DISSERTATION

Automation System Perception
First Step towards Perceptive Awareness

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften unter der Leitung von

O.Univ.Prof. Dipl.-Ing. Dr.techn. Dietmar Dietrich
Institut für Computertechnik – TU Wien

und

Dr. Keith Dimond
University of Kent at Canterbury

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

Von

Clara Tamarit Fuertes
Matr.-Nr. 9926455
Vorgartenstraße 129-143/3/10
1020 Wien

Wien, Juli 2003 ....

Automation System Perception

First Step towards Perceptive Awareness

Automation System Perception
First Step towards Perceptive Awareness

Abstract

The term ‘perceptive awareness’ emerges from the search of a bionic solution to better the behaviour of common automation systems. Based on conscious human behaviour, this work focuses on designing a new automation model, an aiding tool when designing networked automation systems. This new model has been called the perceptive awareness model (PAM). PAM meets the requirements that have appeared in home and building automation during the last years, and which cannot be fulfilled with the existent design tools such as the ISO/OSI model.

Extending the ISO/OSI model, PAM does not just attend to the pure communication between devices but, going a step forward, it covers the high data processing that is required in automation to reach the preventive behaving of the system. The new model is designed in compatibility to present automation technologies and reference tools that are used to implement common automation systems. Additionally the model supports the integration of other kinds of technologies that contribute to a better perception of the environment such as visual and acoustic pattern recognition solutions. In such a way, PAM introduces a new concept in automation based on principles from nature.

Instead of just attending to particular inputs, as it is the case of common automation systems, humans behave by considering the whole situation. In order to equip automation systems with this ability, the new model supports first the collection of large amounts of different data from the environment. Secondly, the model enables the processing of this data in a way that allows the system to recognise and preventively react to the perceived situation not isolated but in the current context.

Once the perceptive awareness model has been designed, implementation efforts concentrate on the faculty of perception. Perceptive systems like visual pattern solutions using IEEE 1394, sound recognition systems like Voice Extreme from Sensory and fieldbus networks such as LonWorks, which are suitable to measure, detect, and monitor different parameters of the environment, are leaded into a ‘team work’.

Automation System Perception
First Step towards Perceptive Awareness

Kurzfassung

Der Begriff ´perceptive awareness´ entstammt der Suche nach einer bionischen Lösung zur Verbesserung des Verhaltens automatischer Systeme. Basierend auf bewusstem, menschlichem Verhalten konzentriert sich diese Arbeit auf das Design eines neuen Automationsmodels, eines Hilfsmittels für die Entwicklung vernetzter automatischer Systeme. Dieses neue Model wird ´perceptive awareness model´ (PAM) genannt. Im Bereich der Haus- und Gebäudeautomation treten vermehrt Anforderungen auf, die mit existierenden Hilfsmitteln wie zum Beispiel dem ISO/OSI Model nicht erfüllt werden können. Hier bietet PAM einen neuen Lösungsansatz.

Ausbauend auf das ISO/OSI Model behandelt PAM nicht nur die reine Kommunikation zwischen den Einheiten, sondern geht noch einen bedeutenden Schritt weiter: PAM umfasst die komplette Verarbeitung von Informationen um ein präventives Verhalten des Systems zu erreichen. Eine entscheidende Voraussetzung für das Model ist die Kompatibilität zu heutigen Technologien in der Automation und zu Tools für die Implementierung dieser Systeme. Darüber hinaus unterstützt das Model die Integration zusätzlicher Applikationen wie zum Beispiel Bild- oder Geräuscherkennung um eine verbesserte Wahrnehmung der Umgebung zu erreichen. Daher bietet PAM ein neues, auf Prinzipien der Natur beruhendes Konzept in der Automation.

Im Gegensatz zu heutigen Automationssystemen, die lediglich bestimmte Inputs beachten, berücksichtigen Menschen die gesamte Situation. Um ein ähnliches Verhalten in automatischen Systemen zu erreichen, besteht der erste Schritt des neuen Modells darin, eine große Menge an Daten über die Umgebung zu sammeln. Diese Informationen werden dann in mehreren Schritten verarbeitet, wodurch eine Erkennung der Situation, und in weiterer Folge, präventives Verhalten erreicht werden.

Nach der Erstellung des Modells für perceptive awareness, liegt der Fokus der Arbeit auf der Wahrnehmungsfähigkeit des Systems: Bildverarbeitungssysteme mit IEEE 1394, Geräuscherkennung mit Voice Extreme von Sensory, und Feldbusse wie LonWorks zum Detektieren und/oder Steuern verschiedener Parameter der Umgebung werden in das Gesamtsystem integriert.

Automation System Perception
First Step towards Perceptive Awareness

Abstracto

El término ‘perceptive awareness’ resulta de la búsqueda de una solución biónica para mejorar el comportamiento de sistemas automáticos comunes. Basado en el comportamiento consciente del hombre, este trabajo se centra en el diseño de un nuevo modelo en automatización, una herramienta de ayuda para el diseño de systemas automáticos en red. Este nuevo modelo se ha llamado ‘perceptive awareness model’ (PAM). PAM cubre los requerimientos que han aparecido en la automatización de edificios y viviendas (domótica) durante los últimos años, los cuales no pueden ser cuviertos con las herramientas de diseño existentes, como por ejemplo el modelo ISO/OSI.

Extendiendo el modelo ISO/OSI, PAM no atiende simplemente la comunicación pura entre componentes sino que, dando un paso adelante, cubre el complejo procesamiento de datos que es requerido en automatización para alcanzar el comportamiento preventivo del sistema. Adicionalmente el modelo permite la integración de otro tipo de tecnologías que contribuyen a mejorar la percepción del medio, como son soluciones para el renocimiento de patrones ópticos y acústicos. De este modo, PAM introduce un nuevo concepto en automatización basado en principios de la naturaleza.

En lugar de simplemente atender a inputs particulares, como es el caso de los sistemas automáticos comunes, el hombre se comporta considerando la completa situación. Para equipar los sistemas automáticos con esta habilidad, el nuevo modelo permite primero la colección de una gran cantidad de diferentes datos del medio. En segundo lugar, el modelo hace posible pensar en un modo de procesar los datos que le permite al sistema reconocer y reaccionar de forma preventiva frente a la situación percibida dependiendo del contexto.

Una vez diseñado el modelo, la implementación se centra en la facultad de percepción. Sistemas perceptivos como por ejemplo reconocimiento de patrones ópticos utilizando IEEE 1394, sistemas de reconocimiento de sonidos como Voce Extreme de Sensory, y redes de buses de campo como LonWorks, los cuales son propios para medir, detectar, y observar diferentes parámetros del medio, son conducidos hascia un ‘trabajo en grupo’.

Automation System Perception
First Step towards Perceptive Awareness

Preface

During the last years, automation has started to find its acceptance and applicability in the home area and with more intensively in the building area. An important contribution to this fact has been the development of fieldbus technologies. Nowadays, lighting, heating, ventilation and air condition (HVAC), sunblind, security systems, energy management systems, and many more applications of this kind can be easily automated by one or another automation technology.

Though much effort has been expended to develop helping tools like the ISO/OSI model to design fieldbus technologies following a kind of standard, there are new requirements in automation that demand an upgrade of these tools. Nowadays it is not enough to attend to aspects such as required data communication and data transmission like the OSI model does. Automation claims for a helping tool to integrate the different existent automation technologies. This tool has to enable an easy cooperative work between these systems and also between applications that will be shortly demanded at homes such as those of video and audio.

Conceptually independent of reference tools and existent technologies there is another important aspect to attend, which is related to the behaviour of the automation system. Analysing conscious human behaviour we see that it is preventive. On the contrary, common automation systems behave depending on specific inputs and following fixed rules. These systems do not consider any other parameters of the environment, which may also be interesting to react in one way or in another to the momentary experienced situation, neither validate reactions before executing them, which may entail undesired side effects sometimes.

In this work the term perceptive awareness is defined on the basis of some aspects of human conscious behaviour. This research works on benefit of a situation-dependent behaviour of the automation systems that enables the system to be conscious of the global momentary situation and consequently capable of selecting the better reaction while considering its side effects before embarking upon it. During the first part of the work the perceptive awareness model is designed to extend the possibilities of automation systems. The second part concentrates on implementing the faculty of perception of the SmartKitchen perceptive awareness automation system (PAAS) on the basis of the designed PAM. The SmartKitchen PAAS has been implemented at the kitchen lab of the Institute of Computer Technology (ICT) at Vienna UT.

The implementation of a perceptive awareness automation system is completed with the work presented by Gerhard Russ in his dissertation: ‘Situation-dependent behaviour in building automation’. His work covers the recognition of the perceived situations and selection of the proper responses in front of them.

Automation System Perception
First Step towards Perceptive Awareness

Acknowledgements

I am indebted to Prof. Dr. Dietrich, who motivated me to start my PhD studies at the Institute of Computer Technology and together with Dr. Dimond gave me the feedback and advices to elaborate a consistent work.

I thank my colleague Gerhard Russ for the pleasant teamwork during the last two years and his patience and support especially during the programming time.

I express special thanks to my family, especially to my parents, for supporting me in every step of my educational career, and to my boyfriend, Jürgen, whose tenderness held me up in those moments of weakness.

In addition I want to thank Markus Falkner, Merfat el Mansit and Peter Wistuba for his help during the implementation phase, and Peter Skiczuk and Markus Vincze for their explanations concerning pattern recognition.

