster head takes more resources, it rotates the role of cluster heads to other nodes so that the resource consumptions is equally distributed among the nodes. Nakamura proposed a hybrid solution (multi hop-hierarchical), in which the leaf nodes might communicate using multi-hop with the cluster heads. And the communication among the cluster-heads can also use multi hop. They proposes that reactive algorithm that assigning roles only when an event is detected and using the shortest path for notifications to perform data aggregation(Nakamura, Oliveira et al. 2006).

A well known approach is to treat the whole WSN as a virtual DBMS which is known as data-centric approach. This approach does not address node individually but it only address specific data or events. Retrieving information from WSN is done by broadcasting a "SQL-like" query which will subscribe to interesting events in a specific region. When nodes in the region sense the corresponding events, they forward the data to the sink through other nodes and the nodes lies in the paths aggregate the data. An example of this approach is TinyDB, Cougar (Yao and Gehrke 2002), MiLAN(Heinzelman, Murphy et al. 2004), SINA (Jaikaeo, Srisathapornphat et al. 2000), DSWare(Li, Son et al. 2003).

Another work by Wang investigated the use of mobile agent for sensor fusion in WSN (Wang 2004). Mobile agent program migrate from node to node carrying the saved state of a program which is then continued when the agent arrived at the destination node. He claims that compares to client-server model, mobile agent save network bandwidths and network latency when the number of nodes is large.

5.3 Existing Methods for Information Fusion

Techniques and methods in fuse data involve a various traditional disciplines such as digital signal processing, statistical estimation, control theory, artificial intelligence, and classic numerical methods.

5.3.1 Inference

Nakamura classifies the existing methods based on their purposes into several classes such as inference, estimation, classification, feature maps, abstract sensors, and compression.

The famous inference strategies that are often used in information fusion are Bayesian inference and Demspter-Shafer Belief Accumulation theory. Bayesian inference solves uncertainty by applying conditional probability theory (Bayes Rule) to the evidence obtained from the observations.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

P(Y|X) represent the posterior probability that wanted to be predicted based on the priors. P(Y) represents the prior probability of Y. P(X|Y), represent the current likelihood of X given Y, and 1/P(X) is a normalizing factor. The problem of Bayesian inference is that P(X) and P(X|Y) have to be estimated or guessed. Bayesian inference has been applied for several works for instance, to reach better accuracy and robustness an advance driving assistance fuses data from different sensors such as laser, radar and video (Coue, Fraichard et al. 2002), to determine whether an electricity voltage is stable by observing three stability indicators of a power system (Sam, Nwankpa et al. 2001).

Dampster-Shafer framework generalizes Bayesian theory. The advantage of this framework is, it can be used for incomplete knowledge representation. It is more flexible than Bayesian inference since it allows each source for contributing information with different levels of detail. Moreover, in Dampster-Shafer inference, an a priori probability to unknown preposition is not needed. The probabilities are only assigned when the supporting information is available. However Bracio et al. pointed out that there is a trade-off between the flexibility of Dampster-Shafer and the accuracy of Bayesian inference(Bracio, Horn et al. 1997).

Yu et al. use Dempster-Shafer to build a dynamic picture of battlefield for situation assessment(Yu, Sycara et al. 2004). In DSWare, every decision is associated with a confidence value that is calculated based on Damster-Shafer theory(Li, Son et al. 2003). Nakamura also used this framework for detecting routing failures and trigger topology construction in his Topology Rebuilding Algorithm(Nakamura, Nakamura et al. 2005).

Fuzzy logic uses approximate reasoning to draw conclusions from imprecise premises. Each input is fuzzified by a membership function. The fuzzy rules produce an output that is defuzzified by output rules. (Cui, Hardin et al. 2004) uses fuzzy logic to deal with incomplete and uncertainty of information in a control system for localizing hazardous contamination sources. (Chan Yet and Qidwai 2005) fused information from ultrasonic sensors to detect potholes and obstacles for an autonomous robotic navigation system. Another example was a work of Gupta et al. and Halgamuge et al. who use fuzzy logic to infer the best cluster heads in WSN. Halgamuge used node concentration, energy level, and centrality as features (Halgamuge, Guru et al. 2003; Gupta, Riordan et al. 2005).

For automatic target tracking (ATR), neural network has also been used since it offers several degree of parallel processing. Baran uses neural network as associative memory guiding the pattern matching process for target recognition (Baran 2002). Cain et al. classifies targets using neural network based on the information obtained from infrared sensors and a ultraviolet laser radar (Cain, Stewart et al. 1989). Besides, ATR, neural network has also been used for other purposes such as fusing audio–visual information in speech recognition (Lewis and Powers 2002) and fusing edge maps from image, optical, and infrared sensors(Yiyao, Venkatesh et al. 2001).

Friedlander introduced a In-network inference process by exchanging only the semantic interpretations(Friedlander 2005). Semantic information fusion needs a knowledge base to infer and fuse data. In his work he explains, that extracting semantic information from sensor network was done by converting sensor data to a formal language. The result is then compared with the known knowledge. He applied this idea to recognize a behavior of a robot based on its trajectory. Whitehouse defined a system that allows users to formulate queries over semantic values. Consequently, the answers are semantic interpretation obtained by the in-network inference processes. Another example is done by Liu and Zhao who decompose declarative queries into graphs that are used for composing services (Liu and Zhao 2006).

5.3.2 Estimation

Estimation plays an important role in the level 1 of JDL model. Estimation techniques in data fusion reduce the uncertainty of sensed data as a result of sensors physical limitation such as noise, accuracy, reliability, and coverage. the most well known techniques to estimate the raw data includes maximum likelihood, least squares, moving average filter, kalman filter, and particle filter. Maximum likelihood estimate (MLE) tries to find the parameter values that most likely have produced the data. It is suitable to estimate state that is not an outcome of a random variable (Brown, Durrant-Whyte et al. 1992). Xiao et al. improves unreliable communication link in WSN by using distributed and localized MLE, in which every node calculate a local estimate that converge into a global maximum likelihood(Xiao, Boyd et al. 2005). MLE is normally used in location discovery such as obtaining distance, direction, and angle estimations (Patwari, Hero et al. 2003; Lei, Wenliang et al. 2005).

Another method based on Bayesian theory is called Maximum a Posteriori (MAP). Similar to MLE it tries to find the most likely value for a state, nonetheless MLE assumes that the state is fixed to unknown parameter space, while MAP assumes the state as an outcome of random variable whose prior probability distributed function is known. Schmitt et al. used MAP for finding the joint positions of a robot and to track positions of moving objects (Schmitt, Hanek et al. 2001). Rachlin claims that the implementation of MAP is too expensive for current sensor nodes(Rachlin, Negi et al. 2006). However, Shah et al. proposed a more efficient implementation by using the estimators as maximum concave function which allows the use of simple numerical maximization algorithm.

Least square is an optimization method that finds the best fits function for a set on input measurement by minimizing the sum of the square error between the values generated by the function and the values obtained from the measurement. Minimizing the square error can be done by using ordinary squared error, Huber loss function, and root mean square error. The least square error is suitable if the parameter to be estimated is fixed. Rabat and Nowak shows that Huber loss function is more suitable in real cases where noisy measurements happen frequently than ordinary square function (Rabbat and Nowak 2004). Willet used a least squares algorithm in spatial sampling algorithm to define a subset of sensor nodes that provide an initial estimate of the environment(Willett, Martin et al. 2004). This technique can be used to build a spatial map and mobile node guidance while building the map(Singh, Nowak et al. 2006).

Moving average filter comes from digital signal processing technique. It computes the mean of a measurement values to produce the output signal. The number of measurement to be averaged depends on the window size. Deciding the window size poses a tradeoff between the ability of the algorithm to reduce noise and the ability to detect the change in signal level. Yang et al used this filter for reducing error on target location in a trekking application. Nakamura used this filter to estimate data traffic in WSNs. Another variant of this filter is named Exponentially Weighted Moving Average, which has multiplying factors on each value and this factors decrease exponentially. Blumenthal used this variant to estimate distances in localization algorithms.

Kalman-Filter is one of the most famous fusion methods. Kalman filter is an optimal recursive data processing algorithm. Kalman-filter finds the most optimum averaging factor for each consequent state. Also somehow remembers a little bit about the past states. Kalman filter incorporates all information that can be provided to it. It uses knowledge of the system and measurement devices, statistical description of the system noises, measurement errors, uncertainty, and any initial condition of the variable of interest. Kalman-Filter is suitable for fusing low level redundant data which has a linear model and its error can be modeled as Gaussian, however when dealing with non-linear models, Extended Kalman Filter or Unscented Kalman Filter should be used. Kalman Filter has been used extensively for source localization and tracking in robotics (Brown, Durrant-Whyte et al. 1992) and in WSN it is used to refine location and distance estimates (Savvides, Park et al. 2003; Hongyang, Deng et al. 2005) as well as for tracking different sources (Li, Ekpenyong et al. 2006).

Particle filters is also a recursive implementation of signal processing. It is also known as sequential Monte Carlo method (SMC)(Gilks, Richardson et al. 1996). Particle filter is an alternative approach of Kalman filter for non Gaussian noise. Particle filter builds a large number of random samples (particle) which is propagated sequentially combining sampling and re-sampling steps. At each time, re-sampling discards some particles to increase the relevance of regions with high posterior probability. In particle filter, each sample is given weight indicating its importance, and the estimate is a weighted sum of all samples of a state. It utilizes 2 phases, prediction and update. in prediction, all particles are modified according to the model. Then in the update, the weight of each particle is re-evaluated based on the actual sensor measurements. Thus, particles with small weight are discarded. Examples of application in sensor fusion include computer vision(Isard and Blake 1996), multi-target tracking (Crisan and Doucet 2002), location discovery in WSN (Gustafsson and Gunnarsson 2003). Aslam used particle filter to explore target tracking using binary detection model (1 bit for moving closer and moving away). Coates used distributed particle filter for target tracking in a hierarchical sensor network(Coates 2004). Wong investigated a particle filter for fusing multi modality sensors in a target tracking application(Wong, Wu et al. 2004). Sheng proposed two distributed particle filter for multiple target tracking(Sheng, Hu et al. 2005). Another applications in WSN is for determining mobile nodes' locations (Hu and Evans 2004). Miguez et al. proposed using

Monte carlo-particle filter for both target tracking and nodes localization(Miguez and Artes-Rodriguez 2006).

5.3.3 Feature map

Applications such as guidance systems require spatial information about environment. Feature map provides feature analysis of an environment. An example of feature map includes occupancy grid and network scans. Occupancy grid (also called occupancy map) estimates the occupancy of environment in a multidimensional space. The observed space is divided into multidimensional cells (e.g.: square, cubes) which have probability of being occupied. The probability is computed based on information obtained from sensors, using different inference methods such as Bayesian, fuzzy set, or Dampster-Shafer. Occupancy grid was initially used for ultrasonic sensors determining static environments. Arbuckle introduced a temporal factor in occupancy grid allowing spatial environment being modeled according time properties. Hoover and Olsen model 2D raster as an occupancy map. Another application of occupancy grids include a position estimation, robot's location perception, and navigation. Network scans show the geographical distribution of network resources and activities of a WSN. Zhao used it to retrieve information about residual energy of each node. It work follow: first, it forms an aggregation tree. Second, each node computes its local residual energy and location. When the energy drop significantly, each node sends the report to the sink. The reports are aggregated based on the region if they have a similar residual energy.

5.3.4 Aggregation

Data aggregation in WSN aims at solving overlap and implosion problems. Implosion happens when data is duplicated many times when it is being routed (e.g.: simple flooding technique). Overlap happens when more sensors disseminate a same data of an event. Aggregation techniques are often used together with data-centric model in WSN such as TinyDB. Intanagonwiwat et al. investigated the latency of greedy aggregation in a dense network. The tradeoff between energy consumption and accuracy was discussed by Boul et al. in aggregation one can suppresses redundant data by discarding duplicates. Another technique for reducing the network overhead caused by frame headers is by placing several reports into a frame such as what TCP window does. This technique is not considered as information fusion, though.

5.3.5 Compression

Compression techniques that utilize information fusion in WSN take advantage of spatial correlation among sensor nodes and the correlation of the sensed data. An example of these techniques is called Distributed Source Coding (DCS) which is extended by Kusuma et al. and Pradhan et al which is called Distributed Source Coding using Syndrome (DISCUS). The main idea is to reduce the number of bits sent from node A to B if their data is correlated. And the possible values of observation can be grouped and indexed. For instance suppose sensor node A and B observation is coded with 3 bit words [000, 001, … ,111]. A and B correlate such that the Hamming distance between A and B is at the most 1. Since the possible states have been grouped and indexed as in Fig.. , A can send only the index (10) to B, and B can infer the value of A based of B's own value and the correlation function (Hamming Distance <=1).

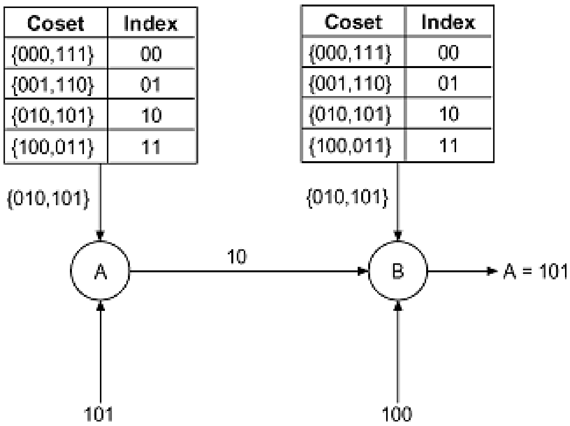


Fig. 5-6. Index table of DISCUS

Hua and Chen improved Viterbi algorithm for DCS that takes advantage of known parity bits at the decoder for error correction. A framework that optimize the tradeoff between compression and network performance for designing and analyzing distributed joint source and network coding was presented by Zhang et al and Ahlswede.

Another compression technique is known as Coding by ordering. The idea is that every node in the region of interest (RoI) sends its data to a border node that collects all data in a big packet that will be sent to the sink. The border node can suppress some packets and sort the rest of the packets in such way that the order indicates the value of the suppressed packets. E.g.: supposed we have 4 nodes (A, B, C, D) whose observation is an integer ranged from 0-5. The border node encodes the packet of node D as follow:

Packet Ordering	Observation from node D
$\{A,B,C\}$	0
$\{A,C,B\}$	1
$\{B,A,C\}$	2
$\{B,C,A\}$	3
$\{C,A,B\}$	4
$\{C,B,A\}$	5

Fig. 5-7. Ordering table

There exist as well other techniques such as EasiPC that reduces redundancy within a single packet to be transmitted, Compressive Wireless Sensing by Bajwa explored a source-channel communication based on compressive sampling (Candes and Wakin 2008) to estimate sensor data that contain structural integrity. Several approaches use wavelet for compression for instance Ciancio and Ortega decouple data among nodes by exchanging information with neighboring nodes (Ciancio and Ortega 2004). Tang used wavelet in a source broadcast scheme for extracting significance bits from sensor data (Tang and Raghavendra 2006).

5.4 Summary

Multi-sensor data fusion is a technology that has been formalized since 70s specifically in military applications. Many architecture, model, and algorithms have been developed addressing various data fusion applications. JDL model is the most used and well known model. It provides a partition of sensor fusion functions. However it does not describe any process model and architectural design. Thus, in ebbits, it is necessary to complete this model to have the common understanding of working processes ebbits domain. For the sake of simplicity, further discussion will be based on the JDL Model.

The most mature area of data fusion is level 1 of the JDL model such as target tracking, position, velocity determination, and object classification (friend or foe). All of these applications tried to estimate certainty of information obtained from sensor data. Nonetheless there is still no general solution that is able to overcome challenges such as object densities, rapid movement, and signal propagation. Many algorithms in this level are required in ebbits scenarios for instance, object tracking is useful for tracking animals in farms and goods in factories, increasing the certainty of wireless transmission and sensors readings in places with harsh conditions. Algorithms have been maturely used in this level includes: filtering algorithms, aggregation and compression.

Level 2 and 3 fusions are dominated by knowledge based such as rule based, fuzzy logic, intelligent agents, and Bayesian beliefs. Although these areas are quite promising to provide an intelligent information, unfortunately these techniques are still immature and do not provide any stable operational systems (Liggins, Hall et al. 2009). The main challenges in this are establishing a common and reliable knowledge base and representing it uniformly, many works on fuzzy logic have shown promising results, though. Another challenge that ebbits contributes to this area is that it aims at massive scale of distributed and heterogeneous systems that have diverse knowledge representations. Thus, ebbits requires defining a new framework that also reflects process model as well as architectural design of information fusion. Furthermore, the ebbits must also define a concept to handle diverse knowledge representations distributed in the internet of things. An example where level 2 and 3 processing will be needed in ebbits is e.g.: for inferring intelligent context for energy savings purposes.

Many multi-sensor fusion models have included control theory that discusses relationship of sensing and control as well as sensing-control loop for performance assessment of the system. Up to now, the main challenges in this area are to model task objectives, manage resources based on the objectives, and provide information that satisfies the decision makers' needs. ebbits also aims at a self regulating system, which need an automatic control and real time system assessments. On the other hand, if the control and assessment is done manually through human intervention, the human computer interaction concepts must also be defined. Control theory and its usage in ebbits will be discussed more in section 8.

5.5 Bibliography

Baran, R. (2002). A collective computation approach to automatic target recognition, IEEE.

Bedworth, M. and J. O'Brien (2000). "The Omnibus model: a new model of data fusion?" Aerospace and Electronic Systems Magazine, IEEE **15**(4): 30-36.

Bracio, B. R., W. Horn, et al. (1997). Sensor fusion in biomedical systems. Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE.

Brokmann, G., B. March, et al. (2001). Integrated multisensors for industrial humidity measurement. Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on.

Brown, C., H. Durrant-Whyte, et al. (1992). Distributed data fusion using Kalman filtering: A robotics application. Data fusion in robotics and machine intelligence. M. A. Abidi and R. C. Gonzalez, Academic Press Professional, Inc.**:** 267–309.

Cain, M., S. Stewart, et al. (1989). Object classification using multispectral sensor data fusion, Bellingham, WA, Society of Photo-Optical Instrumentation Engineers.

Candes, E. J. and M. B. Wakin (2008). "An Introduction To Compressive Sampling." Signal Processing Magazine, IEEE **25**(2): 21-30.

Chan Yet, W. and U. Qidwai (2005). Intelligent sensor network for obstacle avoidance strategy. Sensors, 2005 IEEE.

Ciancio, A. and A. Ortega (2004). A distributed wavelet compression algorithm for wireless sensor networks using lifting, IEEE.

Coates, M. (2004). Distributed particle filters for sensor networks, ACM.

Coue, C., T. Fraichard, et al. (2002). Multi-sensor data fusion using Bayesian programming: An automotive application. Intelligent Vehicle Symposium, 2002. IEEE.

Crisan, D. and A. Doucet (2002). "A survey of convergence results on particle filtering methods for practitioners." Signal Processing, IEEE Transactions on **50**(3): 736-746.

Cui, X., T. Hardin, et al. (2004). A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization. Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on.

Dasarathy, B. V. (1997). "Sensor fusion potential exploitation-innovative architectures and illustrative applications." Proc. IEEE **85**: 24-38.

Elmenreich, W. (2002). Sensor fusion in time-triggered systems. Institut für Technische Informatik. Vienna, Vienna University of Technology. **Ph.D.**

Frankel, C. B. and M. D. Bedworth (2000). Control, estimation and abstraction in fusion architectures: lessons from human information processing. Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on.

Friedlander, D. (2005). "Semantic information extraction." Distributed Sensor Networks: 409–417.

Gilks, W., S. Richardson, et al. (1996). Markov Chain Monte Carlo in Practice. , Chapman & Hall.

Gupta, I., D. Riordan, et al. (2005). Cluster-head election using fuzzy logic for wireless sensor networks. Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual.

Gustafsson, F. and F. Gunnarsson (2003). Positioning using time-difference of arrival measurements. Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on.

Halgamuge, M. N., S. M. Guru, et al. (2003). Energy efficient cluster formation in wireless sensor networks. Telecommunications, 2003. ICT 2003. 10th International Conference on.

Hall, D. L. and J. Llinas (1997). "An introduction to multisensor data fusion." Proceedings of the IEEE **85**(1): 6-23.

Heinzelman, W. B., A. L. Murphy, et al. (2004). "Middleware to support sensor network applications." Network, IEEE **18**(1): 6-14.

Heinzelman, W. R., A. Chandrakasan, et al. (2000). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8, IEEE Computer Society**:** 8020.

Hongyang, C., P. Deng, et al. (2005). A robust location algorithm with biased extended Kalman filtering of TDOA data for wireless sensor networks. Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on.

Hu, L. and D. Evans (2004). Localization for mobile sensor networks, ACM.

Intanagonwiwat, C., R. Govindan, et al. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. Proceedings of the 6th annual international conference on Mobile computing and networking. Boston, Massachusetts, United States, ACM**:** 56-67.

Isard, M. and A. Blake (1996). Contour tracking by stochastic propagation of conditional density. Computer Vision — ECCV '96. B. Buxton and R. Cipolla, Springer Berlin / Heidelberg. **1064:** 343-356.

Iyengar, S. S., K. Chakrabarty, et al. (2001). "Introduction to special issue on "distributed sensor networks for real-time systems with adaptive configuration"." Journal of Franklin Institute **338**: 655-668, 651-653.

Jaikaeo, C., C. Srisathapornphat, et al. (2000). Querying and tasking in sensor networks, SPIE.

Kokar, M. M., M. D. Bedworth, et al. (2000). Reference model for data fusion systems, SPIE.

Lei, F., D. Wenliang, et al. (2005). A beacon-less location discovery scheme for wireless sensor networks. INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE.

Lewis, T. W. and D. M. W. Powers (2002). Audio-visual speech recognition using red exclusion and neural networks. Proceedings of the twenty-fifth Australasian conference on Computer science - Volume 4. Melbourne, Victoria, Australia, Australian Computer Society, Inc.**:** 149-156.

Li, S., S. H. Son, et al. (2003). Event detection services using data service middleware in distributed sensor networks. Proceedings of the 2nd international conference on Information processing in sensor networks. Palo Alto, CA, USA, Springer-Verlag**:** 502-517.

Li, T., A. Ekpenyong, et al. (2006). "Source Localization and Tracking Using Distributed Asynchronous Sensors." Signal Processing, IEEE Transactions on **54**(10): 3991-4003.

Liggins, M. E., D. L. Hall, et al. (2009). Handbook of Multisensor Data fusion, Theory and Practice. Boca Raton, FL, CRC Press.

Liu, J. and F. Zhao (2006). Towards semantic services for sensor-rich information systems, IEEE.

Luo, R. C., Y. Chih-Chen, et al. (2002). "Multisensor fusion and integration: approaches, applications, and future research directions." Sensors Journal, IEEE **2**(2): 107-119.

Luo, R. C. and M. G. Kay, Eds. (1995). Multisensor integration and fusion for intelligent machines and systems, Ablex Publishing Corp.

Madden, S. R., M. J. Franklin, et al. (2005). "TinyDB: an acquisitional query processing system for sensor networks." ACM Trans. Database Syst. **30**(1): 122-173.

Mascolo, C. and M. Musolesi (2006). SCAR: context-aware adaptive routing in delay tolerant mobile sensor networks. Proceedings of the 2006 international conference on Wireless communications and mobile computing. Vancouver, British Columbia, Canada, ACM**:** 533-538.

Miguez, J. and A. Artes-Rodriguez (2006). A Monte Carlo method for joint node location and maneuvering target tracking in a sensor network, IEEE.

Nakamura, E., F. Nakamura, et al. (2005). "Using Information Fusion to Assist Data Dissemination in Wireless Sensor Networks." Telecommunication Systems **30**: 237-254.

Nakamura, E. F., A. A. F. Loureiro, et al. (2007). "Information fusion for wireless sensor networks: Methods, models, and classifications." ACM Comput. Surv. **39**(3): 9.

Nakamura, E. F., H. A. B. F. d. Oliveira, et al. (2006). On Demand Role Assignment for Event-Detection in Sensor Networks. Proceedings of the 11th IEEE Symposium on Computers and Communications, IEEE Computer Society**:** 941-947.

Patwari, N., A. O. Hero, III, et al. (2003). "Relative location estimation in wireless sensor networks." Signal Processing, IEEE Transactions on **51**(8): 2137-2148.

Rabbat, M. and R. Nowak (2004). Distributed optimization in sensor networks. Proceedings of the 3rd international symposium on Information processing in sensor networks. Berkeley, California, USA, ACM**:** 20-27.

Rachlin, Y., R. Negi, et al. (2006). On the interdependence of sensing and estimation complexity in sensor networks. Proceedings of the 5th international conference on Information processing in sensor networks. Nashville, Tennessee, USA, ACM**:** 160-167.

Sam, D., C. Nwankpa, et al. (2001). Decision fusion of voltage stability indicators for small sized power systems. Power Engineering Society Summer Meeting, 2001. IEEE.

Savvides, A., H. Park, et al. (2003). "The n-hop multilateration primitive for node localization problems." Mob. Netw. Appl. **8**(4): 443-451.

Schmitt, T., R. Hanek, et al. (2001). Cooperative probabilistic state estimation for vision-based autonomous mobile robots. Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on.

Sheng, X., Y. Hu, et al. (2005). Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network, IEEE.

Singh, A., R. Nowak, et al. (2006). Active learning for adaptive mobile sensing networks. Proceedings of the 5th international conference on Information processing in sensor networks. Nashville, Tennessee, USA, ACM**:** 60-68.

Tang, C. and C. Raghavendra (2006). Wavelet based source broadcast for in-network processing in sensor networks unknown side information, IEEE.

Wald, L. (1999). "Some terms of reference in data fusion." Geoscience and Remote Sensing, IEEE Transactions on **37**(3): 1190-1193.

Wang, X. (2004). High accuracy distributed target detection and classification in sensor networks based on mobile agent framework, University of Tennessee**:** 204.

White, F. E. (1991). Data fusion lexicon. Technical Panel for C3. San Diego, CA, U.S. Department of Defense.

Willett, R., A. Martin, et al. (2004). Backcasting: adaptive sampling for sensor networks. Proceedings of the 3rd international symposium on Information processing in sensor networks. Berkeley, California, USA, ACM**:** 124-133.

Wong, Y., J. Wu, et al. (2004). Collaborative data fusion tracking in sensor networks using monte carlo methods, IEEE.

Xiao, L., S. Boyd, et al. (2005). A scheme for robust distributed sensor fusion based on average consensus. Proceedings of the 4th international symposium on Information processing in sensor networks. Los Angeles, California, IEEE Press**:** 9.

Yao, Y. and J. Gehrke (2002). "The cougar approach to in-network query processing in sensor networks." SIGMOD Rec. **31**(3): 9-18.

Yiyao, L., Y. V. Venkatesh, et al. (2001). "A knowledge-based neural network for fusing edge maps of multi-sensor images." Information Fusion **2**(2): 121-133.

Yu, B., K. Sycara, et al. (2004). Uncertain information fusion for force aggregation and classification in airborne sensor networks, Citeseer.

6. Current Approaches to Mobile Sensing

With the ubiquity and ever-increasing capabilities of mobile devices (cell phones, PDA etc) there is a great potential to provide intelligent service structures pushed on by mobile sensing. For steps in the process chain that are labeled with some specific dynamic phenomenon it is beneficial to gather data in some mobile, ad-hoc manner and afterwards aggregate and analyze it at some backbone system.

At first, this chapter motivates how mobile sensing can be applied for large-scale distributed e-business environments. Then, it explains the benefits of mobile sensing in general, and introduces the concepts of participatory and opportunistic sensing. Not least, example scenarios for mobile sensing applications in five different domains are outlined.

6.1 Mobile Sensing for ebbits

In the sense of ebbits project mobile sensing is a relevant technological approach to push on intelligent service structures. In particular, mobile sensing can be leveraged to monitor many products (perishable goods, auto parts etc) during the last end of the life-cycle the status. In the following it is described how we initially elicited the benefit of mobile sensing for ebbits during requirements workshops that have been organized for both domains (manufacturing and traceability scenario).

6.1.1 Manufacturing scenario

During the ebbits requirements workshop that has been organized to work on the manufacturing scenario, we found out that there are totally different reasons for applying mobile sensing within the manufacturing process than commonly argued.

For Greenfield development, i.e. a plant is built from scratch, it is easy to install new stationary sensors into a manufacturing, as no constraints imposed by prior networks exist. Though, for Brownfield development, i.e. plant is already running and comprises several machines, it is quite more challenging (Hopkins and Jenkins 2008), as you will need to deploy new sensors and actuators in the immediate presence of existing (legacy) systems.

In this sense, it is slightly similar to (Kanjo, Bacon et al. 2009), as in both cases the mobile sensing approach is applied due to financial constraints. However, in manufacturing plants usually the budget is not the constraint to install stationary sensors that may provide more accurate data instead of enhancing stations with mobile sensors. In fact, the risk is to stop each station and further to face technical problems during the set-up of new sensors; for instance, smart meters for measuring the energy consumption that has become in the recent years an important Fig. to track. If such a risk becomes reality, it might cause a big loss in the progress of the plant's production.

Besides, the risk to affect the production process, we were told that sensing things wirelessly is a smooth solution upon to send data over the wire, as a mass of cable or rearing respectively can be avoided. Hence, the deployment of mobile sensors communicating data over various wireless communication protocols (Bluetooth, ZigBee or Wi-Fi) is motivated as long as manufacturing constraints are met.

6.1.2 Traceability scenario

During the ebbits requirements workshop that has been organized to work on the traceability scenario the following benefit of mobile sensing has been elicited:

In the traceability scenario the subject to observe are pigs, in fact 'from farm to fork'. Due to transports, climatic changes, or improper nutrition etc animals are susceptive to momentous diseases. Hence, to avoid the widespread of disease pigs wear mobile sensor nodes pigs, so that at an early stage a (contagious) disease can be identified. Sensors can measure following health parameters: heart beat, temperature, time of ingestion etc.

For instance, a monitoring system can indicate to a farmer that two of his pigs staying in the barn show critical symptoms. Meaningful events are propagated via an RFID Tag that communicates the events to a gateway in the barn that forwards it to the backbone monitoring system of the farmer. Thereupon the ill animal gets its medical treatment, for example a deterrent could be injected. After the treatment the farmer uses a PDA device to communicate with the RFID tag in order to scan the ID of the pig. Then the application on his PDA links the disease to the ID of the pig, and finally invokes the storage of the disease of the given animal in the history being hosted on a dedicated backbone system.

Such a scenario shows for the case that in later phases of consumption a severe disease is detected, the health process could be traced backed in order to find relevant indicators. Currently, in Denmark the animals are tagged on badge level, i.e. if one piece of meat from one pig has been considered as harmful to health, all other pieces of pigs that belong to the same badge of the harmful pig are disposed of the market. With the strength of ebbits platform it is planned to make one step forward in this context, i.e. to go from badge level to a single animal, and thus become more efficient during the disposal of harmful meat.

The communication between PDA and the mobile sensor node of the pig is not constrained to the above explained disease scenario case. It can also be applicable for a scenario when a pig needs to get an inoculation. There also the ID of the pig must be logged in the history in order to provide a complete life history of an animal.

6.2 Technological Development towards Mobile Sensing

In the last decade mobile computer devices have decreased in size, but at the same time have become increasingly powerful regarding their CPU, data storage etc. At the same time recent progresses in sensor network research allow for the deployment of small and cheap sensor nodes. Hence, the two first required steps for a mobile data collection have been achieved to attach different sensors on some carrier medium that carries or wears mobile sensors, e.g. human, vehicle, or robot.

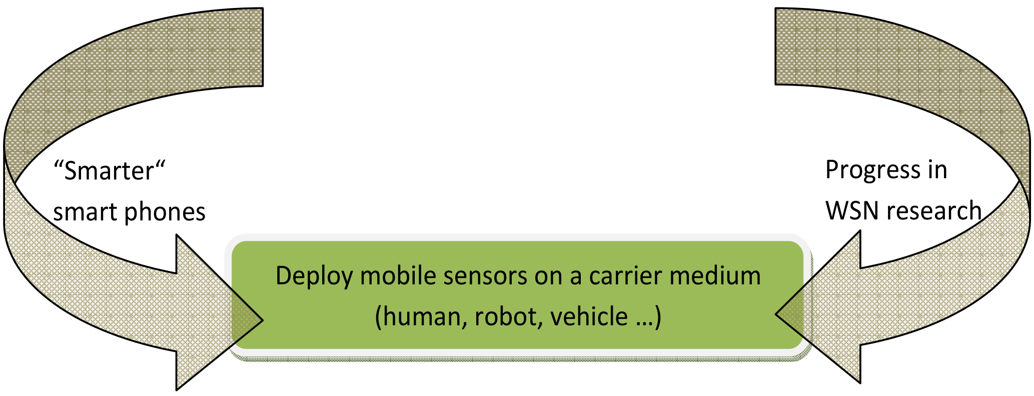


Fig. 6-1 Push to develop mobile sensing applications

Wireless sensor networks allow for observing processes in the real world. Each sensor node can sense changes in its environment, then process on behalf of it, forward it to another sensor node or to a central unit. By deploying several sensors in a given scenario a lot of data is obtained about physical values, e.g. temperature, humidity, lightning condition, energy consumption, carbon dioxide level, soil moisture, magnetic field, heart beat etc, and additionally measurable are acoustic and visual impressions. Furthermore, current characteristics of objects such as speed, direction, and size, the presence or absence of certain kinds of objects, and all kinds of values about machinery, e.g. acceleration, pressure, mechanical stress level, or even noise level. This huge choice of options allows using sensor network applications in a number of scenarios, e.g. environmental monitoring, health care, military surveillance, industrial machinery surveillance, home automation, or tracking animals in wild life or preserved lands.

6.3 Mobile Sensing

By affixing sensors to mobile devices (cell phones, PDA etc) data can be collected from large numbers of people, cars, terrestrial robots, ambient robots, unmanned aerial vehicles etc in ways that were not previously possible. This approach is called *mobile sensing*.

Mobile sensing is also often referred to as *participatory sensing* or *urban sensing*. First term is due to the fact that a lot of mobile sensing applications that have been built yet and published involved human beings for collecting data (6.4.2) and second term as a lot of those applications intend to support urban planning (see 6.4.1).

The crucial benefit of mobile sensing over stationary sensing is to take advantage of the *mobility* of a carrier medium (human being, vehicle, robot etc) while collecting data over a large region and for a given period of time. The collected data of each mobile sensor can later be aggregated, shared, and analyzed in order to cope better with highly-dynamic, complex phenomena, such as traffic jams in and before large cities.

Furthermore, mobile sensing can provide coverage in spaces where it is hard to deploy and maintain static sensors due to natural conditions or even industrial constraints. In (Kanjo, Bacon et al. 2009; Kwok 2009) further advantages of mobile sensing are elaborated.

6.3.1 Participatory Sensing

The term participatory sensing describes the collection of data via mobile sensor nodes in cooperation with the owner of a device. In this scenario the human being is fully involved in data collection, and can decide consciously and actively which amount of data will be input of a specific system or not. Hence, the data collection for one's own represents no real benefit for the common good. Only, if this data is shared among other, aggregated and analyzed new perspectives arise. However, employing handheld devices as sensor nodes poses new challenges for privacy, data security, and not least ethics. To which extent a human is willing to share data or react to sensing request from other nodes in the wireless sensor network or remote stations puts a user in a difficult situation, but is a central question that has been partly answered in the paper of (Lane, Eisenman et al. 2008). Further, in (Shilton, Burke et al. 2008) the concept of *participatory privacy regulation* is introduced that argues that privacy must be a participatory process that takes into account both individual preferences and social settings. As the needs and preferences of an individual change according to social situation, these negotiations cannot be separated from a user's context. Such approach allows humans within urban sensing systems to negotiate with data capture, presentation, and disclosure.

6.3.2 Opportunistic Sensing

In contrary to participatory sensing, the concept of opportunistic sensing reflects the data gathering by a human without being aware when data is collected while he is interacting with his environment (Roggen, Forster et al. 2009). Of course, the user must have the possibility to configure the system in order to provide a minimal degree of privacy. But, compared to participatory sensing the user does not intervene in the application process whilst data collection (see Fig. 6-2). The advantages and disadvantages of opportunistic sensing compared to participatory sensing are obvious. Regarding participatory sensing the user needs clearly to recognize his personal or economic value. For applications that need to acquire continuously data it is rather an obstacle, if a user would permanently need to react to requests of a system. However, there might be specific applications where input from users is required in order to proceed in a given use case, e.g. if a bystander takes a photo of a disaster site and an application on his mobile needs his confirmation to send it to the next command post.

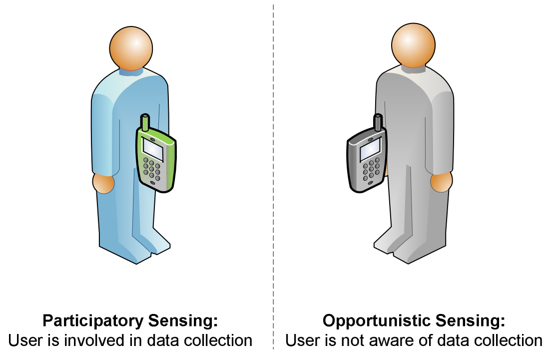


Fig. 6-2. Differences between Participatory Sensing and Opportunistic Sensing

6.3.3 Cell Phones as Sensor Nodes

Cell phones have managed to be leveraged permeable in our everyday life:

In the year 1973, the first handheld cellular device weighing 2 kg was demonstrated by Martin Cooper a leading inventor at Motorola, Inc.[4]. Then, in the year 1990 12.9 million subscribers to mobile networks have been registered. Finally, in 2009 according to the GSM Association[5] it was announced that about 85% of the world population, in detail 4.9 billion, that means 370 times more than in 1990, has mobile phone connection. And not least, it is expected to reach six billion subscribers by the year 2013 (Kwok 2009).

Hence, a life without a cell phone is nearly impossible for most of us. These handheld devices steadily become smaller and at the same time smarter: as their form factor has been decreased enormously, but at the same time their computation power highly increased, long ago cell phones serve merely as wireless phones as originally foreseen. For instance, in India more people access the Internet from cell phone than from PC (Weber 2009).

Increasingly, cell phones are equipped with a set of sensors that may comprise: GPS, compass, temperature, acceleration, integrated camera, face detection, light resistance, gyroscope, pressure, proximity etc. Enhancing simple cell phones with sensors to such so called *smart phones* offers excellent possibilities, as each sensor allows for gathering potentially rich data of the environment in that some person wearing a specific carrier medium is interacting in. This data can be merged over sensor networks, and later be used for different analysis at some backbone system. As persons carry their cell phone nearly everywhere they go, smart phones as sensor nodes themselves represent a huge potential to collect meaningful data in favor of the common good.

A significant feature smart phones provide, is that they can locate via a GPS sensor themselves, and thus the location of their owner. By this data can be collected related to a certain context considering location and time. For the authenticity of the data they are at least that important as the identity of the cell phone's user (Burke, Estrin et al. 2006).

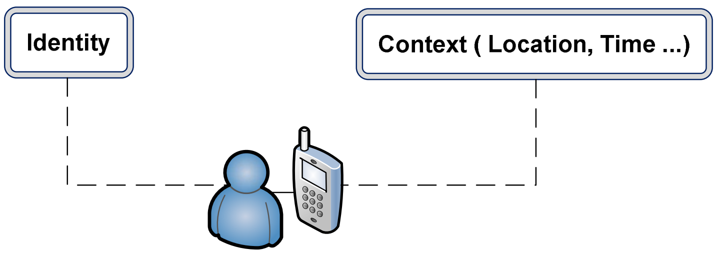


Fig.6-3 Smart Phones GPS functionality supports context-aware system development

[4] Motorola Inc., www.motorola.com.

[5] GSM (Global System for Mobile Communications), www.gsmworld.com/

Examples for smart phones are: the iPhone from Apple[6], Nokia N95, or Motorola Milestone. Till the year 2009 180 million units of smart phones were sold, and for 2014 the number of 400 million has been prognosticated (Parks Associates 2010). With the ubiquity of cell phones and their networks already an infrastructure exists where diverse application can be built on. The mobile instrumentation of sensors nodes facilitates new, so far unprecedented possibilities of acquiring information.

6.4 Applications of Mobile Sensing

Mobile sensing applications have been applied in a number of scenarios, the observation of pollution and the control of traffic flows are just a few to mention. The leading organization in research of mobile sensing is the CENS[7] institute. In the following five different application domains are described. All domains provide good examples for how mobile sensors can be deployed to collect data in new ways in order to cope better with complex phenomena and its benefits for the common society; the last two application domains have been chosen, as they are tangent to both application scenarios we need to evaluate for the ebbits platform.

6.4.1 Improving the Estimation of Traffic Flows

Estimating traffic state is still a huge challenge for the transportation community in large cities. In order to estimate properly in space and time the state of freeways it is necessary to develop efficient control strategies to improve the traffic conditions. At this, stationary sensors, such as loop detector stations, already give support for collecting data. Using this data the density field can be reconstructed to certain accuracy. However, deploying such an infrastructure is expensive, and its reliability varies on public funding. In the Mobile Millenium[8] project mobile sensors were used additionally to stationary sensors to reconstruct traffic flows on freeways. Given incentives at the project's web site participants can download the free release of software and install it on their smart phones. The software provides by means of different colors (see Fig. 6-4) real-time traffic information as speeds, congestions, or travel time estimates.