Index

1 INTRODUCTION 1

1.1 TECHNOLOGY PROGRESS 1
1.1.1 Electrical and Automation Evolution 1
1.1.2 Computer Science 3
1.1.3 Process Automation 6

1.2 LEARNING FROM NATURE: BIONICS 9
1.2.1 Biological and Technical Systems 9
1.2.2 Automation Systems and Nervous System 12
1.2.2.1 Systems' Structure and its Evolution 12
1.2.2.2 Human Behaviour: Cognition 14

1.3 THE NEW CHALLENGE: PERCEPTIVE AWARENESS SYSTEMS 16
1.3.1 Perceptive Awareness: from Biology into Automation 17
1.3.2 Extension of the Automation Model 21

2 STATE OF THE ART 25

2.1 HOME AND BUILDING AUTOMATION 25
2.1.1 Evolution in Home and Building Automation 25
2.1.2 Principal Technologies in Home and Building Automation 30
2.1.2.1 EIB/KNX Protocol 30
2.1.2.2 LonTalk Protocol 31
2.1.2.3 BACnet Protocol 32
2.1.2.4 Backbone and Internet Connection 33
2.1.3 Interoperability in Home and Building Automation 35

2.2 CONSCIOUSNESS AND TECHNOLOGY 36
2.2.1 Basic Theories 36
2.2.2 Artificial Consciousness Models 38

3 PERCEPTIVE AWARENESS MODEL 51

3.1 SCOPE OF PAM 52

3.2 BEGINNING AND FRUITION OF PAM 54
3.2.1 Inter-Industries Function 54
3.2.2 Layers Resemblance 58
3.2.3 Composing Scenarios 59
3.2.4 Sense Organs 61
3.2.5 PAM: Attributes - Objects 62
3.2.6 Dynamic Module 63

3.3 END RESULT 64
3.3.1 Environment-System Interface layer 65
3.3.2 Basic control functions layers 66

3.3.3 Perception Layers ... 68
3.3.4 Recognition Layer ... 70
3.3.5 Reaction Layer ... 71
3.3.6 System Management Unit ... 71
3.3.7 Database ... 72
3.3.8 Data Processing ... 73

4 THE FUNCTION OF PERCEPTION IN PAM ... 77

4.1 HUMANS PERCEPTION AND TECHNICAL ANALOGIES ... 77

4.2 AUTOMATION SYSTEM PERCEPTION ... 80
4.2.1 Local Perception ... 80
4.2.2 Remote Perception ... 83

4.3 PAAS PERCEPTION PROCESS ... 85
4.3.1 Modularity ... 85
4.3.2 Adapting Individual Perception Systems - Communal Data Access ... 86
4.3.3 Data Comparison – Data Redundancy ... 89
4.3.4 Data Association ... 90
4.3.5 Storage of Data ... 92

5 IMPLEMENTATION OF PAM PERCEPTION SYSTEM ... 93

5.1 PERCEPTION SYSTEMS ... 93
5.1.1 Somatic Perception ... 94
5.1.1.1 PAAS Somatic Perceptions ... 95
5.1.1.2 Implementation and Integration in the Perceptive Awareness Automation System ... 96
5.1.2 Olfactory Perception ... 98
5.1.3 Acoustic Perception ... 99
5.1.3.1 PAAS Acoustic Functions and Software Solutions ... 100
5.1.3.2 Implementation and Integration in the Perceptive Awareness Automation System ... 101
5.1.4 Visual Perception ... 105
5.1.4.1 PAAS Visual Functions and Software Solutions ... 106
5.1.4.2 Implementation and Integration in the Perceptive Awareness Automation System ... 107
5.1.5 Remote Perception ... 109
5.1.5.1 PAAS Remote Perception and Possible Solutions ... 110
5.1.5.2 Implementation and Integration in the Perceptive Awareness Automation System ... 111

5.2 GLOBAL PERCEPTION ... 113
5.2.1 Perception and Attention Interfaces ... 114
5.2.2 Transformation Layer ... 115
5.2.3 Validation Layer ... 121

6 RESULTS ... 127

6.1 INDIVIDUAL PERCEPTION SYSTEMS 127
6.2 GLOBAL PERCEPTION 131

7 CONCLUSION AND FURTHER WORK 139

7.1 CONCLUSION 139
7.1.1 Perceptive Awareness Model (PAM) 139
7.1.2 SmartKitchen Perceptive Awareness Automation System 141
7.2 FURTHER WORK 142

ABBREVIATIONS 145

BIBLIOGRAPHY 149

WEB PAGES 1

Chapter 1

Introduction

This chapter is divided in three main sections. An overview of the technological progress particularly in the field of automation is presented during the first section. The term bionic and the similarities between automation systems and nervous systems are the central parts of the second section. The goal of this work is exposed in section three in sense of next progress in automation (in connection to the first section of the chapter) while using some principles from nature (in relation to the second section of the chapter).

1.1 Technology Progress

The history of technology began when humans appeared. Though technology has been always seen as a helping tool, at the beginning survival was the prime importance. Nowadays its uses and new developments are extended to all sectors of human life [Bur 99]. Observing home life, the use of electricity was one significant milestone that contributed to improve life quality in this sector. Nowadays, with the progressive introduction of automation at homes a second important step starts to be experienced towards bettering home life quality.

1.1.1 Electrical and Automation Evolution

The large number of scientific and technical discoveries that took place during the last decades of the XIX century strongly influenced humans' way of living [Ned 02]. The use of energy improved life quality by means of improving house comfort - light and heating, by means of making possible some physical-chemical transformations like cooking, or by improving the power of human muscular effort by equipping machines and vehicles with motors.

All through the years, the variation of the energy forms used in common urban activities has also contributed to improve live quality [Tra 02]. Cooking, which during many centuries was done using wood, changed to make use of coal, petrol, gas, and in most recent cases electricity. Heating is also mostly supported by means of wood, coal, petrol, gas and electricity. Illumination is carried out using of oils, greases, fuel, wax, petrol, gas and clearly also by electricity. Transport, which during some centuries was done by using animals, changed to make used of vehicles equipped with coal motors. Later coal powered motors were in most cases replaced by electrical motors (using first dry batteries, and afterwards wet batteries or electricity supplied by means of external cables, as it is the case of underground, trams, and electrical-bus).

Nowadays electricity plays an irreplaceable role in human life. This form of energy appeared at the beginning of XIX century, with the creation of the first electrical battery. And at the end of the century, with the invention of the dynamo the uses of electricity started a cruise without end. Notorious changes occurred in common transport media, e.g. electrical tram (1879) and electrical railway (1895) [Ica 02]. Furthermore electricity also influenced data transport media. Communication media experienced a first large revolution with the birth of telegraph and telephone.

It was also during the XIX century when automation appeared. Automation was defined as the realisation of tasks and functions by using stand-alone machines, without a direct human intervention, letting machines doing the hard work either requiring physical effort or monotonous tasks. At it is exposed in [Out 02] first uses of automatic machines emerged in the industry and aimed to stimulate serial production.

As it is exposed in [Fis 02], first automation systems were controlled by means of the relay logic type systems. The connection complexity was directly related to the number of devices of the system. The drawback of relays was that the control system had to be changed anytime production changed. When changes became more frequent task reconfiguration costs automatically increased. Moreover connections between hundreds or thousands of relays implied enormous efforts when designing and afterwards for maintenance. One answer to this problem, as Dietrich mentions in [Die 00], was the programmable logic controller (PLC). The development of the (PLC) also reduced the wiring complexity and consequently the high costs derived from any reconfiguration, maintenance, or unexpected stop of the complex control systems based on relays and contactors.

The development of data communication abilities influenced the next step of the evolution of automation technology [Out 02]. This development made it possible for the PLCs to communicate with others and allowed them to be located away from the machines that they controlled. However the lack of a standard together with a continuous technological change led to PLC communication chaos. Multiple physical systems and incompatible protocols appeared. In the middle of the seventies with the introduction of the first proprietary fieldbuses emerged the peer-to-peer communication concept [Fis 02]. All devices connected to the bus shared the same communication capabilities. These structured communication solutions made possible the growths of information exchange and contributed to the emergence of distributed systems in process automation.

Automation was first limited to the industry sector, but later it was integrated in buildings and then in homes. The advantages of automation, focusing on home and building automation, provide increased comfort, greater security and safety, energy efficiency and supporting communication facilities. In order to cover this new sector new automation, protocols such BACnet (Building Automation Control network) [Bac

00] and EIB (European Installation Bus) [Eib 00] started to be developed in the late 1980's. The most relevant technologies in present home and building automation are LonWorks fieldbus protocol [Loy 01] developed by Echelon, EIB/KNX protocol, which is a development the of EHS (European Home System), EIB, and BatiBUS communication protocols, and the BACnet protocol, which is supported by the BACnet committee.

1.1.2 Computer Science

For more than twenty centuries humans have been thinking about the way to make easier the tasks of counting and computation. The first known artefacts that were invented to help humans in their calculus were the counting board around 300 BC and the abacus in China as early as 500 BC [Pen 96]. As it is exposed in [Com 90, Hoy 03] these two devices were the predecessors of the two existent types of computers: the analogue machine and the digital computer. Though these devices made calculation easier the biggest conceptual difference between them and their successors refers to the word automatic. In an automatic calculator enough control and interpretative power are incorporated to allow the device to perform more than one arithmetic step without the direct intervention of the operator.

According to [Gol 77] the history of computers started at the beginning of XVII century when the first digital calculators were designed and built by Wilhelm Schickard and Blaise Pascal. These machines could do completely automatically the operations of addition and subtraction. In the case of Schickard's machine, it could partly automatically multiply and divide. Some years later the Leibniz wheel, based on the machine of Blaise Pascal, was also suitable for the totally automatically multiplication and division. As it can be read in [Com 90], after more than one-century pause of further progress it was during the early days of the nineteenth century when Charles Babbage developed the 'difference machine' in 1822. The difference machine was suitable not only for adding, subtracting, multiplying and dividing but also to solve polynomial equations. A decade later, in 1833, he conceived the basic idea for his masterpiece: the 'analytical machine'. This new conception should be moreover capable of storing and selecting information, solve problems and give printed results back. Imagined by Babbage as a composition of engines that worked together in harmony, the analytical machine was too advanced for those days. Contemporary to Babbage, Joseph Jacquard also contributed to the actual concept of computers. As it is cited in [Con 90], in 1805 Jacquard had invented a system of punch cards and hooks to move threads in mechanical looms. A group of cards constituted a program, which created textile designs.

A combination of both works made by Herman Hollerith gave rise to an electromechanical engine for recording, reading and sorting data in 1890. According to [Boo 65], in contrast to Babbage's design, which used holed cards just to give instructions to the machine, the 'automatic targets perforating machine' of Hollerith used the idea of Jacquard of using the punch cards also for storing information. Hollerith's machine, first accepted and used for handling censuses, was mass-produced and sold by the International Business Machinery (IBM), company that had been found by Hollerith. A few years later came the development of the 'automatic targets perforating machine'.

While the idea of Babbage for a digital counting machine had been almost forgotten, around one century later the idea of a program-controlled computer was considered. In Germany Konrad Zuse developed his model Z1 without any knowledge about the work of Babagge. The basic structure of Z1 matched the concept of Babagge and worked on the basis of the pure mechanic. The work of Zuse brought some innovations, which are enumerated in [Sch 88], that were an essential contribution to the future development of calculating machines:

- Calculation in binary floating point, much more appropriate for the electronic structure of computers than the traditional decimal system
- Use of bistable control elements
- Calculation using the logical basic operators AND, OR, and NOT

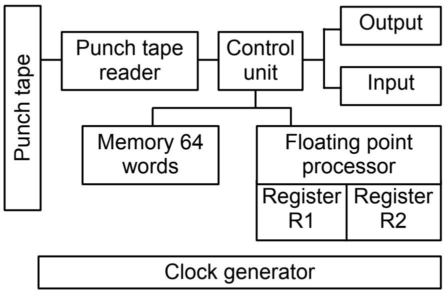


Figure 1 Z3 architecture of Zuse

In 1941 Zuse presented the first officially acknowledged program-controlled computer of the world, the Z3 [Roj 99]. As it can be seen in Figure 1, the structure of the Z3 presented similarities to the structure of a modern computer. Z3 consisted in separated units, such as a punch tape reader, control unit, floating-point arithmetic unit, and input/output devices, and contained organs like a arithmetic, memory-storage, control and connection with the human operator. The calculating machine contained a special operating modus consisting on various instructions. For example, with the instruction Lu the input device was activated and the program stopped. Consequently the human operator could check, among others, the registers R1 and R2 of the arithmetic unit and he could make intermediate calculations before the program proceeded [Zus 99]. As it is exposed in [Des 66] Z3 was built for a leading German aeronautical research centre, the ‘Deutsche Versuchsanstalt für Luftfahrt’. Due to the war Zuse's work was not generally known until years later.

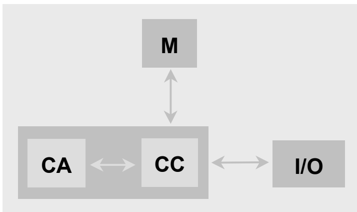


Figure 2 Von Neumann architecture

Without knowledge of the work of Zuse in America the first program-controlled computer was developed three years after the Z3. In 1944 the ‘automatic sequence controller calculator’ or ‘Mark I’ was put into operation in Harvard [Hur 53]. This machine was electrically supplied and information and instructions were introduced by means of punch cards. As Davis writes in [Dav 77], making use of relays ‘Mark I’ could not compete in speed to the next large-scale machine, the ‘Electronic Numerical Integrator and Computer’ (ENIAC). ENIAC was the first totally electronic computer.

The successor to the ENIAC was the 'Electronic Discrete Variable Automatic Computer' (EDVAC), the first stored-program computer. This innovation - the storing-program - became habitual in computer engineering. In 1945 von Neumann, author of these ideas, presented the first American written description of the stored program concept and explained how a stored-program computer could process information. As it is exposed in [Tau 63], the report organised the computer system into four main parts: the central arithmetical unit (CA), the central control unit (CU), the memory (M), and the input/output devices (IO) (Figure 2).

Technological progresses were mostly related to improvements in electrical components, which resulted in more compact and faster computers [Gol 77]. For example, ENIAC employed tubes, consumed considerable amount of energy, these gave way to smaller and much more efficient transistors. The next step forward was the integrated circuits, which enable many logic circuits to be incorporated onto a silicon structure. According to [Dav 77] these developments have made possible to convey to the actual state of computer technology from the first huge computers in the early 1950's, mostly just devoted to applications of the federal government. As Davis mentions in [Dav 77], modern computers' key-features are digital operation, stored program capability, self-regulation or self-controlling capability – dependent on automatic modifiable stored programs, automatic operation and reliance on electronics.

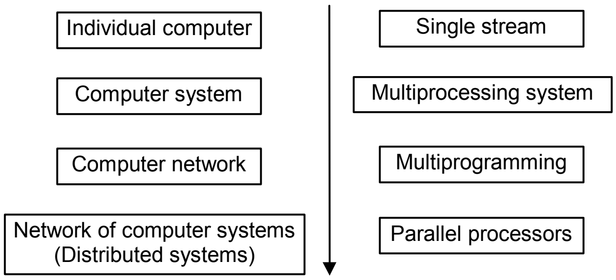


Figure 3 Developments highlighting complexity of interconnection or function

The computer engineering and computer architectural advances coupled with evolutionary changes in mathematics and logic lead to distinct lines of development and change of the environment of computers. Figure 3 shows the engineering development highlighting the complexity of interconnection or function. Computers' changes experienced according to the size of computers are reflected in Figure 4. While Figure 5 refers to progress concerning packaging for end-use or application.

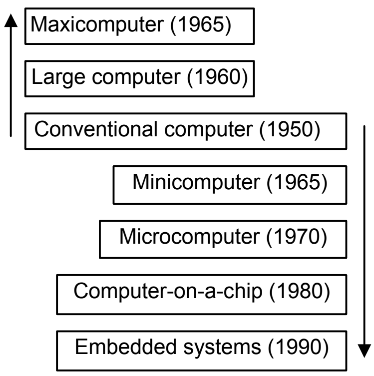


Figure 4 Developments highlighting the size of computers

Latest developments have been seen concerning refining the ability to process information, to compute, to store and distribute data, and to transfer data through long distance by means of world-wide communication networks [Loy 01]. Nowadays processors are not just limited to helping humans calculate, but also carry out decision-making in many different sectors such as robotics and process automation.

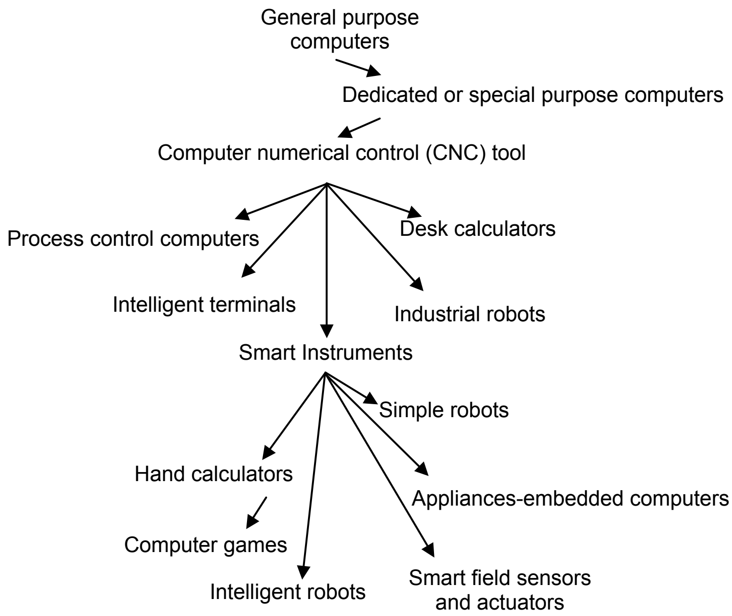


Figure 5 Engineering developments highlighting the packaging of computers for end-use or application

1.1.3 Process Automation

During the last decades of XX century, and as result of unifying technological fields, appeared process automation. Under this terminology computer science and classical electrical engineering came together combining data transfer with classical automation (Figure 6).

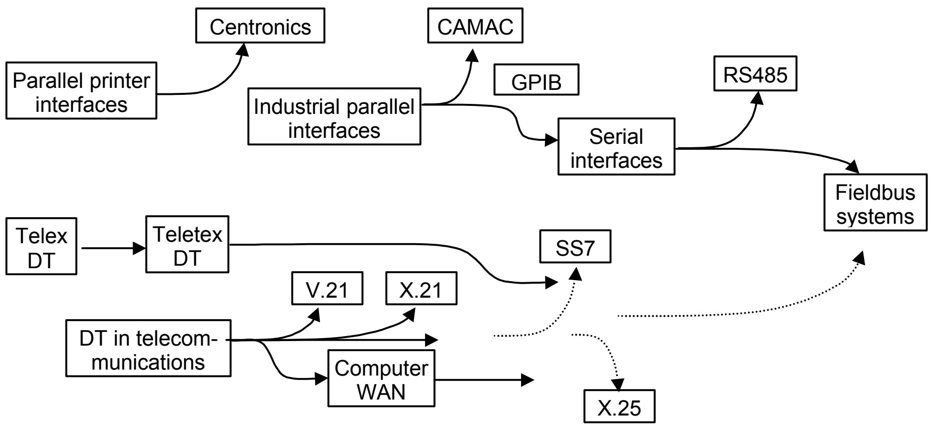


Figure 6 Roots of fieldbus systems up to the 80's [Die 00]

As it is exposed in [Die 00], one foundation of automation data transfer is seen in the classic telex networks, as well as in standards for serial data transmission, e.g.: V.21 for data transmission over telephone line and X.21 for data transmission over special data lines. Due to the limited computing power of the initially the first protocols were very simple and mostly described in forms of state machines diagrams.

At the same time another development was taking place. Hardware engineers worked on interfaces for standalone computer systems: for memories and printers as well as for process control and instrumentation equipment. Computer Automated Measurement And Control 'CAMAC' (in nuclear science) and General Purpose Interface Bus 'GPIB' (IEEE 488) are some of the applications were devices were first interconnected [Gir 95]. These bus systems had parallel data and control lines attending to the limited data processing speed and synchronisation requirements. These were later extended to serial point-to-point connections of computer peripherals to support longer distances and finally multidrop arrangements. With the development of the RS485 protocol, fieldbus technology had a basis. RS485 [Per 99] allows a bus structure with more than just two connections and provides higher noise immunity due to the use of differential signal encoding.

By this time these two important evolutions took place, as Dietrich explains in [Die 00] computer systems were gaining acceptance and were becoming more common. There was a need for computers to be able to be connected together. At the same time the communication systems of national telephone companies changed gradually from analogue to digital systems. This change brought the possibility of transferring large amounts of data from one point to another. Together with a more reliable physical layer, the first powerful data transmission protocols appeared, e.g. X.25 [X25 00] or SS7 [Isd 00].

From that time on, a key contribution to present automation came from networking with the introduction of the ISO/OSI model. The Open Systems Interconnect (OSI) model [Fol 86] was presented by the International Standards Organisation (ISO) [Iso 00] as network communication model to help producers to develop network communication applications, which were interoperable. The ISO/OSI model is the reference and starting point of many communication protocols. The model was first used in the automation field when defining Manufacturing Automation Protocol (MAP). MAP was presented as a tool for comprehensive control of industrial processes, and it resulted on a powerful and flexible, though complex, protocol. From this developed the definition of the successful Manufacturing Message Specification (MMS) protocol. As Foley exposes in [Fol 86], defining the cooperation of different automation components by means of abstract objects and services MMS was used as starting point for other fieldbus protocols.

In addition to this development in computer science, technological progresses in the area of microelectronics enabled the creation of many different integrated controllers. The requirement of interfaces to interconnect integrated controllers in a specific and cheap way motivated electrical engineers to define simple buses like I2C [Kah 93] (Figure 7). Though presenting quite simple protocols, which were not based on the layered ISO/OSI model, nor on another reference architecture [Car 96] the defined buses were and are still widely used in telephones, radios, in the board levels of TV sets.

Another important precursor was the MIL STD 1553 bus, which was defined in the middle 1970's. This bus was developed looking for a reduction of cabling weight in avionics and space technology and showed many typical features of modern fieldbus systems: serial transmission of control and data information over the same line, master/slave structure, adequate for covering long distances and integrated controllers [Sbs 98]. Under the same or similar requirements - weight and cost reduction - later protocols appeared not only in automotive industry but also in the automation area. As Dietrich explains in [Die 00], functionally defined as the classical interfaces, these protocols focus on the lower two layers of the ISO/OSI model paying no or almost no attention to the application layer definition. The addition of these definitions was done sometimes to make these systems applicable to other areas.