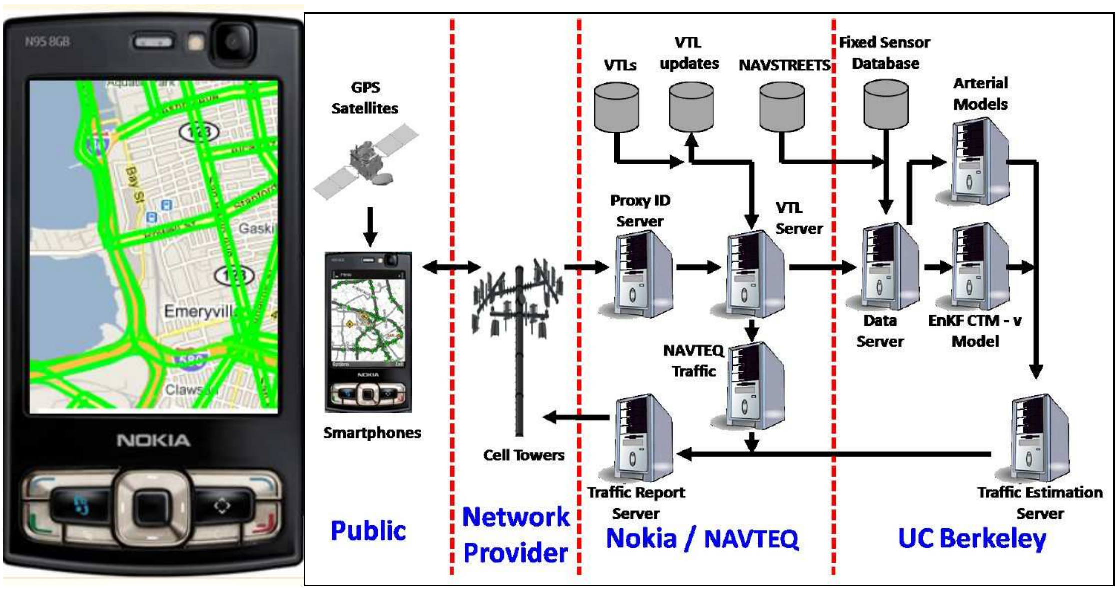


Fig. 6-4 Mobile Millenium: (a) Traffic Flow Software. Image Resource: http://traffic.berkeley.edu, (b) System Overview; image source: (Work, Tossavainen et al. 2009)

Two algorithms are applied and evaluated to use data collected from GPS enabled mobile phones. The first approach is based on data assimilation methods (Newtonian relaxation), and the second is based on Kalman filtering. Results show that the both algorithms appropriately incorporate the new data, improving significantly the accuracy of the estimates for traffic flows that consider loop

[6] iPhone, Apple, http://www.apple.com/iphone/

[7] Center for Embedded Networked Sensing, http://research.cens.ucla.edu/

[8] Millenium Project, http://traffic.berkeley.edu/index.html

detector data only (Amin, Andrews et al. 2008; Work, Tossavainen et al. 2008; Work, Tossavainen et al. 2009). The Mobile Millenium project enables the set-up of intelligent service structures that involves the interplay between people, things and services. In particular, it does well in two decisive criteria referring to the mobile sensing concept:

- First, it manages to deploy dynamically on the road traffic sensors by means of GPS equipped smart phones, where it is difficult and expensive to install stationary sensors, and thus excellently shows how mobile sensing can help to cope better with complex phenomena than previously done.
- And second, in addition to stationary loop detector sensors, it launches successfully in the field a system that consists of a privacy-preserving data-gathering infrastructure including smart phones with GPS. By this it shows how handy stationary sensing can be enhanced with mobile sensing in order to support a given task, i.e. the Mobile Millenium project fuses data from static sensors with data from cell phones.

6.4.2 Environmental Monitoring

The goal of environmental monitoring is to sense the environment; to collect such data is extremely helpful for biology research. In the CommonSense[9] project mobile phones are utilized as personal environmental sensors that allow citizens to collect pollution data in their neighbourhoods and actively participate in order to influence environmental regulations and policies. Smart phones are equipped with gas sensors for carbon monoxide (CO), NOx and Ozone (O3), and further sensors for measuring temperature and humidity. The sensor board communicates every 60 seconds the data via Bluetooth protocol to the cell phone (see Fig. 6-5 a). From there the data is send via SMS including the GPS position to a central remote server where it can be rendered into visualization (see Fig. 6-5 b).

Fig. 6-5 CommonSense: (a) Environmental Handheld Device (b) Air quality of San Francisco

With a similar goal in (Paulos 2008; Paulos, Honicky et al. 2008) it is described how taxi drivers attached environmental sensors on the outside of their cabs, in order to collect dynamically data over the city of Acra in Ghana, and further shared this information among each other. The live heat map (see Fig. 6-6) might not represent an accurate solution, though it is approach of better than nothing. Finally, in (Kanjo, Bacon et al. 2009) a similar measuring and Bluetooth communication pattern for smart phones is installed to cycling couriers in the city of Cambridge, England. The couriers collect data throughout the city for analyzing data of polluted areas.

[9] CommonSense project, http://www.communitysensing.org/

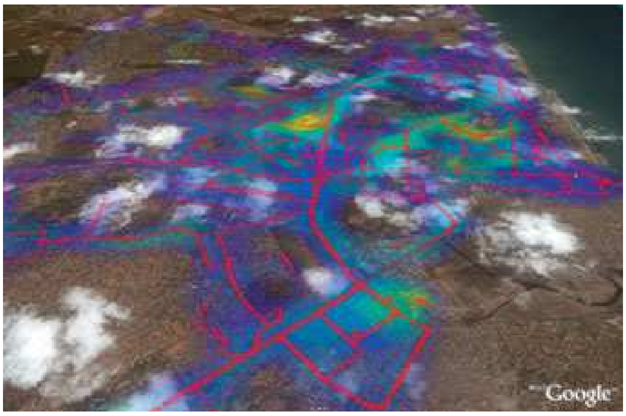


Fig. 6-6 A heat map visualization of carbon monoxide readings across the city of Accra rendered atop Google Earth

6.4.3 Emergency and Health Care

In health care, mobile sensing can be applied for long-term monitoring of patients outside the hospital. There a mass of sensors can be deployed to monitor vital functions of patients. In this sense, the MobiHealth[10] project enables patients to be through and through mobile while being monitored for vital functions. This is achieved by an infrastructure that facilitates continuous monitoring of patients outside the hospital environment by developing the concept of a 3G-enabled body area network (BAN), i.e. infrastructural services are based on GPRS and UMTS technology for wireless broadband data transfer, that encapsulates all the complexity related to security, hand-over or quality-of-service (Wac, Bults et al. 2009). In (Gravina et al. 2008; Seto, Martin et al. 2010) an opportunistic sensing approach is applied for environmental health monitoring. Heart beat and temperature sensors are attached to the human body (see Fig. 6-7a). Via a customized phone system interface—supplied by Intel Research—the patient's cell phone can communicate over ZigBee with other sensor nodes (see Fig. 6-7b).

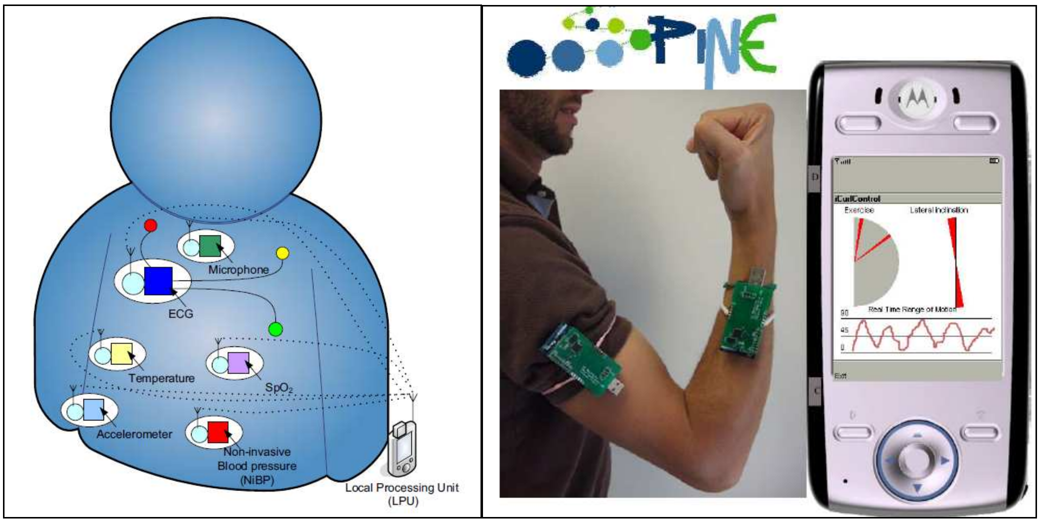


Fig. 6-7 Body Sensor Network (a) Sensors placed on human body(Lo, Thiemjarus et al. 2005) (b) SPINE: BSN using ZigBee enabled phone as communication gateway

For this the SPINE framework is utilized to establish a body sensor network (BSN) that handles the intercommunication between heterogeneous devices (see Fig. 6-8). SPINE distinguishes itself as a mobile platform for BSNs by providing flexibility in integrating different sensors, or on-the-fly reconfiguration of the BSN.

[10] MobiHealth project, http://www.mobihealth.org/

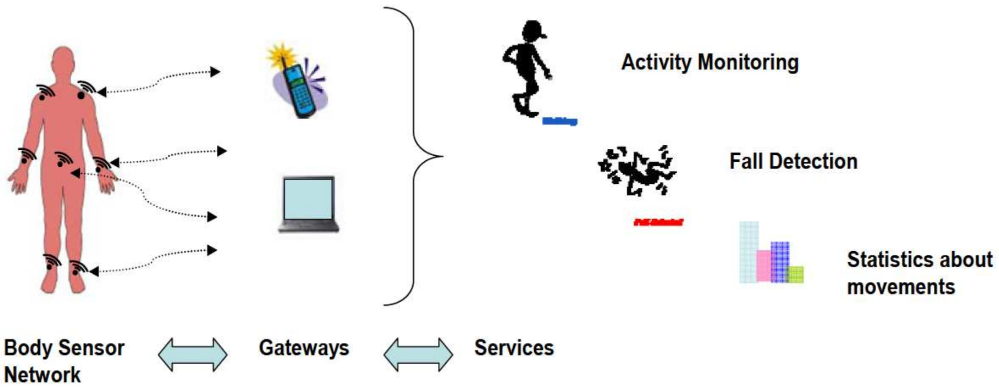


Fig. 6-8 SPINE: Architecture of BSN

Furthermore, a critical issue in healthcare is to establish a seamless transfer of patients between first aid personal, first responders, and doctors at hospital. At this a seamless monitoring and data transfer can take place. CodeBlue (Lorincz, Malan et al. 2004) exactly strives for supporting the previously described subject. It is a system that enhances emergency medical care with the target to have a seamless transfer of patients between a disaster site and hospital. For this body sensors are used to track the health state and location of a patient. Current body sensors are able to obtain heart rate, oxygen saturation, serum chemistries measurements, including serum glucose. The network of CodeBlue does not rely on any infrastructure, quite the contrary it is built ad hoc to span a large disaster area. The ad hoc network supports a heterogeneous set of devices from body sensors over location beacons till PDAs (see Fig. 6-9). At this point it should be mentioned that CodeBlue applies a prioritization concept for dealing with critical data: vital signs from patients at risk or a SOS message from a fire fighter have priority over other traffic.

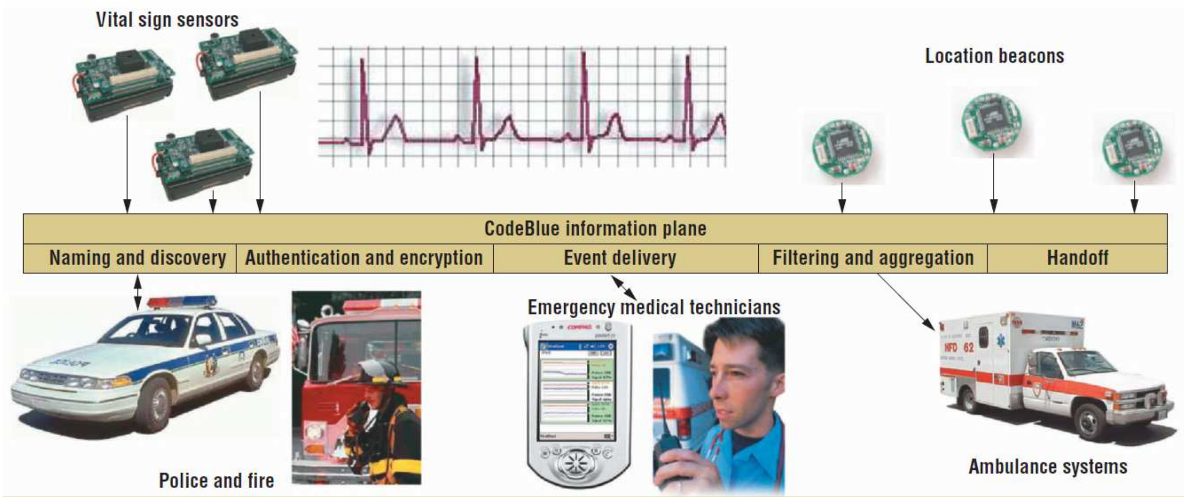


Fig. 6-9 CodeBlue: Ad-hoc Network Infrastructure for Heterogeneous Devices

6.4.4 Manufacturing

Publications of mobile sensing for manufacturing are scarce, as wireless communication has for a long time been banned from manufacturing sites due to many reasons. Mainly security and resilience: these environments are often difficult, very noisy (electrical noise, strong magnetic fields, etc.). Some regulations, i.e. security (TÜV standards for fire alarms, disallow the use of wireless technologies). Nevertheless, this sub chapter provides some general info why sensor networks are useful for manufacturing. Manufacturing relies on an increasingly number of machines that work together in highly complex processes. A breakdown of a machine may interfere severely the production, resulting in a financial loss. Hence, using also wireless sensor networks to observe the machinery might be beneficial in the long run. To prevent unplanned breakdowns it is reasonable to conduct maintenance for machinery in a manufacturing plant. For this different approaches can be applied:

- Run-To-Fail Maintenance: This is the worst approach that does not include any maintenance work at all. A machine is repaired if possible or replaced when it breaks down.
- Scheduled Maintenance: In regular cycles upon some specific schedule maintenance work is conducted. For example, links of a machine has to be oiled every 3 month, or a blade of a machine must be sharpened after cutting 300 doors for some specific type of vehicle. While estimating the proper lifetime of a machine scheduled maintenance is based on expert knowledge, but not on the actual state of a machine. Therefore, this approach lacks accuracy. On the one hand the maintenance may be undertaken too late resulting in irreversible damage of a machine. And, at the other hand it is possible that a machine is maintained, although it does not require maintenance resulting in unnecessary costs and process downtime.
- Condition-based Maintenance: This approach fills the gap of the improper accuracy scheduled maintenance does have, i.e. a machine is maintained regarding its actual state of condition. To find out the proper state to do maintenance sensor networks can put things right, i.e. additional sensors are affixed to components of the machinery in order to detect some malicious state. For instance, the company Rockwell Scientific has launched a condition-based maintenance program (Rockwell Automation 2010). For the implementation of condition monitoring services program diagnostic routines and expert systems were ported to wireless sensor network hardware. Furthermore, the software from now on had to support autonomous data collection and analysis. The main challenge for software development was to abstract and design algorithms independent from a specific type of machine, so that the sensors could be deployed independent from a type of machine. In (Bradley 2009) Rockwell Scientific describes a flexible and cost saving condition monitoring system for wind turbine manufacturers (see Fig. 6-10).

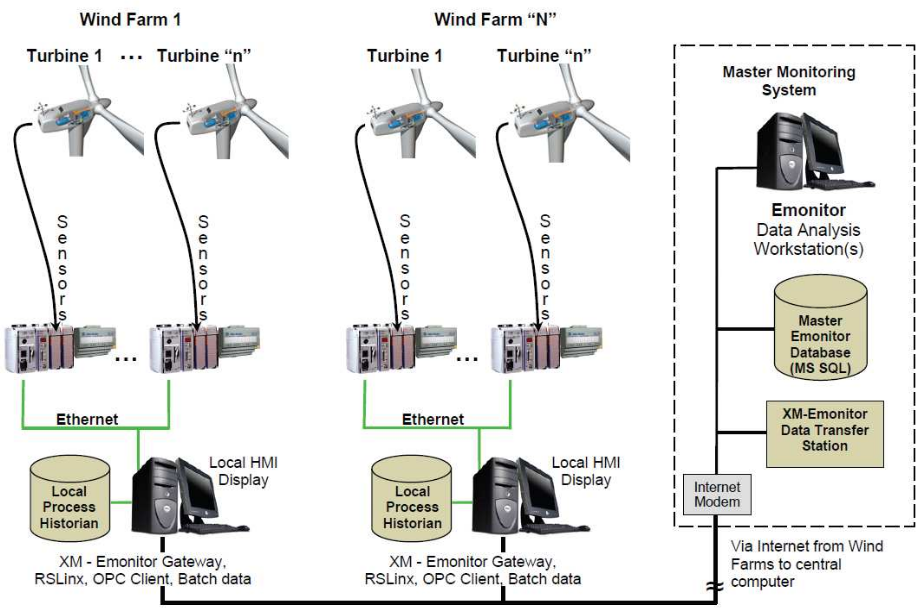


Fig. 6-10 Rockwell: Communication Concept for Condition Monitoring System for Wind Turbine Manufacturers

6.4.5 Monitor Animals in Living Space

The main objective of sensing animals in their living space is to track and observe their behavior, habits and gathering in daily life. Traditionally, for this researchers had to hide and observe the wild life, and later cameras were used for tracking. However, these techniques are intrusive and uncomfortable, and to organize long-term observations is difficult and expensive as well. In ZebraNet (Zhang, Sadler et al. 2004) project a less intrusive approach has been applied. There human interaction with the animals is only needed for the set-up and removal of sensors. At first, workers anesthetize zebras with tranquilizer gun. Then, the sensor nodes are integrated into a collar (see Fig. 6-11 a) that is attached to the animals themselves. The main objective of ZebraNet is to understand the migration patterns of zebras and how these patterns are affected by changes, such as environmental conditions as the weather or plant life.

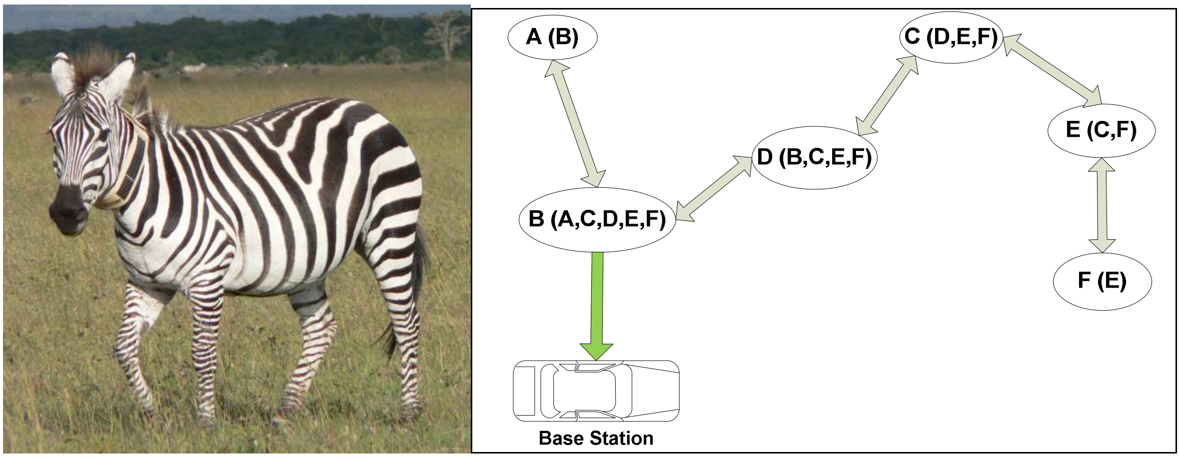


Fig. 6-11 ZebraNet: (a) Zebra with fixed sensor collar, (b) Energy-Efficient Peer-to-Peer Ad Hoc Networking

As a zebra herd reflects a quite large spatial extent, though sensor nodes are constrained in communication range, the base station of ZebraNet has to be *mobile*, besides the sensors. It had been realized by a jeep that regularly drives through. Each hour a detailed zebra activity log is taken for three minutes. The collar is equipped with solar panels for recharging the battery of the sensor node, though it can also operate five days without solar recharge. The sensor nodes in ZebraNet collect biometric data, for instance body temperature, heart rate, and how many times zebras eat per day. Further, to understand the migration pattern a GPS sensor had to acquire regularly the GPS position. In order to save energy and in particular to deliver data reliably, each sensor node sends data to a peer sensor node, and stores his own data and the data coming from other sensor nodes. Whenever, the base station is available sensor nodes upload data to it. The advantage of this is that as data is swapped from node to node, i.e. it is not mandatory to collect data from each zebra in order to receive the whole amount of data. This style of peer-to-peer communication spans an ad-hoc network where data is aggregated back to the researcher base station (see Fig. 6-11 b). As on each sensor node data storage is limited a prioritization algorithms is applied.

6.5 Summary

This chapter enlightened what new possibilities may emerge from applying a mobile sensing concept. New possibilities emerge for data collection, if you take advantage of a carrier medium, i.e. human being, robot, vehicle etc wearing or carrying mobile sensors that can be sent over a communication channel. In particular, the deployment of smart phones as sensor nodes facilitates applications that allow for observations of phenomena, which previously were hard to perceive or even impossible.

6.6 Bibliography

Amin, S., S. Andrews, et al. (2008). Mobile century-using GPS mobile phones as traffic sensors: a field experiment. In World congress on ITS, pages 16--20.

Bradley, A. (2009) "Condition Monitoring Application: Condition Monitoring System for Wind Turbine - White Paper ", 10.

Burke, J., D. Estrin, et al. (2006). Participatory sensing. In Procs. of Sensys, Worldwide Sensor Web Workshop, ACM.

Gravina et al., R. (2008). Demo abstract: Spine (signal processing in node environment) framework for healthcare monitoring applications in body sensor networks. Proc. of the 5th European conference on Wireless Sensor Networks**:** pp. 8.

Hopkins, R. and K. Jenkins (2008). Eating the IT Elephant: Moving from Greenfield Development to Brownfield, IBM Press.

Kanjo, E., J. Bacon, et al. (2009). "MobSens: Making Smart Phones Smarter." IEEE Pervasive Computing **8**(4): 50-57.

Kwok, R. (2009). "Phoning in data." Nature, Macmillan Publishers **458**(News Feature): 3.

Lane, N. D., S. B. Eisenman, et al. (2008). Urban sensing systems: opportunistic or participatory? Proceedings of the 9th workshop on Mobile computing systems and applications. Napa Valley, California, ACM.

Lo, B. P. L., S. Thiemjarus, et al. (2005). BODY SENSOR NETWORK – A WIRELESS SENSOR PLATFORM FOR PERVASIVE HEALTHCARE MONITORING. Adjunct Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005), **:** pp.77-80.

Lorincz, K., D. J. Malan, et al. (2004). "Sensor Networks for Emergency Response: Challenges and Opportunities." IEEE Pervasive Computing **3**(4): 16-23.

Parks Associates (2010) "Number of smartphones worldwide exceeded 180 million units in 2009, at Connections Europe Summit 2010."

Paulos, E. (2008) "Environmental Monitoring with Mobile Phones (Ghana), Wireless Technology for Social Change: Trends in Mobile Use by NGOs, Case Study 10."

Paulos, E., R. Honicky, et al. (2008). Citizen Science: Enabling Participatory Urbanism. Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City. Hershey, PA, IGI Global.

Rockwell Automation, I. (2010). "Condition Monitoring Services-Optimize the Results from Your Condition-based Maintenance Program." from http://www.rockwellautomation.com/services/conditionmonitoring/.

Roggen, D., K. Forster, et al. (2009). OPPORTUNITY: Towards opportunistic activity and context recognition systems. World of Wireless, Mobile and Multimedia Networks & Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a.

Seto, E., E. Martin, et al. (2010). Opportunistic strategies for lightweight signal processing for body sensor networks. Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments. Samos, Greece, ACM.

Shilton, K., J. A. Burke, et al. (2008). Participatory Privacy in Urban Sensing, eScholarship Repository.

Wac, K., R. Bults, et al. (2009). Mobile Patient Monitoring: the MobiHealth System. Engineering in Medicine and Biology Society (EMBC). IEEE. Minneapolis, MN.

Weber, T. (2009). "How Smart Does Your Phone Need to Be?"

Work, D., O.-P. Tossavainen, et al. (2008). An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices in Proc. of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, December 2008, pp. 2141-214.

Work, D. B., O.-P. Tossavainen, et al. (2009). Lagrangian sensing: traffic estimation with mobile devices. Proceedings of the 2009 conference on American Control Conference. St. Louis, Missouri, USA, IEEE Press.

Zhang, P., C. M. Sadler, et al. (2004). Hardware design experiences in ZebraNet. Proceedings of the 2nd international conference on Embedded networked sensor systems. Baltimore, MD, USA, ACM.

7. Technologies for Knowledge Derivation from Sensor Information.

This chapter documents the inference strategies and algorithms for the derivation of knowledge from sensor information based on works that have been published in (Jerzak, Klein et al. exp 2011) and (Huang and et 2008).

7.1 Introduction

Decisions in daily life are based on the accuracy and availability of information. Sensor networks can significantly improve the quality of information as well as the ways of gathering it. Sensor networks can help to get higher fidelity or hard-to-obtain information, acquire that information in real time, and reduce the cost of getting it. Because of sensor networks' cost efficient way of gathering information, they carry a great potential for business-critical applications.

Sensor networks consist of a number of sensor nodes, endowed with physical sensing abilities, limited processing and memory. These nodes collectively form a network and forward the gathered information as data streams to a data sink. Intelligent decision-making algorithms usually do not trigger activities based on a single event (such as an individual observation or sensor reading) but often have to consider correlations among events. Frameworks and Toolkits such as open-source project ESPER[11] or DROOLS[12] are already available for complex event processing such as event stream.

Since sensors usually provide vendor specific protocols some pre-processing of the raw sensor data is necessary before it can be used for knowledge derivation. This also applies if the data is derived from different sensor sources. However, there are already standardization efforts in order to provide a neutral format to define the various sensors and systems, their interfaces, the type of information they convey and their communications. Some sensor languages are described in detail in ch. 13.2.

7.2 Sensor Languages

We will now look at the sensor languages EEML, PML and SensorML in detail.

7.2.1 EEML

The Extended Environments Markup Language (EEML)[13] is an XML-based language that describes the data output of sensors and allows for the addition of metadata about the origin of the data, which enables searching for data streams without having to know the exact details of the source.
The developers claim that it also allows for making "spontaneous connections between streams from different sources", however this would require a commonly accepted semantics of the metadata. A design goal of EEML is sharing of sensor data between remote environments in real time.

The most important application of EEML is Pachube[14], a data brokerage platform for the Internet of Things, whose goal it is to store, share and discover real-time sensor data. The system provides measurements of electricity, weather, building management systems, air quality, and radioactivity, while metadata comprise timestamps, geo location, units, and tags.

According to its organizers, it manages millions of data points per day from thousands of organizations and individuals. With recently posted forum messages and news, the web page gives the impression of being more vivid than that of EEML.

[11] http://www.espertech.com/products/esper.php
[12] http://www.jboss.org/drools/
[13] www.eeml.org
[14] http://www.pachube.com

Although from the feature definition and design goals EEML looks promising, it gives the impression of being unfinished. There is an XML schema definition with version number "005", which consists of merely 134 lines.

A formal explanation of the meaning of the EEML elements and attributes is not available. It appears there are no scientific publications or standard documents for EEML available. Several subtopics on the EEML homepage are labeled "in progress 2008". Moreover, Pachube seems to be the only application.

7.2.2 PML

The Physical Markup Language (PML)[15] is an XML-based language developed at the MIT whose main purpose is the description of physical objects and environments. Its applications include inventory tracking, SCM, machine control, and object-to-object communication. The language itself is intended to be general; therefore it is kept rather simple.

With detailed descriptions of XML entities like "date", "location", or "physical properties", it appears to be more sophisticated than EEML. However, XML schemata are not available for download.
Moreover, the latest updates on the PML homepage are from 2002, and many important pages like "API", "Applications", or "Contact" are empty, thus it appears that PML is not used in applications, and the development of the language has been halted.

7.2.3 SensorML

The Sensor Model Language (SensorML)[16] is an XML-based language that was developed by the Open Geospatial Consortium (OGC). It allows for describing geometric, dynamic, and observational characteristics of sensors. For example, it is possible to capture that a sensor is moving or is measuring a remote phenomenon, which unit of measure is used, and how accurate the readings are. Thus, SensorML is a very powerful and complex language, covering a large range of sensors "from simple visual thermometers to earth-orbiting satellites".

Unlike EEML or PML, SensorML is described by a detailed standard document(Botts. 2007), containing over 60 pages of XML schema definitions for the components of the language. The standard also mentions that the semantics of the phenomena to be observed is supposed to be defined by the respective communities and to be referenced by URIs, which fits very well with the intention of combining sensor data with Linked Data.
In the following, we briefly introduce the most important SensorML data types.

Data Component

A data component represents, for example, measured values. The most important subtypes are count, i.e. integer values, and quantity, i.e. real-number values. Besides the value itself, a data component can be assigned a quality value, a unit of measure (only for quantity), a textual description, and a definition as a URI. Data components also can be time values, quantity and count ranges, and more general types like text, numericals, and Booleans.

Position

A position is given via a reference frame that is denoted by a URI, a time and a location vector. For moving sensors, additional vectors like velocity, acceleration and orientation are possible.

Process

Slightly counter intuitively, sensors are represented as processes. This is because processes comprise both physical processes, which involve interaction with the environment, and pure processes, which merely consist of mathematical-logical operations. A process can be an atomic process or, if several steps of data processing are involved, a process chain. All processes can have

[15] http://web.mit.edu/mecheng/pml
[16] http://www.opengeospatial.org/standards/sensorml

inputs, outputs, a name and a textual description. Physical processes, i.e. sensors, are also called components. They additionally can have a position and a spatial as well as a temporal reference frame.

Phenomenon

The semantics of a measured value can be defined via reference to a phenomenon, like temperature, pressure, or count. This allows for referencing dictionaries or ontologies like NASA's Semantic Web for Earth and Environmental Terminology (SWEET)[17].

7.3 Learning Strategies

Machine learning is the science of computer modeling of learning processes. The most common definition from Mitchell (Mitchell 1997) states:
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

The different approaches are decision tree learning, association rule learning, artificial neural networks, genetic programming, inductive logic programming, support vector machines, clustering, Bayesian networks, reinforcement learning. Often-applied learning techniques in sensor networks are Bayesian inference systems, Dempster-Shafer inference or rule-based systems. Therefore, we briefly describe those in this paragraph.

7.3.1 Bayesian inference

Bayesian inference makes use of probability rules for measuring the uncertainties of hypotheses. The likelihood with a value between 0 (absolute disbelief) and 1 (absolute belief) describes whether a certain information is true. Bayes' rule (Bayes 1763) offers a algorithm to calculate the conditional probability P(A|B) of information A, given information B. The prior probability for information A, the conditional probability of B given A and the prior probability of information B have to be estimated beforehand in order to do the calculation. Therefore this inference method is restricted to data processing with the help of statistical databases.

Bayes' theorem can be used to determine the belief of information A based on the result of observing information B. The method can also be used the other way around. During high-level processing information can be validated as either trustworthy or not using Bayesian inference. The inference algorithm to determine missing data from inactive nodes with the help of Bayesian inference was presented in(Hartl and Li 2005).

7.3.2 Dempster-Shafer inference

Dempster-Shafer inference is based on the Theory of Evidence introduced by Dempster (Dempster 1968) and Shafer(Shafer 1976). Their theory provides a formalism that can be used for incomplete knowledge representation, belief updates and evidence combination(Provan 1992). "Dempster-Shafer theory is more flexible than Bayesian Inference for it allows each source to contribute information with different levels of detail." "…in contrast to the Bayesian Inference, the Dempster-Shafer theory allows us to fuse data provided by different types of sensors." (Nakamura, Loureiro et al. 2007)The probabilities are assigned only when the supporting information is available so that a priori probabilities don't have to be assigned to unknown propositions, like in the Bayesian method. The tradeoff between Bayesian accuracy and Dempster-Shafer flexibility is described in (Bracio, Horn et al. 1997).

[17] http://sweet.jpl.nasa.gov

7.4 Business Rules Engines

Rule-based systems

Rule-based systems are used as a means to store and manipulate knowledge to interpret information in a useful way. They are a relatively simple model that can be adapted to any number of problems. As with any artificial intelligence system, a rule-based system has its strengths as well as limitations that must be considered before deciding if it's the right technique to use for a given problem. Overall, rule-based systems are really only feasible for problems for which any and all knowledge in the problem area can be written in the form of if-then rules and for which this problem area is not large. If there are too many rules, the system can become difficult to maintain and can suffer a performance hit. They consist of a knowledge base which contains a set of facts representing the initial working memory, a rule base, which contains a set of if-then-statements including a termination condition that determines that a solution has been found or that none exists and a inference engine (or semantic reasoner), which infers information or takes action based on the interaction of input and the rule base.

Business Rules Management Systems

Business Rules Management Systems (BRMS) provide the ability to easily express the rules in a simple and understandable way by using abstractions, such as flowcharts, decision trees and decision tables as well as scoring models and textual if-then rules(Brett and Gualtieri 2009). Therefore, the benefit of BRMS lies in the fact that the change in the rules can be easily reflected in the system by a non-technical user. BRMS, in contrast to CEP solutions, are a mature offering with good market penetration.

Business rules engines (BRE), which constitute the core of the BRMS offerings, are in most cases designed to accept discrete events which typically have a complex payload (multiple elements)(Brett and Gualtieri 2009). There exist currently a number of commercial offerings for business rules platforms, including, but not limited to: Tibco Business Events[18], UC4 Automation Engine[19] or ruleCore CEP Server[20]. There exist also non-commercial or open source systems, example being: the already mentioned JBoss DROOLS[21] or Jess[22]. The common denominator for most of the business rules engines is the use of the Rete algorithm(Forgy 1982) to process incoming events against the stored rules. The use of the Rete algorithm allows reuse of the common parts of rules and thus reducing the number of operations which are necessary to match incoming events. Defining business rules and processes in SOA environment has been standardized by OASIS by using WS-BPEL.

7.4.1 WS-BPEL (Web Services Business Process Execution Language)

WS-BPEL[23] is a specification that models the behavior of web services in a business process interaction. It is based on the XML grammar which describes the control logic required to coordinate web services participating in a process flow. An orchestration engine can interpret this grammar, thus it can coordinate activities in the process. WS-BPEL is a layer on the top of WSDL (Web Services Description Language). WSDL defines the specific operations and WS-BPEL defines how the operations can be sequenced. Every WS-BPEL process can be considered as a web service using WSDL describing the public entry and exit points for the process. WSDL data types are used within a WS-BPEL process to describe the information that passes between requests. WSDL might be used to reference external services required by the WS-BPEL process. WS-BPEL provides support for both executable and abstract business processes. The executable process models a private workflow. The abstract process specifies the public message exchanges between parties. The executable processes

[18] http://www.tibco.com/software/complex-eventprocessing/businessevents
[19] http://www.truviso.com/
[20] http://rulecore.com/
[21] http://jboss.org/drools
[22] http://www.jessrules.com/
[23] http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

provide orchestration support while the business protocols (abstract processes) focus more on the choreography of the services.

Support for basic and structured activities is included. The basic activities might be receiving or replying to message requests as well as invoking external services. The structured activities specify what activities should run in what order – the whole process flow. These activities also provide support for conditional looping and dynamic branching. The structured activities might specify that certain activities should run sequentially or in parallel. *Containers* and *partners* are two important elements within WS-BPEL. A container is a variable for exchange in the message flow. A partner could be any service that the process invokes or any service that invokes the process. Each partner is mapped to a specific role that it fills within the business process. This is managed by containers.

In WS-BPEL, a set of activities can be grouped into a single transaction – it means that the steps enclosed in the scope should either all complete or all fail. Within this scope, the developer can then specify compensation handlers that should be invoked if an error occurs. WS-BPEL provides a robust exception handling mechanism through the use of throw and catch clauses, similar to the Java programming language. Examples of WS-BPEL engines include NetWeaver Process Integration (SAP), Apache ODE (OpenSource) , Microsoft Windows Workflow Foundation (Microsoft), WebSphere Process Server (IBM), and ActiveBPEL(OpenSource).

7.5 Planning

In AI planning research concentrates on defining a set of actions which, when executed in a correct order, will reach a desired state (goal).

Planning activities are very often domain specific and cannot easily be adapted in another domain, although there is some work also done in this area, e.g.(Wilkins 1984).
One major area of planning is the domain of robotics. Combined with visual and/or tactile sensors, planning is for example used to move a robot in a certain environment(Georgeff and Lansky 1987).
A second main area for planning is the sector of logistics. Urban delivery companies usually acquire their orders in bidding and auctioning processes on short-notice. Furthermore, current traffic conditions instantly influence planned delivery routes, so that they need to be adapted. In order to deal with dynamism, motor carriers must have timely access to all relevant information about the transportation system, i.e. information about current transportation processes as well as information about transportation requests. Fleet telematics systems can be used to track vehicles and obtain information about the state of drivers and vehicles. Data obtained by such fleet telematics systems can be automatically transferred to logistics systems as shown in(Erkens and Kopfer 2001), (Gruhn et al 2003), and(Goel 2007). Goel and Gruhn(Goel and Gruhn 2006a),(Goel and Gruhn 2006b) , (Goel and Gruhn 2006c)show how real-time data can be automatically analyzed and utilized within collaborative problem solving systems.

7.6 Summary

This section dealt with deriving knowledge from sensor data. One of the techniques is by tagging sensor data with metadata. The tagging process is defined in sensor language such as SensorML and EEML. Metadata can be the processed to derive the meaning of the data. Processing the metadata can be done by inference algorithms such as Bayesian and Dempster-Schafer, by using business rule engines. The main drawback for rule engines is that rules are not flexible meanwhile in the real world there exist exceptions and exceptions of an exception which make rules do not apply anymore. Thus, in ebbits we would like to investigate the business rule engines, that are commonly used in industry and hybrid solution between rule engine and inference approaches. WS-BPEL provides a a standard language that is used by rule engines in SOA.

Deriving knowledge from raw data is needed in ebbits especially in multi-sensor fusion framework discussed in Section 5. In which data are processed into information and knowledge in a distributed and collaborative way.

7.7 Bibliography

Bayes, T. R. (1763). An essay towards solving a problem in the doctrine of chances. In T. R. Society, *Philosophical Transactions of the Royal Society 53* (pp. 370-418).

Botts., M. (2007). *OpenGIS sensor model language (SensorML) implementation spec-*. Retrieved from http://www.opengeospatial.org/standards/sensorml

Bracio, B. R., Horn, W., & Moller, D. P. (1997). Sensor fusion in biomedical systems . *Proceedings of the 19th Annual International Conference of the IEEE* , 1387 - 1390 vol.3 .

Brett, C., & Gualtieri, M. (2009). *Must you choose between business rules and complex event.* Forrester Research.

Dempster, A. P. (1968). A generalization of Bayesian inference. *J. Royal Stat. Soc., Series B 30* , 205—247.

Erkens, E., & Kopfer, H. (2001). WAP-LOG: Ein System zur mobilen Fahrzeugeinsatzsteuerung und Auftragsfortschrittkontrolle. In H.-J. Sebastian, & T. Grönert (Eds.), *Logistik Management - Supply Chain Management und e-Business.* Stuttgart: Teubner.

Forgy, C. (1982). Rete: A Fast Algorithm for the Many Patterns/Many Objects Match Problem. *Artif. Intell.* , 17-37.

Georgeff, M. P., & Lansky, A. L. (1987). Reactive Reasoning and Planning. *The Sixth National Conference on Artificial Intelligence.* Seattle.

Goel, A. (2007). *Fleet Telematics - Real-Time Management and Planning of Commercial Vehicle Operations.* Springer.

Goel, A., & Gruhn, V. (2006a). A Fleet Monitoring System for Advanced Tracking of Commercial Vehicles. *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2006)*, (pp. 2517-2522). Taipei.

Goel, A., & Gruhn, V. (2006c). An Operations Support System for Truckage Companies. *International Conference on Information Systems, Logistics and Supply Chain (ILS 2006)*, (pp. 1213-1222). Lyon.

Goel, A., & Gruhn, V. (2006b). Collaborative Dispatching of Commercial Vehicles. *2nd IEEE International Conference on Cybernetics & Intelligent Systems (CIS 2006)*, (pp. 115-120). Bangkok.

Gruhn et al, V. (2003). Mobile communication systems for truckage companies. In G. Giaglis (Ed.), *The Second International Conference on Mobile Business* (pp. 337–344). Austrian Computer Society.

Hartl, G., & Li, B. (2005). A Bayesian inference approach towards energy efficient . *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)* (pp. 371 - 380). Washington: IEEE.

Huang, V., & et, a. (2008). *SENSEI - WP2 – Sensor Information Services - Deliverable report.*

Jerzak, Z., Klein, A., & Hackenbroich, G. (exp 2011). GINSENG Data Processing Framework. In S. Helmer, A. Poulovassilis, & F. Xhafa (Eds.), *Reasoning in Event-based Distributed Systems.* Springer Engineering Book Series.