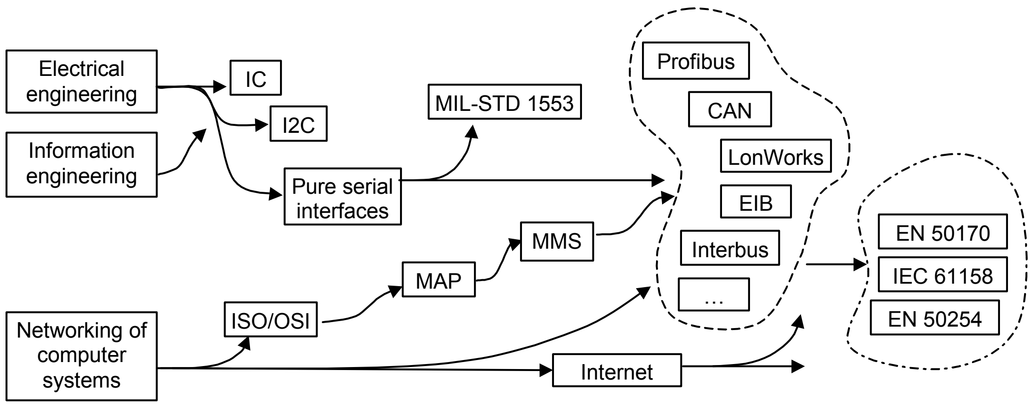


Figure 7 Milestones in the evolution of fieldbus systems [Die 00]

Automation made a great step forward with the programmable logical controllers (PLC) and more intelligent sensors and actuators. From the middle of the 80's on, many fieldbus system were defined for being used in many different application fields. Almost each automation company defined its own bus. As it is exposed in works such as [Die 00, Fis 02], these proprietary fieldbuses supported at most the first two layers of the OSI model, while the rest of the protocol was developed according to the functionality of the automation system. Since producing a small number of nodes could not justify required development and maintenance cost, this first philosophy of proprietary buses changed. Opposite to this philosophy 'open systems' were seen as the only system with possible future in the area of fieldbus automation [Cou 95]. To work out this idea, user organisations were founded to assure the definition and promotion of fieldbus systems independently of individual companies. Consequently the 'open system' philosophy led to the wish for standardisation. At it is defined in [Ghg 97], the idea behind standardisation was to support vendor-independent systems and to assure the trust of the customer in the new technology, which results on securing market position.

Nowadays, despite attempts from organisations such as the International Engineering Consortium (IEC) [Iec 03], which tried to define a universal fieldbus system there are many different fieldbus, which are implemented in distinct areas of automation [Tlc 99]. The failure of the unification is due to different reasons as cited in [Die 00]. First, there are some companies that have already considerable investment in existing and proven systems. Second there are the economical interest of nations and companies. And thirdly there are constraints and demands that prevent the aim of standard organisations to define the one and only fieldbus. Consequently, the key point for the breakthrough of open fieldbuses systems, i.e. the possibility to interconnect devices of different vendors, remains still a topic of consideration. Standards leave room for interpretation, for example the semantics of data objects are not precisely defined, etc. Since this problem concerns the application more than the fieldbuses, it must be handled beyond the ISO/OSI model. Up to now some companies have tried to solve this problem by means of defining profiles [Ech 00, Eib 00, Bac 00]. However, since profiles only partially solve the problem, as it is exposed in [Rus 01], in automation cooperation between systems is still waiting for an answer.

1.2 Learning from Nature: Bionics

Though requests set by nature are much higher than those that can be set and expected from technique new technological solutions can be found by means of analysing nature. It has been a tradition in engineering to take biological systems as models for innovations or improvement of existent developments. Avionics, material science and also architecture have successfully borrowed many concepts from nature [Die 84, Ilg 00a, Ilg 00b, Asf 00]. In addition, it has been observed in [Die 99] that there are great similarities between fieldbus automation systems and biological nervous systems. This supports the idea of using principles of biology to improve on the current limitations in automation (see section 1.1) and extend the possibilities of these systems.

Cognition as defined in [Can 02] by the Canadian Psychological Association, is the ability of the brain to think, to process and store information, and to solve problems. According to this definition this aspect of human nervous system presents special interest in a study towards a bionic solution that aims to improve the behaviour of automation systems. Works such as [Die 01, Tam 01, Rus 02] are partial-results of this study.

1.2.1 Biological and Technical Systems

In order to take advantage from nature it is necessary to analyse those principles of biological systems and to find how this might be realised using technology.

High availability – availability is defined in [Iee 90] as the degree to which a system or component is operational and accessible when required for use. In other words [Öno 01], availability means the ability of a unit to be capable of fulfilling at one point in time or within one time interval its function. This degree is calculated in the following way:

$$\text{Availability} = \left(1 - \frac{\text{Stop time}}{\text{Operating time}}\right) * 100$$

When considering life-supporting functions such as respiration or blood circulation of the human body, such systems must be at each point in time in a status that guarantees the preservation of life. Consequently the degree of availability of these functions is 100. In biological systems while sleeping the organism experiences a relaxation phase, in which the metabolism sinks and also all other activities and functions are reduced to a minimum. The aim of this status is to ensure a high availability of the biological system when awake. This possibility of the body could be technically compared with an autonomous maintenance of a technological system.

High reliability - By reliability one understands the ability of a system or component to perform its required functions under stated conditions for a specified period of time [Iee 90]. Biological systems are over-dimensioned, i.e. in normal operation they never reach their capacity limit, which benefits reliability. Additionally, when greater demands are applied to the body for longer periods, for example by strengthened manual labour, the body is able to react to it. For example, the body changes its structure: muscle tissue can be built up or diminished, and bones can be adapted or strengthened to the specific load [Fal 95].

The human body, as example of biological system, has not only to be suitable for maintaining the vitally functions but also for reacting to outside attentions in a fast and efficient way. Both, the appropriate reaction and the necessary reserves, in order to ensure the sufficient nutrition of an organism, are necessary for the preservation of a biological system. At this level the network of the body - the nervous system – plays a basic role. As soon as the nervous system does not properly work and the control to some organ is interrupted, failure is inevitable. However, not every organ is controlled via the central nervous system or via the brain. The heart, lungs as well as the majority of the internal organs are subjected to the control of the vegetative nervous system and follow other mechanisms [Kan 00a]. A completely different structure of technical systems is needed to set this concept in technique. This work presents a new model for automation systems as first step towards this required new structure.

Robustness - Robustness is the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environment conditions [Iee 90], and it is one of the prerequisites for availability and reliability. The organism has to be capable of bearing and reacting perturbations coming from the outside so that it can keep on working properly. Temperature, pressure, fluctuations of the oxygen content in air, physical forces of most varied type influence the body daily. The system has to be as insensitive as possible to such variable disturbances. The principle of over-sizing is also relevant here. For example, humans breathe approximately 0.5L air at a time, however the lung volume is about 5L, which can ensure a relatively constant supply of oxygen [Kun 00].

In technology robustness is not a new a topic of research. In many works such as [Gol 01, Suw 01, Lyo 91] different methods are presented to achieve robustness.

High stability - It is a fundamental characteristic of biological systems to remain in equilibrium. The preservation of this equilibrium is also a prerequisite for a high availability and reliability. For the preservation of equilibrium, errors must be detected as fast as possible and they have to be battled by suitable mechanisms. In this case sufficient information is absolutely necessary. This requirement will be meet in automation in the next years due to the tendency to integrate an enormous large number of field devices in the system [Die 00]. As it is presented in works such as [Mah 02], there are very low cost and low power wireless ad-hoc sensor networks [Sun 01, Rab 00], providing much more reliable information about the environment, which are suitable for integration into existing fieldbus systems.

Control loops also contribute to the stability of the system. Temperature regulation is an example of such a biological automatic control loops. The temperature of the body must be kept constant with 36,5°C to 37°C. All metabolic functions depend on the temperature and the heart frequency increases by is 10 beats per minute per 1°C temperature rise. Therefore when the body temperature sinks, metabolism also decreases, which entails less oxygen consumption [Kun 00]. This demonstrates that the human body is dependent in its operability on the stability of its biological automatic control loops.

Error tolerance – According to [Iee 90] error tolerance is the ability of a system or component to continue normal operation despite the presence of erroneous inputs.

The quality of biological systems, as it is found for example in the human body, is unattainable for the technological today's state of the art. This fact specially concerns the aspect of error tolerance. Taking this fact into account, the quality of technical systems can be improved by means of working out error tolerance in technology. In order to achieve this goal on the basis of natural principles, it is first required to analyse which are the concepts, mechanisms and strategies that help biological organism to support error tolerance.

Up to now, the basic concept of a complex organism in nature like the human body is entirely different to technical systems. The most important difference is the method of information processing. A biological system like the human body does not work automatically, but it is controlled by a subsystem of the organism. The collected information and the internal messages are the basis for the error-tolerant operation. Without information the human body could not function at all. This information is derived from and transformed by sensory organs, and transported over nerve cells and processed in one of the many centres of the brain [Kan 00e]. Though common automation systems collect information from the environment and react to it, too, at present there is a lack of high data processing in comparison to humans information processing.

The nervous system, as a distributed system, is responsible for processing and analysing information, as well as for generating appropriate reactions. It plays a crucial role by means of supporting the communication between the organism and environment. As part of the nervous system the brain is not an individual organ but consists of a multiplicity of different subsystems, which process information autonomously and in parallel, i.e. analyse and transform. The brain is also the control centre of the organism and its subsystems [Gol 95].

As described in section 1.1.3 automation has evolved towards distributed systems. Taking human nervous system as reference, the next step in the evolution of automation systems was to concentrate on the data processing to enable cooperation between the different units of the system.

In nature the process of acquisition and processing information is more multi-layered and complicated than in electronic technology. This fact leads to a substantially more efficient system, in terms of error tolerance. As error tolerance measures biological organisms support distributed systems, parallel redundancy, and regulation processes and tendency to stability. Redundancy is one the most frequently used error-tolerant methods. Another crucial factor is the capability of repairing cells, particularly in those cases like the nerve cells, in which there is no regeneration. Moreover, in case of failure of a system there is also the possibility of transferring the functions of this system to other entirely different system. It is well known [Kun 00] that the brain is capable of supporting such methods of operation.

In terms of tolerance to errors it is not possible with current technology to emulate the performance of biological systems. To obtain the best possible performance it is necessary to analyse fully the mechanisms that generate errors. Though not expecting to reach the performance of biological systems, data redundancy is a first possible method of error tolerance in technology. As it is exposed in [Die 00] in reference to redundancy, the tendency in automation systems is towards integrating an enormous large number of field devices in the systems. This large amount of data will just require the appropriate data processing to support the redundancy that make automation systems error tolerant.

1.2.2 Automation Systems and Nervous System

In biological systems information is processed by the nervous system, which allows animals to adapt themselves and react to their environment. The more complex the biological system is, the more decentralised its nervous system. As Dietrich defends in [Die 99], this is similar to automation systems, where evolution tended to a decentralisation.

The evolution of the nervous system has contributed to improve the intelligence of the systems and consequently to improve their behaviour [Tuf 02]. Until now almost all automation systems are reactive in their operation, by this is meant that the system reacts to responses, similar to the amoeba. It is hoped that using techniques found in nature it will be possible to develop improved automation systems.

1.2.2.1 Systems' Structure and its Evolution

An important feature of animal cells is their capability of irritability and excitability, which allows them to feel changes around them and react in front of these changes. For example, the amoeba is exited when touching any object and reacts by contracting its cytoplasm; thanks to its crude nervous system it responds only at the point stimulated, communicating the information sluggishly though the rest of its body. A more advanced organism is the paramecium, which has an array of oar-like hairs – regulated by microscopic nerves - that enable it to move rapidly through water by means of acting in a coordinated way and transmitting messages from one part of the cell to another.

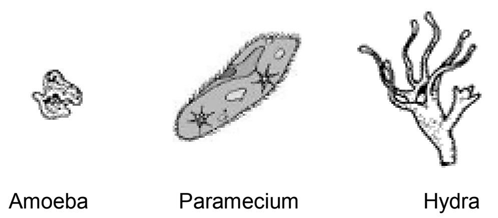


Figure 8 First steps of the evolution of the nervous system. Amoeba, paramecium, hydra

Despite the biochemical complexities of a single cell, it can manifest only simple intelligence. To become smarter, to evolve an intricate nervous system, a single cell would require elaborate sense organs to inform it, as well as developed muscles to carry out its instructions. The road to greater intelligence requires many cells, but not just a haphazard accumulation of many independent cells; clusters of millions of independent cells are no more intelligent than one. For example, although a sponge is clearly multicellular, most of its cells act independently. A sponge has no central nervous system and thus it is really not much more 'intelligent' than an amoeba. For some reasons, sponges failed to profit by their multicellularity. As a result, they have produced no higher forms of life. The sponge is an example of a life form that long ago became an evolutionary dead end [Shi 98].

A favourable mutation was needed to allow an accumulation of many cells to work together as a community. With the multicell organisms, the nervous system evolved towards specialisation of cells and group of cells turned into organs that fulfilled specific functions [Tuf 02]. The organism called hydra is a good example of a multicellular system that did evolve considerable intelligence (Figure 8).

During this process neurones appeared. These cells were not only capable of becoming exited but moreover they were suitable for conducting the stimuli. Mutations and natural selection caused the nerve cells to gradually retreat below the surface of early organisms. In turn, the clustering of such nerve cells led to a primitive central nervous system. Nervous tissue dispersed all around the organism and framed the communication net that constituted the nervous system. The formation of the sense organs and their integration with a centralised brain is the next important aspect in the evolution of the nervous system. They contributed to support remote information transfer. The last steps entail the enlargement of different parts of the central nervous systems like for example the cerebral cortex not only for enhancing survival, but also evolving intelligence.

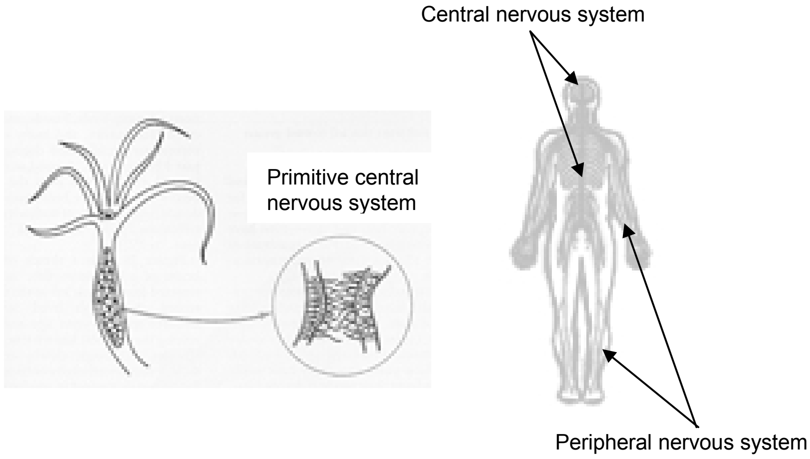


Figure 9 Evolution of the nervous system. From primitive nervous system to human nervous system

Biological systems evolved to complex human organism: containing central nervous system and peripheral nervous system (Figure 9). Humans do not just react to stimuli from the environment but also contemplate the consequences of their actions before embarking upon them. Nervous system has evolved towards supporting this preventative behaviour.

In analogy to natural organisms, automation systems are also stimulated by changes of the environment and react to them. Observing these systems one finds certain similarities between the way they evolved and the last steps of the evolution of the biological nervous system. As it is shown in Figure 10 automation systems also developed from a central to a distributed structure [Ger 97].

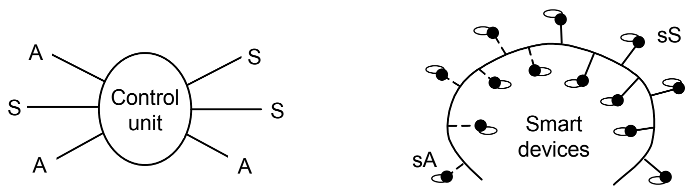


Figure 10 Evolution of control systems. Central control system and distributed control system. S: sensor, A: actuator, sS: smart sensor, sA: smart actuator

Structural and functional features of distributed automation systems such as fieldbus systems show some aspects that are characteristic of complex nervous systems. For example, the bus line itself could be compared to the nerves. And the sensors of the fieldbus network are the sense organs of the system, while the actuators can be resembled to the locomotive organs. However, as it is explain in [Tam 01, Die 01], in contrast to human behaviour the behaviour of automation systems is still pure reactive. In order to improve this aspect of automation systems functional coordination between the units of the system is needed. The 'cognition' of automation systems has to be worked out.

1.2.2.2 Human Behaviour: Cognition

Humans interact with reality processing environmental information as well as information from their own. This information experiences and generates transformations and influences human responses in an interdependent process of cognition, emotions and behaviours. Human behaviour is understood, as the result of genetics, personal history and environment, and experiences are the keys to the fundamental mechanism of association of information [Ott 02]. These experiences are a part of memory, which contributes to human behaviour by means of supporting the cognitive process to process information adding it to the brain, codifying it, storing it and afterwards recuperating it. Working memory includes manipulation of the information concerning actual experience and it has limited capacity and duration. An attention control system is expected to select strategies to process the input information. Long time memory keeps knowledge about the real world and its representations, rules and concepts' meaning. As Kandel et al mention in [Kan 00a] the mind selects some of the data it can access from the environment, it remembers some information and forgets some other.

Recent knowledge about the differentiated structure of some regions of the central nervous systems supports existent conceptions defending that cognitive functions are related to structured systems, presenting dynamic development and related to psycho-neurobiology and social interrelations [Ins 03, Sag 01, Kan 00, Sac 87]. Human's psychic character is not only involved with the innate natural biological aptitudes of the central nervous system but also with its interaction with environment and society [Gen 02]. During its evolution, and while increasing his activity in relation to the environment and to his own, the resulting representations of this activity are stored in the brain [Sac 87, Per 02]. In such a way psychical processes, which seem to be simple functions that have been always executed like for example perception, are the result of a complex historical development.

The actual conception considers highly differentiated areas in the cortical region, which work together solving each new exposed problem by means of new relations and associations. As Marquez et al defend in [Mab 00], the superior functions depend on basic anterior processes. Complex concepts could not be developed if perceptions and sensorial representations were not implemented with solidity. Though the formation of the superior centres depend on the inferior ones, once already formatted the superior centres they organise the work of the inferior centres, presenting a hierarchical control organisation.

As it is defended by neural-scientific studies such as the ones presented in [Kan 00] each individual cognitive function requires some resources from the central nervous system to be developed. Since these resources are not unlimited it is demanded to consider a strict selection of the stimulus, relations, situations and tasks while ignoring or eliminating some others. Though this selection is sometimes conscious, there are also not conscious processes and not conscious ways of processing. In publications such as [Del 01, Duc 96] it is defended that cognitive processes work by means of a modular structure where each component is expected to execute a special kind of transformation. These modules can either receive the information direct from the environment or from other modules, multiplying therefore the global processing capability. These are all strategic flexible systems allowing developing tasks through different ways: more automatic or more controlled and conscious. Each element of the system is capable of working simultaneously or in parallel, with continuous feedback and with frequent references to already existent knowledge structures or schemes.

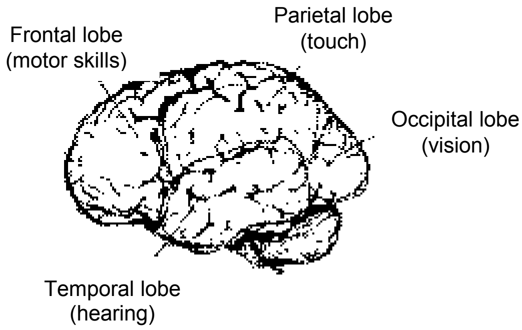


Figure 11 Major divisions of human cerebral cortex and some of their primary functions

These predetermined schemes, which are related to situations and activities, and which are stored in the memory, give an original hint to each process at the time that personalise the results of the cognitive functions.

Attending to the central nervous system and as it is exposed in [Dec 00], there are five different kinds of cortical brain from a functional point of view, (Figure 12, Figure 11). A sensorial stimulus arrives to the primary sensorial cortex being elaborated at the secondary sensorial cortex. The contained information is processed at the association cortex and the possible answers are elaborated at the secondary motor areas being executed by the primary motor cortex.

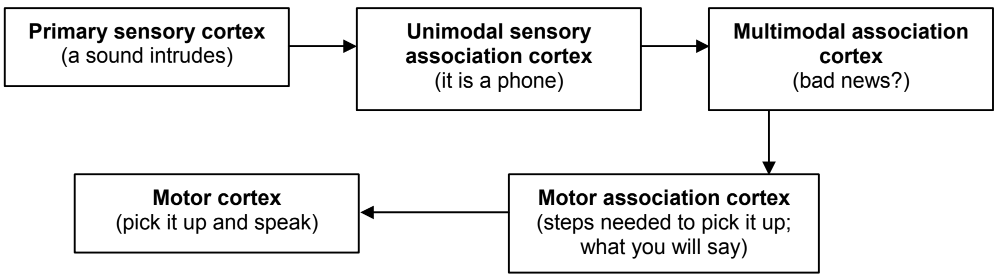


Figure 12 Brain cortex. Example of their functionality

However, if attending to the association cortex, three different kinds of third cortex are distinguished. The function of the nucleus of the parietal-temporal-occipital association cortex is mostly dedicated to integration of data coming from the different primary and secondary sensorial areas as well as to transferring information from the areas corresponding to one sense to the areas corresponding to another sense.