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill.

Nakamura, E. F., Loureiro, A. A., & Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv. 39, 3 (Sep 2007)* .

Provan, G. M. (1992). The validity of Dempster-Shafer belief functions. *Int. J. Approx. Reasoning 6, 3 (May)* , 389—399.

Shafer, G. (1976). *A Mathematical Theory of Evidence.* NJ: Princeton University Press, Princeton.

Wilkins, D. E. (1984). Domain independent planning: representation and plan generation. *Artificial Intelligence* , 22:269-301.

8. Control and Resource Management

8.1 Definition

Dorf and Bishop defines *control system as an interconnection of components forming a system configuration that will provide a desired system response* (Dorf and Bishop 2008). Control engineering concerns with analysis and design of goal oriented design. In modern control theory, many approaches concern with self-organizing systems aiming at robustness, adaptive and flexibility of complex systems. In modern industrial applications, feedback control is a fundamental theory that has been applied extensively. An example of feedback control in a closed loop can be found in an automatic steering of a car as shown in Fig. 8-1, automatic production lines using robotic arms, and self-regulate power generators.

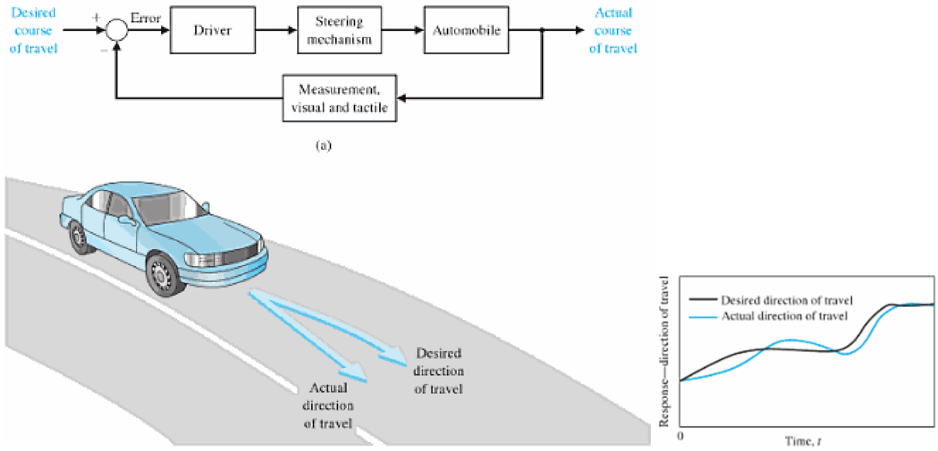


Fig. 8-1. Steering control of a car (Dorf and Bishop 2008)

The mathematical foundation of feedback control system is the differential calculation that compares the desired response with the actual response. Differential equations can be as simple as estimating the voltage differences to linear and time-variant approximation of physical system that can be solved by obtaining linearized differential equation, transform the differential equation using Laplace transform, and solving the algebraic equation for the transform of the variable of interest.

8.2 Industrial Control System

Industrial control system covers several standard solutions that have been used in automatic manufacturing system such as Supervisory Control and Data Acquisition System (SCADA), Distributed Control System (DCS), Advance Process Control, and smaller controller units such as Programmable Controller Logic (PLC). SCADA usually supervises and coordinates an entire sites or even sites in a country. However it does not control in real time. SCADA normally consists of Human-Machine Interface, a supervisory system (computer), remote terminal units (RTU), PLC, and communication modules that work over protocols such as IEC 60870-5-101 or 104, IEC 61850, DNP3 serial, and DNP3 LAN/WAN. Few works have begun offering web service integration (Lipnickas, Rutkauskas et al. 2009). Most control actions are only basic overriding and supervisory that performed automatically through RTUs and PLCs. For instance, a PLC may control the flow of cooling water through part of an industrial process, but the SCADA system may allow operators to change the set points for the flow, and enable alarm conditions, such as loss of flow and high temperature, to be displayed and recorded.

PLC is a digital computer used for automating electromechanical processes that evolves from automotive industry. It allows reconfiguration of electromechanical machines such as robots for assembly lines. PLC utilizes a real time operating system and it is programmed with ladder logic. Modern PLCs allow other programming languages such as C, BASIC, and State Logic. Another form of a controller is called Programmable Automation Control (PAC). PACs serve the same purposes as PLCs, they do however offer open and modular architectures. (Opto22 2008)

Fig. 8-2. Programmable Automation Controller with Multiple Ethernet and Serial Ports SNAP-PAC-S2 (Fig. is taken from Opto22)

DCSs are dedicated for controlling automated processes of batch productions. DCSs are intended to distribute intelligence in the plants by establishing processing in pro7cess control. Nowadays the combination of functionalities of DCS and PLC are offered as all-in-one solution by many vendors which is called "Process Automated System". Nonetheless the complexity of manufacturing processes needs more integration with other technologies such as database integrity, system maturity, etc. Recently, DCS evolved into collaborative process automation system (CPAS) that becomes the primary source for collaborative manufacturing management (CMM) applications (Hollender 2009). The principles of CPAS include:

- Extraordinary Performance
- Continuous Improvement
- Proactive Execution
- Common Actionable Context
- Single Version of the Truth
- Automate everything that should be automated
- Facilitate Knowledge Workers
- Common Infrastructure based on standards

Safety controls in manufacturing equipments are regulated by *Integrated Control and Safety System* (ICSS). The ICSS usually includes three types of safety system such as Process Shutdown System, Emergency Shutdown/Depressurization, and Fire-Gas Safety System. The emergency shutdown protects personals by shutting down processes immediately and in a safe state when emergency condition rises. Fire-Gas should detect gas leak in early stage to avoid fire and explosions. ICSS controls equipment through a safety instrument system module that is equipped with sensors capable to detect anomalies and dangerous events.

8.3 Control as resource management

In contrast to multi-sensor fusion that functions to perform estimation, resource management functions as part of control. In mathematics, control and estimation are expressed as duality. (Liggins, Hall et al. 2009) describes the relationship between resource management and information fusion as a control loop, which is named Information-processing cycle (IPC) (depicted in Fig. 8-3). IPC consists of an information fusion component that support decision making process and a resource management in which decision about resource usage takes place. They also clarify that the role of resource management is to transform decision-makers' information needs to affect real-world actions in a manner that produces data and information that will satisfy those needs.

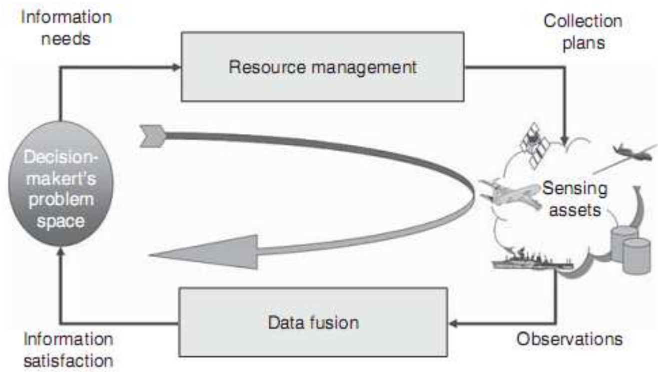


Fig. 8-3. Information - Processing Cycle.

Resource management is a transformation process started with understanding the objectives, the observables that are expected to satisfy those objectives, and the sensing tasks required to gather those observables. The tasks contain sensing requirements for instance: time, location, geometry, accuracy, signatures. TRIP (Transformation of requirements for information process) model suggests a framework the stages that decomposes objectives into sensing requirements such as sensors selection and tasks assignment. The transformation means the translation of the needs into specific technical means or database searches (e.g.: location, time, conditions). The model describes the stages in control paradigm including information elements, effects, dependencies, and assumptions.

Decomposition of user requirements is represented in multiple levels as depicted in Fig. 8-4. *Level 3. Information needs* describes the needs of information about environment or situation such as location, behavior, condition of entities. Such needs do not dependent on the means of data gathering however they could include specific space and time requirements. *Level 2. collection objectives* describes the partitioning process of objective into tasks in which tasks' dependencies are considered especially when resolving concurrent request of resources. *Level 1. Observable* provides ways of determining methods of observations including sensing modalities, parameters, and physical constrains. *Level 0. Task and Plan* describes one or more collection of tasks to a selected sensor/source/platform combination and constrains under which the tasks are to be completed. *Level 4. dynamic re-planning* describes provides control and update to level 0 - level 3 in a dynamic environments according to the change of priorities, constrains, and contingencies.

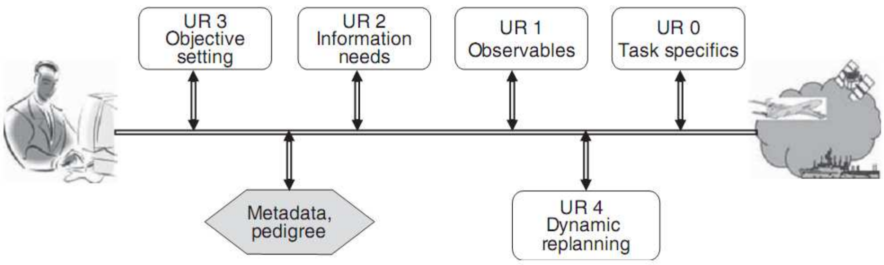


Fig. 8-4. TRIP Model(Liggins, Hall et al. 2009)

The complementary relationship between resource management and data fusion can be described as follow: resource management seeks to maximize the value of the collected information, whereas data fusion seeks to minimize the uncertainty of the information. The levels described in Fig. 8-5 can be mapped to the JDL model as depicted in the table.

Dual Data Fusion and Resource Management Processing Levels

Level	Data Fusion Level	Resource Management Level
0	Signal/feature assessment: estimation of signal/feature states	Signal management: management of resource emissions/observables
1	Entity assessment: estimation of entity attributive states	Resource response management: management of individual resources
2	Situation assessment: estimation of entity relational/situational states	Resource relationship management: management of resource relationships
3	Impact assessment: estimation of the impact of fused states on mission objectives	Mission objective management: management of mission objectives
4	Process assessment: estimation of MOP/MOE states	Design management: management of system engineering and operational configuration

Fig. 8-5. Data Fusion and Resource Management Duality

8.3.1 Resource management for energy conservation in WSN

(Anastasi, Conti et al. 2009) defined taxonomy of energy conservation techniques in WSN. They classify the current techniques into three main approaches: duty cycling, data-driven, and mobility. The approach that corresponds to a control strategy is Duty-cycling. In local sense, it manages nodes' resources by putting the network radio into low power sleep mode whenever it is not needed to communicate. This technique is called *power management.* It can be classified into (1) *sleep-wake up protocols* that is implemented on top of MAC and (2) *MAC protocol with low duty cycle* that is implemented as part of the MAC layer. In global sense of sensor network, duty cycle could act as a *topology control* that activates only a minimum subset of a network that is necessary to ensure connectivity within the network. This could only be applied if there is redundancy in the network. The subset of selected nodes is adaptively rotated to ensure that all nodes deplete their energy supply at almost the same time. Topology control is divided into location driven and connection driven techniques.

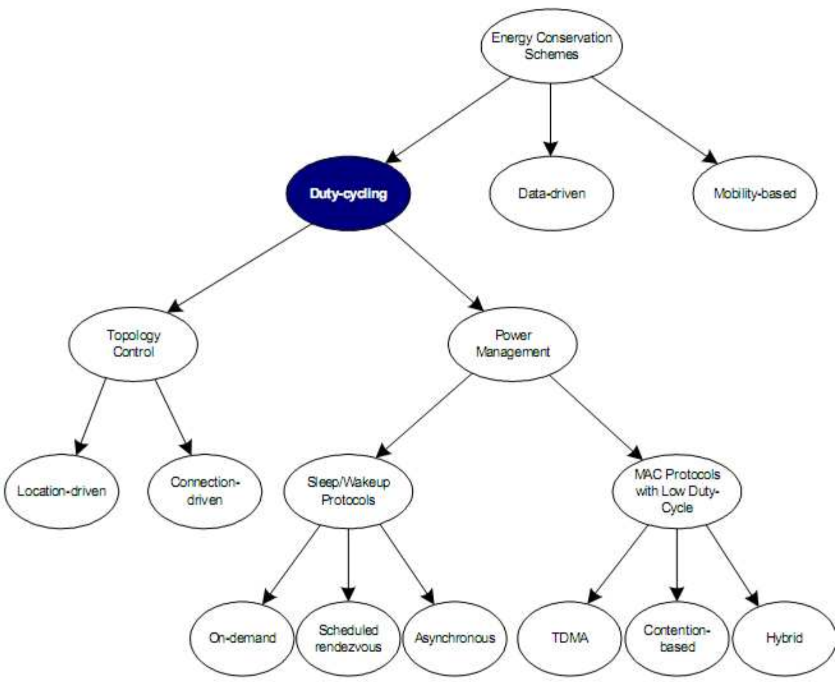


Fig. 8-6. Classification of Duty Cycling Techniques (Anastasi, Conti et al. 2009).

Topology control

Topology control is of course more effective if the network is quite dense and offers a high degree of redundancy. This might the case, if the network deployment is done randomly (e.g.: dropping sensor nodes from an aircraft), since to ensure connectivity, a larger number of nodes than necessary are deployed. Topology control can exploit several criteria to activate / deactivate nodes for instance GAF (geographical adaptive fidelity) takes advantage of location to decide which subset is to be activated while preserving a constant level of routing fidelity (Xu, Heidemann et al. 2001). GeRaF use nodes position and redundancy to take decision (Zorzi and Rao 2004). SPAN uses connectivity for adaptively electing coordinators, which have to stay awake all the time and perform multi-hop routing (Chen, Jamieson et al. 2002). ASCENT approach is to let a node to decide whether to join the network or continue to sleep based on information about connectivity and packet loss that are measured locally by the node itself (Cerpa and Estrin 2004). The basic idea is that most of the nodes are initially only listen to packet but do not send. If the sink experiences heavy message loss, it send "help" messages to trigger neighboring nodes become active.

Power Management

Sleep/wake-up protocol

Sleep/wake-up protocol can be applied to different component of the node (e.g.: Radio subsystem, sensor interface). Sleep/wake-up protocol can also be divided into three types: on-demand. Scheduled rendezvous, and asynchronous schemes.

In On-demand, nodes or components should wake up only when another node wants to communicate with it and vice versa. The main problem is that when sleeping, nodes are not able to know when other node tries to communicate with them. Several approaches make use of a low powered second radio for listening to beacons. STEM (Sparse Topology and Energy Management) uses two different radios for wakeup signal and data packet. Each node turns its wakeup radio asynchronously for a period of time. The source node sends a stream of beacon on the wakeup channel. Once the target node receives a beacon, it sends back an acknowledgement and turns its data radio on. A study indicated that combination of GAF and STEM can reduces 1% of energy consumption which increases the network lifetime by a factor of 100 (Schurgers, Tsiatsis et al. 2002).

An alternative is to schedule a rendezvous time at which neighboring nodes supposed to wake up and remain active for a period of time to listen if other nodes try to reach them. To ensure nodes wake up at the same time, clock synchronization among nodes is needed. An example of this kind of algorithm is *Fully Synchronized Pattern,* in which all nodes wake up periodically at T_{wakeup} and remain active for a fixed T_{active} (Keshavarzian, Lee et al. 2006). T_{wakeup} must be significantly larger than T_{active}. This techniques is also used in TinyDB, TASK (Buonadonna, Gay et al. 2005)and several MAC protocol such as S-MAC (Ye, Heidemann et al. 2004) and T-MAC (Van Dam and Langendoen 2003). Keshavarzian et al. also proposes further technique such as Staggered *Wakeup Pattern*, *Two-Staggered Pattern, and Crossed Staggered Pattern.* The idea of staggered wake up is that nodes located at different levels of data-gathering tree wake up at different time. The active time of adjacent nodes must partially overlap to allow nodes communicate with their children and the active parts of different levels are arranged so that the T_{active} of a node uses to communicate with its children is equal to the portion it uses to communicate to its parent as depicted in . staggered patterns are used in TinyDB, TAG(Madden, Franklin et al. 2002), and D-MAC(Lu, Krishnamachari et al. 2004).

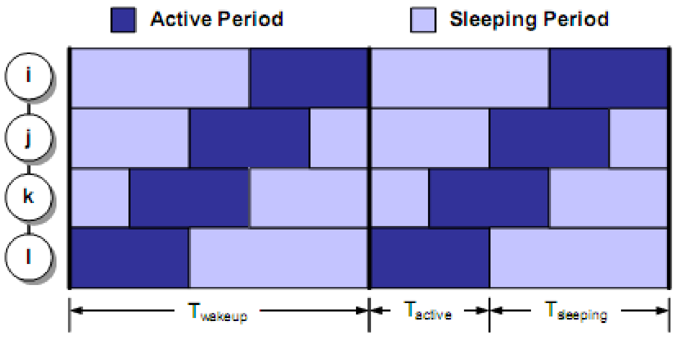


Fig. 8-7. Staggered Wake Up Pattern (Anastasi, Conti et al. 2009).

The drawback of synchronized wakeup method is collision is likely to happen since most of the nodes are transmitting almost at the same time.

Another approach is to use asynchronous wakeup (Tseng, Hsu et al. 2003; Zheng, Hou et al. 2003). The idea is nodes can wake up whenever they want and able to communicate with its neighbors and by guarantying that neighbors always have an overlapping active time. Zheng et al. proposes an Asynchronous wakeup protocol. Each node maintains a function that generates a wake up schedule. The wakeup function takes advantage of a combinatorial design problem.

Duty Cycle in the MAC protocol

TDMA (Time division Multiple access) schemes uses time slots for enabling duty cycle. Each node only turns on the radio on the time. TDMA has been used mainly with hierarchical topology, in which the cluster heads assign the time slots for their leaf nodes as in Bluetooth (Haartsen and BV 2000) and LEACH (Heinzelman, Chandrakasan et al. 2002). (Rajendran, Obraczka et al. 2006) proposed TRAMA that divides time in two portions, a random access and a scheduled access period. The random access period is used to reserve time slots using contention technique. The contention is also used to derive information of 2-hop neighbor to avoid collision. Each node gets a priority to own a specific slot, and the node with the highest priority becomes the owner. FLAMA reduces traffic for deriving information by using pull mechanism.

A well known contention based approach is called B-MAC (Polastre, Hill et al. 2004) that is shipped with TinyOS (Levis, Madden et al. 2005). B-MAC makes use of Low Power Listening (LPL), in which nodes wakeup periodically to check channel's activity. If there is activities, nodes power up and stay awake. If no packet is received, the timeout tells node to go back to sleep. A famous MAC protocol for sensor network is S-MAC. It exchanges sync packets to organize the sleep/wakeup periods. Each node can randomly follow the neighbors' schedule or create its own schedule. Nodes having the same schedule form a virtual cluster. A node can also follow several schedules and thus be a bridge to several clusters. IEEE 802.15.4 is a standard for personal area network which is used by Zigbee and 6lowpan. It supports two mode, beacon enabled and non-beacon enabled. In beacon enabled mode, coordinator defines a *superframe structure* in beacon. As depicted in Fig. 8-8, each superframe consists of an active period and an inactive period. In the active period, devices communicate with the coordinator they are associated with. The active period can be further divided in a contention access period (CAP) and a collision free period (CFP). During the CAP a slotted CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) algorithm is used for channel access, while in the CFP a number of guaranteed time slots (GTSs) can be assigned to individual nodes. During the inactive period devices enter a low power state to save energy. In the non-beacon enabled mode there is no *superframe* structure, i.e., nodes are always in the active state and use an unslotted CSMA/CA algorithm for channel access and data transmission.

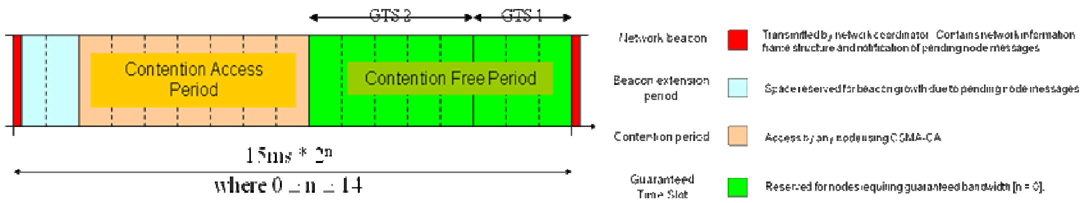


Fig. 8-8. IEEE 802.15.4 Superframe Structure

Hybrid approach switches the protocol between TDMA (Time Division Multiple Access) and CSMA/CA depending on the level of contention. An example of this approach is Probabilistic TDMA(PTDMA) (Ephremides and Mowafi 2006) and Z-MAC (Rhee, Warrier et al. 2008). PTDMA was designed to work in one-hop WLAN environment. It adjusts nodes' ownership to the time slots depends on the number of senders. In contrast, Z-MAC is designed to work in multi-hop WSN. In setup phase of Z-MAC, each node maintains (i) 2-hop neighbors information to ensure nodes within 2-hop range are not assigned the same slots and (ii) a local time frame that depends on the number of neighbors. The local slot assignment and the local time frame are forwarded to its two hop neighbors. Thus each node has slot and time information of its neighbors and all synchronize to a reference slot. After the setup phase is finished, nodes can be in either *Low Contention Level (LCL)* mode or *High Contention Level (HCL)* mode*.* All nodes are initially in LCL, in which any nodes may compete accessing the channel. However the owners have a higher priority. When a node experiences high contention, it sends an *Explicit Contention Notification (ECN)*. Nodes receiving ECN within the last T_{ECN} period change to HCL mode. In HCL, only owners of the current slot and one-hop neighbors are allowed to compete for the accessing the channel.

8.4 Mobile Sensing and Control Applications

Mobile sensors can enhance resource and control management. In this sub chapter two applications for controlling resources while deploying mobile sensors will be described:

- Using sensor information to reduce the carbon footprint caused by food.
- Using sensor and actuator networks to control cattle for protecting sensitive areas.

8.4.1 Utilizing Sensor Information for reducing the carbon footprint caused by perishable goods

Perishable goods, such as meat, fish, dairy products, fruits represent vital parts of human's nutrition. In (Goldman et al 2002; Tsiros and Heilman 2005) it is shown that their availability, presentation, and perceived quality are more relevant to consumers' store choice than the availability of branded products. In industrial countries "food-conscious" society has widely acknowledged the importance of perishable goods. However, the management of food is still quite challenging for retailers and supply-chain partners: transport, improper storage, and specific regional conditions can result in products that are over-expensive, and in worst case need to be thrown away. For instance, 10% of the total industrial and commercial waste in the United Kingdom is due to perishable food products (UK Department for Environment 2007). Furthermore, the food supply's environmental footprint is a major concern: in Europe, between 20% and 30% of the greenhouse gas emissions are due to the production, transport, preparation, and storage of perishable goods (European Commission 2006).

Wireless sensor networks can help to support a more efficient food supply chain, and this is what the ebbits project is targeting on. As perishable food products are susceptible to environmental changes, it is sense making to measure environmental parameters, such as humidity, temperature, or vibration and shock. As, temperature is one main cause for letting food perish, a lot of companies started to use temperature logger inside transport containers. For this the industry uses analog Partlow recorders[24] to measure the return air flow to get an indication of the cargo's state. Usually, the recorder's display is outside of the container. Currently, decisions to accept or reject shipments are often made on the basis of such a Partlow chart.

[24] Partlow Temperature Recorder, http://www.partlow.com/content.aspx?id=12.

Fig. 8-9.Temperature tracking technologies. (a) Partlow process and temperature recorder, and (b) Semi-passive RFID temperature logger tag from Caen.

Though, goods inside a container are not exposed to a uniform temperature level. For instance, Chiquita found out that the temperature inside a single container may vary up to 35% from palette to palette (O'Connor 2006). Those variations have lead in past to rejections of shipments. So far, only the ambient temperature has been measured, but this is insufficient. The industry needs to apply a more fine-grained temperature tracking concept, i.e. details have to be considered as sun exposure, respiration heat of perishable goods, total air circulation rate and distribution. For example, a semi-passive RFID tag from Caen called A927Z[25] can be equipped with a temperature sensor that ranges up to 10 meters. Such sensor tag has a three to five year battery life, and allows for reading the temperature history through a RF interface.

Based on the totality of sensor information fused to some meaningful index a first-expire-first-out (FEFO) policy can be applied. As always, when new technology is planned to be deployed, there is a fear that the additional costs (in this case the manufacturing, transporting, and disposal of sensors) might be in the final too high. In the paper of (Ilic, Staake et al. 2009) the value of sensor-based replenishment configurations with respect to profit and emissions is reported on the basis of a case study. Their results show that emission reductions from avoiding waste and reducing the number of shipments are more than compensate for the emissions due to sensor manufacturing, i.e. referring to companies that use sensor technology effectively money can be saved, and further the carbon footprint being reduced.

8.4.2 Wireless Sensor Actuator Networks (WSAN) for protecting environmentally sensitive areas

Advances in wireless sensor and actuator networks (WSAN) have the potential to radically shift in future the management of agriculture:

The earth is permanently confronted with the risk of considerable and irreversible damage. An immense strain is placed on our environment due to increasing populations and consumption of natural resources. As climate has significantly changed in the last decade, there is a strong attempt to find better ways to control our resources in order to sustain our lifestyle.

A major threaten for our environment is the damage caused by cattle herds. Many landscapes exist where it would be difficult and expensive to install conventional fencing. For instance, along river banks, scattered riparian areas, or steep terrain regions the installation of static sensors would be costly and cumbersome as well. These areas are either large districts that are inappropriate for grazing or sensitive districts where cattle may cause severe damage, such as erosion along river banks or the destruction of valuable plant species in riparian zones. The global impact of animal agriculture is enormous, as circa 1.3 billion cattle cover grassland areas in a total of 10 million square km (Food and Agriculture Organization of the United Nations (FAO) 2007). Hence, livestock management must be performed in a sustainable way in order to avoid significant and widespread environmental degradation in sensitive areas.

Several recent studies have built reliability into long-term, environmental sensor networks, and others have focused on embedding mobile nodes with animals and monitoring them. In (Wark, Swain et al. 2009) systems are described that go beyond simple monitoring of animals and their surrounding environment, such as ZebraNet project (see 6.4.5). In fact, they can change via

[25] Caen Temperature Logger, http://www.caen.it/rfid/syproduct.php?fam=tag&mod=A927Z.

actuator networks the behavior of animals in order to assist environmental stewardship. This is achieved through 'virtual fencing' keeping cattle away from specific areas where no physical fence is installed; the dashed red line in the Fig. below represents the virtual fence separating the cattle on grassland from the exclusion zone.

Fig. 8-10. Virtual Fencing (Wark, Swain et al. 2009): Mobile actuator nodes to control large cattle herds

To obtain observation from the environment static nodes networks are deployed that measure air temperature and soil moisture. All mobile nodes affixed on the cows can undertake actuation on the basis of observations made by other nodes. For pointing cows that they would cross a virtual border, a combination of auditory and mild electrical stimuli is applied by a position-aware device worn by the animal; strict animal ethics guides this experiment to meet animal welfare requirements. For instance, for the flight-response state, i.e. when an animal runs into the exclusion zone rather than pausing or turning back, the system will automatically deactivate the virtual fence until the desire to rejoin the herd makes the animal to return into the non-exclusion zone.

Such an experiment proves the feasibility of spatial control of large cattle herds using sensor and actuator networks, and thus designates a new way of environmental protecting in cattle production. Finally, we conclude that since sensor and actuator networks provide fair enough reliability and performance, there might be on this basis a potential to shift farm enterprise paradigms.

8.5 Summary

Control mechanisms presented in this chapter cover the control theory that have developed in manufacturing as well as controlling resources in wireless sensor devices that have been foreseen as the future technology which will be used in manufacturing. Wireless sensor and actuator network (WSAN) propose a promising technology to replace and complete the existing technology in the manufacturing plans as well as food production because wireless offers a better flexibility in deployment than wired solution. However, wireless communication introduces higher complexity in communication means as it is more vulnerable against interferences. There have been many theoretical research and work being done to increase the wireless network resilience. However exits only limited works on the field. Thus, further studies applying wireless networks in the manufacturing plants and food production in context of ebbits project should be conducted. The existing control theories always assume a perfect that network reliability which is not the case in wireless communication therefore in ebbits there is a need to introduce communication parameters within the existing control theories. Energy supply in WSAN is an important factor. In this section, many works have been explored to conserve energy supply of WSAN by controlling the nodes and network behaviors. In ebbits we need to study the communication patterns in order to conserve the energy supply effectively.

In section 8.4 some real world examples of mobile sensing and control are elaborated. These examples give an overview how the application of WSAN in ebbits will look like.

8.6 Bibliography

Anastasi, G., M. Conti, et al. (2009). "Energy conservation in wireless sensor networks: A survey." Ad Hoc Networks **7**(3): 537-568.

Buonadonna, P., D. Gay, et al. (2005). Task: Sensor network in a box, IEEE.

Cerpa, A. and D. Estrin (2004). "Ascent: Adaptive self-configuring sensor networks topologies." IEEE Transactions on Mobile Computing: 272-285.

Chen, B., K. Jamieson, et al. (2002). "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks." Wireless Networks **8**(5): 481-494.

Dorf, R. and R. Bishop (2008). Modern control systems, Pearson Prentice Hall.

Ephremides, A. and O. Mowafi (2006). "Analysis of a hybrid access scheme for buffered users-probabilistic time division." Software Engineering, IEEE Transactions on(1): 52-61.

European Commission (2006). Environmental Impact of Products (EIPRO): Analysis of the Life Cycle Environmental Impacts Related to the Final Consumption of the EU-25, EUR 22284 EN.

Food and Agriculture Organization of the United Nations (FAO). (2007). from http://faostat.fao.org.

Goldman et al, A. (2002). "Barriers to the Advancement of Modern Food Retail Formats: Theory and Measurement." J. Retailing **78**(4): 14.

Haartsen, J. and E. BV (2000). "The Bluetooth radio system." IEEE Personal Communications **7**(1): 28-36.

Heinzelman, W., A. Chandrakasan, et al. (2002). Energy-efficient communication protocol for wireless microsensor networks, IEEE.

Hollender, M. (2009). Collaborative Process Automation Systems, Isa.

Ilic, A., T. Staake, et al. (2009). "Using Sensor Information to Reduce the Carbon Footprint of Perishable Goods." IEEE Pervasive Computing **8**(1): 22-29.

Keshavarzian, A., H. Lee, et al. (2006). Wakeup scheduling in wireless sensor networks, ACM.

Levis, P., S. Madden, et al. (2005). "Tinyos: An operating system for sensor networks." Ambient Intelligence: 115-148.

Liggins, M. E., D. L. Hall, et al. (2009). Handbook of Multisensor Data fusion, Theory and Practice. Boca Raton, FL, CRC Press.

Lipnickas, A., R. Rutkauskas, et al. (2009). Interoperability of SCADA system applications with Web Services, IEEE.

Lu, G., B. Krishnamachari, et al. (2004). An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks.

Madden, S., M. Franklin, et al. (2002). "Tag: a tiny aggregation service for ad-hoc sensor networks." ACM SIGOPS Operating Systems Review **36**(SI): 131-146.

O'Connor, M. C. (2006). "Cold-Chain Project Reveals Temperature Inconsistencies." RFID Journal.

Opto22 (2008) "Understanding Programmable Automation Controllers (PACs) in Industrial Automation."

Polastre, J., J. Hill, et al. (2004). Versatile low power media access for wireless sensor networks, Citeseer.

Rajendran, V., K. Obraczka, et al. (2006). "Energy-efficient, collision-free medium access control for wireless sensor networks." Wireless Networks **12**(1): 63-78.

Rhee, I., A. Warrier, et al. (2008). "Z-MAC: a hybrid MAC for wireless sensor networks." IEEE/ACM Transactions on Networking (TON) **16**(3): 511-524.

Schurgers, C., V. Tsiatsis, et al. (2002). "Optimizing sensor networks in the energy-latency-density design space." IEEE Transactions on Mobile Computing: 70-80.

Tseng, Y., C. Hsu, et al. (2003). "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks." Computer Networks **43**(3): 317-337.

Tsiros, M. and C. Heilman (2005). "The Effect of Expiration Dates and Perceived Risk on Purchasing Behavior in Grocery Store Perishable Categories." J. Marketing **69**(2): 15.

UK Department for Environment (2007). Report of the Food Industry Sustainability Strategy Champions' Group on Food Transport, Food and Rural Affairs.

Van Dam, T. and K. Langendoen (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks, ACM.

Wark, T., D. Swain, et al. (2009). "Sensor and Actuator Networks: Protecting Environmentally Sensitive Areas." IEEE Pervasive Computing **8**(1): 30-36.

Xu, Y., J. Heidemann, et al. (2001). Geography-informed energy conservation for ad hoc routing, ACM.

Ye, W., J. Heidemann, et al. (2004). "Medium access control with coordinated adaptive sleeping for wireless sensor networks." IEEE/ACM Transactions on networking **12**(3): 493-506.

Zheng, R., J. Hou, et al. (2003). Asynchronous wakeup for ad hoc networks, ACM.

Zorzi, M. and R. Rao (2004). "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance." Mobile Computing, IEEE Transactions on **2**(4): 337-348.

9. Consolidation of Distributed Information and Knowledge

This chapter gives an overview of approaches that bring together and access information from distributed sources. This general task is of importance in multiple research and technology fields. We limit the discussion on the fields of Information Retrieval, Databases, Web-based systems as well as selected research projects we participated in that dealt with consolidating distributed knowledge in enterprise environments.

9.1 Approaches from Information Extraction

Information Extraction refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources (Sarawagi 2007). As such, it deals with the consolidation of knowledge distributed across various unstructured and structured sources to make sense of it and provide useful applications and services. Especially relevant aspects in this context are data integration and duplicate detection.
Data integration in information extraction currently follows a number of different approaches. (Mohania 2008) describes new trends in this area. This includes *Symbiotic Context Oriented Information Integration (SCORE)* (Roy et al. 2005). With SCORE, SQL queries are transformed into keyword queries. This allows the integration of documents and relational databases within one query system. *Entity Recognition in the Context of Structured Data (EROCS)* (Venkatesan et al. 2006) is another approach to linking text documents to relevant structured data. Further strategies include *Event Based Information Integration (EBII)* for event-based updates of data warehouses as well as *Stream Information Integration (SII)* for integrating streaming data, as common in telecommunications. All these approaches consolidate data from heterogeneous sources by integrating data from unstructured sources into structured, relational databases or at least provide links between the two data types.

Another important step to consolidate information extracted from distributed sources is *duplicate extraction*. Duplicates can be detected with numerous algorithmic solutions. (Charikar2002) and (Gionis et al. 1999) describe the LSH algorithm that leverages locally sensitive hash functions on a k-dimensional vector space in order to determine similarity values for two documents. (Xi et al. 2005) describe SimFusion that provides similarity comparison for arbitrary data objects. Furthermore, DSC (Broder2000), SIF (Manber1994) oder I-Match (Chowdhury et al. 2002) can be used for duplicate detection. These approaches primarily focus the clustering of similar text documents. Research in the area of duplicate detection is less concerned with the foundation than with the applicability of individual algorithms in relation to data integration of heterogeneous data sources, as well as with optimizing the approaches for this task.

9.2 Approaches from Databases and Information Management

Consolidating data from distributed sources has been a research topic in Information Management for many years. In our context, approaches for data fusion as well as the relatively recent emergence of the concept of the dataspace are of particular interest.

Data Fusion (Bleiholder and Naumann 2008) aims at fusing records from multiple structured sources (e.g. databases), provide conflict resolution strategies to applications and handle issues like contradictory information and incompleteness. Data Fusion supplements earlier data integration steps of schema mapping (transformation of data structures into a common representation) and duplicate detection (finding possibly inconsistent representations of the same real-world object with approaches as discussed above). Research in the database community especially focused on performing data fusion by applying relational database operators in a scalable way. Strategies include approaches based on join, based on union as well as other techniques, like "considering all possibilities" or "considering only consistent possibilities" (Bleiholder and Naumann 2008).

A more recent concept within the databases community is *dataspaces* (Franklin et al., 2005). Dataspaces provide an architecture around which ongoing research on reference reconciliation, schema matching and mapping, data lineage, data quality and information extraction are unified (Halevy et al. 2006). In contrast with other information integration systems, dataspaces systems offer best-effort answers before complete semantic mappings are provided to the system. A key idea of dataspaces is that the semantic cohesion of a dataspace is increased over time by different parties providing mappings. This is the same pay as you go data integration approach that currently emerges on the Web of data (see below). Rather than providing new algorithms for consolidating distributed knowledge, dataspaces offer a new process for organizing the knowledge in a flexibly way with the right balance between the need for centralized integration and the advantages of decentralized management.

9.3 Approaches from the Web

Since its invention, the Web has provided the world with a platform of distributed knowledge of growing importance. Its major principle is the decentralized publication of information and subsequent linking as a first form of consolidation. We give a brief overview of more advanced forms of consolidation of the Web's unstructured and structured data.

Advanced forms of consolidation have been applied to the still predominant *unstructured content* of the Web by applying the information extraction approaches listed above. One example for such work is the YAGO project (Kasneci et al., 2008), which aims at gaining precise knowledge from different sources. YAGO relies on few core sources that are assumed to provide correct information and semantically connects this information. Core extractors use rules to derive the knowledge base. The extracted facts are further restricted to those validated using the WordNet taxonomy. In a second two-step process, additional information is gathered from Web resources that is then judged with regards to the existing knowledge base. The gathered and consolidated knowledge is then leveraged in query facilities that can answer questions like "Which Nobel prize winners had an academic advisor who graduated from the same university?" and the like.

The Web is increasingly used as a platform for applications on top of distributed *structured content* sources. Early examples are the APIs and content feeds in lightweight formats as RSS or JSON, that are combined together in what has been termed "Web 2.0 applications". The combination of information from different feeds is facilitated with a number of mash-up platforms like Yahoo! Pipes (Fagan 2007), Mozilla Ubiquity (Raskin 2008), SAP Rooftop (Hoyer et al. 2009), DERI Pipes (Le Phuoc et al. 2009). The actual knowledge consolidation is however usually not well supported and needs to be realized by the applications, possibly with the help of the mentioned techniques from the database field.

While these first approaches emphasis the effort of consolidating knowledge in the data consuming applications, *Semantic Web* approaches foresee the use of the Web a platform for consolidating knowledge outside the applications. In correspondence to the structure of the Web, knowledge is represented with graph data models like RDF and a number of models that build on RDF, standardized by the World Wide Web Consortium (W3C). One aspect, with which the Semantic Web goes beyond the mentioned Web 2.0 approaches with respect to knowledge consolidation, is that it also allows for distributed querying approaches of the Web as a whole. This has been demonstrated with research prototypes like SemaPlorer (Schenk et al. 2008) and SearchWebDB (Tran et al. 2008), that both support the distributed execution of SPARQL queries over multiple stores and endpoints.
More recently, best practices for distributed publishing and consolidating data on the Web, coined as *Linked Data,* have emerged and already demonstrated great success (Bizer et al. 2009). Linked Data aims to create a "Web of data" as a complement to the current "Web of documents", in accordance with the Semantic Web goal to achieve a global Web of machine-readable data. Just as entire Web pages are referenced and accessed via URI's with standard Web protocol HTTP, entities like e.g. a product are now identified with standard URI's. Machine-readable descriptions of them are then accessed via HTTP in formats like GoodRelations (Hepp 2008). The product data can then be referenced by arbitrary other entities, as e.g. an offer for selling the product. Linked Data follows the vision of leveraging Web scalability for data management by using the Web as a giant collaborative database, on top of which we can built applications. Early projects include DBpedia (Auer et al.

2008) that extracts structured information from WikiPedia and exposes it via a SQL-like interface. With respect to the task of consolidating distributed (i.e. independently managed data) data, Linked Data can be seen as a set of technologies and practices for realizing a dataspace (cf. above). An important form of consolidation that goes beyond the practices mentioned so far is the task of semi-automatically generating links for combining independent data sets, as described in (Volz et al., 2009).

9.4 Approaches in Enterprise Application Research

We conclude this section by presenting selected research projects SAP research participated in that dealt with consolidating distributed knowledge in enterprise environments. The Aletheia project is concerned with semantically harmonizing information on products along the product lifecycle, whereas in the OKKAM project, we dealt with the task of consolidation identifiers from distributed data sources in an enterprise.

9.4.1 Aletheia

Aletheia is a German publicly funded project (Aletheia 2010) that deals with semantically harmonizing information on products. Today, product-related information can be found in various data sources and formats across the product lifecycle. Exploiting this information requires the federation of these sources, the extraction of implicit information, and the efficient access to this comprehensive knowledge base. Existing solutions for product information management (PIM) are usually restricted to structured information, but most of the business-critical information resides in unstructured documents. Within Aletheia, a generic architecture and a prototype was developed for federating heterogeneous information from various sources (Wauer et al. 2009). The architecture provides a framework for applying approaches from the disciplines of Information Extraction, databases and the Web in a centralized system.
The general harmonization strategy in Aletheia with respect to the different schemas (e.g. different integrated databases, different XML formats of integrated application interfaces) builds on the approach described in (Angele and Gesmann 2006). Aletheia first overcomes the syntactic heterogeneity of the distributed sources by lifting different media types to a common ontology language. The resulting ontologies are called *technical ontologies*, as they are usually automatically generated. Semantic heterogeneity is then revolved with mappings to a central human-engineered *domain ontology*. The depicted screenshot in Fig. 9-1 shows such a mapping for the example of data from the automobile domain, using the Ontostudio ontology engineering IDE.