As it is explained in neurological works such as [Kan 00, Mab 00], these areas play an important role in transforming determined perception in abstract thought, in the organisation of internal schemes and in the organised memorisation of the experience. These areas organise the discrete impulses coming from the various regions and turn the successive stimulus into processed groups simultaneously, being the only mechanism that gives an answer to the synthetic character of perception. The frontal cortex is suitable for making stable planes and intentions capable of controlling the consequently conscious behaviour of the subject by means of regulating the activation processes located at the base of voluntary attention and the cortical tone. Prefrontal areas of the cortex support connections not only to inferior structures of the brain, thalamus and the diencephal, but also to almost all the rest of cortical regions. They are a superstructure upon every other part of the brain cortex, developing a more universal function during the general regulation of behaviour.

Besides neural-scientific studies psychological studies are also a helping tool in understanding humans' consciousness. Some psychological literature has been taken as basis when concerning consciousness modelling. For example, in 1986 Kahneman and Miller published the Norm Theory [Kah 86]. This theory is centred in social psychology and demonstrates that the norms of social behaviour are suitable for manipulation by modifying past experiences. From the different experiences authors concluded defending that 'specific objects or events generate their own norms by retrieval of similar experiences stored in memory'. Though it is not claimed that the norms become themselves it is defended that they determine some of the contents of consciousness.

Also in reference to past experiences and their influence in consciousness appears the area known as categorisation. Studies such as the ones made L. W. Barsalou [Bar 87] defend this context dependence of prototypes of categories. Barsalou concluded that 'the concepts that people use are constructed in working memory from knowledge in long-term memory by a process sensitive to context and recent experience'. Experiments made by Witherspoon and Allan [Wit 85] concerning word exposure's duration shown that conscious experience of words, which were already known by the persons had been altered by the prior experience of exposure to those words. Similar to these experiments are those related to the 'false-fame' effect of Jacoby and Whitehouse [Jac 89]. Amplifying knowledge about how human consciousness works there are also neuroanatomical references attending to the way information flows [Sch 80]. For example, the idea that functions of the thalamic reticular nucleus could support winner-take-all competition is supported by different causes presented in [Pos 90, Ber 90, Lli 92].

All these studies, either psychological or neuroanatomical, are not only basic information to understand human consciousness. Results obtained from these different researches are significant material for projects that expect to find or better technological solution on the basis of natural principles, as it is the case of the present work.

1.3 The new Challenge: Perceptive Awareness Systems

As it has been exposed in the first sections of this chapter, since humans appeared technology progress has contributed to better life quality. Trying to reach the next step of progress in automation, this work aims to better automation systems on the basis of biological principles. The already existing analogies between automation systems – fieldbuses – and nervous systems [Die 99] and successes of bionic solutions in different technological domains [Ilg 00a, Ilg 00b, Asf 00] support this idea.

The principal target of the work is the design of a new automation model that help to surpass the existing limitations towards a systems' cooperative work in automation - a helping tool when designing networked automation systems. The model has been called Perceptive Awareness Model (PAM) [Tam 01]. PAM assists enlarging the possibilities of automation systems by enabling automation systems to behave considering the whole situation, which contributes to enable the preventative behaviour of the systems. This fact means surpassing the 'pure reactive behaviour' of today's automation systems, and its correspondent disadvantages [Die 01, Tam 01, Rus 02].

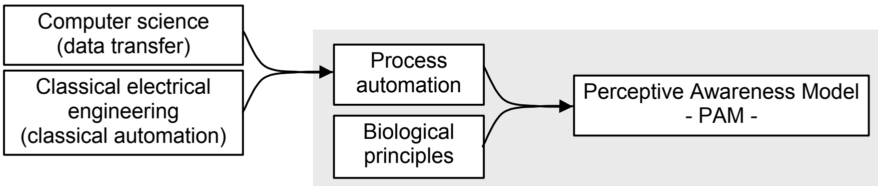


Figure 13 Concept of the Perceptive Awareness Model

Considering humans, the fact that a person is conscious of the current situation allows him/her to follow a preventative reaction [Kan 00a]. In such a way, it is justified the used of the noun awareness to refer to this new capability for automation systems. By adding the adjective perceptive the complex, large meaning of awareness is limited. Perceptive awareness concerns the part of awareness related to data perception and posterior processing of the captured data, no considering any other of the complicated aspects that consciousness presents in human beings [Fra 99].

As it is shown in Figure 13, the next step in automation results again from combining disciplines. The new model is defined on the basis of neuroscience studies concerning the way human nervous system process information, and it is designed in compatibility to present automation technologies and reference tools that are used to implement common automation systems. Going a step further in comparison to common automation system, PAM also covers the high data processing that is required in automation before reaching the application level.

1.3.1 Perceptive Awareness: from Biology into Automation

Observing complex biological systems like humans there are two first aspects of special interest when thinking on extending automation systems on the basis of biological principles. On one hand it is that of coordination between the different units of the organism and on the other that of error tolerance. Both aspects are handled by human nervous systems and contribute to enable human preventative behaviour.

Attending to the first aspect when analysing humans' comportment two main behaviours can be distinguished. Depending on the complexity of the process behind there are reflex behaviours and cognitive behaviours. Though all behavioural functions of the brain, e.g. sensory information processing, programming of motor and emotional responses, storing information, are carried out by a specific set of interconnections and communication between neurons, depending on the art of action - reflex or cognitive - communication happens along a shorter or more complicated path. As it is mentioned in [Kan 00a], while reflex actions require shorter and more direct communication between sensory neuron and motor neuron, cognitive comportment demands more complex communication processes.

Attending to the first ones, reflex actions, works such as [Kan 00c] defend that reflexes are involuntary coordinated patterns of muscle contractions and relaxation elicited by peripheral stimuli. If external conditions remain the same, a given stimulus will elicit the same response time after time. Reflexes are isolated in animals in which motor pathways from higher brain centres to the spinal cord have been cut. To understand the principles of reflex behaviour the knee jerk can be analysed. In this case a transitory imbalance of the body, which puts a stretch on the extensor muscles of the leg, produces sensory information that is conveyed to motor cells, which in turn convey commands to the extensor muscles to contract so that balance will be restored. Tapping the tendon of the kneecap - patellar tendon - just bellow the patella pulls the quadriceps femoris. This initiates a reflex contraction of the quadriceps muscle to produce the knee jerk. An extension of the led is smoothly coordinated with a relaxation of the opposing flexor muscles (Figure 14).

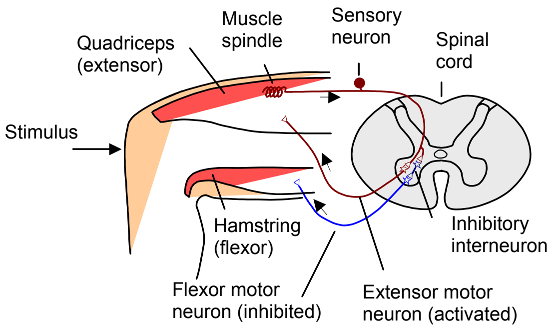


Figure 14 Knee jerk reflex [KAN00f]

However, though reflexes has been traditionally considered as automatic, i.e. as stereotyped movements in response to stimulation of peripheral receptors, recent experiments show that under normal conditions reflexes can be modified to adapt to the task [Kan 00d]. In such a way reflexes are not totally determined but present some kind of flexibility allowing them to be incorporated into complex movements initiated by central commands. Defending this proposition the action of stretching the muscles of a wrist of one arm while a subject is kneeling or standing can be exposed. The reflex response of the elbow extensor of the opposite arm depends on the task being performed by the arm, such as holding a cup or the edge of a table (Figure 15).

When referring to conscious behaviour the complexity of the process increases. Large neuron nets support the cognitive behaviour. While being interconnected, these nets give rise to the complex circuit that originates mental activity. As exposed in [Kan 00a], mental activity consists on perception, planned action, and though. Therefore, in order to understand how cognitive behaviour occurs it is need to know how sensory information is perceived and how perceptions are assembled into inner representations and formulated into plans for immediate behaviour or concepts for future action.

As Kandel mentions in his speech [Ker 00], progress in understanding the major functional systems of the brain has benefited from a reductionism approach to mental functions. The functions concerning mental activity emerge from the biological properties of nerve cells and of their pattern of interconnections. In such a way the mind can be considered a set of operations carried out by the brain, an information-processing organ made powerful by the enormous number, variety, and interactions of its nerve cells and by the complexity of interconnection among these cells.

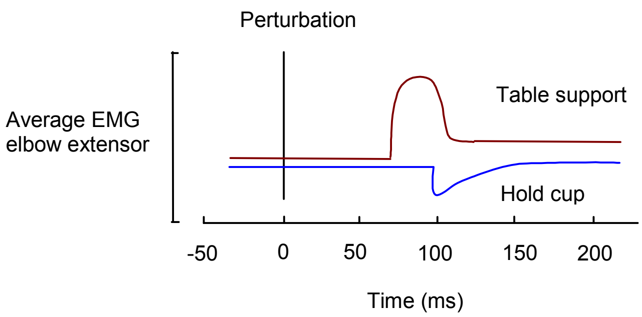


Figure 15 Reflex responses can change depending on the task

Several discrete functional systems, e.g. touch, hearing, vision, taste and smell, form the central nervous system (Figure 12). As it is described in neurological works each of these functional systems involves several brain regions that carry out different types of information processing. Information is transformed at every step, and the output of one stage of a functional system is rarely the same as its input.

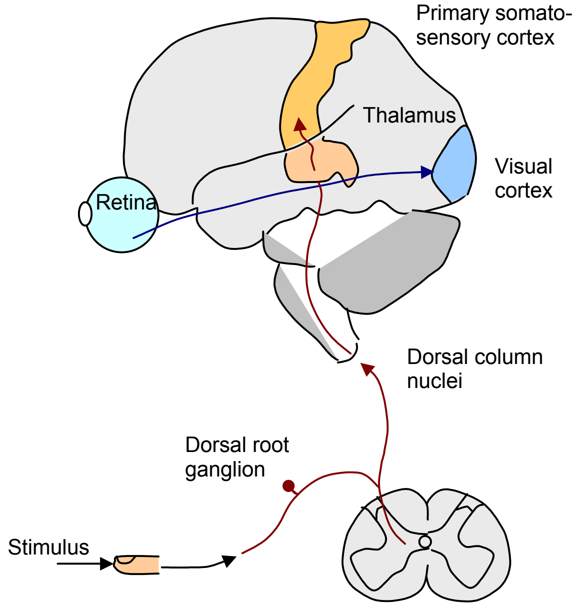


Figure 16 Flow of sensory information

Information may be amplified at some step, attenuated at other, etc. Moreover, as it is mentioned in [Kan 00d], the hierarchical organisation of the functional systems entails hierarchical organisation of information processing too. In the same way that a relation is established between reflex actions and spinal cord, humans' cognitive behaviour is connected to cerebral cortex. Organised in functional layers, the cerebral cortex is responsible for much of the planning and execution of actions in everyday life. Many areas of the cerebral cortex are concerned with processing sensory data or delivering motor commands at the same time that each one of these areas consists on several specialised zones playing different roles in processing information

According to [Kan 00e], attending to its functionality the neocortex has been divided into six layers that organise inputs and outputs. The neocortex receives inputs from the thalamus, from other cortical regions of the brain and from other sources. At the same time the output of the neocortex is directly related to several brain regions like basal ganglia, thalamus and spinal cord. Different inputs are processed in a different way (Figure 16) and the outputs arise from different populations of neurones. Information passes serially from one processing centre to another. The layering of neurones contributes to efficiently organising the input-output relationship.

Looking for the analogy in technology, human reflex behaviour is suitable of comparison to the behaviour of common automation systems, particularly those systems based on fieldbus technologies - not only in the principle of the actions but also in the way they are materialised. These systems bear a similarity to works in neurology [Sac 87, Hei 98]. The next question, which has been already formulated in different works such as [Tam 01, Die 01] is: 'Could this similarity be extended to partly cover cognitive behaviour too?'

As Kandel mentions in [Kan 00a] cognitive behaviour consists on three main functions:

- Perception of the situation
- Recognition of the situation
- Selection and execution of the proper response depending on the situation

In order to equip automation systems with perceptive awareness (PA), each one of these functions has to be supported by the system. This task entails to meet different requirements on the basis of principles from biological systems (see section 1.2.2.2). Some of these requisites are:

- The system consists on different functional units.
- The system needs stored representations of the environment.
- The functional units work together while relating and associating data.
- The system selects the stimulus that are further processed.
- Units receive information not only from the environment but also from other units.
- The system supports different kinds of reaction (in humans reflex action and conscious actions).

Analysing briefly each one of the three functions enumerated by Kandel, perception of the current situation consists on two main tasks: data collection and data processing. Considering human beings, different sources of perception - senses organs - allow the person to capture data about the current situation. Each of these organs captures different kinds of data allowing the person to perceive the situation from various points of view. This fact contributes on one hand to assure the validity of the information, by reducing errors concerning collected data and on the other hand to reach a proper perception by means of allowing data joining [Sac 87, Kan 00a]. Data processing means to submit collected data to different processes such as comparison and association. As it is exposed in [Lin 96, Kan 00e], among other things data processing allows the system to perceive complexes from the single events. The end-result of this process is the perception of the global situation.

As Kandel et al. describe in [Kan 00d], the second function 'situation recognition' demands a previous knowledge since something unknown can not be recognised. In such a way the first required task is to equip the system with a memory. Additionally data processing is needed to relate this previous knowledge with the perceived information. Once humans recognise the current situation, by means of relating it to previous experiences, they know how to react in front of it. In an analogue way and as it is explain in [Rus 02], the automation system has also to select a proper reaction for each perceived situation by means of direct connections: 'recognised situations – proper reactions'.

The working out and implementation of these three principal functions is a task that comprised in two dissertations. The work presented in this dissertation deals mainly with perception while recognition and reaction are principally handled in the dissertation of Gerhard Russ [Rus 03].

1.3.2 Extension of the Automation Model

As Ballard mentions in [Bal 90] layered models provide a powerful conceptualisation tool and aid in understanding the operation of complex systems. Functions are partitioned into vertical set of layers. Each layer performs a subset of the overall required functions. The model relies on the next lower level to perform more primitive functions and to conceal details of those functions. Each layer provides services to the next higher layer while well-defined interfaces are provided for each layer both up and down. A major advantage of a model is that the detailed internal implementation of a layer can change without affecting the overall operation of the model. Layered models have proven to be effective modelling tools for communication systems, computing operating systems, complex biological systems and robotic systems [Bal 90, Leo 00]. Even automation learnt to take advantage of layered models as reference tools during the last decades of its evolution as it is next exposed

- **Model Evolution and Present State**

Automation technology evolves and at the beginning of the 60s, with the development of communication systems. As Fischer mentions in his dissertation [Fis 02], the aspect of data communication, which continuously increases its significance in automation, had clear repercussions in the evolution of these systems and introduced some requirements that still have to be met such that of communication interoperability between devices.

Until the middle of the seventies the systems that are used in building automation consist of analogue controllers. Making use of the standardised signal-interfaces 0-10V or 4-20mA (cf. [DIN 66258]) sensors and actuators from different manufacturers could be connected on the physical layer, though for each communication link a single cable was required. During these days, the complexity of the wiring strongly increased depending on the number of components. As soon as the systems became larger new solutions were required. According to [Die 00] the introduction of the centralised controllers was the next important milestone in automation.

At the same time, the development of microprocessors, the progressive enlargement of computers' capacity and the reduction of the hardware cost led to the first DDC-devices (Direct Digital Control). Due to the increasing amount of communication, protocols were introduced – although proprietary at this time. In the middle of the seventies with the introduction of the first proprietary fieldbuses the peer-to-peer communication concept emerged. All devices connected to the bus had the same communication rights. During the late 1970's, the International Organisation for Standardisation (ISO) developed the layered Open Systems Interconnect (OSI) model, which was adapted also by fieldbus systems [Sta 98]. This progress is shortly summarised by successive model structures shown in Figure 17.

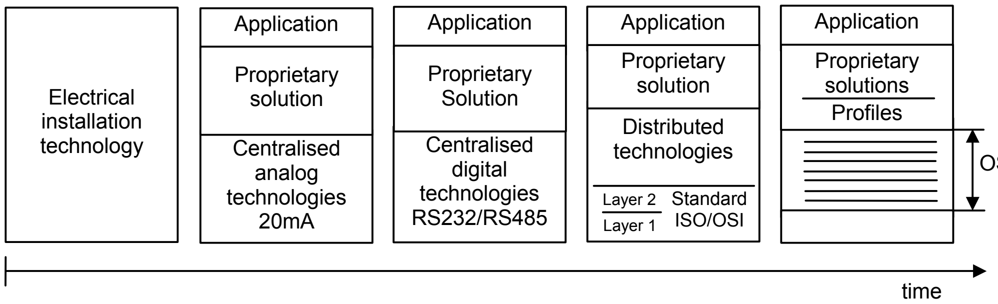


Figure 17 Structures that reflect the evolution of automation systems

During this development the aspect of interoperability between different devices and systems was gaining significance [Fis 02]. In the beginning, devices from different manufacturers could be interconnected, but as soon as central control appeared, the first problems concerning lack of interoperability arose. Under pressure of the market, communication standards were developed. However, soon, there were a variety of standards available. Thus, there was a need for universal development tools to manage projects, which combine different systems

Committees such as CEN TC247, CEN TC247 WG4, CLC TC65, LONMARK, LNO-D AKII defend the importance of the layers of the communication models for the future. But the reference models and reference tools in communication technology, like the OSI-Model, the automations pyramid, definition of profiles, SDL in telecommunications or UML, are not longer sufficient to transcribe complex automation systems in an efficient way and for a reasonable price.

- **Proposed Extension**

Aiming to solve this problematic the first outcome of this work is the design of the Perceptive Awareness Model (PAM) as extension of the automation reference tools to close the existent gap in automation theory, which is mentioned in [Die 98, Die 00b, Die 01]. The Perceptive Awareness Model is represented in terms of a layered model similar to the ISO/OSI model [Die 98, Die 00b]. This method of representation has been chosen since the communication of parallel processes are the basics of both models [Tam 01]. However, the ISO/OSI-principle covers only the pure communication between the nodes whereas PAM tends to an independent unit. That unit is to represent a logical system with one or more specific tasks (Figure 18). Though several attempts have been made to describe functions above layer 7 of the ISO/OSI model, as it is exposed in [Rau 99, Mar 99, Rat 01], PAM is a new way based upon an autonomous model which incorporates processing on top of the communication function.

Besides, while meeting this requirement of higher processing PAM introduces a new concept in automation in relation to the comportment of the system. Nowadays there is a shadowed aspect, which will gain more significance in future: today's automation systems are purely reactive, as it is described in [Die 01, Tam 01, Tam 02, Rus 02]. That means that they only react to input, if the relevance and influence of these inputs are defined explicitly in the data processing. Other factors, which are, however, not directly assigned to the application, are simply ignored. PAM surpasses this limitation by means of taken some principles from human behaviour (see section 1.2.2) to enable the automation systems to support preventative behaviour. Preventative behaviour is a situation-dependent behaviour, which means that the system does not just react to particular inputs but considers the whole situation.

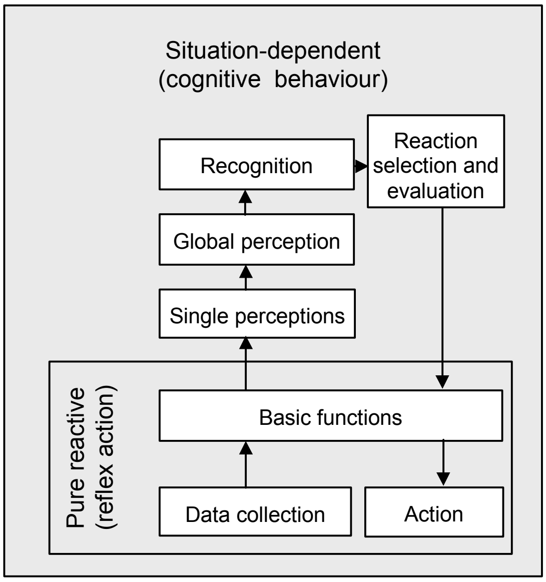


Figure 18 PAM basic architecture

Every required function, which has to be implemented in order to equip the system with the capability of perceptive awareness, is fixed positioned along the different modules of PAM. This modularisation allows working out each part independently from the others. Besides, in order to support perceptive awareness the automation system demands many more ways of processing than most of the current common automation systems. For example, in order to provide rapid response reflex actions and basic control functions, such as are already found in fieldbus systems, are to be supported. On the other hand above of these basic control functions the possibility of more complex control functions has to be integrated.

Chapter 2

State of the Art

Chapter 2 principally consists on two main parts. During the first part the state of the art in home and building automation in exposed. The section aims not only to present the possibilities of today's solution but also on showing the current low or high tendency towards systems' cooperative work in automation. The second section focuses on the aspect of artificial consciousness models. Different works are presented and discussed in base to this work.