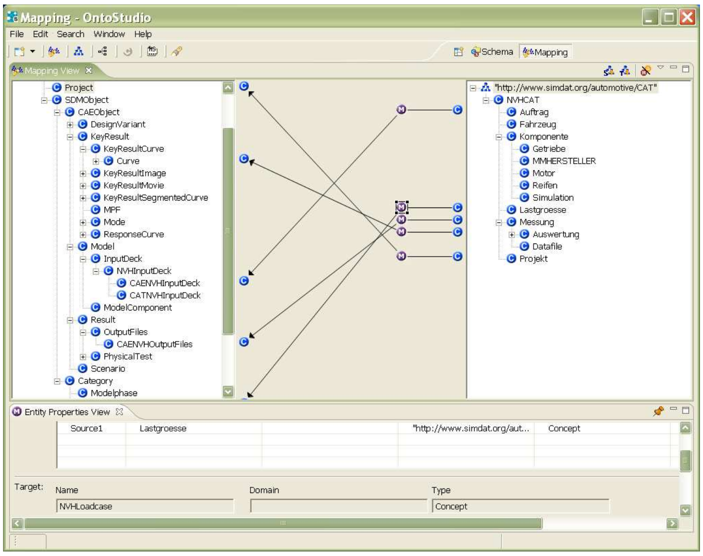


Fig. 9-1. A mapping example in Aletheia

9.4.2 OKKAM

OKKAM is an FP7 EU project (OKKAM, 2010) that deals with Knowledge Management – one of the most important tasks in any organization. Essentially all applications access data or produce data in a form or another. In recent years, the unstructured information sources such as documentation, user generated content and even the World Wide Web have started to play an increasing role in the development of the new business applications. In particular, business domains such as *Business Intelligence* and *Product Information Management* are only some of the areas that require support for the management and understanding of unstructured data.

In this context, the major challenge is to develop methods and techniques able to connect the corporate structured enterprise data with unstructured data. Since in general, the structured data has been designed to capture information about real world entities, the natural approach is to attempt to recognize and extract the occurrences of the very same entities in text. Nevertheless, this is not at all a trivial task: if the entities defined and used in a database or a business warehouse are homogenous and follow an original design, in the case text documents entities can occur in various forms, referred through the use of various names, abbreviations or identifiers.

In order to address this issue, the first step is to maintain a pool of entities of interest, unambiguously identified and associated with a minimum of information necessary to discriminate between them when no appropriate identifier is known.
Documents refer to the same entities as those defined in the corporate structured data but in a much fuzzier way; applications resort more and more to unstructured data in order to augment, complete or improve the functionality they offer. In order to reduce the costs associated with operations such as entity extraction, entity disambiguation, etc. and to enable the interoperation among the applications concerned with the same enterprise entities, a consistent, centralized view on these entities has to be maintained. Such a view has to be flexible enough in order to fully match various application contexts without need for change in the business logics of these applications. That is, this view should include only essential information sufficient for the identification and recognition of these entities.

By using the OKKAM ENS, such a view on the enterprise entities can be created and maintained. Each entity is associated with a unique identifier and a profile, containing only as much information as needed to recognize the entity when the identifier is not known. In fact, at a closer analysis, one can recognize that this profile represents nothing else than the metadata of that particular entity.
A very important aspect of OKKAM in this context is represented by its public deployment. One of the major pain-points in managing the entire ecosystem around the company's products and solutions, is making sure that business partners and customers have access to (and use) up-to-date information about the products. By using a public OKKAM node, the company has one more instrument to proactively disseminate correct information about its solutions and use it as support in solving enterprise information integration issues.

A more detailed introduction can be found in the (OKKAM, 2009) deliverable. Most of the discussions of the application scenarios, as well as the adopted solutions that are being illustrated in the deliverable (OKKAM, 2009) have been previously published to several conferences and workshops in the area of unstructured information management – see, e.g., (Brauer et al., 2010).
The deliverable (OKKAM, 2009) analyzes four main application scenarios where entity recognition plays a major role and where such a central entity management and identification system can bring real added value, particularly

- *Self-support for Customers,*
- *Analytics of Unstructured Community Data,*
- *Supplier Relationship Management* and
- *Text2Query*.

The first two are the use case scenarios prescribed by the OKKAM description of work, while the other two are scenarios have been identified in work results. All of them have proven the relevance of an entity-centric organizational knowledge management.
Additionally, some of the main technical aspects needed to enable such an entity-centric approach, e.g. entity extractors (i.e. okamaziers), public and private entity name systems (i.e. public/private OKKAM nodes), are described. The main concepts described in (OKKAM, 2009) to enable entity-centric organizational knowledge management are

- Entity extraction and matching,
- Data models for text analytics,
- Public/Private OKKAM nodes with the problems of
    - Search,
    - Profiles, and
    - Synchronization.

Results from the OKKAM project show that the so-called *enterprise entities* are key enablers for information integration, data aggregation or analytical tasks. It can be learned from the project results that the continuously growing importance of the unstructured information sources however, has significantly changed the way the enterprise entities need to be managed and handled.

9.5 Summary

This section discussed data and information mining from distributed sources. There exist many approaches evolving from database domain, web, and heterogeneous sources. New approach in database domain is called *Dataspace* that increases semantic cohesion over time by assuming that different parties provide mappings Thus providing an unified architecture for reference reconciliation, schema matching and mapping, data lineage, data quality and information extraction. In web domain semantic web allows software agents / crawler to query information from distributed sources examples of this approach are SemaPlorer and SearchWebDB. Coming from web 2.0, many mesh up platforms such as yahoo pipe are used to aggregate information and services. Aletheia, a German national project for harmonizing product information uses semantic abstraction and human interference for resolving conflict. OKKAM, an FP7 EU ongoing project, tries to connect the corporate structured enterprise data with unstructured data such as product documentations that are written from humans.

This section have presented state-of-the-art of information mining in centralized and distributed environment that can be adopted and extended to fuse information from heterogeneous sources as foreseen in ebbits use cases. These approaches can be used in level 2-3 JDL multi-sensor fusion model.

9.6 Bibliography

Aletheia (2010): Aletheia project Homepage (accessed on 2010-11-03). http://www.aletheia-projekt.de/

Angele, J., Gesmann, M. (2006): Data Integration using Semantic Technology: A use case, in Second International Conference on Rules and Rule Markup Languages for the Semantic Web, Athens, GA, ISBN 0-7695-2652-7, p. 58-66

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z (2008): DBpedia: A Nucleus for a Web of Open Data, in The Semantic Web, Springer, ISBN978-3-540-76297-3

Bizer, C., Heath, T. Berners-Lee, T. (2009): Linked Data - The Story So Far. In: International Journal on Semantic Web & Information Systems, Vol. 5, Issue 3, p. 1-22

Bleiholder, J., Naumann, F. (2008): Data fusion, in ACM Computing Surveys (CSUR), Volume 41 Issue 1

Brauer, F., Huber, M., Hackenbroich, G., Leser, U., Naumann, F., & Barczynski, W. (2010). Graph Based Concept Identification and Disambiguation for Enterprise Search. WWW 2010

Broder, A. Z. (2000): Identifying and Filtering Near-Duplicate Documents, OM '00, Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching, Springer-Verlag, 2000, 1-10

Charikar, M. S. (2002): Similarity estimation techniques from rounding algorithms STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, ACM, 2002, 380-388

Chowdhury, A.; Frieder, O.; Grossman, D. & McCabe, M. C. (2002): Collection statistics for fast duplicate document detection ACM Trans. Inf. Syst., ACM, 2002, 20, p. 171-191

Fagan, J.C. (2007): Mashing up Multiple Web Feeds Using Yahoo! Pipes, Computers in Libraries, v27, n10, Information Today, Medford, NJ p. 10-17

Franklin, M., Halevy, A., Maier, D. (2005): From Databases to Dataspaces: A new Abstraction for Information Management, Sigmod Record, ACM, 34(4):27, p. 33

Gionis, A.; Indyk, P. & Motwani, R. (1999): Similarity Search in High Dimensions via Hashing VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., 1999, p. 518-529

Halevy, A., Franklin, M., Maier, D. (2006): Principles of Dataspace Systems, in Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ISBN:1-59593-318-2

Hepp, H. (2008): GoodRelations: An Ontology for Describing Products and Services Offers on the Web, in Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008), Acitrezza, Italy, Springer, p. 332-347

Hoyer, V., Gilles, F., Fleischmann, K., Dreiling, A., Stanoevska-Slabeva, K. (2009): SAP Research RoofTop Marketplace, in Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web, Madrid, Spain

Kasneci, G., Ramanath, M., Suchanek F., Weikum, G. (2008): The YAGO-NAGA approach to knowledge discovery. SIGMOD Rec., 37(4):41/47

Le Phuoc, D., Polleres, A., Morbidoni, C., Hauswirth, M., Tummarello, G. (2009): Rapid Semantic Web Mashup Development Through Semantic Web Pipes, Proceedings of the 18th World Wide Web Conference (WWW2009)

Manber, U. (1994): Finding similar files in a large file system WTEC'94: Proceedings of the USENIX Winter 1994 Technical Conference on USENIX Winter 1994 Technical Conference, USENIX Association, 1994, 2-2

Mohania, M., Bhide, M. (2008): New trends in information integration, in ICUIMC '08: Proceedings of the 2nd international conference on Ubiquitous information management and communication, ACM, New York, NY, USA, p. 74-81

OKKAM (2009): OKKAM Deliverable D9.2. Lessons learned for Organizational Knowledge Management

OKKAM (2010): OKKAM project Homepage (accessed on 2010-11-03). http://www.okkam.org/

Raskin, A. (2008): Introducing ubiquity, Mozilla Labs Blog, 2008, http://labs.mozilla. com/2008/08/introducing-ubiquity.

Roy, P., Mohania, M., Bamba, B., Raman, S. (2005): Towards automatic association of relevant unstructured content with structured query results, in CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management, ACM, New York, NY, USA, p. 405-412

Sarawagi, S. (2007): Information Extraction, in Foundations and Trends in Databases, Vol. 1, No. 3 (2007), p. 261–377

Schenk, S., Saathoff, C., Baumesberger, A., Jochum, F., Kleinen, A., Staab, S., Scherp, A. (2008): SemaPlorer - Interactive Semantic Exploration of Data and Media based on a Federated

Cloud Infrastructure, in Web Semantics: Science, Services and Agents on the World Wide Web, Volume 7, Issue 4, 298-304, Semantic Web challenge 2008

Tran, T., Wang, H., Haase, P. (2008): SearchWebDB: Data Web Search on a PayAsYouGo Integration Infrastructure, Technical Report, Universität Karlsruhe (TH)

Venkatesan, T., Chakaravarthy, Gupta, H. Roy, P., Mohania, M. (2006): Efficiently linking text documents with relevant structured information, in VLDB '06: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, p. 667-678

Volz, J., Bizer, C. Gaedke, M., Kobilarov, G. (2009): Discovering and Maintaining Links on the Web of Data. International Semantic Web Conference (ISWC2009), Westfields, USA

Wauer, M., Schuster, D., Meinecke, J., Janke, T. Schill, A. (2009): Aletheia - Towards a Distributed Architecture for Semantic Federation of Comprehensive Product Information; IADIS International Conference WWW/Internet

Xi, W.; Fox, E. A.; Fan, W.; Zhang, B.; Chen, Z.; Yan, J. & Zhuang, D. SimFusion (2005): measuring similarity using unified relationship matrix SIGIR '05: Proceedings of the 28th annual international ACM SIGIR con

10. Technologies for Semantic Management of Data and Services

This chapter discusses current concepts and tools that enable the modeling and management of contextualized data. Furthermore, it presents approaches for semantically resolve inconsistent or incomplete data coming from different sources.

10.1 Modeling of Inconsistent and Incomplete data

Knowledge models are used to model various complex domains. They include data and relations between them. When models are created, data are acquired from sources in the real world. This is done by converting (mining) knowledge from electronic documents (these can be created by both human or machines) into a Knowledge representation. However, as these documents are very often incomplete or erroneous as for the knowledge they contain, we have to find a way to deal with incompleteness and inconsistencies in knowledge models. Paradigm of mathematic logic does not provide satisfactory tools for formulation of a solution of many of problems of knowledge representation and reasoning. Mathematical logic's domain is idealized presentation of mathematical results (Barwise and Etchemendy 1998). One of these is the reasoning with inconsistent knowledge. New non-monotonous logics were (and still are) developed and researched.

There are two main approaches how to deal with, inconsistency in a knowledge representation. First approach deals incompleteness as an erroneous state and tries to find solutions to eliminate that state. Second approach deals with the inconsistency as an extra knowledge property and tries to find ways, how to work with knowledge untouched by modified reasoning approaches (Huang et al 2005). Incompleteness of data can be treated by *Open World Assumption* or *Closed World Assumption*. By adding a new knowledge we can keep the validity of the existing knowledge in the model (monotonous logic) or not (non-monotonous logic).

Several methodologies for eliminating inconsistencies in ontologies were proposed, including *Minimal Unsatisfiability Preserving Sub-ontologies* (MUPSs) (Kalyanpur, et al. 2006) and *Maximally Concept-Satisfiable Sub-ontologies* (MCSSs) (Meyer, et al. 2006).

Database and knowledge bases may be inconsistent for various reasons. For example, during the construction of an expert system, we may consult many different experts (Grant and Subrahmanian 1995). Each expert may provide us with a group of rules and facts which are self-consistent. Some inconsistency may arise if one expert is correct, while another made a mistake, or the two experts have a genuine disagreement between them. In either case, we are forced to reason in the presence of an inconsistency. In both cases, we need to be sure that the existence of the inconsistency does not affect our reasoning. In particular, the fact that an expert made a mistake may be uncovered only after the expert system has been in use for some time. During this time, the answers provided by the expert system should not be affected by the inconsistency.

According to (Gomez-Perez 1999) the evaluation of an ontology includes the inspection of its taxonomy, which should be checked for

- inconsistency,
- incompleteness, and
- redundancy.

Inconsistency. It tests check, if a contradictory definition of an individual can be found or if contradictory knowledge can be derived from other definitions and axioms given in the ontology. We distinguish between circularity errors, partition errors, and semantic errors. Circularity is identified if a class defined in an ontology is a specialization or generalization of itself. Partition Errors in taxonomy subconcepts of a class can be defined as a disjoint partition of the generalizing class. A class partition error occurs, if a class is defined as a common subclass of several classes of a disjoint partition. Analogously, a common instance of two disjoint classes resembles an instance partition error. Semantic inconsistencies occur if the developer of the ontology asserts incorrect semantic classifications; e.g., if a class is falsely defined as a subconcept of another class, such that there is no semantic relation between the concepts. Semantic inconsistencies often are difficult to find using automated methods, but they may be detected by a manual inspection of the ontology.

Incompleteness. Ontological knowledge can be incomplete, if is-a relationships between concepts are imprecisely defined or possible information about decompositions is missing. Typically, incompleteness occurs if (probably) important concepts are omitted during the definition of the taxonomy. Furthermore, partitions are incompletely defined, if knowledge about the disjointness or exhaustiveness of a partition is omitted. Unfortunately, it is very difficult to provide automated methods for detecting such errors.

Redundancy. For a given ontology we can detect redundant class/instance definitions or redundant subclass relations (is-a) in the taxonomy.

People may find it difficult to understand the logical meaning of the underlying description logic, and potential inferences in ontologies; hence people may fail to formulate axioms which are logically correct, or may specify contradictory statements, thus leading to inconsistencies as errors (Lam et al. 2006). Other typical scenario of inconsistency is inconsistencies as knowledge overwrite. Inconsistency is resolved by rewrite one of the axioms. Therefore, it is necessary to give the user guidance on how to rewrite axioms in order to resolve the contradiction.

Ontologies play a crucial role in the Semantic Web. However, this makes SW technology highly dependent on the quality and correctness of these ontologies (Schoblau and Huang 2005). State-of-the art Description Logic reasoners can efficiently detect inconsistencies even in very large ontologies. A major shortcoming of ontologies is their inability to represent and reason with uncertain, incomplete data (Costa et al. 2006). Almost all current ontology formalisms are based on classical logic, and thus provide no support for plausible reasoning. There is a current line of research focused on extending OWL so it can represent probabilistic information (e.g. Ding & Peng 2004, Gu et. al. 2004). The approach involves augmenting OWL semantics to allow probabilistic information to be represented via additional markups. Another option for representing uncertainty in OWL is to focus on OWL-DL, a decidable subset of OWL that is based on Description Logics (Baader et. al. 2003). Probabilistic extensions have been developed for description logics (e.g. Koller et al., 1997, Giugno & Lukasiewicz 2002) Description logics are highly effective and efficient for the classification and subsumption problems they were designed to address but are restrictive in their limited ability to represent constraints on the instances that can participate in a relationship.

PR-OWL is a probabilistic extension to the OWL Web ontology language. The OWL specification is a Recommendation issued by the Worldwide Web Consortium (W3C). PR-OWL provides constructs for writing ontologies containing probabilistic information that could be processed by diverse tools (e.g. Netica, Hugin, Quiddity*Suite, JavaBayes, etc.) employing different representation formalisms (e.g. PRMs, BNs, OOBNs, etc). That level of flexibility can only be achieved using the underlying semantics of first-order Bayesian logic, which is not a part of the standard OWL semantics and abstract syntax.

Another proposal of Non-Deterministic Distance Semantics for Handling Incomplete and Inconsistent Data was proposed in (Arieli and Zamansky 2009). Unlike 'standard' semantics, in which conclusions are drawn according to the models of the premises, reasoning in distance-based semantics is based on the valuations that are 'as close as possible' to the premises, according to a pre-defined metric. As this set of valuations is never empty, reasoning with inconsistent set of premises is not trivialized.

In (Hitzler 2009) author proposes possible solution for unsoundness and incompleteness of classical reasoning algorithms for logic-based knowledge representation languages. Approximate reasoning for the Semantic (Densel and Harmelen 2007) is the solution. Approximate reasoning shall not replace sound and complete reasoning in the sense that the latter would no longer be needed.

10.2 Context Modeling

According to (Strang and Linnhoff-Popien, 2004), to model the context, several most typical approaches can be used. These can be classified by scheme of data structures which are used to exchange contextual information.

- Key Value Models

- Markup Scheme Models
- Graphical Models
- Object-Oriented Models
- Logic-Based Models
- Ontology Based Models

Most simple data structure for modelling the context information is Key-Value Model. In (Samulowitz, Michahelles and Linnhoff-Popien, 2001), (Schilit, Adams and Want, 2005) used key-value pairs to model context by providing the value of context information (location). This approach is used in distributed service frameworks, where services itself are described with a list of simple attributes. Approach is easy to manage, but simple.

Markup Scheme models are hierarchical data structures consisting of markup tags with at- tributes and content. In particular, the content of the markup tags is usually recursively defined by other markup tags. Typical approaches are profiles based upon Standard Generic Markup Language (SGML), or Profile Description Language (PPDL). This XML based language allows inclusion of contextual information in limited scale.

Most typical Graphical Model is the Unified Modeling Language (UML) which provides a strong graphical component in form of UML diagrams. UML is also appropriate to model the context. This is shown in (Bauer, 2003), where contextual aspects relevant to air traffic management are modeled as UML extensions. Another example is context extension of Object-Role Modeling (ORM) (Halpin, 2001), where involves identifying appropriate fact types and the roles that entity types play in these.

Object-Oriented Models are based on OO approach. Main focus is to use the concepts of encapsulation and reusability to cover parts of the problems arising from the dynamics of the context in ubiquitous environments. Cues (TEA Project) represents such approach (Schmidt and Larehoven, 2001). A cue is regarded as a function taking the value of a single physical or logical sensor up to a certain time as input and providing a symbolic or sub-symbolic output. A finite or infinite set of possible values is defined for each cue. The output of each cue depends on a single sensor, but different cues may be based on the same sensors. The context is mo- deled as an abstraction level on top of the available cues. Thus the cues are objects providing contextual information through their interfaces, hiding the details of determining the output values.

We will briefly describe common Ontology Based Models. The use of ontologies brings forward several benefits and additional functionalities. One approach comes from CoOL – Context Ontology Language (Strang, Linnhoff-Popien, Frank, 2003). Entities are characterized by ContextInformation instances which in turn are defined and interlinked by use of the aspect-scale-context (ASC) model. ASC provides an umbrella vocabulary to transfer arbitrary context models and is therefore a strong approach with respect to the comparability criterion. In fact, relating different scales for the same context aspects and deriving and aggregating new scales from existing ones is one of the motivations for this ontology. CONON defines core classes to model Person, Location, Activity and Computational Entities (Gu et al., 2004). to develop a context model based on ontologies because of its knowledge sharing, logic inferencing and knowledge reuse capabilities. Wang et al. created an upper ontology which captures general features of basic contextual entities and a collection of domain specific ontologies and their features in each subdomain. The CONON ontologies (Wang, Zhang, Gu, Pung, 2004) are serialized in OWL-DL. This enables consistency checking and contextual reasoning using inference engines developed for description languages. CoDAMoS (Preuveneers et. Al, 2004) points in the same direction by defining an ontology around four concepts used to model the profiles of Users, the Environment, Platforms and Services. Both approaches focus on the modeling of profiles for human users and applications, and might be limited with respect to future context-awareness tasks in service-service interaction models.

Hydra (Hydra 2010) presents a hybrid-approach to context-awareness, providing the capability for modeling, reasoning and interpretation of contextual data with both a lower-level key-value approach, and through higher-level ontologies, that permit more powerful, albeit more computationally expensive, reasoning capabilities. Hydra basically categorize context into application, device, and semantic. Device context represents a data source for acquiring data through pulls or events mechanism. Semantic Contexts represent location, environment, and entity. The Application Context acts as a means of grouping together certain contextual information for the purposes of a higher level application, so that it can be easily queried and retrieved.

10.3 Summary

Data populated from distributed sources, documents and any other sources might present erroneous knowledge. There are 2 ways to deals with this kind of data, first to purge erroneous state and secondly by finding ways to work with it by modifying the reasoning approach. Erroneous might be caused by inconsistency, incompleteness, and redundancy. Another drawback of ontology is the inability to represent and reason upon uncertainty. There exist few works trying to present uncertainty e.g.: OWL-DL and PR-OWL (Probabilistic information in OWL).

Modeling context can be done by Key Value Models, Markup Scheme Models, Graphical Models Object-Oriented Models, Logic-Based Models, and Ontology Based Models. Hydra follows a hybrid approach that models low level data through key-value and high level context through ontology model. Hydra introduced three types of context: data, semantic, and application.

Improvement of hydra context modeling that will be done in ebbits is to incorporate the ability to represent and reason uncertain knowledge using probabilistic functions.

10.4 Bibliography

Arieli, O., A. Zamansky (2009). Non-deterministic distance semantics for handling incomplete and inconsistent data. Proc. 10th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09), LNAI 5590, pages 793-804, Springer

Barwise, J., Etchemendy, J. (1998). Computers, visualization, and the nature of reasoning. In: The Digital Phoenix: How Computers are Changing Philososphy (T.W. Bynum, J.H.Moor, eds.), pp. 93–116

Bauer, J. (2003). Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic

Baumeister, J., D. Seipel (2005). Smelly Owls - Design Anomalies in Ontologies. In: Proc. FLAIRS 2005, pp. 215-220

Costa, P. C. G., K. B. Laskey, G. Alghamdi (2006) Bayesian Ontologies in AI Systems. Proceedings of the Fourth Bayesian Modelling Applications Workshop, held at the Twenty Second Conference on Uncertainty in Artificial Intelligence Cambridge, MA, USA

Ding, Z., Y. Peng. (2004). A probabilistic extension to ontology language OWL. in 37th Annual Hawaii International Conference on System Sciences (HICSS'04). Big Island, Hawaii

Fensel, D., F. van Harmelen (2007). Unifying reasoning and search to web scale. IEEE Internet Computing, 11(2):96, 94-95.

Giugno, R., T. Lukasiewicz (2002). P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the Semantic Web. in European Conference on Logics in Artificial Intelligence (JELIA), Cosenza, Italy: Springer

Gomez-Perez, A. (1999). Evaluation of Taxonomic Knowledge on Ontologies and Knowledge- Based Systems. In Proceedings of the 12th International Workshop on Knowledge Acquisition, Modeling and Management

Grant, J., V.S. Subrahmanian (1995). Reasoning in inconsistent knowledge bases, Knowledge and Data Engineering, IEEE Transactions on , vol.7, no.1, pp.177-189

Gu, T., P.H. Keng, Z.D. Qing (2004). A Bayesian approach for dealing with uncertainty contexts. In Second International Conference on Pervasive Computing. Vienna, Austria: Austrian Computer Society

Gu, T., X. H. Wang, H. K. Pung, D. Q. Zhang (2004). An Ontology-based Context Model in Intelligent Environments, In Proc. of Communication Networks and Distributed Systems Modeling and Simulation

Halpin, T. A. (2001). Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufman Publishers

Hitzler, P. (2009). Towards Reasoning Pragmatics. In Proceedings of the 3rd International Conference on GeoSpatial Semantics (GeoS '09), Krzysztof Janowicz, Martin Raubal, and Sergei Levashkin (Eds.). Springer-Verlag, Berlin, Heidelberg, 9-25

Huang, Z., van Harmelen, F., ten Teije, A. (2005). Reasoning with inconsistent ontologies. In Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'05

Hydra Consortium (2010), D3.16 Final Context Awareness Report, , in *Hydra Project Deliverable*. IST 2005-034891

Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B. (2006). Repairing Unsatisfiable Concepts in OWL Ontologies. In Proc. of the Third European Semantic Web Conference.

Koller, D., A. Y. Levy, A. Pfeffer (1997). PCLASSIC: A Tractable Probabilistic Description Logic. Paper presented at the Fourteenth National Conference on Artificial Intelligence (AAAI-97) 27-31. Providence, RI, USA

Lam, J., Pan, J., Sleeman, D., Vasconcelos, W. (2006).Ontology inconsistency handling: ranking and rewriting axioms, Technical Report:AUCSTR0603

Meyer, T., K. Lee, R. Booth, J. Z. Pan (2006). Finding maximally satisfiable terminologies for the description logic ALC. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06), July 2006

Preuveneers D. (2004). Towards an Extensible Context Ontology for Ambient Intelligence. In 2nd European Symposium on Ambient Intelligence

Samulowitz, M., F. Michahelles, C. Linnhoff-Popien (2001). An architecture for context-aware selection and execution of services. In New developments in distributed applications and interoperable systems, Kluwer Academic Publishers.

Schilit, B. N., N. L. Adams, R. Want (2005). Context-aware computing applications. In IEEE Workshop on Mobile Computing Systems

Schlobach, S., Z. Huang (2005). Inconsistent ontology diagnosis: Framework and prototype. Project Report D3.6.1, SEKT, 2005.

Schmidt, A., K. V. Laerhoven (2001). How to Build Smart Appliances. IEEE Personal Communications

Strang, T., C. Linnhoff-Popien (2004). A Context Modeling Survey, In Proc. of Workshop on Advanced Context Modelling, Reasoning and Management

Strang, T., C. Linnhoff-Popien, K. Frank (2003). Applications of a Context Ontology Language. In Proceedings of International Conference on Software, Telecommunications and Computer Networks

Wang, X. H., D. Q. Zhang, T. Gu, H. K. Pung (2004). Ontology Based Context Modeling and Reasoning using OWL. In Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications

11. Service-Oriented Architectures for Distributed Intelligent Service Structures

This chapter reports on the current state of Service-Oriented Architectures (SOA) and their use for distributed intelligent service structures as the basis for the ebbits platform. It further discusses the potential contemporary use of a Resource-oriented Architecture (ROA) implementation and RSS feeds.

As the ebbits research and development will be performed in a number of work areas and influence its platform SOA, it is important to clarify what technological and enterprise research challenges there are: SOA for a maximum interoperability between heterogeneous structures and with end to end characteristics, a distributed discovery architecture and unique physical identification of loosely coupled objects, scalable network technologies integrating wired and wireless technologies using structured P2P networking layers in a transparent and seamless way, software and services with goal oriented orchestration and support for semantic interoperability, context awareness, and distributed decision support including workflow management and business rules processing. The challenges also concern security and privacy technologies enabled for cloud computing with models for decentralized identification, authentication and trust and enterprise framework and business socio-economic performance in dynamic business constellations and with sustainable business models.

11.1 Principles of SOA

The World Wide Web Consortium (W3C) defines SOA as "A set of components which can be invoked, and whose interface descriptions can be published and discovered". No universally agreed definition is available, but the term is generally considered to imply that application functionality is provided and consumed as sets of services which can be published, discovered and accessed and are loosely coupled as well as implementation and technology neutral. SOA encourages loose coupling among the interacting software systems. A service is used only via the published service description and the service consumer does not address a specific implementation or deployed instance of the service. Changes to the implementation do not affect the service consumer and the service consumer can change the instance of the service that is used (changing location or implementation of the service, e.g. when two service providers offer the same service) without modifying the client application. By abstracting the service from the implementation, the developer will not need to consider which technique was used to implement the service. Parallel implementations of the service may be available, and the actual version used is transparent to the consumer. The use of standardized protocols for publishing, discovering and accessing services allows the service to be provided on any platform that can implement these protocols. In orchestrating a SOA solution, services that are (internally) implemented with different languages, architectural styles and on platforms from different vendors, can be used together transparently (W3C SOA).

In ebbits, the presumed platform is built as a SOA implementation that transforms every device into a service. The SOA will also be based on a semantic resolution of services and objects that allow services to semantically discover, configure, and communicate with each other in a ubiquitous way. The architecture also requires that providers and requestors of services can communicate business logic information freely with each other despite separate heterogeneous communication and information infrastructures. The ebbits architecture will by that provide an interoperable end-to-end IP communication between systems, subsystems, devices, sensors, users and data repositories in one all-inclusive platform. For this purpose, SOA is foreseen as the most suitable architecture approach.

The principles of SOA basically defines how to integrate widely separated applications for a web based solution and where the functionality of SOA-based distributed systems is to split itself into services instead of objects or tightly coupled components. In SOA, these services are autonomous and independent of each other as software building parts. Also, in order for these services to locally work they need to be defined for service invocation, description, discovery, coordination and composition (Oberle 2006). Therefore, a SOA implementation in ebbits has to be a collection of services that communicate with each other with the aim to have interoperable and loosely coupled services distributed in the ebbits network and as proposed, to be realized by the Hydra Middleware. In this context, a service will be a function that is well-defined, self-contained, and does not depend on the context or state of other services.

It is strongly believed that SOA allows development to align business with information technology in a dual effective way and that it is a bridge between them creating a symbiotic and synergistic relationship. This is regarded as more powerful and valuable than anything else as SOA enhances the business results achieved from having better alignment between the business and IT. In that way, SOA introduces two high-level abstractions where enterprise business services represent existing IT capabilities (aligned with the business functions of the enterprise) and where business processes (orchestrating business services) define the overall functioning of the business.

The SOA approach in ebbits therefore allows for distribution of intelligence between the edge network and the centralized business/process information system and applications to be launched as bundles of service components. Each of these applications can then be assigned to a network node (e.g.: manufacturing cell level, shop floor level, operator level, point-of-sale, home environment), which is optimal for the overall execution of the application. The ebbits platform thus offers an open system as an alternative to centralized or centrally owned and administered infrastructure components and may in that way eliminate centralized gatekeeper and lock-in of critical business or process functionalities.

11.2 Distributed intelligent services

SOA is most commonly recognized as a kind of network architecture where services integrate widely separated applications for a web based solution. It is therefore relevant to give a detailed coverage in order to give an understanding of the different structures and network architecture types.

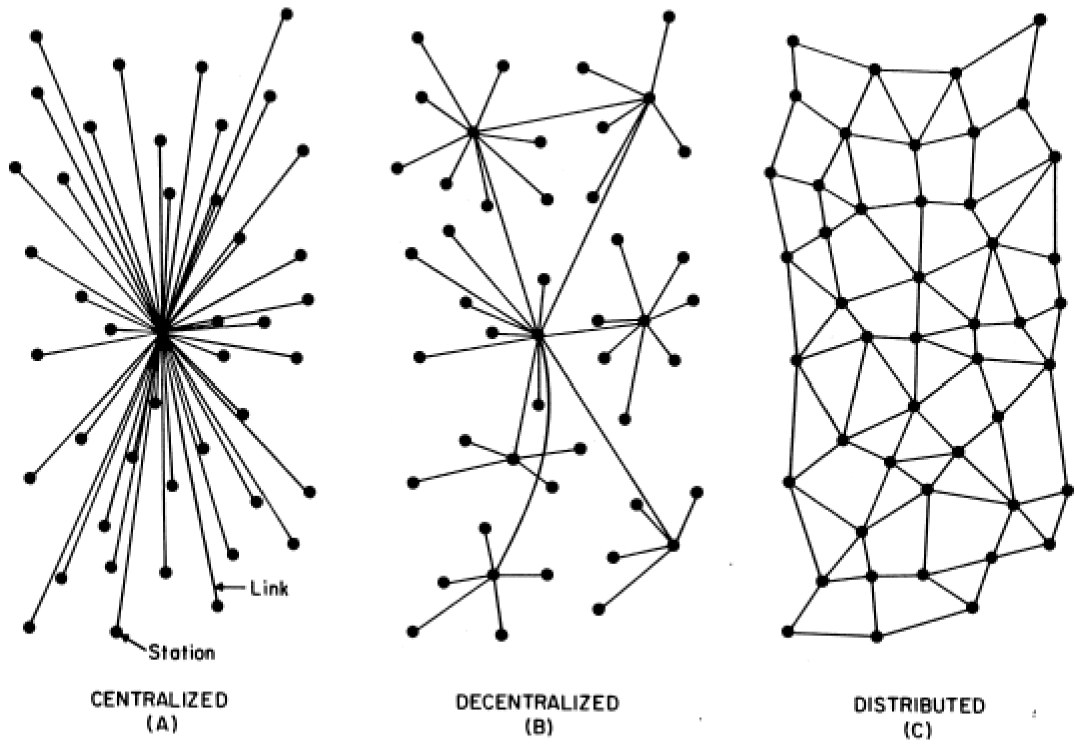


Fig. 11-1 Three different architecture types.

11.2.1 Centralized architecture

The centralized architecture (model A in Fig. 11-1) exploits itself and its set of sources of information in a single location in both a simultaneous and non-simultaneous way. The advantage of a centralized architecture type is that is presumed to provide results that are aggregated in an optimal way by taking into account the whole knowledge that is locally available instead of when it is distributed. Since the original and local information is fused without diversions of attributes, states etc., it is regarded as a minimal information loss and therefore better represented if not provided by any autonomous source. Simultaneously, there is a risk when a source in that set makes an error or a faulty value and whereby this source affects the whole data set. This risk creates a decreased quality in the whole set. Centralized architecture is preferred when data retrieved is supposed to be used at the scene of the set of source collection.

11.2.2 Decentralized architecture

Compared to the centralized architecture the decentralized architecture (model B in Fig. 11-1) instead offers a large modularity and flexibility as it, by its analogue name of being autonomous, processes independently of each of its sources of information. This is true all the way until it fuses representation on a higher semantic level and at a later stage in the data fusion process, i.e. clustering of relevant data. A decentralized architecture should be chosen when there are communication problems such as limited bandwidth, insecure communications or if the acquisition rate of sources changes a lot, and where an adoption of a decentralized architecture can avoid re-processing of all the sources. The drawback of a decentralized architecture compared to a centralized is that it has lower global quality and lower information content. Also, in a decentralized architecture, the databases or any retrieval from these are left completely separated and will only perform the data fusion when the user requires or needs the information.

11.2.3 Hybrid/distributed architecture

The hybrid architecture (model C in Fig. 11-1) is as shortly described earlier a combination of the two previous ones. The benefit of this architecture combines the ones of previous architectures and as advantages, the architecture features: high accuracy, systemic error robustness and the possible distribution of computational load. As possible disadvantages in the hybrid architecture there is a higher computational load because the measures sometimes might be processed two times but where this can be stripped down by proper architecture design.

11.2.4 Web Services

Web Services commonly provide a standard, simple and lightweight mechanism for exchanging both structured and typed information between distributed services in a network. By tradition, SOAs are based on a client and server architecture where a server application (hosted by an always-on end system) provides services to many other client applications but this may differ depending on how one chooses to implement the SOA. Web Services are further discussed per architecture model in chapter 9.

11.3 Examples of SOA projects in the EU

There are many projects and solutions made with SOA, in this context it is relevant to review three of them which all are made under the flag of European Union.

11.3.1 SOA4All

Service Oriented Architecture FOR All (www.soa4all.eu) is a large-scale integrating project funded by the 7th Framework Programme and aiming at realizing a world where billions of parties are exposing and consuming services via advanced Web technology. The main objective of the project is to provide a comprehensive framework that integrates complementary and evolutionary technical advances (i.e. SOA, context management, Web principles, Web 2.0 and semantic technologies) into a coherent and domain-independent service delivery platform (see Fig. 11-2).

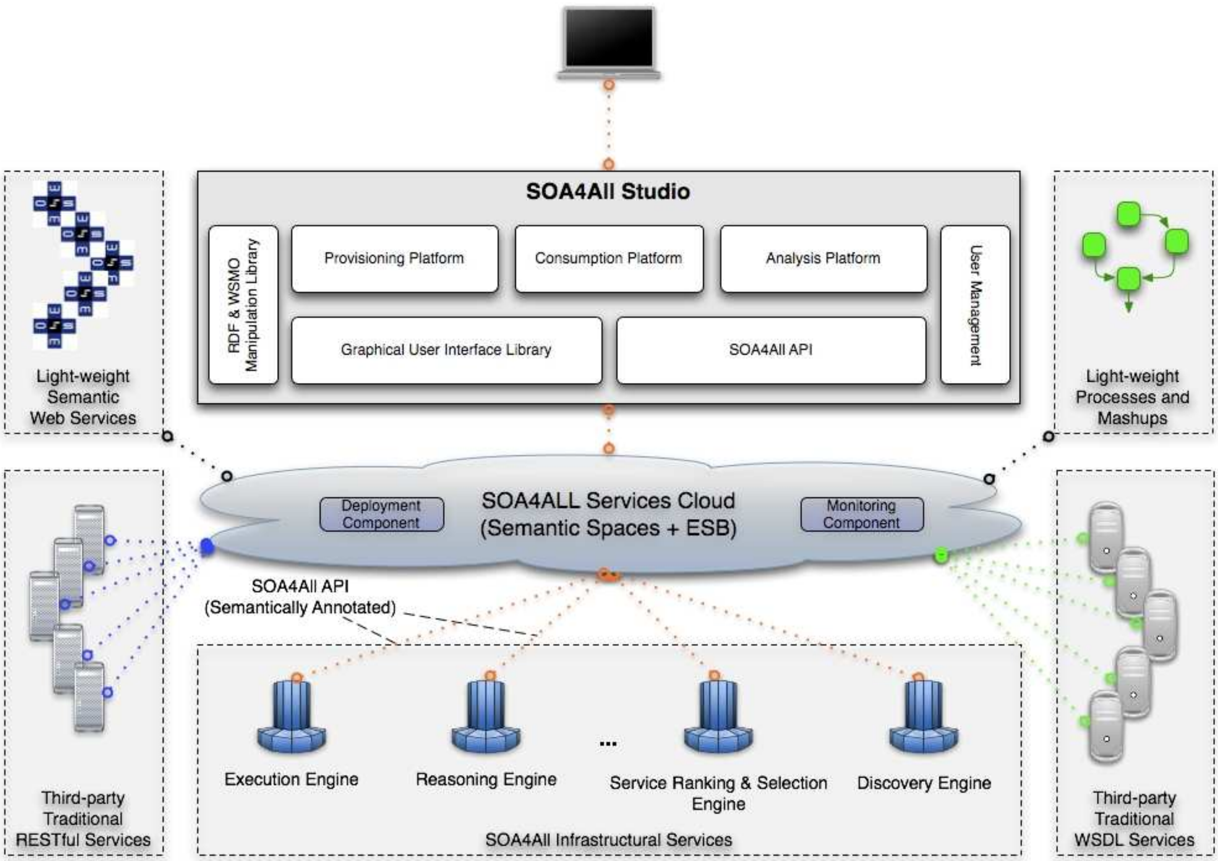


Fig. 11-2 High-level SOA4All architecture concept.

The project also aims at transforming service-oriented environments into architectures of billions of services and with this purpose it is said to integrate the principles which made the Web such a successful platform for the worldwide sharing of content. SOA4All seeks that everybody could be able participate either as a provider or consumer of information beyond the boundaries of enterprises, meaning that different roles will be made possible depending on the circumstances, and also means that the provisioning and modification of services must be under the distributed control of peers rather than being controlled by a central authority.

Another goal is that service operations such as discovery, selection, composition, mediation and invocation in a world of numerous services all need an advanced as self-managed as possible infrastructure. In order to achieve this, the project need to provide semantic descriptions for Web Services at many different levels where the creation of service locators based upon these descriptions and the creation of grounding mechanisms for invoking these services are processed.

SOA4All will further incorporate context in SOA as a means to customize service usage and provisioning on a global scale so that to enable service consumption customization from the user perspective and from the provider expectations. The mix of semantic and context technologies in the SOA4All infrastructure will be a key enabler of dynamic adaptation of services to their context of use and to facilitate the deployment of semantic services on a global scale, context need to cover a series of aspects ranging from system and location information to social settings and legal regulations.

Finally, the project has the intention to make use of Web 2.0 technology as a means to generate and access the semantic service layer and thereby properly include human interaction and cooperation that will enable users to provide solution to certain tasks such as service ranking or mediation (otherwise unfeasible) (SOA4All).

11.3.2 SOCRADES

Another project is SOCRADES (www.socrades.eu) that has the aim to create new methodologies, technologies and tools for the modeling, design, implementation and operation of networked systems made up of smart embedded devices. In order to achieve any enhanced system intelligence by co-operation of such heterogeneous smart, they need to interact seamlessly and intensively over a (wired or wireless) network.

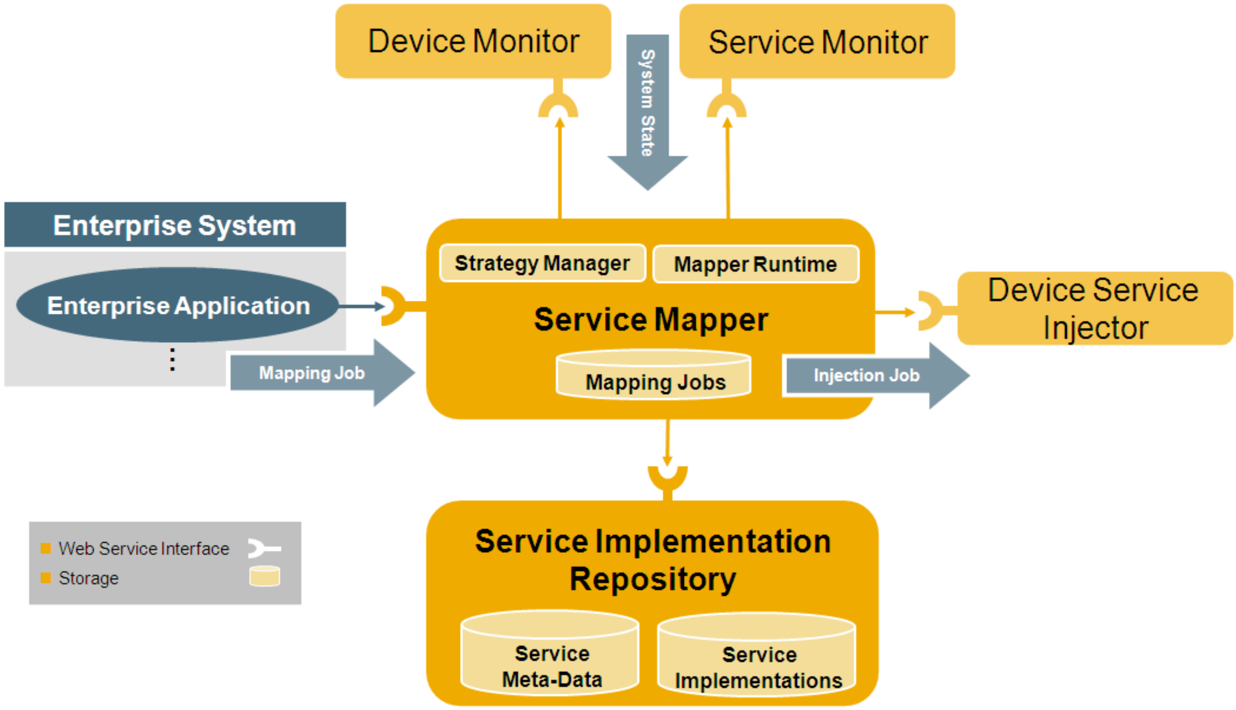


Fig. 11-3 shows deployment planning and execution within the middleware where a created business process gets represented by its hosted services in the middleware device layer. It generates service mappings that describes within the business process executable services.

The developed middleware technologies in SOCRADES are based on the Service-Oriented Architecture (SOA) paradigm, encompass both wired and wireless networking technologies and provide open interfaces enabling interoperability at the semantic level. A SOCRADES service is a software component encapsulating device-specific functionality which is advertised to the outside world, so as to be located and invoked by other networked devices and/or applications without the latter being aware of how the functionality is implemented (SOCRADES).

11.3.3 Hydra

Hydra (http://hydramiddleware.eu) is an EU FP6 Integrated Project that has developed a middleware for networked embedded systems which allows developers to rapidly create intelligent applications. Systems developers are provided with development tools for easily and securely integrating heterogeneous physical devices into interoperable distributed systems. Semantic self-configuration tools for devices, semantic resolution, knowledge discovery modules and basic security models. All of this will be integrated into the ebbits platform through the use of Hydra middleware. The Hydra project can also be seen to be concerned with ICT enabled business processes in devices which in turn can be categorized as actuators, sensors, or processors that are installed and operated in the context of an ICT application.

The work will partly build on refined results from the Hydra project by further extending the Hydra middleware and Semantic Model-driven Architecture (SeMDA) researched and applied within this project.

The Hydra middleware further allows developers to incorporate heterogeneous physical devices into their applications by offering easy-to-use web service interfaces for controlling any type of physical device irrespective of its network technology such as Bluetooth, RF, ZigBee, RFID, WiFi, etc. Hydra incorporates means for Device and Service Discovery, Semantic Model Driven Architecture, P2P communication, and Diagnostics. Hydra enabled devices and services can be secure and trustworthy through distributed security and social trust components of the middleware.

The choice of a service-oriented architecture in the Hydra project turned out to be a viable and successful approach for the platform as SOA applies both to the implementation of the middleware managers themselves, as well as for the higher-level device interfaces in terms of software proxies, i.e., devices are also web services in Hydra. The SOA implementation also works well across platforms as well as network boundaries. The system behind Hydra is implemented on two main IDEs, Eclipse and .NET and also provides P2P device interoperability across networks. As the Hydra middleware is based on SOA, to which the underlying communication layer is transparent, it also includes support for distributed as well as centralized architectures, security and trust, reflective properties and model-driven development of applications.

Further, the HYDRA middleware is deployable on both new and existing networks of distributed wireless and wired devices, which operate with limited resources in terms of computing power, energy and memory usage where the embedded and mobile Service-oriented Architecture provide interoperable access to data, information and knowledge across heterogeneous platforms, including web services, and support true ambient intelligence for ubiquitous networked devices. The second objective of the HYDRA project was to develop a Software Development Kit (SDK). The SDK can be used by developers to develop innovative Model-Driven systems such as the ebbits platform. Also using SOA in ebbits will enable the software architecture to start from the middleware Hydra which in turn will provide a service-oriented approach (e.g. by using the managers found in Fig. 11-4) to networks of heterogeneous embedded devices deployed in ebbits.

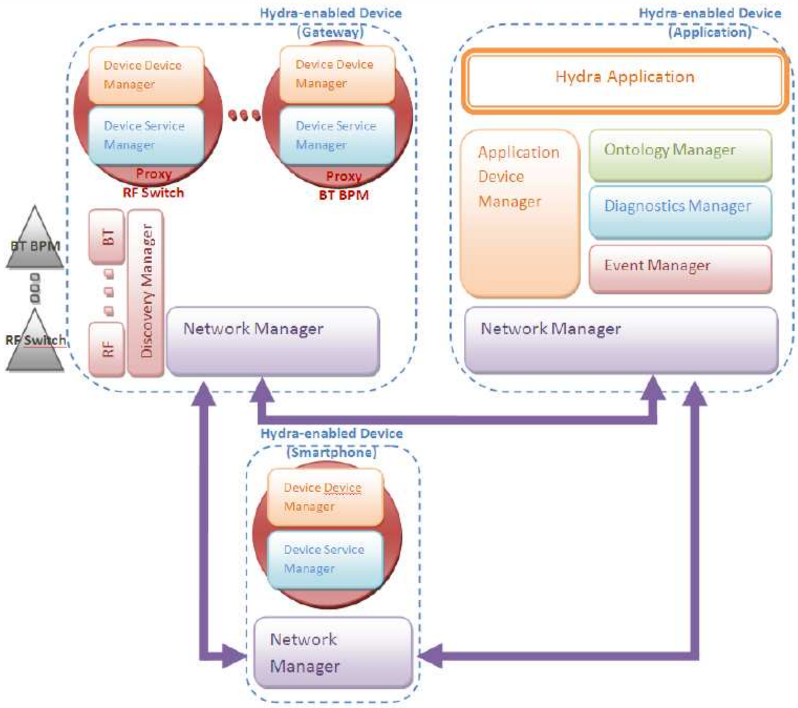


Fig. 11-4 Deployed Hydra SOA managers.

The SOA in Hydra represents an architectural style where the primary concept is the use of loosely coupled, implementation-neutral services supporting a business process as building blocks where service consumers use the service by means of its published interface-based service description without dependence on implementation, location or technology. The process of building, combining and sequencing services to provide more complex services is known as orchestration and as Hydra is based on a SOA solution that is built of a set of services orchestrated by clients or middleware to realize an end-to-end (business) process it gives openness to the architectural style that in turn also allows for ad-hoc service consumers and flexible and dynamically re-configurable processes (Hydra Middleware).

11.4 Resource-Oriented Architecture

The term Resource Oriented Architecture is generally considered as a specific set of guidelines for an implementation of REpresentational State Transfer (REST). An approach established in the doctoral thesis "Architectural Styles and the Design of Network-based Software Architectures" by Roy Fielding (Fielding 2000). REST describes a series of constraints that exemplify how the web utilized the design emerged HTTP but it is hard to review REST without correlating its implementation to the ROA architectural principals. To put it simply, ROA is an ideal approach for applications that are resource-based such as RSS readers where feeds create the basis of the point of interests.

REST is often associated with ROA while SOAP is often associated with SOA but they are not mutually exclusive, because it is possible to implement a SOA architecture through a RESTful implementation. Such an implementation follows architectural principles including:

- Services are stateless, i.e., requests from client to server must be self-contained including all the information necessary to interpret the request. No assumptions can be made as to the state or context on the server.
- Uniform services interface. Commonly based on the HTTP operations: GET, POST, PUT, and DELETE.
- REST-based architectures are built from resources) that are uniquely identified by URIs.
- REST components manipulate resources by exchanging representations of the resources. For example, a purchase order resource can be represented by an XML document, which might be updated by posting an XML document containing the changed purchase order to its URI.

In a REST-based architecture communication is primarily through the transfer of representations of resources, whereas in a SOA implementation with web services/SOAP, communication is based on the invocation of application/domain specific actions. While the SOA focuses on exposing many verbs (actions/services), the ROA instead focuses on exposing many nouns (resources), applications might require a combination of both and have actions as well as resources. Normally, the application predominance should dictate the chosen architecture but if a platform requires a blend of actions and resources it may be useful to consider assembling the two architectures into one (deeper described in chapter 11.4.1).

The W3C describes the Resource Oriented Model (ROM) which focuses on aspects of the architecture that relate to resources which in turn are fundamental concepts that supports much of the web and much of Web Services. As a Web Service is a special kind of resource that is important to the ROA, the ROM focuses on the key resource features that are relevant to the concept of resource and is independent of the role the resource has in the context of Web Services. Instead it focuses on issues such as the ownership of resources, associated policies etc. inherited as resource properties by the Web Services (W3C WS-ARCH).

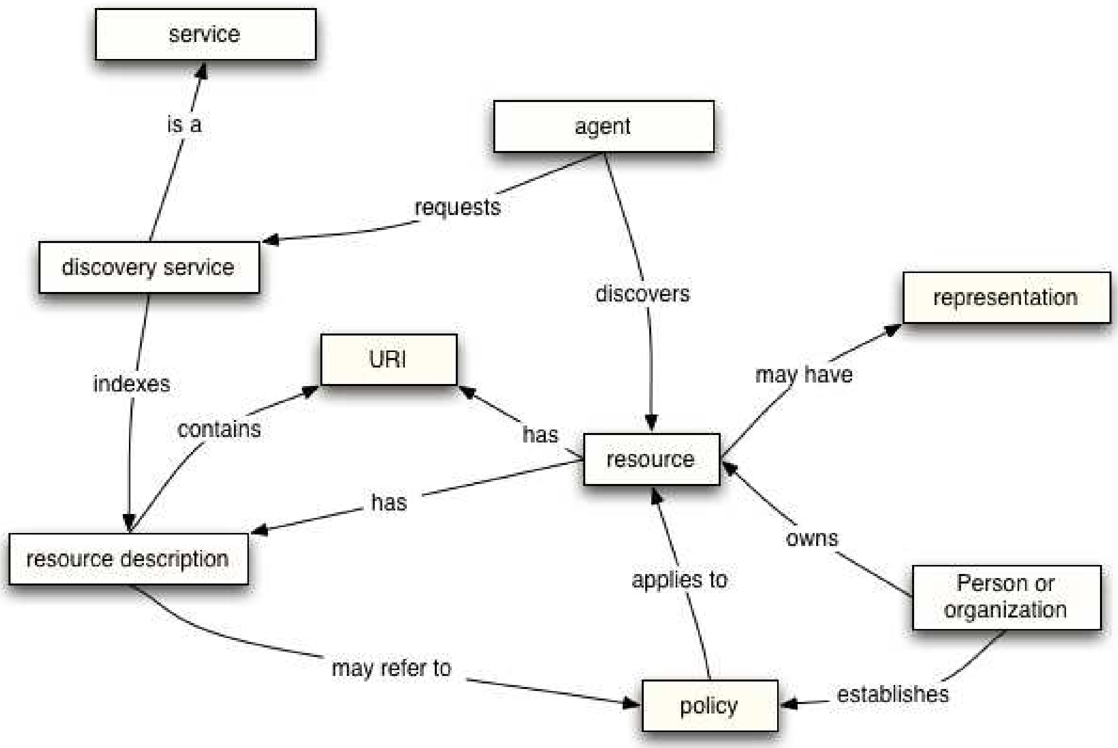


Fig. 11-5 Basic concepts and relationships in the Resource Oriented Model.

11.4.1 Hybrid SOA-ROA architecture

As reviewed so far, there are two architectural approaches commonly applied: SOA and ROA and when having a point of choice for a platform that is developed, the decision itself is all about which one of these are more appropriate for the functionality that is intended and whether it involves part of or the entire platform.

For example, SOA describes how to design a system by modeling business processes as services or actions that occurs within. Each service or action here is a distinct functional unit that does not interact with other services at first hand. This prevents calls between services in SOA and as the goal in SOA is to allow large pieces of functionality (data) to be rearranged on a higher level to form new applications (i.e. out of services), it also might reduce the functional granularity as seen at each action point. Therefore, SOA services are typically created to process business logic and while the two architectural patterns actually share the same common problem domains such as versioning and application manageability, the ROA services or RESTful services are used to bring resources to the user. Therefore, the hybrid architecture implementation becomes practical when using the two architectures together and use them for their intended but separated purposes. As with cloud computing and enterprise mash up applications[26] are becoming more common, the more applications are being based on the hybrid architecture of SOA and ROA and should for ebbits be taken into consideration.

When looking at it technically, a web service can be seen as a software system designed to support interoperable client-server interaction and vice versa over a network and traditionally the web service for SOA has been supplied by the Simple Object Access Protocol (SOAP). In the making of web 2.0, Representational State Transfer (REST) (i.e. the design paradigm melted in REST makes Resource Oriented Architecture (ROA)) which instead focuses on using the web as it is by following

[26] A mash up is a technique for building applications that combine data from multiple sources to create an integrated experience.

its basic principles, is becoming more and more popular as an alternative or a complement to SOA (MSDN Blogs 2009).

The main advantage of SOAP-based SOA over ROA is that SOA comes with more developed tool support and that it includes type-safe XML requests. Certainly, ROA developers can also use XML even though this might not be the first choice for a ROA developer as the ROA pertaining REST base its actualization by performance with better cache support, lightweight requests and responses and easier response parsing. So while SOA allows loosely coupled device and process interoperability, the ROA allows easier implementation, design agility, mobile clients and servers and reduced network traffic.

As an example and a hypothetical attempt to follow the ISO 10303-239 PLCS (Product Life Cycle Support)[27], where a hybrid SOA/ROA architecture in ebbits could implement the following from each approach:

- ROA – Atom Web Services (RSS feeds) that emits events and actions that are subscribed according some business process scheme both locally (at manufacturer site) and globally by any consumer interest, e.g. to follow how an ordered car is manufactured in detail.
- SOA – A SOAP tunneling approach (service invocation) that addresses the problems where actual WS communications require direct connection between the client and the server. Here, devices may be presented as UPnP devices, although UPnP discovery information is usually restricted to Local Area Networks, this solution proposes a way to replace the client-server architecture for a distributed P2P platform (Milagro et al 2008). In this approach, all peers will act as clients and servers at the same time.

Of course, such an example would need to combine the architectural approaches into one explanatory one with all the overlapping dependencies and relationships defined throughout the technological and enterprise research challenges described in the very beginning of chapter 9.

11.5 SOA in Product Lifecycle Management

The collective concept of Product Lifecycle Management (PLM) targets product information management in various aspects and from an information context, PLM should provide resources for the management of the information throughout the product lifecycle (including multiple domain views, different business processes scattered across enterprises and different representations of a multitude of native product-, resource- and process-models). From a system point of view, the PLM deals more with the management of information that is created in dispersed systems and where each one provides their own management scope, implementing their own business rules and many times hiding the information from data exchange capabilities. Generally, it is essential for any virtual organization to maintain its competitive advantage by: a) owning their own business processes, b) have their own local master PLM information base, and c) use the computer systems being most suited for the company.

As ebbits will implement SOA there is a potential impact of the platform that will be made visible by the ability of ebbits IoPTS applications to manage production optimization with special emphasis on including energy consumption in the optimization process. SOA need here to be adopted to strengthen the information management by different representations of a multitude of native product-, resource- and process-models.

By modeling SOA for a PLM environment (Fig. 11-6), the value of information can be increased when one piece of information can be seen in the context of other information. An example could be a virtual product that is defined by information in many lifecycle domains but where each domain governs its view of the product and where the traceability or relation, between the provided services and the requirements domain as marked as global glue. So, looking deeper into the information being used in a PLM environment, we can see that it is comprised of content and context (assumptions by mostly implicit and managed informalities). That is, the involved computer systems

[27] An international standard that specifies an information model that defines what information can be exchanged and represented to support a product through life.

each apply their own assumptions, rules and interpretations on the governed information but for the information to keep its integrity and quality it must have a formal context representation.

This might for example become useful in the ebbits platform for smart home environments integrated in the IoPTS where its business application enables consumers to achieve not only comprehensive and reliable information of the life-cycle history but also useful information such as product composition, number of workers involved in making the product and even the company's environmental strategies etc. This could greatly enhance the consumer's experience. But it could also give feedback on consumer opinions that might affect the business processes. The way this is made possible is by the following: as SOA normally implies that services are available for use by clients not known on beforehand, from a PLM perspective a SOA can be used to make access to in-house and commercial PLM systems in an integrated environment. That is, a scalable SOA would require the adoption of new business processes with a minimum of impact on the existing PLM infrastructure but where dealing with heterogeneous environments, the use of a standardized PLM interface would make each user (i.e. consumer) independent of which system is used to deliver the service. Further, this standardization would benefit clients that need to access PLM information from many sources and thereby aggregating PLM information online. To further accommodate system-to-system integration there is a need to have a standardized API that defines events, which can be sent to a message bus. Hence, if PLM information can be sent (via events) each subscribing system can apply their business (context driven) logic. This work is undertaken the ISO 10303 STEP modules.

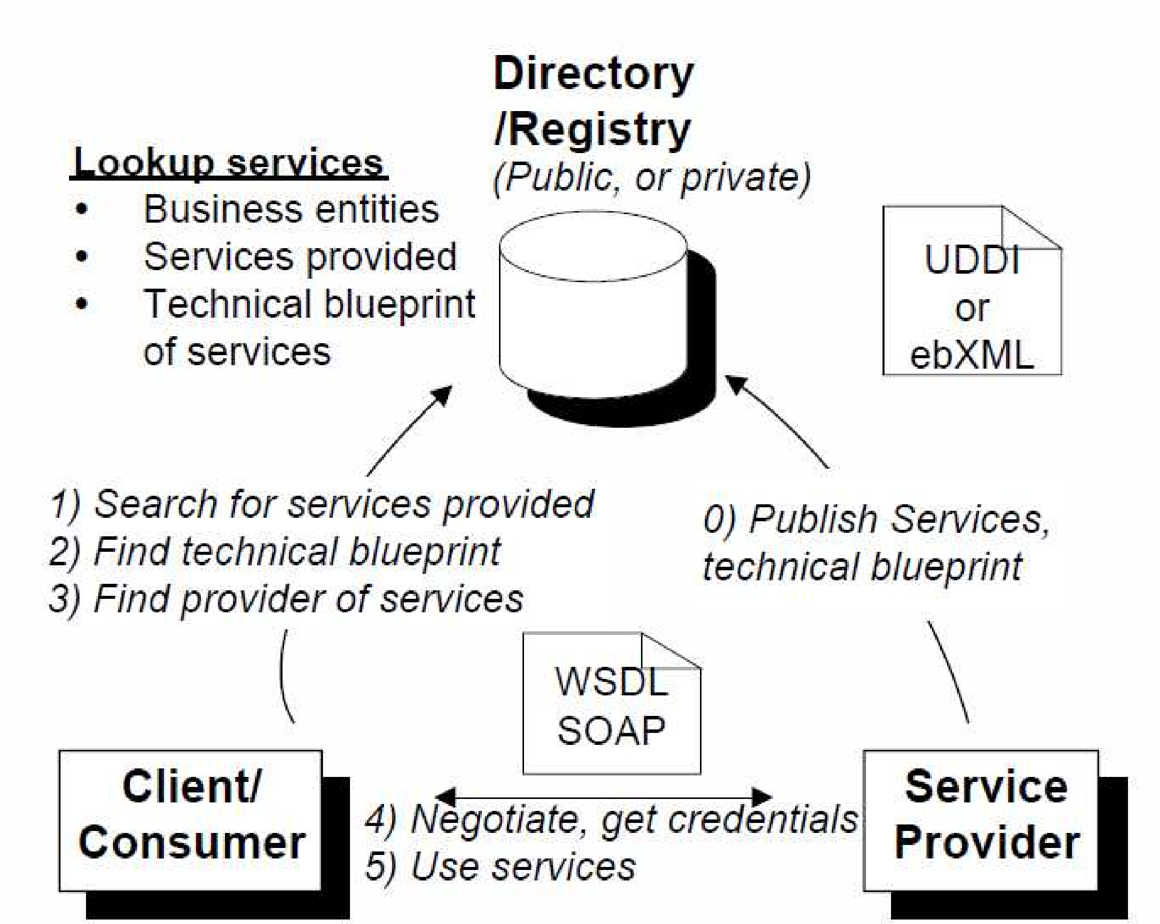


Fig. 11-6 Basic SOA where a Service Provider publish its services to a Directory where Consumers can both look up the available services and get technical blueprints as well as find the service provider. This establishes closer business and consumer relationships(Rosén 2006).

A model driven development of a PLM integration platform would enable the possibility of an ebbits platform talkback such as described by a standardized PLM interface. That is, to plug in an existing PLM system into the PLM environment it needs to expose some functionality according to a given standardized PLM API, e.g., based on the ISO STEP PLM standards. By adopting a standard here would enable a core information model to exist that represents product models throughout the lifecycle in an integrated way. It would also capture content and context and allow business process dependent characteristics to be separated from the actual product model content. And from a system point of view where own business rules and product model representation are established, these can still exchange and share the product data to other systems without changing the semantics. When PLM information is available to clients from many sources it need to be able to configure chunks of information that is subject for configuration control and since many systems govern pieces of information that can be put together in a collaborative context, the assembled information needs to be managed outside each local master system. This makes it possible to introduce traceability and change management that is distributed across different systems and business processes (OASIS PLCS) (Rosén 2006).

Application protocol (AP) in ISO 10303 for PLM and Manufacturing are summarized in the following APs:

Manufacturing APs:

- AP 219, Dimensional inspection information exchange
- AP 223, Exchange of design and manufacturing product information for cast parts
- AP 224, Mechanical product definition for process plans using machining features
- AP 238 - Application interpreted model for computer numeric controllers
- AP 240, Process plans for machined products

Life cycle support APs:

- AP 239, Product life cycle support
- AP 221, Functional data and schematic representation of process plants

11.6 Summary

Access to device features may be radically different among various devices, ranging from human-only access (e.g. screens and buttons), to exclusive M2M (machine-to-machine) communication over standard protocols and wireless transports. This wide array of access mechanisms might be acceptable for the individual device manufacturer, but not for the developer who wants to build solutions based on a vast number of devices from different manufacturers with obscure access protocols, that for the most remain proprietary or unknown. This makes it almost impossible for existing devices to communicate and to interchange useful information and/or commands structures, without the direct involvement of the manufacturers of the devices in one way or the other.

The way to access intelligent services across a distributed network is for the ebbits platform (see Fig. 11-7) to create a ubiquitous communication infrastructure that automatically and dynamically connects to sensors and devices in the physical world in e.g. manufacturing facilities or in private smart homes. It further connects to mainstream backend information systems, public authentication systems and regulatory information sources using semantic web services and finally connects to human users in dispersed geographical locations (e.g. professional users in technical support, field service and other business environments as well as ordinary consumers in shops or at home).

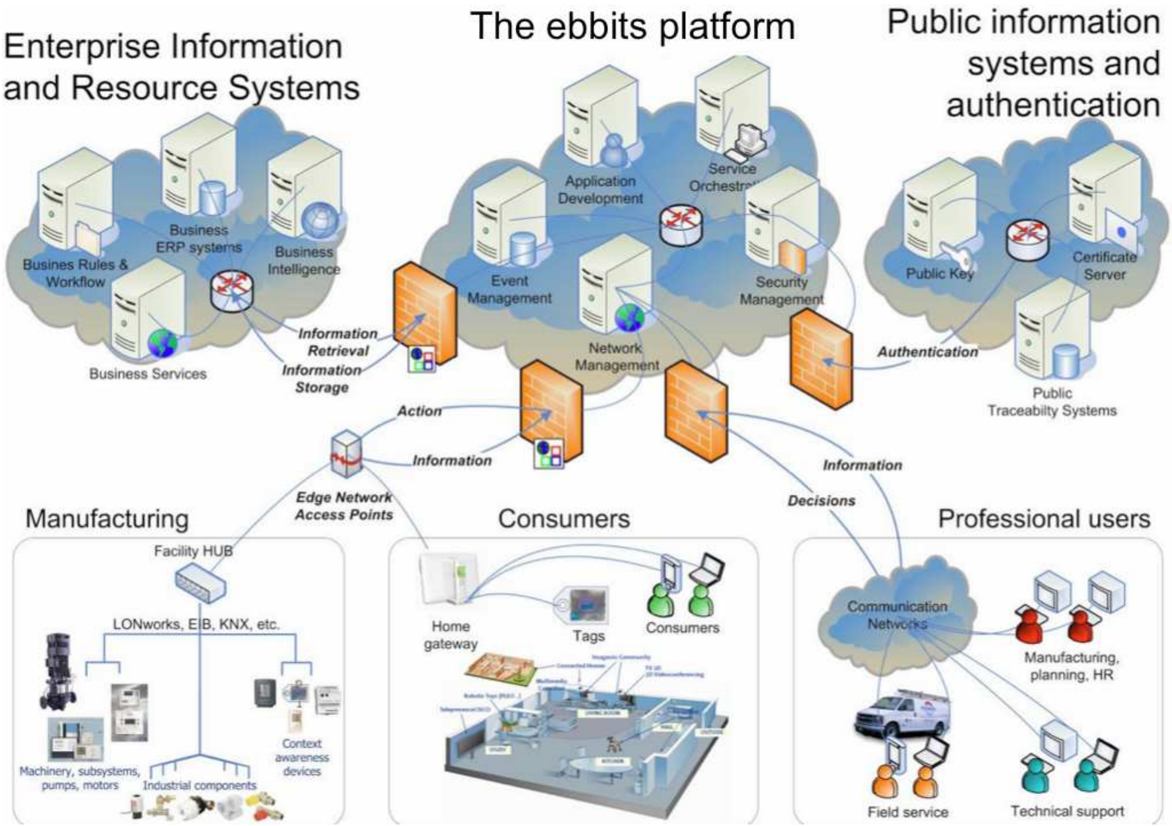


Fig. 11-7 The ebbits platform with a prototypical actualization of SOA.

The ebbits platform consists of subsets of production servers for data management, event management, security, application execution and communication where all servers are to interoperate in an open architecture on the basis of web services. The ebbits SOA is completely platform agnostic and scalable. The implementation of SOA for product lifecycle management and manufacturing should follow ISO 10303 specifically the following application protocols:

Manufacturing APs:

- AP 219, Dimensional inspection information exchange
- AP 223, Exchange of design and manufacturing product information for cast parts
- AP 224, Mechanical product definition for process plans using machining features
- AP 238 - Application interpreted model for computer numeric controllers
- AP 240, Process plans for machined products

Life cycle support APs:

- AP 239, Product life cycle support
- AP 221, Functional data and schematic representation of process plants

Moreover, SOA-ROA hybrid approach should be studied and integrated in ebbits platform supporting different purposes of information access for instance mobile phones are mostly support REST better than SOAP Web Services since REST implementations require less resources while SOAP web services are supported by mot of enterprise applications.

11.7 Bibliography

Fielding, Roy. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral thesis, Irvine: University of California.

Milagro, F., Antolin, P., Kool, P., Rosengren, P., Ahlsén M. (2008). SOAP tunnel through a P2P network of physical devices. In *Internet of Things Workshop, Sophia Antopolis, Sep 2008*.

MSDN Blogs. (2009). Hybrid architecture based on SOA and ROA. URL: http://blogs.msdn.com/b/architectsrule/archive/2009/07/07/hybrid-architecture-based-on-soa-and-roa.aspx [Downloaded: 2010-11-08].

OASIS PLCS. (2004). OASIS Product Life Cycle Support (PLCS) Technical Committee. URL: http://www.oasis-open.org/committees/plcs/charter.php [Downloaded: 2010-11-08].

Oberle, Daniel. (2006). Semantic management of middleware. New York: Springer Science and Business Media, Inc.

Rosén, Jonas. (2006). Federated throuhg-life support, enabling online integration of systems within the PLM domain. In *1st Nordic Conference on Product Lifecycle Management - NordPLM'06, Göteborg, January 25-26.*

SOA4All. (2010). Enabling a Web of billions of services. URL: http://www.soa4all.eu/ [Downloaded: 2010-11-08].

SOCRADES. (2009). Service Oriented Cross-layer infRAstructure forDistributed smartEmbedded deviceS URL: http://www.socrades.eu/ [Downloaded: 2010-11-08].

W3C WS-ARCH. (2004). Web Services Architecture. URL: http://www.w3.org/TR/ws-arch/ [Downloaded: 2010-11-08].

W3C SOA. (2008). Web Open Standards, SOA. URL: http://www.w3.org/2008/11/dd-soa.html [Downloaded: 2010-11-08].

12. Devices and Service Discovery in Intelligent Environments

This chapter documents approaches of discovering devices and services in heterogeneous and highly dynamic environments.

12.1 Web Service Discovery

Any discovery over a network is regarded as the act of locating a machine-process able description of a Web Service-related resource that meets some specific functional criteria. It can also locate previously unknown resources that are open for discovery but usually the discovery itself involves matching a set of functional and other criteria with a set of resource descriptions where the goal is to find an appropriate Web Service-related resource.

By doing so, Web Services provide access to software systems over the Internet using standard protocols. For service providers, publishing a Web Service requires as a minimum as creating a software artifact and make it accessible to others such as consumers or professional users in e.g. ebbits. In order for a consumer or user to be able to use a service, developers usually fix a Web Service endpoint with an interface description by using the Web Services Description Language (WSDL). This endpoint is accessible for the consumer so that services are elucidated. Using the WSDL makes it is possible to describe a particular XML Web Service as the XML Web Service discovery itself is the process of locating or discovering services. In simple words, it is through that discovery process that the XML Web Service clients get noticed that an XML Web Service does exist and where to find its service description document.

Moreover it is possible to explicitly register a service with a Web Services registry such as UDDI or publish additional documents intended to facilitate discovery such as with Web Services Inspection Language (WSIL) documents which are recapped as the service consumers and professional users need to search Web Services manually or automatically. The implementation of UDDI servers and WSIL engines usually is provided as simple search APIs or web-based GUIs. Nonetheless, the current UDDI search mechanism just has the ability to focus on a single search criterion (e.g. business name, business category, service type by name, discovery URL etc.) but when used in a business solution, it becomes useful to conduct multiple UDDI registries or WSIL document searches in order to aggregate any returned results.

Web Services may also be discovered in local network using multicast mechanisms like WS-Discovery which reduces the need for centralized registries. But this is unlikely to be the case for such dynamic and broad work spaces as the one to be found in ebbits (W3C WS-ARCH).

12.1.1 Discovery in SOA

As the software components are services themselves with specially defined implementation-independent interfaces it is important to understand that: 1) services are self-contained and loosely coupled form each other, 2) composite services (i.e. applications) can be built by aggregating other ones, and 3) for this, these services need to be dynamically discovered in the network.

SOA therefore uses the *find-bind-execute* paradigm where service providers register their service in a public registry that is available for use by professional users or consumers. These actors can find specific services by matching certain criteria and if the registry holds that particular service it will provide the actor with a contract and service endpoint address (which was used to access the services). Applications in SOA can therefore be distributed over a multi-tier network but still maintain presentation, business logic and persistence layers.

Realizing SOA with Web Services

In order for SOA to realize the *find-bind-execute* paradigm in a network, it needs to provide web services that are designed to support interoperable machine-to-machine interaction. In SOA, this interoperability is gained through a set of XML-based open standards (some have been mentioned earlier - WSDL, SOAP, UDDI) which provide a common approach that enables the service defining, publishing and the actual how-to-use of Web Services (Qusay 2005).

12.1.2 Semantic discovery

SOA solutions provide modularity and access to service users in the system but an ideal way to offer services is to make use of any business functions by domain modeling and decomposition. As the ebbits platform will feature a SOA-based architecture on open protocols and the Hydra Middleware, it will also effectively transform every subsystem or device into a web service with semantic resolution. This is made with regard to the business purposes within the project.

Using SOA in ebbits not only brings the need to modularize, reuse and provide services but also accurately and intelligently define and infer the descriptions, attributes, parameters and relationships associated with service data. Any user, Middleware, backend application, service provisioning and infrastructure will need the ability to properly infer about (i.e. context of) data that is based on a service request and process. Here, semantics realizes this ability to build the relationships in a dynamic way and to infer where required. To assistance, the semantics have ontologies that describe a formal knowledge representation within a domain and the relationships between any concepts. Within SOA, semantics add value at two levels: service creation, service orchestration, service management and governance but foremost service definition and discovery which are one of the key aspects of any SOA solution by providing efficient service modularization and availability. There are many standards associated with Web Services that are supporting SOA such as WSDL, BPEL[28], UDDI (Verma & Sheth 2007) (W3C OWL-S).

12.1.3 Discovery in ROA

In principle, the ROA service discovery reminds to great extent of the one performed within SOA but with a focus where each domain performs local discovery of its service infrastructure and makes any configuration information available through a RESTful interface. The resources being discovered are usually service descriptions and if a requester entity does not already know what service it wishes to engage, the requester entity must discover one. In ROA this is supported by RESTful Web Services (described later) but also by DNS SD (Domain Name System Service Discovery). The DNS SD specification has been developed to overcome the limitation given by SRV (Service Records), i.e. the lack of service lookup. The DNS SD uses three different record types (PTR, SRV, TXT) to associate various kinds of so called resource records with a given domain and by that provides a service discovery mechanism that can answer: which services are there that support e.g. the Atom Publishing Protocol, service specified by OpenSearch specification, the location of any given service, or what meta data is available for any given service.

RESTful Web Services

When it comes to ROA some say it is just a replacement name for a SOA using REST services but the main advantage with ROA is the use of REST as with RESTful Web Services. To mention first is that as a programming approach, REST is a lightweight alternative to general Web Services and RPC (Remote Process Calls) and much like Web Services, REST comes with services that are platform independent, programming language independent, standards-based (as it runs on top of HTTP) and works fine with firewall presence. As with Web Services, REST cannot offer built-in security features, encryption etc. but can be added by building on top of HTTP.

Discovery is submitted as part of the resources in REST that are the key elements of a RESTful Web Service design. The resources contain 'links' to additional information as the role is to provide a web of resources where single resources are not too large to miss the lightweight objective of having low detail granularity. There is no connection state as all interaction in REST is stateless (even if servers and resources can be stateful) and any new request should carry all the information required to

[28] BPEL describes processes in Business Process Execution Language export and import information by using web service interfaces exclusively.

complete it, and must not rely on previous interactions with the same client. Resources in RESTful Web Services should also be able to cache data whenever this is possible and the protocol (HTTP cache-control headers are used here) must allow servers to explicitly specify which resources may be cached, for how long and clients must respect this. Any standard HTTP proxy can be used as part of the architecture in order to improve performance and scalability.

12.2 UPnP

The UPnP architecture offers peer-to-peer network connectivity of PCs, intelligent appliances and wireless devices. UPnP achieves this by defining and publishing UPnP device control protocols built upon open, Internet-based communication standards. The UPnP architecture is a distributed, open networking architecture that uses TCP/IP and HTTP. It enables seamless networking in addition to data transfer between networked devices at home, in the office and everywhere in between. It enables data communication between any two devices under the command of any control device in the network.

A large number of ICT devices, from PCs to phones, are UPnP compatible today. Recent developments in home automation have also resulted in new types of home appliances that are UPnP-compatible, such as those based on the DLNA (Digital Living Network Alliance[29]). This is an architecture that includes and extends the UPnP protocol, mainly for various media rendering devices.

The main characteristics of UPnP are:

- *Media and device independence*. UPnP technology can run on any medium including phone lines, power lines, Ethernet, IR (IrDA), RF (WiFi, Bluetooth), and FireWire. No device drivers are used; common protocols are used instead.
- *Common base protocols*. Base protocol sets (Device Control Protocols, DCP) are used, on a per-device basis.
- *User interface (UI) Control.* UPnP architecture enables vendor control over device user interface and interaction using the web browser.
- *Operating system and programming language independence*. Any operating system and any programming language can be used to build UPnP products. UPnP does not specify or constrain the design of an API for applications running on control points. OS vendors may create APIs that suit their customer's needs. UPnP enables vendor control over device UI and interaction using the browser as well as conventional application programmatic control.
- *Internet-based technologies*. UPnP technology is built upon IP, TCP, UDP, HTTP, and XML, among others.
- *Extensibility*. Each UPnP product can have value-added services layered on top of the basic device architecture by the individual manufacturers.

The UPnP architecture supports zero-configuration, invisible networking and automatic discovery for a breadth of device categories from a wide range of vendors. Devices can dynamically join a network, obtain IP addresses, announce their names, convey their capabilities upon request, and learn about the presence and capabilities of other devices. DHCP and DNS servers are optional. A device can leave a network smoothly and automatically without leaving any unwanted state information behind.

UPnP is based on the SSDP (the Simple Service Discovery Protocol), a network protocol based on the Internet Protocol Suite for advertisement and discovery of network services and presence information.

As a concrete example of a current device description approach we briefly look at the device modeling and access as defined by the Device Control Protocols (DCP) of UPnP.

[29] http://www.dlna.org

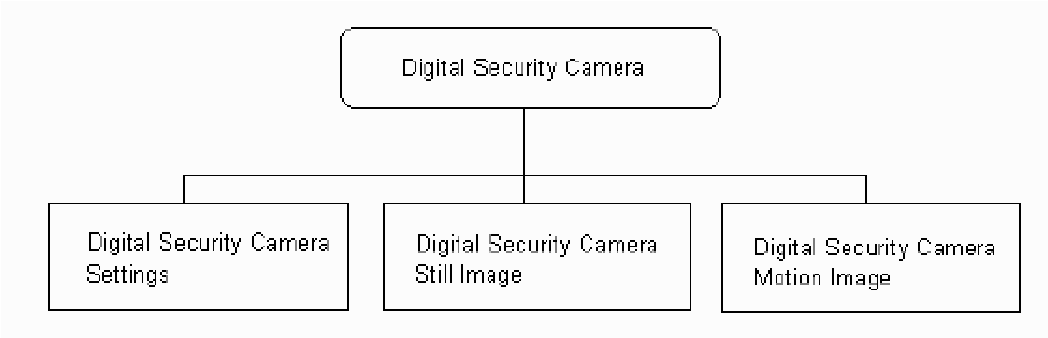


Fig. 12-1 showing inherited functionalities in a digital security camera.

The device (a Digital Security Camera) is described by a simple functional device model, showing the set of services this type of device may expose. In this case there are functions for accessing still and motion images as well as controlling basic camera settings (e.g. white balance). The device description is represented in an XML structure following the UPnP device type format.

```
<root xmlns="urn:schemas-upnp-org:device-1-0">
    <specVersion>
    ....
    </specVersion>
    <URLBase>base URL for all relative URLs</URLBase>
    <device>
        <deviceType>urn:schemas-upnporg:device:DigitalSecurityCamera:1</deviceType>
        <friendlyName>short user-friendly title</friendlyName>
        <manufacturer>manufacturer name</manufacturer>
        <manufacturerURL>URL to manufacturer site</manufacturerURL>
        <modelDescription>long user-friendly title</modelDescription>
        <modelName>model name</modelName>
        <modelNumber>model number</modelNumber>
        <modelURL>URL to model site</modelURL>
        <serialNumber>manufacturer's serial number</serialNumber>
        <UDN>uuid:UUID</UDN>
        <UPC>Universal Product Code</UPC>
        <iconList>
            <icon>...</icon>
        </iconList>
        <serviceList>
            <service>..</service>
            .........
        Declarations for other services added by UPnP vendor (if any) go here
    </serviceList>
    <deviceList>
        Description of embedded devices added by UPnP vendor (if any) goes here
    </deviceList>
        <presentationURL>URL for presentation</presentationURL>
</device>
</root>
```

Table 12-1 showing an UPnP device type model template.

The above description provides a model template for the UPnP device type DigitalSecurityCamera identified by the deviceType URN defined by the UPnP standard. The device description includes a list of service descriptions for the device functionality. A service list may be extended with manufacturer specific services in addition to the ones prescribed by the UPnP device type.

A service description list identifies each service by its service type and ID and list URLs for control and functions. The camera device type supports two services for camera control settings, access to still images and access to motion images.

```xml
<service>
        <serviceType>urn:schemas-upnporg:service:DigitalSecurityCameraSettings:1</serviceType>
        <serviceId>urn:upnporg:serviceId:DigitalSecurityCameraSettings</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
</service>
<service>
        <serviceType>urn:schemas-upnporg:service:DigitalSecurityCameraStillImage:1</serviceType>
        <serviceId>urn:upnporg:serviceId:DigitalSecurityCameraStillImage</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
</service>
<service>
        <serviceType>urn:schemas-upnporg:service:DigitalSecurityCameraMotionImage:1</serviceType>
        <serviceId>urn:upnporg:serviceId:DigitalSecurityCameraMotionImage</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
</service>
```

Table 12-2 showing service type associated with the device type.

An individual service may have a set of state variables on which it operates, e.g., the default encoding for images. Any product, that exposes a device description referring to a specific UPnP device type, is required to implement mandatory descriptive elements and services as described in the corresponding DCP documentation. The UPnP device template framework provides a fairly simple and open model for device description and service access. The model provides for extensibility through the XML-based description model.

12.3 The Hydra approach

This section is based on previous research conducted in Hydra Project (Hydra 2010a).
As stated in the ebbits project DoW, the platform should re-use and be built on results from several highly successful EU RTD projects. It also concludes that while a fundamental technology for Internet of Things is how to connect physical devices and appliances to the Internet, it requires a middleware to run either on the device or as a proxy in a gateway. The middleware must provide a number of services such as providing service interfaces for accessing the device and discovery mechanisms so the device can be discovered by other devices and applications. The Hydra project, which is one of the baselines for ebbits, has pioneered research into service-oriented architectures for networked embedded devices based on a semantic model-driven approach. The ebbits semantic interoperability capabilities will allow business logic to be distributed and executed on the edges of a networks, thus enabling some business processes to be decentralized for the benefit of performance, scalability, and local decision-making. The main contribution of the Hydra Middleware to the ebbits architecture is therefore to support:

- Distributed, open, service oriented architecture with end-to-end characteristics, neutral access, clear layering and resilience to physical network disruption.
- Cloud enabling technology featuring event-driven architecture with autonomous synchronization, interoperability of heterogeneous systems and support for self-properties.
- A decentralized architecture based on peering of nodes and distributed intelligence.
- Mechanisms for gathering fragments of distributed information from a variety of sources, even when those sources are not known a priori, in order to achieve comprehensive end-to-end traceability as far as is permitted.
- Distributed ownership of data, in which objects (enterprises, services, devices, people, etc.) can control which information to share with other objects based on a strict, federated authorization control.

The main contribution of Hydra to the Semantic Web is to bring semantic web technologies down to the device level, i.e. each device can act as a semantic web service accessible by other devices, users and software applications.

12.3.1 Devices

The Hydra middleware platform provides a discovery architecture that builds on UPnP technology. The approach implements a three layered discovery architecture that includes physical device detection, UPnP network announcement and semantic resolution of devices against a device ontology.

Hydra applications are built by programming networked ambient intelligent devices. Devices are made programmable by the Hydra middleware through proxies as well as by embedded components. Whatever the method, it is transparent to application developers, as they access all devices based on a pure service and event based programming model. In order to support open and dynamic networks, the device protocols need to provide descriptions of the capabilities of the supported devices. This includes device identity and functional interfaces (services) and possibly also additional information such as details about the manufacturer, the model and the version. Powerful instruments for device modeling and description are central in the Hydra architecture, as in all networks of devices and the "Internet of things". A number of efforts have been launched or are in pursuit to promote device modeling and management function to facilitate device interoperability in ambient intelligence environments similar to those are supported by Hydra.

A Hydra Device is the software representation of a physical device. This representation is either implemented by a proxy running on a gateway device, or, by embedded Hydra managers on the actual device. A Hydra Device is said to Hydra-enable a physical device. There are five categories of Web (UPnP) services generated for a Hydra Device:

- A Generic Hydra web service, exposing metadata and management functions common to all Hydra Devices
- An Energy web service, providing a set of functions for the monitoring and control of the energy consumption of devices.
- A Memory Service which allows logging and storing of device internal data such as state variables and energy consumption data.
- A Location Service which can be used to query the device about is location and position.
- A device type specific web service, exposing the device type specific functions

12.3.2 Run-time architecture

The Hydra middleware incorporates support for self-discovery of devices. When a Hydra enabled device is introduced then the middleware is able to discover and configure the device automatically. In Fig. 12-2, we see an example of a Hydra device network. Hydra distinguishes between two different devices. More powerful devices are capable of running the Hydra middleware natively and smaller devices that are too constrained or closed to run the middleware. For the latter devices, proxies are used and once proxies are in place, all communication is based on the IP protocol.
If we look again on the Fig. 12-2 below that illustrates the two cases, we will see on the right the terminal that directly can incorporate Hydra middleware and is able to establish communication with services on the ebbits platform. In the situation on the left, the devices cannot operate the Hydra middleware (because they are too resource constrained or have proprietary interfaces). In this case, proxies are created on the BAN or PAN node (in this case a mobile phone). The proxies virtualizes the device vis-à-vis the ebbits platform. Any service will think it is communicating with the device, where in fact it is communicating with the proxy.

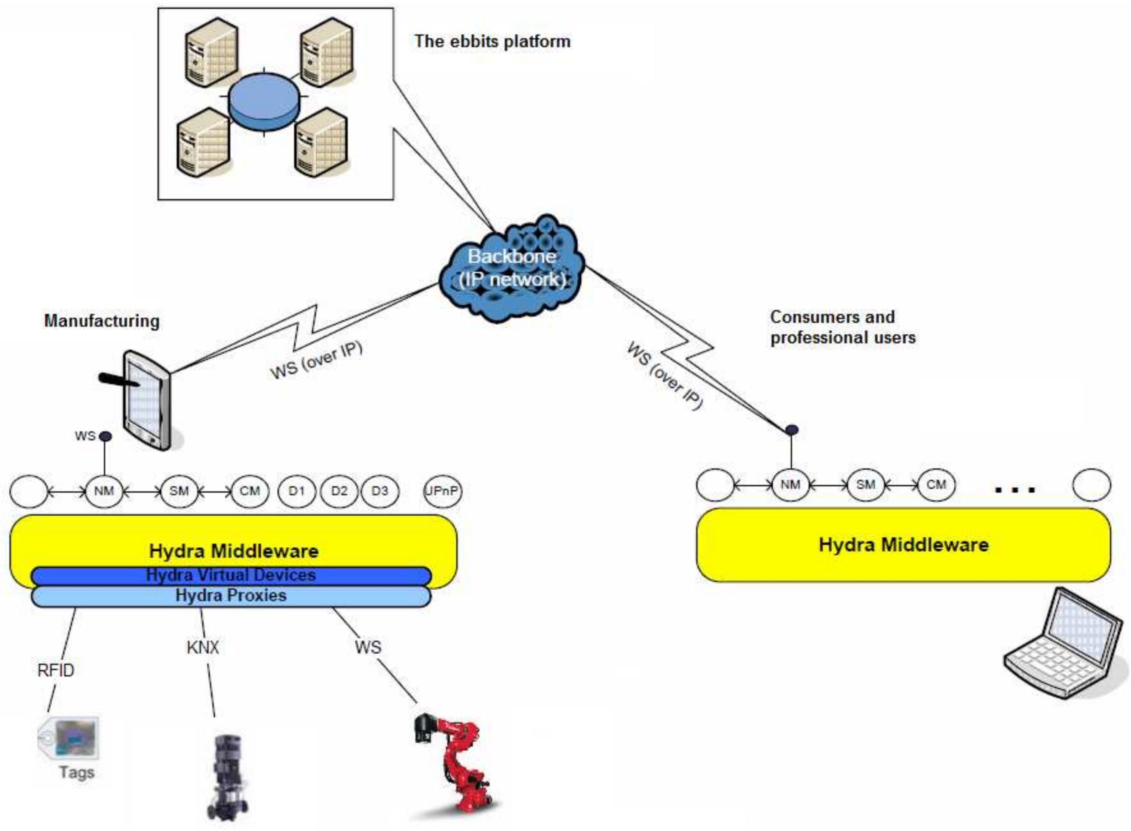


Fig. 12-2 Incorporation of devices using the Hydra middleware.

12.3.3 Device Discovery Architecture

The Hydra model driven architecture (MDA) includes a Discovery Architecture which implements the device discovery process. This architecture is structured in three layers abstracting the discovery functions. The discovery process operates both locally and remotely, so that devices that are discovered in a local Hydra network can also be discovered in a peer Hydra network over the P2P protocol implemented by the Hydra Network Manager.

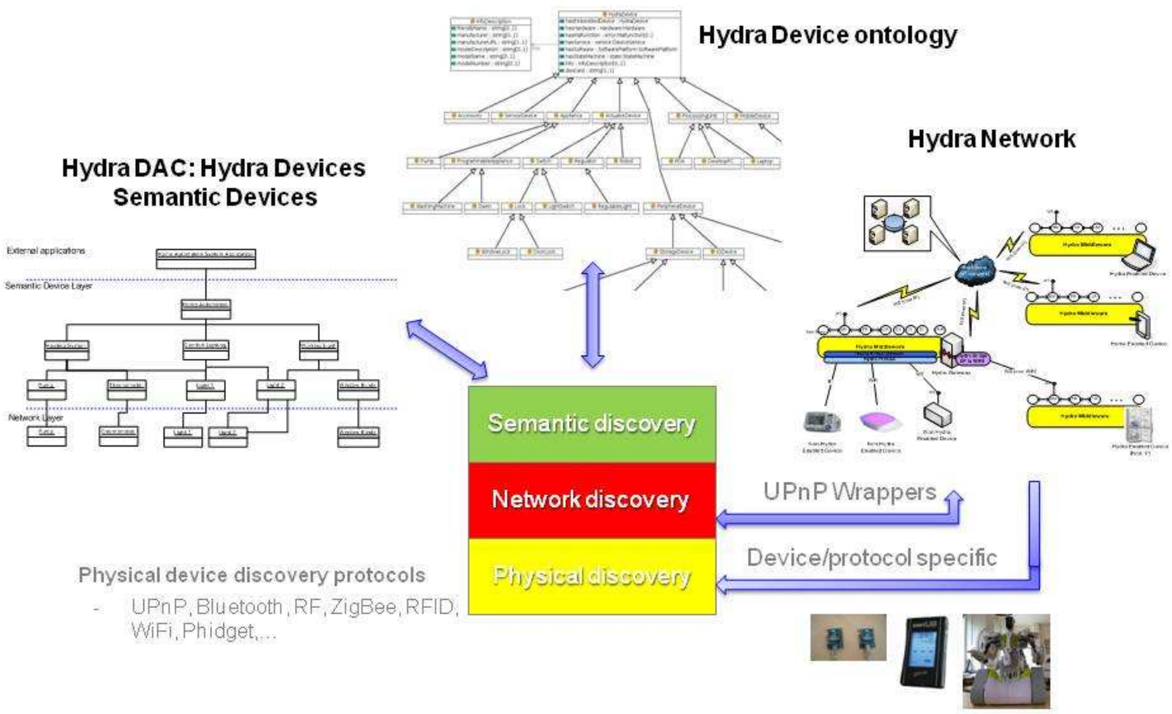


Fig. 12-3 The 3-layered Discovery Architecture is part of the Hydra MDA.

The lowest discovery layer implements the protocol specific discovery of physical devices. This is performed by a set of specialized discovery managers listening for new devices at gateways in a Hydra network. The second layer uses UPnP/DLNA technology to announce discovered physical devices in the local network and to peer networks.

At the top most layer the device type is resolved against the Device Ontology and is mapped to some Hydra Device type. It is then placed in the *Device Application Catalogue* (DAC). If an application subscribes to events regarding this type of device, it will be notified that the device is available and has been placed in the Device Application Catalogue.

The Hydra middleware provides: 1) Discovery mechanism, 2) Low level protocols, 3) Service execution, 4) Virtualization, and 5) security and trust policies which can directly be used by the developer of ebbits applications. The whole process of the Hydra middleware management of devices and services is reviewed on the following page.

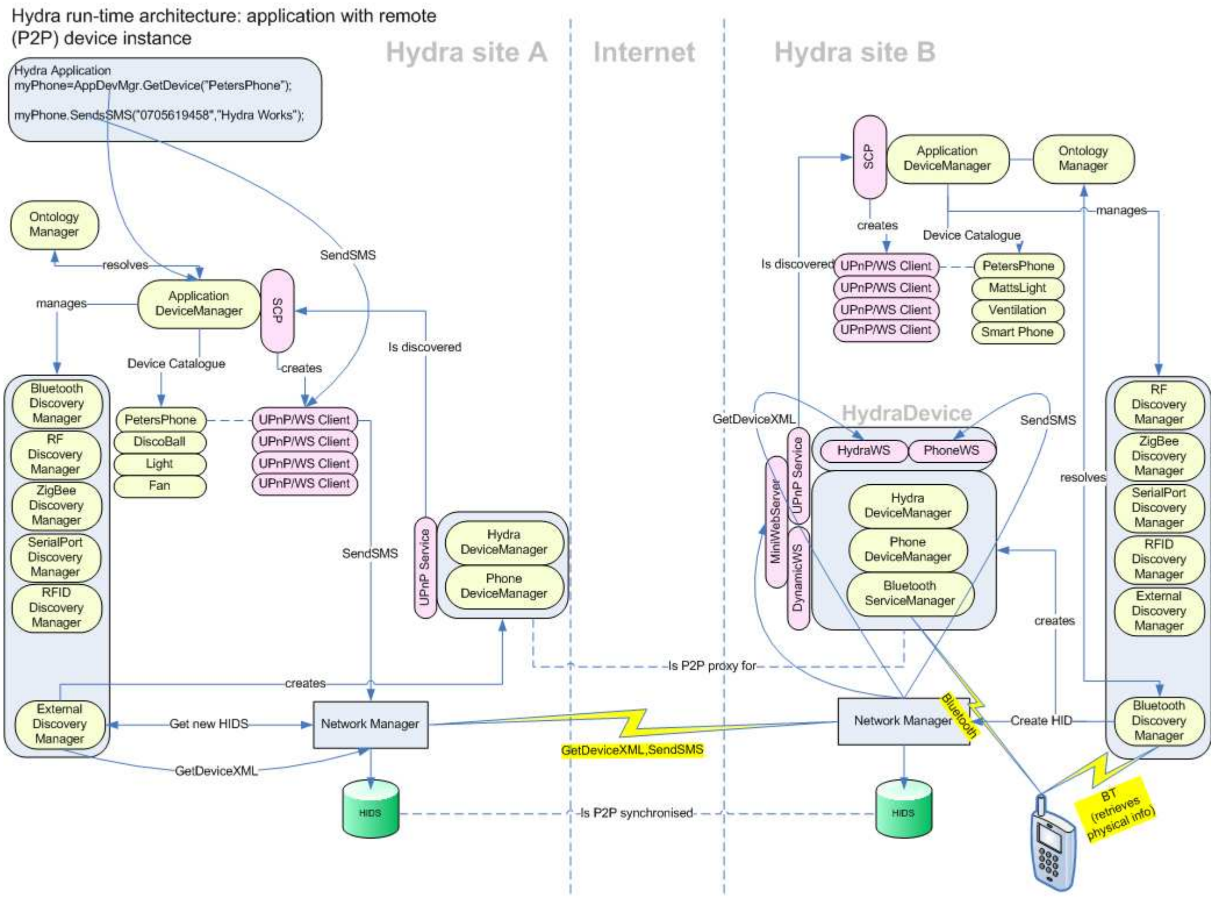


Fig. 12-4 Discovery in the Hydra P2P architecture.

The Fig. above (Fig. 12-4) both illustrates local discovery of physical devices as well as P2P discovery between two local networks and is described by the following points directly below here.

- A Bluetooth phone comes into a local network (lower right).
- The phone is discovered by the Bluetooth Discovery Manager running on site B. The Bluetooth Discovery Manager extracts as much information from the phone as possible and forwards it to the Ontology Manager at site B.
- The Ontology Manager reasons and concludes it has found a device of type Basic Phone, it instructs the Bluetooth Discovery Manager to create a proxy and an interface for such a device.
- The BT Discovery Manager creates a Hydra Device that consists of a Phone DeviceManager and a Bluetooth Device Service Manager. These two modules are encapsulated into a three web

services, a Phone Web Service that exposes the phone functions (SendSMS, ReadSMS etc.), the generic Hydra Web Service and the Energy Web Service.

- In addition to the web services, also the same service but UPnP based are created, so that the device can be accessed either using web services or UPnP depending on the requirements of an application.
- The Discovery Manager then dispatches the Hydra Device and uses Network Manager to create a Hydra identifier, HID, which is registered with the Network Manager.
- The Hydra Device uses an UPnP broadcast message to announce itself in the local network The Hydra Device is discovered by the Application Device Manager, who updates the Device Application Catalogue.
- The Hydra Device is now fully functional and available for applications and other devices in the local network on site B.
- The Network Managers in site B and in site A are using P2P techniques to synchronize their own databases of Hydra identifiers.
- An External Discovery Manager is running on site A and will discovery that a new device has appeared on site B. It uses the SOAP tunneling mechanism of the Network Manager A to query about the device description of the remote device. Network Manager B receives this request and resolves it using its internal HID database. It results in a local web service call being made to the Phone's generic Hydra Web Service.
- The result of the WS call is then returned to the External Discovery Manager at site A, which now has enough information to create a local proxy for the Phone on site B.
- In the same way this local Hydra Device is discovered in the local network at site A and registered in the Device Application Catalogue.
- One thing that now needs to be done is to bind the Phone to a local application identifier. Applications running in site A needs to be able to refer to the devices they need to use, without knowing their physical address or IP-address. These bindings are set up by the Application Device Manager through a rule set provided by the application developers.
- In this case the Phone is now bound to the local identifier "Peters Phone" and an application on site A can now invoke services on the phone referring to it as "Peters Phone". When the call SendSMS comes from site A it is routed using P2P, SOAP tunneling and local web service invocation on site B.

12.3.4 Hydra Device XML

Since all metadata and the state of a device is communicated using an XML structure it is fundamental to understand this structure and how it can be used. Below is an example of the Hydra Device XML for a device. The Hydra Device XML is an extension of the UPnP SCPD XML (Service Control Point Document) vocabulary. Elements with the namespace "hydra" are the Hydra-specific extensions.

```
<root xmlns="urn:schemas-upnp-org:device-1-0">
<specVersion>
        <major>1</major>
        <minor>0</minor>
  </specVersion>
 <device>
        <deviceType>urn:schemas-upnp-org:hydradevice:enhanchedswitchdevice:1</deviceType>
         <hydraidDynamicWS xmlns="hydra">0.0.0.6189708676876140718</hydraidDynamicWS>
        <energywsendpoint xmlns="hydra">http://212.214.80.144:8080/hydradevice/8619ff3a-af98-
        44a9-85da-7f5f18f7e562/energy</energywsendpoint>
        <hydraidStaticWS xmlns="hydra">0.0.0.6592261886889156134</hydraidStaticWS>
        <discoveryinfo
        xmlns="hydra"><tellstickdevice><name>PetersLight2</name><vendor>Nexa</vendor><devicei
        d>2</deviceid></tellstickdevice></discoveryinfo>
         <hydraidUPnPService_urn_schemas-upnp-org_memoryservice_1
        xmlns="hydra">0.0.0.4695383175879738995</hydraidUPnPService_urn_schemas-upnp-
        org_memoryservice_1>
```

```
 <networkmanager
xmlns="hydra">http://localhost:8082/services/NetworkManagerApplication</networkmanager
>
 <hydraUDN xmlns="hydra">PetersLight2</hydraUDN>
 <standbytime xmlns="hydra">60</standbytime>
 <status xmlns="hydra">web service initiated</status>
 <hydraidStaticWSDescription
xmlns="hydra">PetersLight2:StaticWS</hydraidStaticWSDescription>
<hydraidUPnPService_urn_schemas-upnp-org_locationservice_1
xmlns="hydra">0.0.0.8817877591614169464</hydraidUPnPService_urn_schemas-upnp-
org_locationservice_1>
 <hydraidUPnPService_urn_schemas-upnp-org_energyservice_1
xmlns="hydra">0.0.0.410334127518851262</hydraidUPnPService_urn_schemas-upnp-
org_energyservice_1>
  <hydraWSEndpoint xmlns="hydra">http://212.214.80.144:8080/hydradevice/8619ff3a-af98-
44a9-85da-7f5f18f7e562</hydraWSEndpoint>
 <UPnPEndpoint xmlns="hydra">http://212.214.80.144:64277/</UPnPEndpoint>
 <hydraidUPnPService_urn_upnp-org_serviceId_switchservice_1
xmlns="hydra">0.0.0.7715272012937744631</hydraidUPnPService_urn_upnp-
org_serviceId_switchservice_1>
 <dynamicWSEndpoint xmlns="hydra">http://212.214.80.144:64277/</dynamicWSEndpoint>
 <wsendpoint xmlns="hydra">http://212.214.80.144:8080/0/EnhancedSwitchWS</wsendpoint>
 <hydraidHydraWS xmlns="hydra">0.0.0.713272519360667694</hydraidHydraWS>
  <DACEndpoint
xmlns="hydra">http://212.214.80.144:8080/ApplicationDeviceManager</DACEndpoint>
 <hydraidUPnPDescription xmlns="hydra">PetersLight2:UPnP</hydraidUPnPDescription>
  <hydraidHydraWSDescription
xmlns="hydra">PetersLight2:HydraWS</hydraidHydraWSDescription>
<securityinfo xmlns="hydra"><securityInfo xmlns="hydra"><property
name="tellstick.api.version"><value>2.1</value></property><property
name="switch.mode"><value>2</value></property><property
name="EncryptionProtocol"><value>None</value></property></securityInfo></securityinfo>
 <hydraidUPnPService_urn_upnp-org_serviceId_1
xmlns="hydra">0.0.0.6339391984478104269</hydraidUPnPService_urn_upnp-org_serviceId_1>
<hydraidEnergyWSDescription
xmlns="hydra">PetersLight2:EnergyWS</hydraidEnergyWSDescription>
 <gateway xmlns="hydra">BLONDIE</gateway>
 <hydraidUPnP xmlns="hydra">0.0.0.3263501067198386232</hydraidUPnP>
<hydraidEnergyWS xmlns="hydra">0.0.0.3952190387415366563</hydraidEnergyWS>
 <friendlyName>PetersLight2</friendlyName>
<manufacturer>Telldus</manufacturer>
<manufacturerURL>http://www.telldus.se</manufacturerURL>
 <modelDescription>Remote switch</modelDescription>
 <modelName>Tellstick</modelName>
  <modelNumber>X1</modelNumber>
  <UDN>uuid:8619ff3a-af98-44a9-85da-7f5f18f7e562</UDN>
</device>
</root>
```


Fig. 12-5: Device XML

The following element is an example of a standard UPnP element. It specifies the device type:

```
<deviceType>urn:schemas-upnp-org:hydradevice:enhanchedswitchdevice:1</deviceType>
```

This element is an example of a Hydra-specific extension. It specifies the gateway where the device is running:

```
<gateway xmlns="hydra">BLONDIE</gateway>
```

There are a number of methods that allows for searching of devices in the network. These require XPath expressions as parameter. This Xpath expression is evaluated against the Hydra Device XML for each device to decide if the match the search criteria or not.

The various elements can be grouped into categories:

- PID

  The hydraUDN element represents the PID (Persistent ID) that has been assigned to this particular device.

  ```
  <hydraUDN xmlns="hydra">PetersLight2</hydraUDN>
  ```

- Hydra-IDs

  The following elements represent the different Hydra IDs (HID) for different device services. The hydraidStaticWS is the normal HID to be used, while hydraidHydraWS is the HID to access the generic Hydra services of the device.

  Note the element hydraidUPnPService, for each UPnP service a HID is created with the format hydraidUPnPService_serviceid (where the element by the service id : has been replaced with _as in "hydraidUPnPService_urn_schemas-upnp-org_energyservice_1")

  ```
  hydraidStaticWS
  hydraidDynamicWS
  hydraidHydraWS
  hydraidEnergyWS
  hydraidUPnPService_
  ```

- Endpoints

  The endpoint elements represent the endpoint to the device service. Normally this should not be used. Use the corresponding HID instead.

  ```
  energywsendpoint
  wsendpoint
  dynamicwsendpoint
  UPnPendpoint
  ```

- Other Hydra elements

  The DACEndpoint element represents the DAC that has discovered and created the Hydra Device. It "owns" the device

  ```
  <DACEndpoint
  xmlns="hydra">http://212.214.80.144:8080/ApplicationDeviceManager</DACEndpoint>
  ```

  The gateway element represents the gateway where the device is running:

  ```
  <gateway xmlns="hydra">BLONDIE</gateway>
  ```

- UPnP elements

  The following elements are standard UPnP elements

  ```
  <deviceType>urn:schemas-upnp-org:hydradevice:enhanchedswitchdevice:1</deviceType>
  <friendlyName>PetersLight2</friendlyName>
  <manufacturer>Telldus</manufacturer>
  <manufacturerURL>http://www.telldus.se</manufacturerURL>
  <modelDescription>Remote switch</modelDescription>
  <modelName>Tellstick</modelName>
  <modelNumber>X1</modelNumber>
  <UDN>uuid:8619ff3a-af98-44a9-85da-7f5f18f7e562</UDN>
  ```

  It is possible to extend the Hydra Device XML to incorporate your own meta data and state information. Simply call the method SetProperty in the Hydra WS, then you can add properties to the device which will be available in the Hydra Device XML and can be used as part of your search expressions.

Calling myDevice.SetProperty("myproperty","value1"), will create the following element in your Hydra Device XML:

```
<myproperty xmlns="hydra">value1</myproperty>
```

You can then easily select devices in the network that has myproperty="value1".
For instance the following call with get a Hydra encoded URL to the Energy WS for the all devices that has myproperty="value1".

```
m_applicationdevicemanager.GetHydraURLsFromXpath
        (".//*[name()='myPropety' and .='value1']" , "hydraidEnergyWS", "");
```

12.3.5 Discovery and management of devices in Hydra

In Hydra, the Application Device Manager (Fig. 12-6) manages all knowledge regarding devices that have been discovered and are active in the Hydra network. The Application Device Manager knows about devices from a network perspective but does not handle the locations or context of the devices. The Application Device Manager's main functions are discovery of new (and existing) devices, semantically resolves the device type and available services based on the Device Ontology, creates a service interface for the device, manages semantic device descriptions, provides semantic device aggregation and manages the Device Application Catalogue.

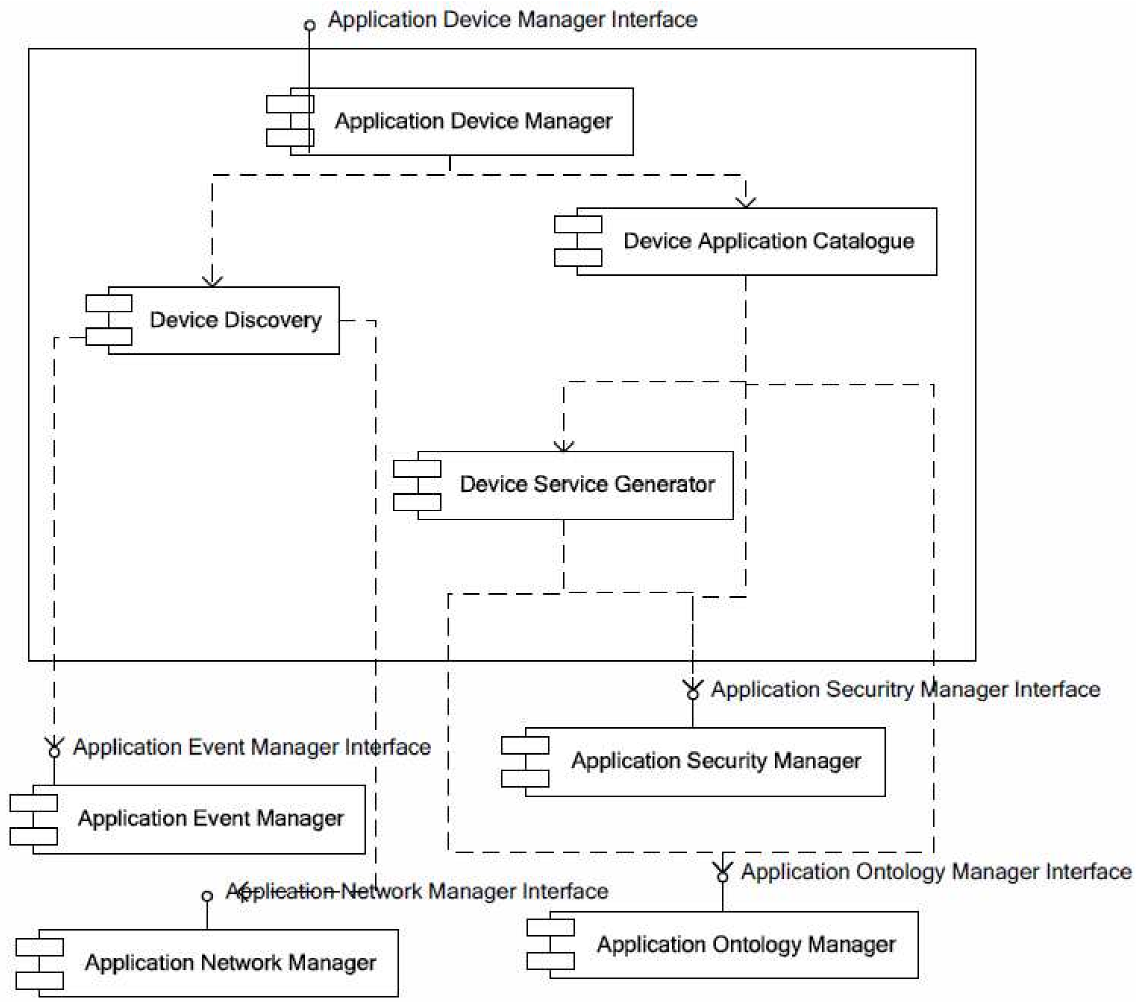


Fig. 12-6: Hydra Application Device Manager components

Device Discovery is one of the major functions of the Application Device Manager and the aim is to discover new devices in the network. It will support user-initiated discovery as well as automatic

schemes. For each device protocol such as Bluetooth and ZigBee there is a dedicated discovery module (Fig. 12-7) that manages the protocol specifics. This can be extended by need. Discovery managers run on Hydra gateways where they look for remote devices such as Bluetooth devices.
Device Application Catalogue keeps track of and manages all devices that are currently active within one application. It is a view on the Device Ontology. It can be queried about existing devices and their status. It can also provide service interfaces for the different devices upon request. The Device Application Catalogue will also keep track of when the device entered the system, when it was last heard of and its current state. The Device Service Manager is responsible for generating a service interface for a certain device. It will create web services as well as UPnP services.
Discovery Manager is the base class for all discovery managers in the Hydra Middleware. A discovery manager is part of the Application Device Manager.

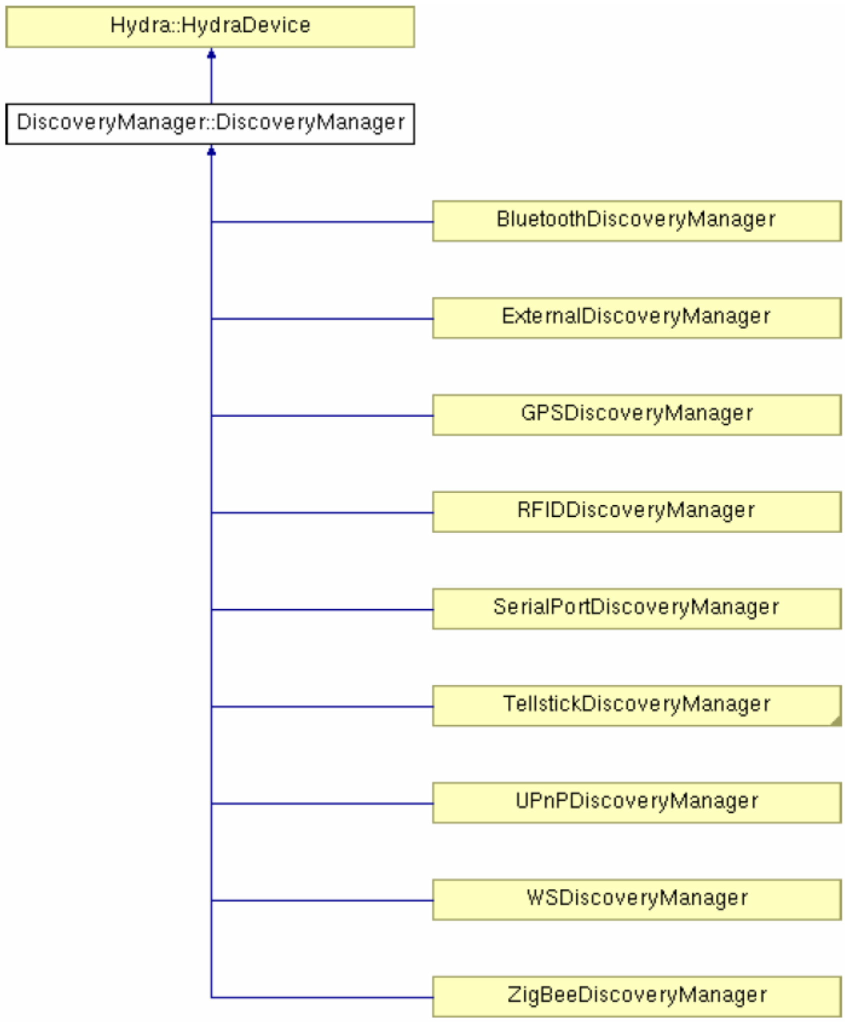


Fig. 12-7: Hydra discovery managers class hierarchy

A discovery manager keeps track of the devices it has discovered. As long as the devices are unresolved they are treated as embedded devices of the Discovery Manager. A discovery manager runs locally on a gateway/PC where it looks for remote devices such as Bluetooth or RF switches devices. The discovery manager has direct access to the device objects it has created. The Discovery Manager is part of the implementation (a sub-manager) of the Application Device Manager. This (sub-) manager also implements the base class for all protocol specific discovery managers in Hydra.
A discovery manager keeps track of the devices it has discovered. As long as the devices are unresolved they are treated as embedded devices of the Discovery Manager. A discovery manager

runs locally on a gateway/PC where it looks for remote devices such as Bluetooth devices (Hydra Middleware).

In summary, the Hydra middleware provides: 1) Discovery mechanism, 2) Low level protocols, 3) Service execution, 4) Virtualization, and 5) security and trust policies which can directly be used by the developer of ebbits applications. The integration of Hydra Middleware will provide a number of core software modules that can be implemented in the ebbits architecture.

12.4 Summary

ebbits will use discovery mechanism provided by the Hydra middleware. Hydra introduced 3-layer discovery mechanism for existing devices. In the first layer, Hydra discovers devices using device specific discovery protocols such as Bluetooth discovery and ZigBee discovery. After physical devices have been discovered, in the second discovery layer, Hydra then creates a proxy that uses UPnP protocols to advertise itself in the IP layer. In the third layer, Hydra tries to classify the device based on the information it advertises and maintain the device in a catalogue (Device Application Catalogue) that can be accessed by Hydra application from different side of the network. Hydra offers semantic abstraction of devices that allow high flexibility such as addressing devices by their properties. This is required in such a dynamic environment where devices enter and leave the network unpredictably.

12.5 Bibliography

Christensen, H.B., Corry, A., and Hansen, K.M. (2004). *An Approach to Software Architecture Description using UML.*. Technical Report, Computer Science Department, University of Aarhus. http://www.daimi.au.dk/~hbc/publication/christensen2004archdesc.pdf (Accessed 2008-05-05)

Hydra Consortium. (2010a), D6.11 Hydra SOA and Model-Driven Architecture, in *Hydra Project Deliverable*. 2010, IST project 2005-034891.

Hydra Middleware. (2010b). Middleware for networked devices. URL: http://hydramiddleware.eu/ [Downloaded: 2010-11-08].

Verma, K, Sheth, A. (2007). Semantically Annotating a Web Service. In *IEEE Internet Computing March/April, (vol. 11 no. 2), pp. 83-85.*

Qusay H. Mahmoud. (2005). Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI). URL: http://www.oracle.com/technetwork/articles/javase/index-142519.html [Downloaded: 2010-11-08].

UPnP Forum. (2010). URL: http://www.upnp.org/ [Downloaded: 2010-11-09].

W3C OWL-S. (2004). OWL-S: Semantic Markup for Web Services. URL: http://www.w3.org/Submission/OWL-S/ [Downloaded: 2010-11-09].

W3C WS-ARCH. (2004). Web Services Architecture. URL: http://www.w3.org/TR/ws-arch/ [Downloaded: 2010-11-08].

W3C SOA. (2008). Web Open Standards, SOA. URL: http://www.w3.org/2008/11/dd-soa.html [Downloaded: 2010-11-08].

13. Opportunistic Peer-to-Peer Communication

Opportunistic Peer-to-Peer is a communication and application approach which can be used to exchange resources (e.g., data) among a set of interconnected objects. Despite its complexity, it can provide advantages with respect to resilience, self-healing and pervasiveness of network connection, especially in environments where connectivity is limited, heterogeneous or partially disrupted, such as in the case of physical-world environments.

Opportunistic Peer-to-Peer is a non-traditional approach and it is not widespread as other, well-known interaction schemes, e.g., client-server. For such reason it is important to analyze carefully the possible non-functional requirements introduced by such approaches on the definition of Intelligent Service Structures.

This section is organized as follows: in first place, an overview of existing Peer-to-Peer systems is provided; in second place, opportunistic communication is defined and explained. Finally, examples of possible uses and impacts of Peer-to-Peer Opportunistic Communication are analyzed.

13.1 Peer-to-Peer systems

Peer-to-Peer is a distributed computing architecture where a group of objects (or peers) can *exchange resources directly with each other* without the support of a centralized entity managing and organizing communication. According to such approach peers play at the same time the *twofold role of client and server* (i.e., the entity providing resources and the entity accessing those resources) and operate in a *cooperative fashion* in order to fulfill one or more services.
Peer-to-Peer systems go beyond the traditional client/server scheme in which simple hosts (users) access resources hosted by powerful, possibly replicated servers. The success of such interaction approach is witnessed by file sharing Peer-to-Peer based applications which have become the most bandwidth-consuming application since the beginning of this century. Initially, the typology of files more frequently exchanged by peers consisted of mp3 music files. Subsequently, they were gradually replaced by video files, characterized by much larger size, exploiting the growth of the bandwidth available to users for accessing the Internet and of the typical storage resources onboard personal computers.

Differently from client-server applications such as web pages requests, in Peer-to-Peer systems, file sharing users - acting as clients - aim to download a known resource, or more generally a set of resources matching some qualification, from one or more unknown users - acting as servers (i.e., *the resource is known, the resource provider is not*). This resource can be identified precisely through some combination of unique identifiers such as, e.g., an SHA-1 hash of the resource contents, but it is not known a-priori where it is physically located. Therefore, the main problem faced by a file sharing system is to identify one or multiple hosts where such resource is available in order to subsequently establish a series of client-server connections between the requesting user and the ones currently sharing the requested resource (*resource location*).

A simple solution to the resource location problem was provided by the first generation of file sharing applications proposed and actually implemented, such as Napster[30]. A powerful central server kept continuously up-to-date a database whose entries contained the name of a specific resource shared in the Internet by some Napster peers and the list of IP addresses of peers currently storing such resource. Though actual data exchange was performed in a Peer-to-Peer fashion, such architectures were still partially centralized since all the relevant pieces of information were recorded in a single tracker. Subsequent file sharing systems which have followed tried to overcome such weakness. The common approach was to distribute among the peers not only the actual resources (i.e., the files to be shared with the Peer-to-Peer community) but also information about their availability and location. Chronologically, Gnutella[31] and Freenet[32] were the first

[30] Napster website, http://www.napster.com
[31] Gnutella documentation, http://rfc-gnutella.sourceforge.net/
[32] Freenet documentation, http://freenetproject.org

contributions and their common idea was a uniform distribution of the control information within the network and the proposal of flooding-based search protocols to localize resources (unstructured resource location). However, these solutions suffered from scalability problems. Significant performance improvements of search and localization of a desired resource in a file sharing system have been achieved thanks to the research activities on structured DHT (Dynamic Hash Tables) Peer-to-Peer solutions. Chor[33], Pastry (Rowstron and Druschel 2001) and Tapestry (Zhao, Ling et al. 2004) are only the most known examples of academic research proposals whose findings have been adopted and implemented by recent and popular file sharing systems such as FastTrack (Kazaa)[34], BitTorrent[35] and eDonkey (eMule)[36]. The basic concept is that an overlay network is built on top of the existing physical network in order to support balanced distribution of keys and efficient search of content location (structured resource location). The existing physical network consists of the actual Internet topology, with physical links among Internet hosts and routers, while the overlay network is a virtual network composed of the Internet nodes running the file sharing application, whose topology is built based on the distribution of keys related to the name of the resource. The many existing systems based on DHT mainly vary with respect to the different organization and maintenance of the overlay network, the searching and retrieval of keys in the overlay network, but also the policies applied in content downloading in terms of distribution of chunks, bandwidth assigned to a peer according to its reputation, and so on.

13.2 Opportunistic communications

Opportunistic networking is an emerging communication paradigm that has been gaining greater popularity in the research community in the last few years. The following are some of the basic features that an opportunistic network scenario can include:

- Large network: an opportunistic network is often composed of a large number of nodes.
- Heterogeneous communication technologies: any single node could be equipped with multiple communication interfaces.
- Intermittent behavior of nodes: nodes can alternate periods of activity to periods in which they are not available, either because they are switched off or because they are out of order/out of reach.
- Tolerance to large delays: opportunistic network applications implicitly have to admit and accept unbounded or, anyway, extremely large latencies
  - Delay Tolerant Network – DTN – is one of the possible declinations of opportunistic networking; equivalently, opportunistic network is one type of DTN.
- Nodes mobility capabilities: nodes are endowed with mobility capabilities, although the specific mobility pattern of the whole network population or of a single node is generally unknown. Opportunistic mobile networks benefit from all data transport opportunities, including the mobility of the users themselves.

In accordance with the features listed above, a reference scenario for opportunistic communications is represented by a network of people in a large city. This concrete example is particularly suitable to introduce opportunistic networking thanks to its simplicity, its popularity and the interest it is receiving. In fact, it is frequently reported as a case-study and as a real common application scenario, and the topic of people networks is at the crossroads of several networking research themes.

---

[33] Chord project, http://pdos.csail.mit.edu/chord
[34] Fasttrack, http://developer.berlios.de/projects/gift-fasttrack/; Kazaa ,www.kazaa.com
[35] Bittorrent website, http://www.bittorrent.com
[36] Bittorrent website, http://www.bittorrent.com; eMule homepage, http://www.emule-project.net/home/perl/general.cgi?l=1

Large network. Every person equipped with a PDA (or a similar device), living, travelling and working within a city can be associated to a node of a large network, being the number of nodes linearly proportional to the number of the city inhabitants. This is a reasonable view since in the last few years there has been a great increase in the number of small devices such as PDAs, laptops and smart phones carried by ordinary people, indistinctly belonging to any urban social class.

Heterogeneous communication technologies. Such devices commonly offer multi-technology radio communication capabilities: besides the native connectivity to cellular networks, they are often equipped with radio chips enabling short-range wireless networking capabilities such as, for instance, Bluetooth and WiFi.

Nodes mobility capabilities. Mobility is inherently associated to a node in such a type of network since people commonly spend a fraction of their ordinary day moving: e.g., from home to the working place (and vice versa), to reach places where they engage in spare time activities, hobbies or fixed habits (purchases, health, …). In addition, mobility patterns in networks of people are partially predictable: firstly, because the sequence of movements from one place to another is often repetitive during (working) days; secondly, because most of such movements always occur approximately in the same time ranges and involve almost fixed time periods, mainly when scheduled public transports are utilized.

Intermittent behavior of nodes. This kind of behavior is easily explained by considering that the electronic devices taking part in a network of people are powered by finite capacity batteries so that a power recharging period is usually required from time to time. In addition, the person carrying the device can autonomously decide to switch it off any time, even when batteries are not exhausted (e.g., at night).

Opportunistic networks of people are also called Pocket Switched Networks (PSNs) (Pietilainen and Diot 2009), the name being a play on "packet switched networks", the fundamental design principle of the Internet architecture. The reference to the term "pocket" comes from the place where people usually put their radio-enabled devices while carrying them.

These networks can be compared to the well-known MANETs (Mobile Ad-hoc NETworks), infrastructure-less networks in which at the same time every node is a source, a destination and a router. However, the key difference is that PSNs are essentially disconnected most of the time and no route can be efficiently constructed among any two nodes. Indeed, differently from MANETs, opportunistic networks do not assume that there exists an end-to-end connectivity between source and destination nodes. While MANETs rely on end-to-end routing protocols such as AODV and DSR (and many others), in opportunistic networks the data is delivered through one hop data transmission opportunistically exploiting node encounters, intermediate node storage and intermediate node mobility. For this reason, taking inspiration from the original "Store-and-Forward" strategy of Internet packet switching, in the literature the "Store-Carry-Forward" scheme is now associated to the opportunistic communication paradigm.

Definitively, even if they are not necessarily completely infrastructure-less (which is the case for MANETs), opportunistic networks focus on mobile ad-hoc DTNs, where routes between source and destination are dynamically built and any possible intermediate node can be used opportunistically to ferry data. In this perspective, opportunistic networks can be interpreted as an evolution of MANETs.

The research issues related to opportunistic networking concern many different aspects, including:

- Nodes mobility prediction and characterization;
- Inter-nodes time contacts characterization;
- Onboard nodes storage resources management.

Usually, nodes belonging to opportunistic networks are severely constrained in terms of storage capacity. While replication of contents during nodes contacts is the most efficient strategy to increase the overall content delivery success rate and to reduce end-to-end delays, assuming that nodes have infinite storage capacity is, unfortunately, not realistic. An effective policy to choose if and when to forward and/or to replicate contents has to be based on a certain optimal trade-off between final content delivery success probability and the amount of storage resources used.

In this respect, several history-based policies have been proposed that exploit statistics about previous contacts among pairs of nodes, which are elaborated and kept up-to-date locally at each node. Frequency and duration of contacts are primarily processed to predict the expected end-to-end delivery delays and therefore used to decide whether to forward the content to a possibly more reliable intermediate node or not. The forwarding policy may also take into account additional metrics, such as the amount of residual energy left onboard the nodes. Energy availability is, in fact, another critical parameter in opportunistic networks. A complete solution dealing with routing in opportunistic networks is CHARON (Soares, Franceschinis et al. November 2010).

Recently, following the success of social networks, social network structure of human has been explored for unicast routing in opportunistic networks (Musolesi and Mascolo 2006). The basic idea is that, in the case of networks of people, social relationships generally vary much more slowly than the underlying network topologies; therefore, more efficient, reliable and scalable forwarding decisions can be taken on the basis of the node's social relationships rather than considering controlled replication and delivery probabilities. This line of research appears promising if we consider that social networks exhibit the small world phenomenon (i.e., individuals are often linked by a short chain of acquaintances).

So far we have mostly focused our attention to networks of people. Nevertheless, several other scenarios exhibit the basic features of opportunistic and delay tolerant networks. The following examples of different contexts, along with related projects, represent a non-exhaustive though significant overview.

Opportunistic ad-hoc networking can provide intermittent Internet connectivity to rural and developing areas. Rural communications represent a challenging scenario in emerging countries like India or Africa where the lack of an existing infrastructure makes communications almost impossible and building an infrastructure from the scratch is a prohibitive, too expensive, investment.

The aim of DakNet project[37] is to design a low-cost asynchronous ICT infrastructure in order to provide connectivity to rural villages in India. In each village, static information kiosks have to be built. They must be equipped with storage capacity and short range wireless interfaces. When mobile access points, mounted on buses travelling between villages and towns, reach the village, kiosks are able to download and upload information to these mobile access points. Using the Store-Carry-Forward paradigm, the information from the town is ferried by the buses to the villages, where local users can download information from kiosks. Similarly, the buses also ferry information from the villages to the town in the same Store-Carry-Forward way.

Through another program, called SARI (Sustainable Access in Rural India)[38], almost 100 rural Internet kiosks were disseminated in the Madurai area of Tamil Nadu.

Finally, according to the Computers On Wheels (COW) project, a set of motorcycles equipped with an Internet-connected laptop travel between very remote villages to collect requests for Internet access and support users communications during the limited time the motorcycle stops at the village.

Wild-life monitoring is another interesting application of opportunistic communications. Specimens of wild animals, suitably provided with a radio tag with sensing and storage capabilities, represent mobile nodes of a network typically deployed on very vast areas. Various radio tags carried by animals measure environment data and send it to a sink node connected to the Internet. Since animal mobility is unpredictable and the area of mobility is large, opportunistic networking is the only efficient and really applicable approach to allow researchers understand remotely animal behaviors and interactions, while keeping disturbance and intrusiveness at minimum. Additionally, researchers can learn how human activities affect the ecosystem. The opportunistic networking solution is cost-effective: indeed, animals are remotely tracked and studied by researchers who are not constrained to physically pursue them, which would result very costly. Instead, they can live in static settlements where they can save much time and invest it more efficiently. Two examples of

[37] DakNet project, http://www.daknet.net
[38] SARI project, http://edev.media,mit.edu/SARI

wild life monitoring are Shared Wireless Infostation Model (SWIM) (Small and Haas 2003) and the ZebraNet project[39].

13.3 Merging Peer-to-Peer and Opportunism: towards Opportunistic Peer-to-Peer communication paradigm

In previous sections some basic concepts have been introduced, defined and discussed. Fundamental keywords of Section 13.1 are file sharing systems and Peer-to-Peer distributed architecture while Section 13.2 focuses on opportunistic communication, delay tolerant applications and node mobility. Since Peer-to-Peer and opportunism features are expected to characterize next generation networks, understanding the process through which they will be merged and made co-exist is of paramount importance to predict how networking will evolve in the near future.

This process is, to some extent, already undergoing nowadays. For instance, opportunistic Peer-to-Peer file sharing application such as BlueTorrent that supports only Bluetooth enabled devices, are emerging. Moreover, although musical and cinematographic contents are still the predominant type of content being shared and exchanged in the Internet, user generated electronic contents are becoming increasingly available. Audio and video programs, as well as published news and photographs, traditionally produced by small groups of professionals, can now be easily originated by ordinary users thanks to the incredibly fast technology evolution. In addition, such users are more and more frequently mobile users. Indeed, the extraordinary success of social networks such as Facebook and Myspace, video and photo displaying systems like Youtube and Flickr, and other new platforms for the generation and the sharing of contents and information as Wikipedia, thematic blogs and podcasts witness that the amount of data created by a larger number of users is becoming larger than the data produced by smaller groups of professionals.

Beside the facts that can be directly observed in the continuous evolution of technology and ICT world, the academic world has started to research these emerging communication scenarios where node mobility, opportunism and Peer-to-Peer architectures co-exist.

A significant example is represented by the PodNet[40] project, focused on wireless ad-hoc podcasting for podcasts sharing and distributing. The system leverages on both traditional infrastructure-based networks and more challenging infrastructure-less networks associated with short-range wireless communication technologies. Local cache of nodes is also considered to enable the system functioning. In PodNet, encounters with mobile devices carried by people are exploited for data dissemination purposes, and the focus is not on unicast routing but on broadcasting the data to a group of destinations. PodNet heavily uses costless short-range wireless communication, in contrast with expensive and capacity constrained cellular networks and WLAN networks, to bring user-generated content service and bulk content distribution. The routing strategy works as follows: instead of routing the content directly to destinations, first the content is replicated at intermediate nodes using a solicitation protocol at application layer, then it is implicitly routed to destinations thanks to node mobility and node relaying. An original feature in PodNet architecture is the absence of network layer; in fact, application and transport layers are implemented directly on top of the MAC layer.

An important contribution is provided by (Helgason, Yavuz et al. 2010) in which the authors propose a middleware architecture for a mobile Peer-to-Peer content distribution system. Nodes mobility, absence of infrastructure, wireless contents dissemination and opportunistic contents exchange are ingredients of the investigated topic and of the proposed system. A publish/subscribe procedure is at the basis of any service access.

In the same field, the Delay Tolerant Network (DTN) Architecture[41] was previously proposed. DTN conceives an overlay layer (the bundle layer) logically operating above the transport layer whose goal is to deliver data units (the bundles) from a sender to a receiver using different transport protocols and exploiting opportunistic connectivity so that nodes store, carry and forward bundles to cope with link outages.

[39] Zebranet project website, http://www.princeton.edu/mrm/zebranet.html
[40] PodNet, http://podnet.ee.ethz.ch/
[41] DTN Architecture RFC, http://tools.ietf.org/html/rfc4838

Haggle communication architecture (Haggle documentation) is another interesting proposal strongly related to PSNs introduced in Section 13.2. It deals with mobile devices, such as smart phones and PDAs, carried by people and studies how to forward messages by opportunistically exploiting nodes contacts. As opposed to PodNet, in Haggle the routing is of unicast nature. In Haggle, mobile devices are the main actors on which the architecture is built and the primary goal of Haggle is to separate application functionalities from the underlying technologies in order to guarantee seamless operation even in case of unavailability of network connectivity due to sudden disruptions. Haggle is thus much more than a simple protocol architecture; its contribution is the deep design of node such that nodes and applications are able to adapt to the network connectivity level. Indeed, Haggle is founded on a data-centric architecture where applications ignore the details of the mechanisms transporting data to the right place. Haggle high level architectures performs the task of propagating data in such a way that applications can automatically take advantage of any connection opportunities that arise, both local neighborhood opportunities and connectivity with servers on the Internet when available. The technical solution for all this is obtained by providing late binding of network interfaces, protocols and names.

Another system showing some similarities with Haggle and the DTN Architecture is Cimbiosys[42], a platform providing content synchronization and replication through opportunistic Peer-to-Peer communication. Content-based filters are utilized to specify which contents are synchronized and shared by the system.

13.4 Summary

Peer-to-Peer is a distributed computing architecture where a group of objects (or peers) can *exchange resources directly with each other* without the support of a centralized entity managing and organizing communication. features of Peer-to-Peer systems which are relevant to the development of intelligent service structures include: (1) peers acts simultaneously as producers and consumers of resources; peers communicate autonomously and directly with each other; (2) peers act in a globally cooperative fashion; (3) resources are identified uniquely (i.e., need for unique identifiers) while resource providers are not (i.e., need for a centralized or distributed resource location service). The solely problem of P2P network is that peers are dynamically enter and leave network this causes problem when a peer is leaving while another peer needs the its resources. Opportunistic communication enhances P2P by trying to establish transmission when there is an opportunity. Examples of previous sensor networks using opportunistic paradigm include Zebranet, PodNet, and Haggle.

13.5 Bibliography

Haggle documentation. from ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/fire/projects-haggle_en.pdf.

Helgason, Ó. R., E. A. Yavuz, et al. (2010). A Mobile Peer-to-Peer System for Opportunistic Content-Centric Networking. Mobiheld. New Delhi, India.

Musolesi, M. and C. Mascolo (2006). A community based mobility model for ad hoc network research. Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality. Florence, Italy, ACM.

Pietilainen, A. K. and C. Diot (2009). Social Pocket Switched Networks. INFOCOM Workshops 2009, IEEE.

Rowstron, A. and P. Druschel (2001). Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems Proc. IFIP/ACM Middleware 2001.

Small, T. and Z. J. Haas (2003). The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). Proceedings of the 4th ACM international symposium on Mobile ad hoc networking \& computing. Annapolis, Maryland, USA, ACM.

[42] Cimbiosys project, http://research.microsoft.com/apps/pubs/default.aspx?id=56258

Soares, J., M. Franceschinis, et al. (November 2010). "Opportunistic Data Collection in Sparse Wireless Sensor Networks", Special Issue." EURASIP Journal on Wireless Communications and Networking(Opportunistic and Delay Tolerant Networks).

Zhao, B. Y., H. Ling, et al. (2004). "Tapestry: a resilient global-scale overlay for service deployment." Selected Areas in Communications, IEEE Journal on **22**(1): 41-53.

14. Existing Messaging Formats and Protocols

14.1 Existing Formats for Low-Level Sensor-Device Communication

Interoperability is a major issue in heterogeneous Wired and Wireless Sensors and Actuators Networks, where devices produced by several manufacturers must interact with each other. Fostering and easing interoperability does not necessarily entail using a common, syntactically uniform protocol among all devices, but, at least some semantically compatible approaches must be followed with respect to representation of data and functionalities. In order to support the analysis of such common possible approaches for data and functionalities representation this section explores the different approaches already employed in existing technologies data exchange between different devices.

The following technologies are considered in the section: ZigBee Application Profiles including an overview of the ZigBee Standard (ZigBee Alliance October 2007) and the Application Profiles with their uses cases; ZigBee Smart Energy 2.0 Profile (ZigBee Alliance and HomePlug Powerline Alliance liaison December 2009) and how this profile runs over IP networks; Bluetooth Profiles(Bluetooth SIG July 2007) with an overview of the contents of the Application Profiles; LonMark Functional Profiles[43] including an overview of the different applications and functional profiles; KNX Profiles[44] together with an overview of the standard and profiles; ISOagriNET[45] with a focus on data formats and the procedures for the data exchange; IEEE 1451[46] family of standards including an overview of the architecture, the related standards and the software architecture; EtherNET/IP (ODVA 2008) including an overview of the standard, how the application data is managed and the description of some uses cases; CANopen[47] including an overview of the specification and of the different modes used to transfer data; MODBUS TCP[48] along with an overview of the conformance classes and the technology applications; PROFINET & PROFIBUS International Application Profiles[49] with an overview of PROFINET and PROFIBUS and the Application Profiles.

The analysis of all the considered technologies is also supported by a use-case example. In conclusion, a summary with an overview of the technologies mentioned.

14.1.1 ZigBee Application Profiles

ZigBee is a low-cost, low-power and low-data rate wireless communication standard developed by the ZigBee Alliance[50] and used in home and building automation, industrial controls, PC peripherals, health-care applications, and games. Smart devices adopting the ZigBee standard are able to cooperate and create an ad-hoc wireless network composed of nodes possibly equipped with sensors and/or actuators and programmed to accomplish a specific task.
The ZigBee specification, whose first release was published in December 2004, describes the complete protocol stack, organized into four fundamental layers: Physical (PHY) Layer and Medium Access Control (MAC) Layer, based on the IEEE 802.15.4-2003 (IEEE October 2003) standard, Network (NWK) Layer and Application (APL) Layer. In particular, the NWK layer supports networking capabilities and the creation of complex multi-hop topologies to extend the radio coverage of the wireless network. The APL layer hosts the applications logic.

To exchange data at Application Layer, ZigBee defines the *ZigBee Cluster Library* (ZCL) (ZigBee Alliance October 2007) which is a collection of pre-defined application messages, also called *clusters*. In particular, a cluster specifies the *attributes* and *commands* defining a communication interface to a specific application level functionality.

[43] LonMark, http://www.lonmark.org
[44] KNX, http://www.knx.org/
[45] ISOagriNET ,http://www.isoagrinet.org/
[46] NIST IEEE-P1451, http://ieee1451.nist.gov/
[47] CANopen, http://www.can-cia.org/index.php?id=47
[48] MODBUS TCP, http://www.modbus.org/
[49] PROFINET & PROFIBUS International, http://www.profibus.com
[50] ZigBee Alliance, http://www.zigbee.org/

An *attribute* is a data entity referring to a specific functionality hosted by a ZigBee device. For instance, an attribute could represent the result of a *physical measurement* performed by a ZigBee device equipped with a temperature sensor or *status information* related to whether a ZigBee-enabled lamp has been switched on or off. Then, *commands* can be used to read/modify the value of the attributes, to perform a discovery of the attributes exposed within a cluster and even to receive asynchronous notifications of some attribute-related events. It is worth observing that the data types used for attributes are specified within the ZCL specification document.

Leveraging on ZCL, ZigBee describes a *device* in terms of *application level capabilities*. For instance, a "*On/Off Switch*" is a standard device able to send on/off and toggle commands to devices (e.g. a lamp) able to interpret such messages. A device usually supports a pre-defined list of clusters.
Moreover, ZigBee defines the *Application Profiles*. They represent a collection of device descriptions related to a particular application scenario and detail the specific features to be supported in the overall protocol stack. The Application Profiles are classified into *Public* and *Manufacturer-Specific*.

- **Public Profiles -** defined by the ZigBee Alliance to provide interoperability between devices from different vendors. The public profiles are the followings:
  - Telecom Services (ZigBee Alliance January 2010)
  - Health Care (ZigBee Alliance March 2010)
  - Home Automation (ZigBee Alliance February 2010)
  - Smart Energy 1.0 (ZigBee Alliance December 2008)
- **Manufacturer-Specific Profiles -** designed by vendors for specific applications that cannot be realized using public profiles.

In the following, a brief description of each public profile is reported.

Telecom Services. The scope of the profile is to provide a list of device descriptions and standard interfaces for applications enabling the delivery of a variety of services in a telecom environment. The main application domains where the profile is expected to be used include information delivery, location based services, peer-to-peer small data sharing, mobile commerce, mobile gaming, voice over ZigBee and chatting. Possible installation scenarios include a single room but also larger environments.

Healthcare. The profile was designed to operate in the followings application domains: Disease Management (Non-critical patient monitoring, Patient alarm monitoring and Drug administration), Personal Fitness (Monitoring / tracking fitness level and personalized fitness schedule) and Personal Wellness Monitoring (Activity monitoring, Safety Monitoring, Living independently). The installation scenarios considered for this profile include residential environments, retirement communities, nursing homes, medical care facilities and fitness centers.

Home Automation. This profile mainly refers to applications required in residential/small office environments: lighting, Heating, Ventilating and Air Conditioning (HVAC), environmental monitoring, energy management, safety, and security.

Smart Energy 1.0. This profile is focused on applications aiming to provide an effective energy consumption monitoring and to enable a more efficient energy usage. Applications of interest include metering, pricing, demand response and load control. Installation scenarios proposed within the profile range from a single home to an entire apartment complex.
The Fig. below presents a typical Smart Energy Profile scenario.

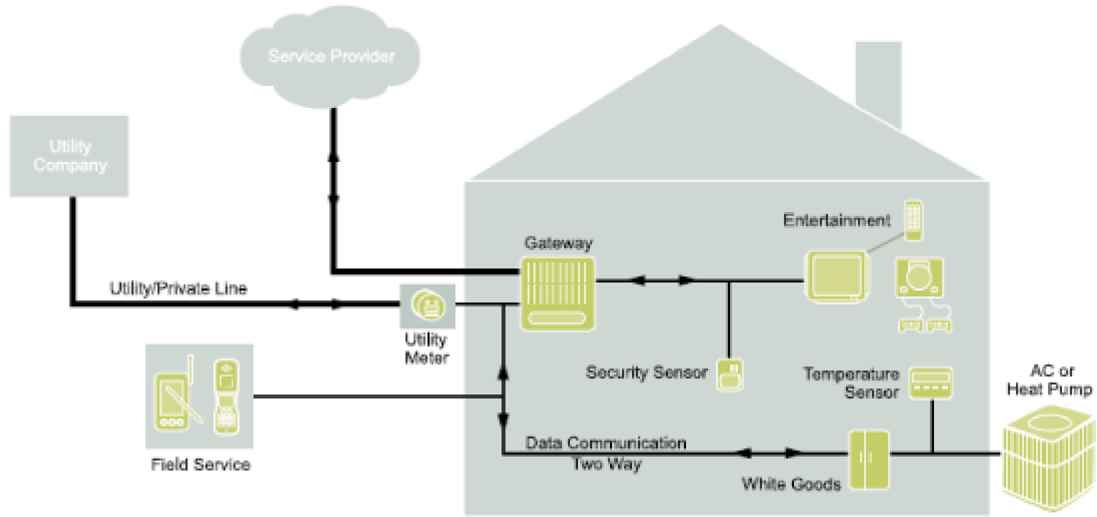


Fig. 14-1. Smart Energy scenario[51] - Copyright © Freescale Semiconductor, Inc

As far as the diffusion of ZigBee solutions in the market is concerned, we should mention that, at the moment, more than 300 companies around the world (especially in US) provide ZigBee-enabled products.

14.1.2 ZigBee Smart Energy 2.0 Profile

The Smart Energy Profile (SEP) 2.0 standard is being designed within a liaison between ZigBee Alliance and HomePlug Powerline Alliance[52]. The main objective is to define a "*networking and application integration platform* for messages" (ZigBee Alliance and HomePlug Powerline Alliance liaison December 2009) between *energy services providers* and *costumer devices* enabling the monitoring, the control and possibly the reduction of energy consumption.
The resulting solution should be able to run on any network relying on IPv6 protocol. In fact, different link layer technologies are expected to be supported including *IEEE 802.15.4 (IEEE 2006; IEEE October 2003)*, *IEEE 802.11 (IEEE June 2007)*, *IEEE 802.3* (IEEE December 2008), *Home Plug* (Power line communications [PLC]) (HomePlug Powerline Alliance December 2005) and *IEEE P1901* (High Speed PLC) (IEEE P1901). At the moment, only few preliminary documents have been published and relate to both technical (ZigBee Alliance and HomePlug Powerline Alliance liaison April 2010; ZigBee Alliance and HomePlug Powerline Alliance liaison December 2009) and marketing (ZigBee+HomePlug Joint Working Group March 2009) requirements.
In the technical documents available, requirements at different layers of the protocol stack have been discussed. In particular, the use of IEEE 802.15.4 is taken into account along with the adoption of the IETF 6LoWPAN[53] standard as the relevant adaptation layer.
Moreover, a first draft of the *application protocol* has been defined. SEP 2.0 basically adopts a *web paradigm*. A SEP 2.0 device can be considered as a server hosting specific *application level capabilities exposed as web resources*. A generic client is able to get, update, extend or delete the above resources. In particular, a client-server interaction model is used and is based on polling mechanisms. It is worth noting that each device can act both as a client and a server. Publish-subscribe solutions have been integrated as well. Accordingly, an event related to a specific "web resource" can be notified to all the interested clients.
The implementation of this web approach actually relies on a *HTTP-based RESTful architecture* where the application level functionalities are exposed as Uniform Resource Identifiers (URIs). The actions (i.e. get, update, extend or delete) on such resources can be performed through the well known *HTTP methods GET*, *PUT*, *POST* and *DELETE*. SEP 2.0 also proposes as an alternative solution to adopt the Constrained Application Protocol (CoAP), being defined within Core IETF WG.

[51] ZigBee Learning Center, http://www.freescale.com/webapp/sps/site/learning_summary.jsp? nodeId=052BDF2C4431B9
[52] HomePlug Powerline Alliance, http://www.homeplug.org/home/
[53] IPv6 over Low power WPAN (6LoWPAN), IETF WG, http://www.ietf.org/html.charters/6lowpan-charter.html

The *data models* used are defined in the International Electro technical Commission's (IEC) 61970-301 (IEC 2009) and 61968-11 (IEC 2009). The common information model 61970 is in charge of the application program interfaces for Energy management systems (EMS). The second information model 61968 provides a way to make easier the inter-application integration between distributed software application systems.

A major concept introduced in SEP 2.0, is the *Function Set*. A function set provides a set of resources and associated transactions. Each function set is assigned a resource name to be used in the resource discovery process. Currently, the defined function sets are:

- **Demand Response & Load Control –** provide methods for effecting reduction of energy use during peak load times.
- **Directed Messaging –** use for send messages to specific devices.
- **Public Messaging –** use for send messages to all devices.
- **Price –** provide pricing publication information.
- **Pre-Payment –** allow customers to pay for usage credit.
- **Metering –** use for send data from metering devices.
- **Mirroring –** push data from a metering device.
- **Plug-in Electric Vehicle –** combine many of other function sets like Demand Response, Price, Messaging, Metering, Mobile network authentication, Billing resolution in order to resolve the complex aspect of supplying electric cars.
- **Distributed Energy Resource Management –** provide a platform to support all types of energy sources.
- **Billing –** give consumers a financial view of their consumption.

The different devices defined within the profile include In-home display, Smart thermostat, Load control, Meter, Plug-In Electric Vehicle, Smart Appliance, Premises Energy Management System, Range Extender, Energy Services Interface, Pre-Payment Terminal, and Inverter.
In the following, an example is reported to describe how SEP 2.0 devices can expose their capabilities leveraging on the web paradigm.
A SEP 2.0-enabled device is given and equipped with an electricity meter, uniquely identified by Id 0. The electricity meter is actually considered as a resource and exposed using the URI below:

```
http(s)://(IPv6 Address)/mtr/0
```

By using the GET method on such URI, a generic client is able to obtain the description of the meter capabilities. The response is formatted as an XML message:

```
<?xml version='1.0' encoding='UTF-8'?>
<MeterReading href="http(s)://(IPv6 Address)/mtr/0/mr"/>
<Status href="http(s)://(IPv6 Address)/mtr/0/s"/>
<TariffProfile href="http(s)://(IPv6 Address)/mtr/0/tp"/>
```

The obtained XML message contains a list of URI that can be used to get further information from the specific meter i.e. the meter reading, the current status and the tariff profile adopted.

14.1.3 LonMark Functional Profiles

LonMark Functional Profiles are defined in the LonMark Interoperability Guidelines(Echelon September 2005) in order to provide a set of generic functions to implement applications in a LonWorks-based network. LonWorks is a network platform based on a communication protocol called LonTalk created by Echelon Corporation[54]. LonWorks is mainly used in control applications and for the automation of several functions (i.e. lighting and HVAC) in smart buildings.

The relevant communication protocol LonTalk is defined by the ANSI/EIA/CEA-709.1(CEA 2010) and EN 14908-1 (ISO/IEC 2006) specifications and defines the format of the messages to transmit between devices and the actions expected by a device when sends or receives and message.

In addition, LonMark Functional profiles - usually called LonMark profiles - describe the application layer interface, defining the network variables, configuration properties, and power-up behaviors. The profiles are used by devices for specific, commonly used control functions. Basically, a functional profile defines a template for a functional block. A functional block is in charge of receiving configuration and operational data inputs, processing the incoming data, and sending the operational data outputs. The network variables and configuration properties specified in each functional profile have the name of functional profile members. Finally, functional profile members can be mandatory or not according to the profile implemented.

At the moment, there are 65 different profiles related with the control of refrigerated display cases, control of lifts and elevators, access control and intrusion, fire systems, HVAC applications, standby power supplies, generic analogue inputs and outputs, lighting applications and generic sensors.

The following figure shows as example how the Switch Functional profile is organized.

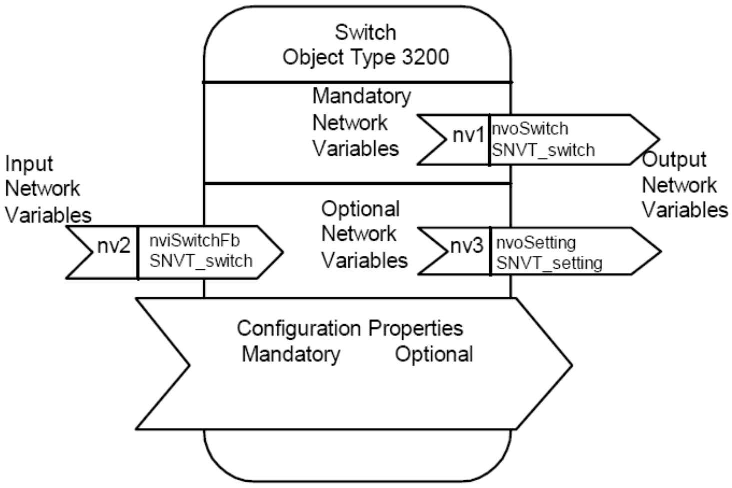


Fig. 14-2. Switch Functional profile - Copyright © LONMARK

Nowadays, more than 350,000 LONWORKS systems and over 100 million devices have been installed around the world.

14.1.4 KNX Profiles

KNX is a network communication protocol for intelligent buildings administrated by the KNX Association. It was standardized in CELENEC EN50090(CELENEC 2010), ISO/IEC 14543-3(ISO/IEC 2000), CEN EN 13321-1(CEN 2006) and GB/Z 20965(CNS 2007). Basically, KNX integrates three standards: European Home Systems Protocol (EHS), BatiBUS, and European Installation Bus (EIB).
It can operate over different communication technologies:

- Twisted pair (KNX TP)
- Powerline (KNX PL)

[54] Echelon Corporation, http://www.echelon.com/

- Radio frequency (KNX RF)
- IP/Ethernet (KNX IP)

The KNX profiles are described in the Volume 6 "Profiles" of the KNX specification. A profile groups a set of features in order to facilitate the design and certification of devices that will be part of the network.

KNX is implemented in home and building control applications including lighting, security control, heating, ventilation, air conditioning, monitoring, alarming, water control, energy management, metering as well as household appliances and audio.

There are almost 7000 KNX certified products provided by more than 200 companies worldwide. The KNX Association has partnership agreements with more than 30,000 installer companies in 100 countries and more than 60 technical universities as well as over 150 training centers.

The following picture shows an example of lighting control with the ABB i-bus KNX based on the KNX specification.

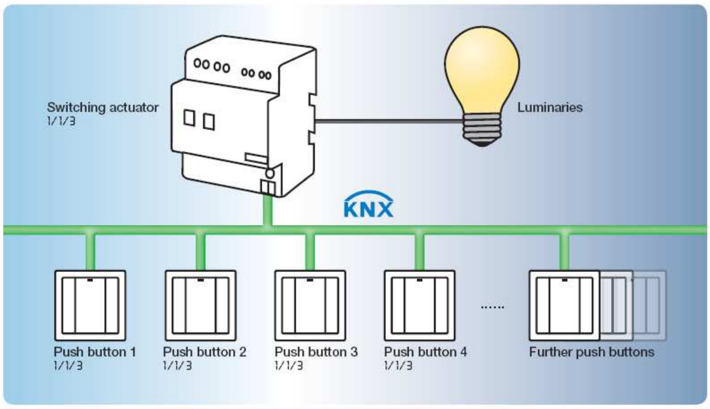


Fig. 14-3 X. ABB i-bus KNX Lighting Control - Copyright © ABB

14.1.5 Bluetooth Profiles

The Bluetooth standard was created in 1994 to replace cables and perform a quick, easy and secure communication between compliant devices. These devices exchange data over short distances forming a Personal Area Network (PAN). Several updates of Bluetooth specification have been released since 1994. At the moment, the most diffused one is the Bluetooth Core Specification Version 2.1 + EDR (Bluetooth SIG July 2007). However, the latest Bluetooth Core Specification is version 4.0: it has been completed on April 2010 and includes high speed and low-power features.
Within the core specification, profiles were created in order to provide interoperability between devices from different manufacturers. In particular, these profiles have been defined for PC peripherals, entertainment devices and mobile devices like cell phones, headsets and laptops. For instance, the Advanced Audio Distribution Profile (A2DP) is used for the wireless communication between a stereo headphone and a laptop or a cell phone.
The information contained in each profile is the following:

- User interface format
- Bluetooth protocol stack configuration
- Dependencies on other profiles

The way Bluetooth works at application layer varies depending on the profile used. For instance, A2DP adopts Audio/Video Distribution Transport Protocol (AVDTP) solution to support audio and/or video distribution connections and streaming of audio or video media over the Bluetooth air interface.
The most common profiles are the followings: Generic Access, Intercom, Generic Object Exchange, Serial Port, File Transfer, Object Push, Synchronization, Headset, PAN and A2DP.

The *interaction model* considered in Bluetooth profiles is Client-Server. For instance, in the Object Push Profile (OPP), two types of devices are defined:

- **Push Server** – device that contains objects;
- **Push Client** – device that pushes and pulls objects to and from the Push Server.

The objects represent a piece of information e.g. a business card or an appointment detail. In this context, a Push Client can push one or more objects to a Push Server but also initiate a function allowing to pull an object from the Push Server. The considered operations are described in the following image taken from.

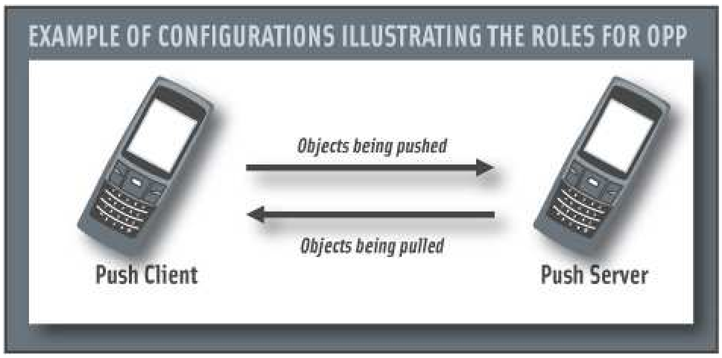


Fig. 14-4. Object Push Profile - Copyright © Bluetooth SIG

Nowadays, Bluetooth is being used in many products and application scenarios. Its main promoters are Ericsson, Intel, Lenovo, Microsoft, Motorola, Nokia, and Toshiba and have more than 13,000 Associate and Adopter member companies in the world.

14.1.6 ISOagriNET

ISOagriNET is an international standard based on ISO-2007 (ISO 2007) and it was developed to support data communication within farm networks, to provide data connectivity to Internet and to enable the implementation of demand-oriented services. In particular, the standard specifies the data formats and the procedures for the data exchange over an Ethernet network.
ISOagriNET uses two parallel data description languages to exchange data:

- **ADIS/ADED**
  - *Agricultural Data Interchange Syntax (ADIS)* - describes "how" data is transmitted. An ADIS file consists of ASCII characters (ISO 8-bit code) containing lines which represent the included data (Line Types) and its functions (Status symbol).
  - *Agricultural Data Element Dictionary Standardized (ADED)* – describes records of data to be transferred. They can be Data Elements (Basic element), a set of them (Data object (entity)) or Headers (Special Data Object).
- **XML/ADED**
  - *eXtensible Markup Language (XML)* - defines the set of rules for encoding documents in machine-readable form.

Nowadays, only few ISOagriNET-enabled products are available and are used for feeding, ventilation and management of pig housing (e.g. PIG NET[55]).

14.1.7 IEEE 1451

The IEEE 1451 is a family of Smart Transducer Interface standards developed to provide an interface between *transducers* (sensors, event sensors and actuators) and *microprocessors*, instrumentation systems, and control networks. The objective is to make easier the interface of transducers to systems and networks, leveraging on existing sensor and networking technologies. In order to reach this objective, the standard introduces an element called *Transducer Electronic Data*

[55] PIG NET, http://202.55.147.39/

Sheet (TEDS) that stores identification, calibration, correction data and manufacturer information for each transducer.

The general structure proposed is based in two main components:

Transducer Interface Module *(TIM):* it contains the transducer, TEDS and related features (signal conditioning, Analog-to-Digital and/or Digital-to-Analog conversion). Some of the TIM main functions include Analog Signal Conditioning, Triggering, Command Processing, TEDS Storage, and Data Transfer.

Network Capable Application Processor *(NCAP):* defines a gateway between the TIM and the network. Its responsibilities are Communications, Interface Control, Message Routing, Encoding and Decoding, TIM Discovery and Control and Interpretation of TEDS Data.

Inside this family of standards, only two are related to the application layer: **IEEE P1451.0** (IEEE September 2007) and **IEEE P1451.1** (IEEE June 1999).

IEEE P1451.0 defines the TEDS, operations and commands for the complete family of IEEE 1451 smart transducer standards. The objective is reach the information contained in any transducer in the network independently of the physical technology used to connect to the NCAP.

IEEE P1451.1 describes the transducers behavior defining a common object model. It defines also the measurement and communication models. The latter includes the client-server and publish-subscribe models.

The software architecture is based in three types of models:

- An object model

  It adopts a hierarchy of object classes divided in four categories:
  - Block classes performing data processing.
  - Component classes encapsulating data.
  - Service classes handling inter-NCAP communications and system-wide synchronization.
  - Non-IEEE 1451.1 object classes satisfying particular requirements of a given application

All Blocks, Components, and Services are also Entities. Generic attributes like Object ID and Object name, are provided by the entity abstract class for all 1451.1 objects that are visible within the network.

- A data model

  The data model defines the data types used to exchange information between Entities inside the NCAP process and across the network. The included primitive data types and arrays of these types are: Boolean, Octet, Integer, Floating Point and String.
- Two network communications models

  Client/Server model for one-to-one communication. Fig. 14-5 (taken from(IEEE June 1999)) shows the object organization in a client/server model.

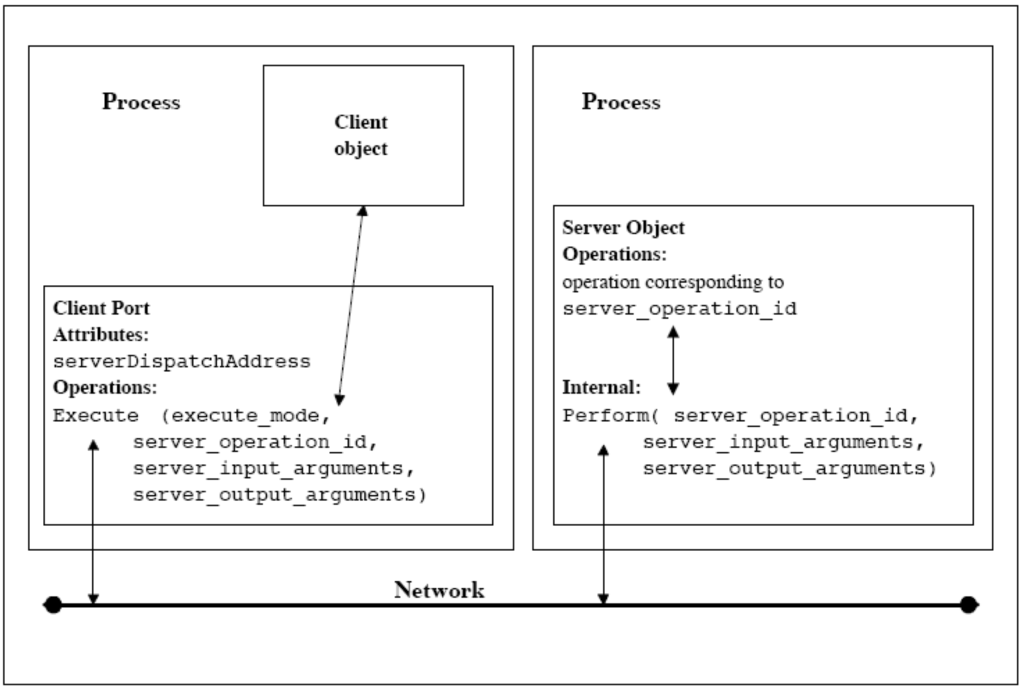


Fig. 14-5.Client-server communication components - Copyright © IEEE

To invoke an operation on a Server Object, the Client Object utilizes an execute operation. Then, the server object uses the perform operation and returns the output results to the client object.

- *Publish/Subscribe model for one-to-many communications*.

Another important standard is IEEE P1451.5 (IEEE October 2007) defining the transducer-to-NCAP interface and TEDS for wireless transducers. The supported communication protocol standards are IEEE 802.11, IEEE 802.15.1 and IEEE 802.15.4.

As mentioned before, the family of standard IEEE 1451 was developed to make possible the interconnection between transducers and networks. The adoption of 6LoWPAN is IEEE 1451 is described in (Higuera and Polo) as a potential use case. In addition, a real implementation of a Wireless Environmental Monitoring System is described in (Kang and Song 2006).

14.1.8 EtherNet/IP

EtherNet/IP is the name given to the **Common Industrial Protocol** (CIP), as implemented over standard Ethernet (IEEE 802.3 and the TCP/IP protocol suite). EtherNet/IP was introduced in 2001 and is managed by Open DeviceNet Vendors Association (ODVA).

CIP is a media independent, connection-based, object-oriented protocol designed for automation applications. It provides a unified communication architecture including a set of communication services for automation applications. This allows the integration of user's applications with the Internet and Ethernet networks. It also defines *devices profiles* and *application objects* that describe behaviors and interfaces enabling an end-to-end communication between devices.

The devices at the CIP application layer are represented by an *object model*. The objects are defined as a group of data and how this data behaves. In fact, the standard considers three groups of objects: *Application*, *Network-specific* and *Communication*. Moreover, the objects can be divided in three different classes:

- **Required Objects** - included in all CIP devices. They contain the Identity Object, the Message Router Object and network-specific objects.
- **Application Objects** - describe how the devices encapsulate the data. They are specific to the Device Type and function.

- **Vendor-specific Objects** - describe specific services related to a manufacturer. They are optional.

  The following picture taken from (ODVA 2008) shows the **CIP Object-oriented view of application data.**

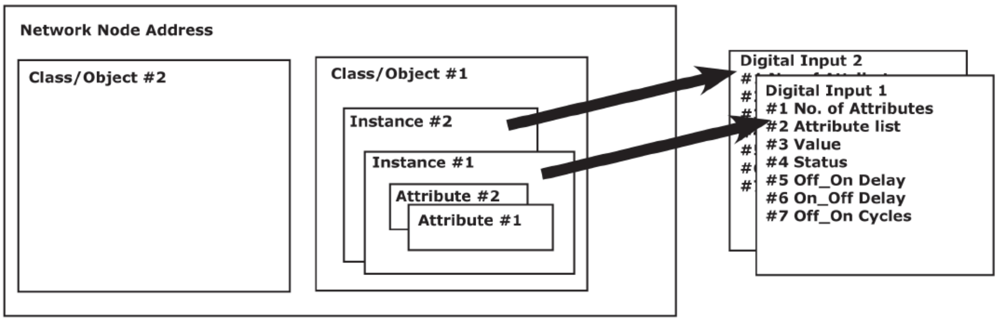


Fig. 14-6. CIP Object-oriented view of application data - Copyright © ODVA

A class is a set of objects that represent the same type of system components. An object instance is when there is more than one 'copy' of an object within a device. This is referred as an object class. A set of attributes is within each instance of the object class. This set is the same for each instance except for the values contained.
In order to access data within a device, the address information usually used is: Device network address, Class ID, Instance ID, Attribute ID and Service code.
In fact, the main purpose of EtherNet/IP is to provide communication between industrial control systems and their components, such as a programmable automation controller, programmable logic controller or an I/O system. At the moment, is commonly used for industrial automation settings using Rockwell Automation's Allen-Bradley-brand control systems. Also other manufacturers like Phoenix Contact, Opto 22 and WAGO Corporation provide EtherNet/IP complaint devices.
Fig. 14-7 shows an EtherNet/IP Typical Configuration used for Rockwell Automation devices.

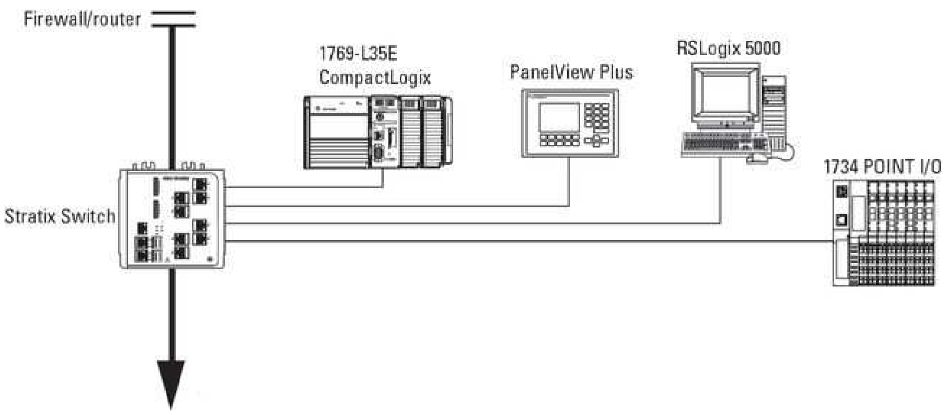


Fig. 14-7. EtherNet/IP Network with complaint devices - Copyright © Rockwell Automation(EESTI 2003)

14.1.9 CANopen

CANopen is a specification for distributed industrial automation systems based on Controller Area Network (CAN). It was developed by CAN-in-Automation (CiA) and is standardized by CENELEC EN 50325-4 (EESTI 2003). The CANopen standard describes the exchange of data in a CAN-based network. This standard defines both the fundamental communication mechanisms (communication profile) and the devices functionalities (device profile). Moreover, an *object dictionary* is specified and is used to offer a standardized access to the "Application Objects" of a device. The functionality and characteristics of a CANopen device are described using an Electronic Data Sheet (EDS).
Basically, the specification provides two different ways to transfer data:

- *The **Service Data Objects** (SDO)* based on a client-server communication model. It is used for configuration of a device, and upload and download of large data blocks. Direct addressing of an object is possible using its index and sub index but requires additional protocol overhead.
- ***Process Data Objects*** *(PDO)* provides an efficient transmission of data according to a producer-consumer model using no additional protocol overhead. The producer-consumer model refers to send data using broadcast. The PDOs can be asynchronous which are event-controlled and are the most common or synchronous which are only transmitted after the reception of a synchronization message.

The following list shows some of the CANopen profiles and gives an idea where CANopen is applied:

- CiA 401: Device profile for generic I/O modules
- CiA 402: Device profile for drives and motion control (servo controller, stepper motor controller, frequency inverter)
- CiA 404: Device profile for measuring devices and closed-loop controllers
- CiA 434: Profiles for laboratory automation systems
- CiA 445: Device profile for RFID reader
- CiA 447: Application profile for special-purpose car add-on devices
- CiA 453: Device profile power supply

As CANopen is based on CAN-based industrial and embedded system solutions, it was broadly accepted and considered a dominating standard in Europe.
The following picture taken from Fig. 14-8 shows some of the devices (PLC's, control panels, drive controllers) that can be connected using CANopen.

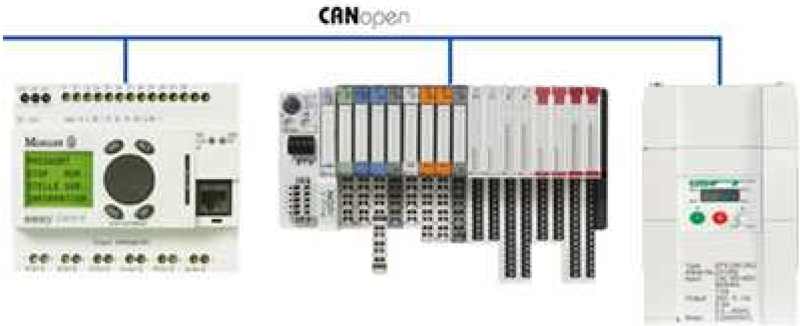


Fig. 14-8. CANopen compliant devices interconnected - Copyright © EATON

14.1.10 MODBUS TCP

MODBUS TCP is based on the MODBUS family. It was created in order to provide the possibility to send MODBUS messages over an Internet environment utilizing TCP/IP. It is usually adopted in industrial environments.
The protocol is very simple and consists in encapsulating MODBUS frames into TCP frames. Transactions are connection-oriented, meaning that every query expects a response.
The messages sent are grouped in "conformance classes" which make a difference between the optional messages and the ones that are universally implemented.
The classes are the following:

- **Class 0 -** Minimum set of functions.
- **Class 1 -** Additional set of functions which are commonly implemented and interoperable.
- **Class 2 -** These are the data transfer functions needed for routine operations such as Human Machine Interface (HMI) and supervision.

The structure of the request and response body remains the same as in MODBUS.
The main use of MODBUS TCP is to connect PLC's and I/O modules to Ethernet networks. For instance, this allows performing maintenance and repair on remote devices or remotely manage geographically distributed systems.

MODBUS TCP has become a de facto standard communications protocol in industry, and is one of the most commonly used standards to connect industrial electronic devices. The main reason for the extensive use of MODBUS is its openness, simplicity, low cost development, and minimum hardware required to support it.
The following picture taken from Fig. 14-9 shows and example of MODBUS TCP configuration.

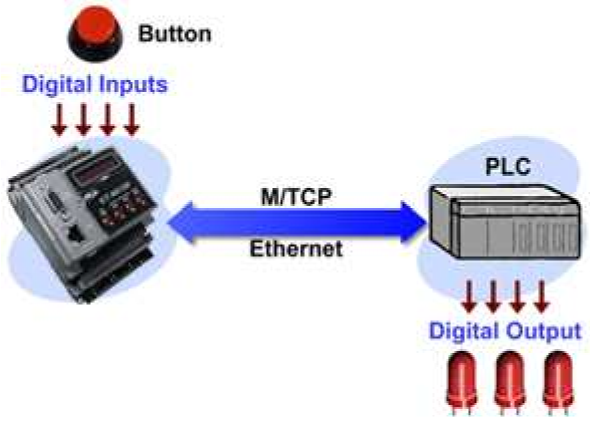


Fig. 14-9. Ethernet network using MODBUS TCP - Copyright © ICPDAS

14.1.11 PROFINET & PROFIBUS International Application Profiles

PROFINET&PROFIBUS International (PI) has specified several Application Profiles for use in PROFIBUS and PROFINET systems.

PROFIBUS is a standard used in automation environments for field bus communication. It is implemented on layers 1, 2 and 7 of the ISO OSI model. At layer 7 (Application), PROFIBUS provides diverse service levels like cyclic exchange of data and diagnosis, acyclic and cyclic data exchange and alarm handling, and isochronous mode and data exchange broadcast. Currently, there are two versions of PROFIBUS: PROFIBUS DP used for factory automation and PROFIBUS PA used for process automation. This technology has the 20% market share (more than 30 million nodes) for field busses.

PROFINET is an automation standard compatible with Ethernet and it uses UDP/IP in the higher layers for demand-oriented data exchange. The minimum data communication requirements provided by the standard are the following: 100 Mbps data communication; Full duplex transmission; Switched Ethernet; Autonegotiation and Autocrossover and Wireless communication (WLAN and Bluetooth). It also provides a "proxy" solution enabling the integration of other networks, especially fielbuses like PROFIBUS DP. The technology is supported by more than 50 different companies worldwide and installed in more than 2 million nodes.

The Application Profiles reside at application layer. PI defines them as specifications of certain properties, performance features, and behavior of devices and systems. Generally, these profiles are referred to specific devices and are divided in two main groups: General Application Profiles and Specific Application Profiles. The current Application Profiles available for PROFIBUS and PROFINET are PROFIsafe, PROFIdrive, PROFIenergy, Encoder, Low Voltage Switch Gear, and Identification Systems. PROFIsafe and PROFIdrive are standardized in the respective specifications: IEC 61784-3 (IEC 2010) and IEC 61800-7(IEC 2007).

The following picture shows an example of the profiles used in a hybrid scenario.

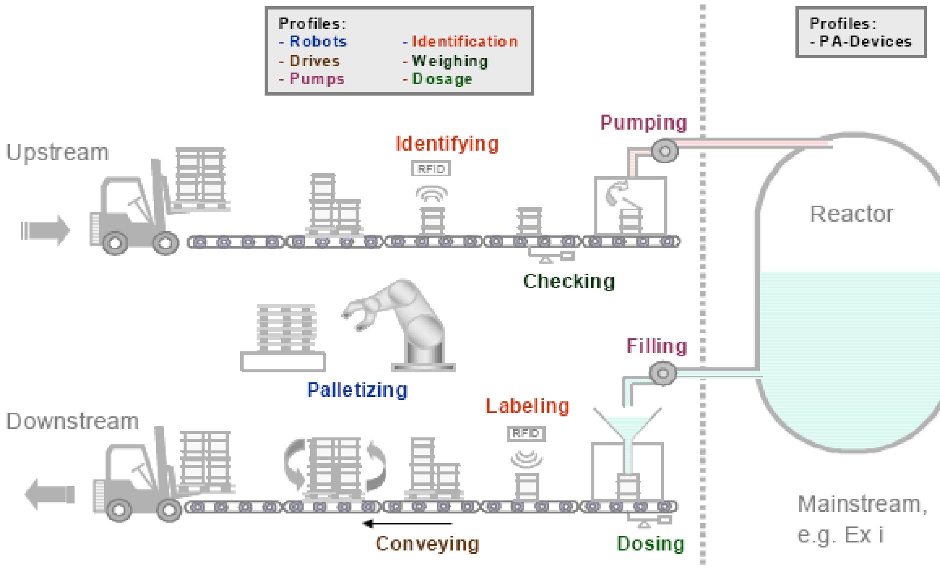


Fig. 14-10 PI devices profiles used on an hybrid application - Copyright © PI

14.1.12 Other Technologies used in manufacturing scenario

Another point to take into account is that there are some technologies that do not provide an application layer leaving the work to software developers. This is the case of IEEE 1394 (IEEE October 2008) where only the two layers of the protocol stack are specified. The interface is best known as FireWire and is mainly used by personal computers and digital audio/video but also in the automotive industry. In (Dallemagne and Ruiz 2000), the authors propose the development of an application layer based on FireWire bus and aimed for industrial purposes.
Another example is IEEE 488 (IEEE December 2003) that was created without defining an application layer. It is a short-range communication bus mainly used for instrumentation equipments but also as computer interface. In order to provide a common command structure and data formats for measurement devices, Standard Commands for Programmable Instruments (SCPI) (SCPI Consortium May 1999) was designed. It can also be used with RS-232 and the Universal Serial Bus(USB) besides being originally created for IEEE 488.

14.2 Messaging Format for High-Level Communication

14.2.1 JSON

According to ECMA 3rd edition, *JavaScript Object Notation (JSON) is a text format for the serialization of structured data. It is derived from the object literals of JavaScript.* JSON has been widely used for exchanging data in many programming languages such as C#, Java, PHP as well as for different purposes such as REST, feeds. *(Crockford 2006)* JSON supports four primitive types (strings, numbers, booleans, and null) and two structured types (objects and arrays) whereas a string consists of Unicode characters. An object is formed by a pair of curly brackets surrounding members that are defined by name-value pairs. Arrays are represented by square brackets that have elements separated by commas. Supported numbers are only integer and float. Strings are represented with quotation marks.

Example:

This is a JSON object:

```
{
"Image": {
"Width":  800,
```

```
"Height": 600,
"Title":  "View from 15th Floor",
"Thumbnail": {
   "Url":    "http://www.example.com/image/481989943",
   "Height": 125,
   "Width":  "100"
},
"IDs": [116, 943, 234, 38793]
```

14.2.2 XML Protocol

XML Protocol (XMLP) [10] from W3C provides simple protocols that can be ubiquitously deployed and easily programmed through scripting languages, XML tools, interactive Web development tools, etc. The goal is a layered system which will directly meet the needs of applications with simple interfaces (e.g. validateCreditCard), and which can be incrementally extended to provide the security, scalability, and robustness required for more complex application interfaces. The XML Protocol Working Group is chartered to design the following four components:

1. An envelope for encapsulating XML data to be transferred in an interoperable manner that allows for distributed extensibility.
2. A convention for the content of the envelope when used for RPC (Remote Procedure Call) applications. The protocol aspects of this should be coordinated closely with the IETF and make an effort to leverage any work they are doing, see below for details.
3. A mechanism for serializing data representing non-syntactic data models such as object graphs and directed labelled graphs based on the data types of XML Schema.
4. A mechanism for using HTTP transport in the context of an XML Protocol. This does not mean that HTTP is the only transport mechanism that can be used for the technologies developed, nor that support for HTTP transport is mandatory. This component merely addresses the fact that HTTP transport is expected to be widely used, and so should be addressed by this Working Group. There will be coordination with the Internet Engineering Task Force (IETF), see Blocks Extensible Exchange Protocol (BEEP)[56] .

Furthermore, the following two general requirements must be met by the work produced by this Working Group:

- The envelope and the serialization mechanisms developed by the Working Group may not preclude any programming model nor assume any particular mode of communication between peers.
- Focus must be put on simplicity and modularity and must support the kind of extensibility actually seen on the Web. In particular, it must support distributed extensibility where the communicating parties do not have a priori knowledge of each other.

XMLP was used as a model for analyzing and evaluating protocols for Web Services and resulted in the endorsement of SOAP.

Simple Object Access Protocol

The Simple Object Access Protocol (SOAP)[57] by W3C organization[58] is XML-based protocol for exchanging information in a decentralized, distributed environment (soap can be viewed as the successor of XML-RPC protocol[59]). SOAP supports different styles of information exchange, including:

[56] W3 org at http://www.w3.org/2001/12/semweb-fin/w3csw

[57] Web Service Semantics (WSDL-S), http://www.w3.org/Submission/WSDL-S/

[58] Semantic Web Services and processes. http://lsdis.cs.uga.edu/projects/meteor-s/

[59] http://www.xmlrpc.com

- Remote Procedure Call style (RPC), which allows for request-response processing, where an endpoint receives a procedure oriented message and replies with a correlated response message.
- Message-oriented information exchange, which supports organizations and applications that need to exchange business or other types of documents where a message is sent but the sender may not expect or wait for an immediate response.

SOAP is independent of protocol, language, platform and operating system and it has support for messages incorporating attachments (using the multipart MIME structure).

A SOAP message consists of a SOAP envelope that contains two data structures, the SOAP header and the SOAP body, and information about the name spaces used to define them:

- The SOAP header is optional. When the header is presented, it provides the information about the request defined in the SOAP body, for example: authentication transactional, security, contextual, user profile information, data encoding, or how a recipient of a SOAP message should process the message.
- The body contains a web service request or reply to a request in XML format. These messages are usually defined using the WSDL specification.

SOAP is typically defined over “firewall friendly” protocols such as HTTP and SMTP. SOAP can be used to exchange complete documents or to call a remote procedure.

Asynchronous Application Service Protocol for SOAP

The purpose of the OASIS[60] ASAP (oasis-open June 2004) is to create a very simple extension of SOAP that enables generic asynchronous web services or long-running web services. It is intended to integrate asynchronous services across the Internet and provide for their interaction. The integration and interactions consist of control and monitoring of the service. The protocol is intended to be lightweight and easy to implement, so that a variety of devices and situations can be covered.

14.2.3 Messaging Format in Representational State Transfer

Representational State Transfer (REST) (Fielding) is an architecture style for distributed networked systems such as www. REST is an architectural style, not standard. REST is intended to evoke an image of how a well-designed Web application behaves: a network of web pages (a virtual state-machine), where the user progresses through an application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use.

A REST web service can be viewed as a SOA based on the concept of resource. Resources are the sources of specific information, each of which can be referred to using a global identifier (an URI). In order to manipulate these resources, components of the network (clients and servers) communicate via a standardized interface (e.g. HTTP) and exchange representations of these resources (the actual documents conveying the information). Any number of connectors (e.g., clients, servers, caches, tunnels, etc.) can mediate the request, but each does so without “seeing past” its own request. Thus an application can interact with a resource by knowing two things: the identifier of the resource, and the action required. The application does, however, need to understand the format of the information (representation) returned, which is typically an HTML, XML, JSON document of some kind, although it may be an image or any other content.

14.2.4 Web Distributed Data Exchange

Web Distributed Data eXchange (WDDX) from OpenWDDX.org is a technology for exchanging complex data structures between programming languages. It has been designed with web applications in mind. WDDX consists of a standard for language-independent representation of

[60] http://www.oasis-open.org/home/index.php

instantiated data based on XML 1.0 and a set of serializer/deserializer modules for every language/technology that uses WDDX.

14.2.5 Data Format in Semantic web

In attempt to respond to this situation, the term "Semantic Web" was coined by Tim Berners-Lee and his colleagues (Berners-Lee et al 2001) referring to a "web for machines" as opposed to a web to be read by humans. In their understanding "*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.*"

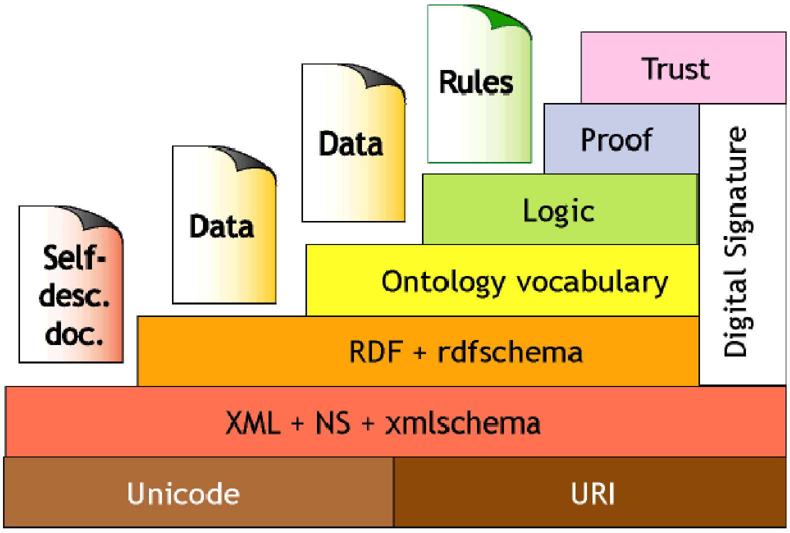


Fig. 14-11 Semantic web layers (from [61])

The Semantic Web is the opportunity for providing, finding and processing information via the Internet with the help of machines (and mostly also for machines) which are capable of dealing with the semantics of the information. The idea is to transform information into something meaningful to actors who seek to enhance their knowledge in order to satisfy a specific concern or accomplish a specific task related to their particular context.

To fulfill the promises and enable semantic technologies to work, there must be a knowledge model (of some part) of the world that is used to provide meaning to information to be processed within an application. The knowledge model has the form of a semantic model which differs from other kind of models (Semantic Technology March 2004):

- Meaning is represented through connectivity. The meaning of terms, or concepts, in the model is established by the way they connect to each other.
- A semantic model expresses multiple viewpoints.
- Semantic models represent knowledge about the world in which systems operate and are shared across applications.
- Several interconnected models could be used to represent different aspects.
- Use of a model is often referred to as "reasoning over the model". The reasoning can range from a very simple process of graph search to intricate inference.

Although the role of a semantic model can be played by a simple taxonomy, nowadays use of semantically richer ontologies (ontological models) dominates.

Although most common definition states that "*An ontology is a specification of conceptualization*", more detailed definitions can make things a bit clearer. One of them states that "*The subject of an ontology is the study of the categories and things which exist in some domain. The product of such*

[61] W3 org at http://www.w3.org/2001/12/semweb-fin/w3csw

study, called an ontology, is a catalogue of the types of things that are assumed to exist in a domain of interest from the perspective of a person who talks about the domain using some language". From the practical point of view, an ontology is a network of connections defining explicit relationships (named and differentiated) between concepts. New knowledge can be derived by examining the connections between concepts. Simple ontologies are just networks of connections, richer ontologies include rules and constraints governing these connections.

The Semantic Web principles are implemented in the layers of Web technologies and standards (see Fig. 14-11). The most common ontology languages are briefly described below (all the presented languages are supervised by the World Wide Web Consortium (W3C).

RDF

RDF (Resource Description Framework) is a standard way for defining simple descriptions. RDF supports semantics - a clear set of rules for providing simple descriptive information. RDF enforces a strict notation for the representation of information, based on resources and relations between them. The RDF data model provides three object types: resources, properties, and statements. Resource may be an object; property is a specific aspect, characteristic attribute, or relation used to describe a resource; statement is a triple consisting of two nodes and a connecting edge. The strength of the language is in its descriptive capabilities, but it still lacks some important features required in an ontology language such as inferences for example. However, ontology languages built on top of RDF as a representation and description format.

RDF Schema

RDF Schema (RDFS) enriches the basic RDF model, by providing a vocabulary for RDF, which is assumed to have certain semantics. Predefined properties can be used to model instance of and subclass of relationships as well as domain restrictions and range restrictions of attributes. Indeed, the RDF schema provides modeling primitives that can be used to capture basic semantics in a domain neutral way. That is, RDFS specifies metadata that is applicable to the entities and their properties in all domains. The metadata then serves as a standard model by which RDF tools can operate on specific domain models, since the RDFS meta-model elements will have a fixed semantics in all domain models.

OWL

OWL is the newest W3C recommendation for ontology definition. OWL enhances RDF vocabulary for describing properties and classes: relations between classes (e.g. subclasses), cardinality, equality, richer typing of properties, characteristics of properties (e.g. symmetry) and instances. OWL is quite a sophisticated language. The most important feature is its capability for description logic (DL) reasoning (Description Logics are a family of logic-based knowledge representation formalisms designed to represent and reason about the knowledge of an application domain in a structured and well-understood way). The OWL language also provides three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full, each offers a different level of expressiveness at the trade-off for simplicity, thus offering a suitable sub language parts available for use according to expressivity needs.

An important constituent of the semantic web is represented by inference engines. Their aim is to reason over ontological models to prove statements or to deduce new knowledge from already explicitly presented knowledge. They are expected to make explicit those facts that are present in the ontology only implicitly. The reasoning over ontology can have the following purposes:

- *Validation*. Validating ontology means ensuring that the ontology is a good representation of the domain of discourse that should be modelled. Reasoning is extremely important for validation. For example, checking whether an ontology is internally consistent is crucial: obviously, no inconsistent theory can be a good representation of domain.
- *Analysis*. In analysis one assumes that the ontology is a faithful representation of the domain, and tries to deduce facts about the domain by reasoning over the ontology. In a

sense of trying to collect new information about the domain by exploiting the ontology. Obviously, analysis can also provide input to the validation phase.

- *Harmonization*. Myriads of ontologies can be used within the semantic web environment. Since each ontology represents a particular point of view, using different ontologies to represent meaning of information within a domain of interest results in mismatches in understanding and dealing with this information. In order to avoid it, semantic mappings between ontologies must be done.

At the beginning, the idea of the semantic web tried just to enhance the current version of the web. It started out with a document oriented approach. The basic idea was to make web pages identifiable by computers as information resources carrying not only information (readable only by humans) but the meaning of this information as well. The meaning was added by annotating these pages with semantic mark-up. Ontologies here define a shared conceptualization of the application domain at hand and provide the basis for defining metadata, that have a precisely defined semantics, and that are therefore machine-processable. The idea of semantically annotated web pages with machine-interpretable description of their content aimed at automated processes of searching and accessing pages enabling human users to better utilise information stored on the web. In addition to human users, the semantic web enables the participation of non-human users as well. These machine agents do not need to deal with whole web pages. Instead of this, they exchange chunks of data with each other. Although they can communicate using different protocols, technology of web services has become a dominant way of communication with and using services of applications in the web environment.

Formerly, the problem of interoperability of different agents was tackled by translation technologies, most commonly by field to field mapping. The semantic web enables agents to exchange chunks of data with meaning associated to the data using semantic technologies. Advanced applications can use ontologies to relate the information to a semantic model of a given domain. In this way semantic technologies offer a new way to integrate different applications. Nowadays, the field of semantic interoperability is the most addressed problem connected with the idea of the semantic web.

14.3 Summary

Messaging data formats are important to ensure syntactic compatibility. Many approaches have been implemented as standard communication protocols in different network layers and application oriented profiles. Chapter 14.1 has provided an overview of existing technologies used to support data exchange in sensor and actuator networks. The installation scenarios considered for ebbits can range from a single room in home environment to a big industrial factory. Some of the considered technologies i.e. ISOagriNET, IEEE 1451, EtherNet/IP, MODBUS TCP, PROFINET and ZigBee Smart Energy 2.0 can be adopted in IP-based networks. This can be considered as an interesting feature since it enables an easier integration with the whole Internet. Table 14-1 provides a summary of the main characteristics identified for the single technologies analyzed along the chapter.

On the higher level of network where computers are mainly used, Communication for enterprise applications is dominated with the use of web services that use SOAP as messaging protocol and WSDL to describe services. REST provides a lightweight alternative solution that utilizes HTTP features. REST uses several forms of messaging formats such as JSON, ATOM, and XML. Several works have taken advantage of semantic web to elevate interoperability to semantic level. The standard formats to define semantic web includes RDF and RDFS to define its semantics, and OWL which is an extension of RDFS.

ebbits may introduce new measuring devices (e.g.: 6lowpan based sensors and actuators) for manufacturing and food production scenarios that have not yet been standardize, thus it is necessary to adopt and extend the existing communication standards for new devices introduced in ebbits to ensure that new applications developed on ebbits platform are interoperable.

Table 14-1. Summary of communication protocol standard for low level devices

Solution	Zigbee Application Profiles	Smart Energy 2.0 Profile	Bluetooth Profiles	LonMark	KNX	ISOAgriNET	IEEE 1451	EtherNet/ IP	CANopen	MODBUS TCP	PROFINET	PROFIBUS
Dependence on Technology[1]	★★★	★	★★★	★★★	★	★★	★	★★	★★★	★	★	★★★
Adopted communication technologies	Zigbee	IP/802.11, 802.15.4 (6LoWPAN), 802.3	Bluetooth	LonWorks	IP/802.3	IP/802.3	IP/802.11, 802.15.4 (6LoWPAN), 802.3	IP/802.3	CAN	IP/802.11, 802.3	IP/802.3, 802.11, FieldBus	FieldBus
Reference Application Scenarios	Heath Care, Telecom Services, Home Automatio n, Smart Energy	Smart Energy	Wireless Personal Communication, Heath Care	Home Automation	Home Automation	Farm Automation	Industrial Automation	Industrial Automation	Industrial Automation	Industrial Automation	Industrial Automation	Industrial Automation
Diffusion in the market[1]	★★ (US)	N/A	★★★	★★	★★	★	★	★★	★★	★★	★★	★★
Data Types	Complex	Simple	Simple	Simple	Simple	Simple	Complex	Simple	Complex	Simple	Simple	Simple
Interaction Model	Client-Server	Publish-Subscribe	Client-Server	Client-Server	Client-Server	Client-Server	Client-Server / Publish-Subscribe	Client-Server	Client-Server	Client-Server	Client-Server	Client-Server

[1]Note ★ = Low ; ★★ = Medium; ★★★ = High ; N/A = Not Applicable(Standard not yet released)

14.4 Bibliography

Berners-Lee et al (2001). "The semantic web - A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American." Scientific American **284**(5).

Bluetooth SIG (July 2007) "Bluetooth Core Specification 2.1 + EDR."

CEA (2010) CEA-709.1 Standard, Control Network Protocol Specification.

CELENEC (2010) CELENEC EN 50090 Home and Building Electronic Systems (HBES)

CEN (2006) EN 13321-1 Open data communication in building automation, controls and building management - Home and building electronic system - Part 1: Product and system requirements.

CNS (2007) GB/Z 20965 The control network home and building electronic system (HBES) technical standard.

Crockford, D. (2006). The application/json media type for javascript object notation (json), RFC 4627, July 2006.

Dallemagne, P. and L. Ruiz (2000). An application layer for using Firewire in industrial applications. Factory Communication Systems, 2000. Proceedings. 2000 IEEE International Workshop on.

Echelon (September 2005). LONMARK® Application-Layer Interoperability Guidelines

EESTI (2003) "EESTI Std. CENELEC EN 50325-4, Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces - Part 4: CANopen."

Fielding, R. Representational state transfer (REST). Chapter 5 in Architectural Styles and the Design of Networkbased Software Architectures, Ph. D. Thesis, University of California, Irvine, CA, 2000.

Higuera, J. and J. Polo Understanding the IEEE 1451 standard in 6loWPAN sensor networks. Sensors Applications Symposium (SAS), 2010 IEEE.

HomePlug Powerline Alliance (December 2005) "HomePlug AV baseline specification."

IEC (2009) "IEC 61968 Application integration at electric utilities - System interfaces for distribution management- Part 11: Common Information Model (CIM)."

IEC (2009). IEC 61970: Energy Management System Application Program Interface (EMS-API) - Part 301: Common Information Model (CIM) Base.

IEC (2010) IEC 61784-3 Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.

IEC (2007) IEC 61800-7 Adjustable speed electrical power drive systems - Part 7-1: Generic interface and use of profiles for power drive systems - Interface definition.

IEEE (2006) "IEEE Std. 802.15.4-2006, IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area etworks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)."

IEEE (December 2003) "IEEE std. 488.1- IEEE Standard for Higher Performance Protocol for the Standard Digital Interface for Programmable Instrumentation."

IEEE (December 2008) "IEEE Std. 802.3, IEEE Standard for Information Technology--Telecommunications and Information Exchange Between Systems--Local and Metropolitan

Area Networks-Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Section One, ."

IEEE (June 1999) "1451.1-1999 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators."

IEEE (June 2007) "IEEE Std. 802.11, IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ."

IEEE (October 2003). "IEEE Std. 802.15.4-2003, IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)."

IEEE (October 2007) "1451.5-2007 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators."

IEEE (October 2008). "IEEE std. 1394 IEEE Standard for a High-Performance Serial Bus."

IEEE (September 2007). "1451.0-2007 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators."

IEEE P1901. ""Draft Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications"." from http://grouper.ieee.org/groups/1901/index.html.

ISO (2007) "ISO 17532-2007, Stationary equipment for agriculture - Data communications network for livestock farming."

ISO/IEC (2006) DIS 14908-1 Standard, Open Data Communication in Building Automation, Controls and Building Management -- Control Network Protocol -- Part 1: Protocol Stack.

ISO/IEC (2000) 14543-1 **Information technology -** Home electronic system (HES) architecture

Kang, L. and E. Song (2006). A Wireless Environmental Monitoring System Based on the IEEE 1451.1 Standards. Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE.

oasis-open (June 2004). Asynchronous Service Access 3 Protocol (ASAP) Version 1.0.

ODVA (2008) "EtherNET/IP Quick Start for Vendors Handbook ".

SCPI Consortium (May 1999) "Standard Commands for Programmable Instruments (SCPI)."

Semantic Technology. (March 2004). "Semantic Technology. TopQuadrant Technology Briefing." from http://www.topquadrant.com/tq_white_papers.htm.

ZigBee Alliance (December 2008) "ZigBee Smart Energy Profile Specification - Revision 14."

ZigBee Alliance (February 2010). "ZigBee Home Automation Public Application Profile."

ZigBee Alliance (January 2010). "ZigBee Telecom Applications Profile Specification."

ZigBee Alliance (March 2010). "ZigBee Health Care Application Profile - Revision 25."

ZigBee Alliance (October 2007) "ZigBee Cluster Library Specification."

ZigBee Alliance (October 2007) "ZigBee Specification - Revision r17."

ZigBee Alliance and HomePlug Powerline Alliance liaison (April 2010) "Smart Energy Profile 2.0 Application Protocol Specification, ZigBee Smart Energy Profile 2.0 Technical Requirements Document."

ZigBee Alliance and HomePlug Powerline Alliance liaison (December 2009) "ZigBee Smart Energy Profile 2.0 Technical Requirements Document - Technical Requirements Document."

ZigBee+HomePlug Joint Working Group (March 2009) "Smart Energy Profile Marketing Requirements Document (MRD)."

15. Conclusion

As introduced in section 2, ebbits deals with a massive and complex heterogeneous network of devices, people, and services as such envisioned in IoPTS. Ensuring such a complex will run, intelligence in each different level of the network is needed. Intelligence comprises of ability to infer information out of data, awareness, autonomy, and collaboration. In WP5, these characteristics are represented by major tasks such as multi-sensor fusion, control management and context modeling. Many works has been done in the internet for business domain, nonetheless they do not provide solution that can satisfy ebbits vision. This overview is very useful to give insight of the present technology and the technology that the future needs.

15.1 Intelligent sensor fusion services

We conclude that JDL Model is able to give ebbits a basic reference model to develop intelligent sensor fusion services since it offers a generic view of sensor-fusion or information fusion in general that have been successfully used in many fields including military and non-military applications. It offers fusion levels that are connected through a BUS, which similar to SOA. However, it is a very abstract model that could lead to various interpretations. Thus, in ebbits we will specify the levels in the model more details that answer to ebbits needs. Many techniques have been used in different levels of the model are discussed in section 6 and section 5.3. including inference algorithms, sensor languages. Ebbits also envision mobile environment, in which sensors cannot be installed at fixed position thus various techniques of mobile sensing is also discussed in section 6. Ebbits will provide a framework in which, these techniques can be used as extensions for different fusion purposes since there is no algorithm that can satisfy all kind of needs. Ebbits will also extend JDL model specifically to provide reference for a massive distributed architecture to achieve this we discussed several promising techniques that could be used for populating distributed knowledge that include techniques from database and semantic web.

15.2 Control management

Control management that is envisioned in ebbits is a machine's capability to take decision and manage it resources based on current and past knowledge that it has obtained from the environment. Resources include sensor and actuator modules, energy supply, and network resources. Ebbits will use and extends feedback control theory by allowing sensor fusion and control system interacting in different levels as suggested in many standard industrial control systems that has been discussed in section 8. Ebbits will also conduct a feasibility study to use an automatic planning from AI for the control system. We have also reviewed several examples of using control management to influence the environment such as controlling container climate when transporting food, redirecting cattle's movements through auditory feedback. An example of intelligent control management that is needed in ebbits is to control production speed based on current energy policy.

Power conservation in WSN is an important aspect that has to be taken account into. Hence, ebbits will also use resource management to optimize power management in the wireless sensor network to maximize the network lifetime as discussed in section 8.3.1. Several algorithms will be evaluated in simulation as well as real world test-bed.

15.3 Semantic and Context modeling

Context information in ebbits plays important role to create intelligent services that are aware of the environment and of the internal system condition. Context information is used to indicate the environment as well as the effect of the system to it and thus can be an indicator for evaluating system performance. Context in muss be modeled according to the application domains thus in ebbits we will develop context models for various purposes such for energy consumption measurement, for feeding system in farms, for tracing food. As described in section 10, these models will consist of context model for application, devices and semantic relationship among them. Context information of the device will model the data acquisition processes in terms of device's events and data pooling that will be used by multi sensor components as described by level 1 of the

JDL model. Semantic model will describe the location and environmental condition related to the application needs (e.g.: temperature, humidity). And the application context will describe high level events specific to the application domain that could be references for fusion processes in the level 2 and 3 of the JDL model. Context model is also used by control framework as knowledge and a reference, based on which the decision is made.

15.4 Collaboration

Collaboration happens in different levels of the ebbits network. In a high level network that consist of resourceful devices (e.g. desktop computers, servers, and gateways) ebbits will make use of hydra and service oriented architecture. We will use and extend hydra P2P network infrastructure with e.g.: opportunistic communication discussed in section 9 to prevent any message loss when the network becomes unreliable. Discovering devices and services is important to initiate collaboration among them. Ebbits will take advantage of semantic resolution based on machine understandable semantic models. This is able to provide flexibility in discovering massive amount of devices and services in dynamic environment based on their capabilities and to allow machines discover each other automatically.

In the sensor network level, where sensor nodes communicate in ad-hoc manner, syntactic interoperability needs to be solved. Ebbits envision in the future of car manufacturing and food production life cycle IPv6 based wireless sensor and actuator nodes (6lowpan) are going to be used more and more. There exists however no definition of standard services as such owned by Bluetooth. Thus ebbits will define a set of services for both domain based on existing standard explored in section 14.1.

16. List of Figures.

17. List of Tables