2.1 Home and Building Automation

In the beginning of XIX century, with the development of the first electrical battery, started a new era of technological possibilities. Among other things, in the second middle of the XIX century appeared the first electrical appliances dedicated to various uses in homes and buildings [Aeg 02]. Few years later, at the beginning of XX century, another important event for the home area occurred in the communications field when television appeared [Whi 89]. Nowadays, due to the continuous technological progress, one even talks about intelligent household electric as part of the home technology. By this it is meant household electric that are connected to the home operating network and that can be programmed and controlled remotely through Internet, e.g. the Screenfridge from Electrolux [Scr 03] or Ariston Digital household electric from Merloni Electrodomestici [Mer 03].

2.1.1 Evolution in Home and Building Automation

During the last century home and building automation has experienced many significant changes. While using a classic wiring, switches and contactors have to be installed to implement the desired electrical applications. In this case power and data concerning the desired state of the device (on/off) run through the same wire.

Electrical current is at a time source of energy and data (Figure 19) [Rie 96]. Such an installation presents considerable disadvantages concerning the amount of data that can be transported. As Dietrich defends in [Die 98], in order to support higher data transmission, power and data should run along different wires. Depending on the field of work and on the technological development, data transfer has been solved in different ways. Particular conditions determine particular specifications and the different engineers have different previous knowledge, experiences and conceptions.

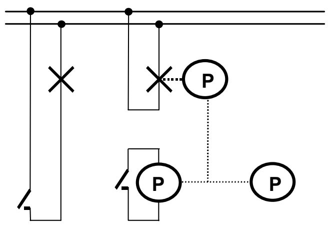


Figure 19: Traditional installation (left side); Network circuit (right side); P: Processor

Home and building automation started with centralised control systems [Fis 02]. In the beginning of the 1980's, more and more manufacturers of HVCA-related equipment began incorporating microprocessors-based controllers in their products. At that point, these controllers were designed to be stand-alone [Cra 02]. They consisted on sensors and actuators, and an intelligent control unit (ICU) between them, as it is shown in Figure 20a. Based on these principles programmable logical controllers (PLCs), which are point-to-point control systems, are designed [Veg 98]. With the increase in popularity of networked building automation and energy management systems, communication ports were added to the stand-alone controllers and various communication protocols evolved [Cra 02].

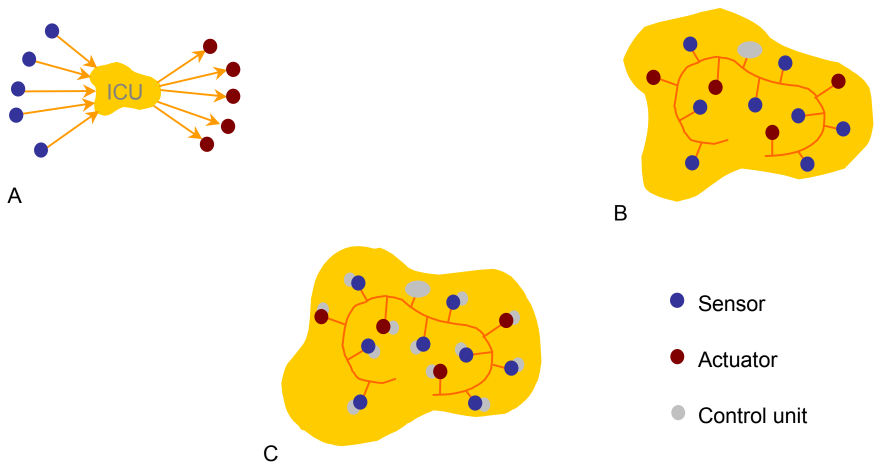


Figure 20: Control systems' evolution: A) Traditional control system B) Central bus control system C) Fieldbus-distributed bus control system

As it is exposed in [Die 98], evolution in automation drove to leave the point-to-point control systems when appearing bus technology, first centralised and later on distributed - fieldbuses. Figure 20b reflects a centralised control bus system, which consists on many dependent remote points (sensors and actuators) all connected to a central processor. The design of distributed control systems (Figure 20c) can be seen as the current last big step of automation evolution. This last development is based on the idea of implementing distributed intelligent systems, which consist on smart sensors and smart actuators, which can work without requiring a central control unit, connected to the bus line.

This evolution does not only entail structural changes of the automation installations but contributes to improve different aspect of the systems such as flexibility, extendibility, functionality and efficiency, as it is next exposed. Point-to-point automation control systems require a couple of wires to connect each sensor or actuator to the ICU. In these cases the required installation is similar to the classic electrical installation where each input-action unit, e.g. a switch, influences specific mechanisms, e.g. a lamp. Moreover these actions take place with neither influence on the rest of the devices nor being influenced by any of them. The global system depends on the ICU and therefore damage or wrong functioning of this unit, influences the work of the global system. As it is mentioned in [Loy 01] the biggest disadvantage of these systems is the complex control.

As it is explained in various works such as [Sch 97, Loy 01], with bus systems disappear some of the disadvantages presented by the point-to-point automation control systems. For example, bus systems support to connect all devices of the system by means of just one couple of wires, which benefits, among other things, the installation and maintenance; and every unit sends and/or receives messages to/from other units of the bus [Ast 88]. But the biggest success of this technology in words of Loy et al [Loy 01] is that ‘they allow an entirely new way of thinking in system design’.

In case of centralised-control bus systems, a central control unit is required to program how the different devices have to behave. However, devices can keep on communicating to the others and some parts of the systems keep on functioning even when the central unit does not work. On the contrary distributed-control bus systems require no central control unit. The idea behind distributed-control bus systems, or fieldbuses, is to develop a system in which units are capable of working by their own. Consequently control modules are less significant in distributed fieldbus technology than in centralised-control bus systems. There are two kinds of fieldbus protocol, proprietary and open protocol. In order to use proprietary protocol expensive licences are required. On the contrary, open systems public all their specification, which are accessible to reasonable prices. As Fischer enumerates in [Fis 02], some other benefits of fieldbuses technology are:

- Reduction of required hardware to implement the control system. Just few logic controllers are needed which reduces not only control panels complexity but also wiring and old complicated connections.

- Since less hardware is required, installation and maintenance costs and times diminish. New installations can be managed in an easier, faster and more secure way, as well as some other tasks like diagnosis and verification of connection errors.

- Since the complexity of the control system is reduced, project and design are more simple, easy and cheap.

- Posterior modifications, amplifications and redesigns are also easier and cheaper.

- Worldwide recognised and accepted systems mean possibility of interchange without requiring technical knowledge concerning connectivity or compatibility aspects.

Fieldbus protocols that are used in homes and building automation such as LonWorks, EIB/KNX and BACnet, have been developed on the bases of the ISO/OSI model (International Organisation for Standards/Open System Interconnection) [Sta 98, Iso 00]. Initially, standard protocols utilised flat data structures of independent and possibly unrelated values (LonTalk Network Variables, OPC items, etc.) [Cra 02]. In a flat data structure, each piece of information stands alone. For example, 25.7 might represent a temperature, but the units and name of that data sample are stored separately. In addition, as it is exposed in [Die 00], usually these fieldbus protocols considered just the four first layers of the OSI model and shown a common lack of poor specifications at the application layer. In some cases this lack has been partially covered by means of defining standard profiles for standard devices, and by specifying points such as which parameters have to be defined for each particular device and the format of these parameters. In this case, the standard device can be changed by a compatible one, without many problems. According to [Cra 02], this change is based on object-oriented programming. As object-oriented programming paradigms gained acceptance as an alternative to flat data structures, a more object-oriented approach toward field data is wished. In an object-oriented data structure, the temperature value 25.7 would be packaged with the units (°C) and the name. In such a way, data integration is assisted and objects' hierarchy is also supported by means of grouping objects together into another object. This tendency towards object oriented programming lead to the present generation of object-oriented protocols such as BACnet [Bac 00], LonMark Functional Profiles [Pro 03], and European Installation Bus Object Interface Specification (EIB - ObIS) [Eib 00]. These are some examples of companies' attempts to cover the exposed lack of specification at the application layer. This attempt is also persecuted in projects like NOAH [Doe 99] or RACKS [Qua 97], which aim providing a fieldbus-independent application interface.

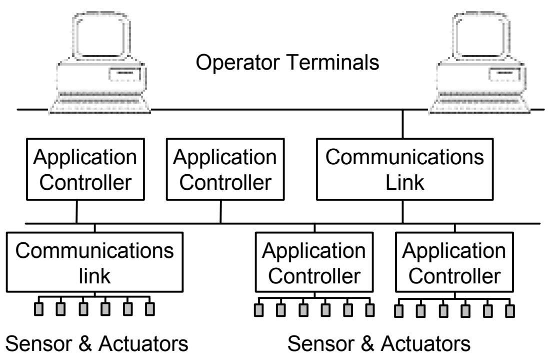


Figure 21 HES Architecture

At the present time, manufacturers build products that support one or more of the available standard protocols. However, as it is defended in [Cra 02], integrating more than one protocol into a single system can be a challenge. Besides, the tendency in building automation towards increasing the number of nodes, as Dietrich defends in [Die 00], will result on a complex control of the system, which will be accentuated with the lack of interoperability, both within a particular protocol and between protocols.

Beside the evolution of the automation systems another important aspect in the field of home and building automation is the definition of official system structures. Nowadays, attending to the automation systems structure, the Home Electrical System (HES) C3B appears as the official accepted current model in home and building automation [Iso 00]. As it is described in the web page about standards and protocols [Cab 03b], the Home Electronic System (HES) is an international standard for home automation under development by experts from North America, Europe, and Asia. The Working Group is formally known as ISO/IEC JTC1/SC25/WG1.

- ISO - International Organization for Standardization
- IEC - International Electrotechnical Commission
- JTC1 - Joint Technical Committee 1, responsible primarily for information standards
- SC25 - Subcommittee 25, Interconnection of Information Technology Equipment
- WG1 - Working Group 1, entitled Home Electronic System

As it is explain in [Hes 99] the HES C3B model represents an amalgamation of proposals submitted by French and United States experts. Figure 21 illustrates the HES C3B model features. This hierarchical description is provided for logical clarification of building control functions. The hierarchy consists of three logical layers:

- Field Level - The lowest level in the hierarchy, the Field Level, is populated primarily by sensors and actuators. Control elements may be located at this level for fully distributed control. The two Application Controllers on the right illustrate this. Typical building automation applications include security, lighting, HVAC (heating, ventilating, and air-conditioning), vertical transport (elevators and escalators), and power distribution.

- Automation Level - Control elements in building automation have traditionally been located at his level. This is especially the case when sensors and actuators do not include local intelligence. Thus, the two Application Controllers on the left side of Figure 1 are concentrators for less intelligent devices at the Field Level. Also, related functions located in different parts of the building may be coordinated at the Automation Level.

- Management Level - This is where the building systems spanning many applications are managed and scheduled. Typically, operator stations are located here. However, autonomous management without operator intervention is possible, depending on the implementation.

Devices may be physically connected to a single controller, or may be connected via a Communications Link to the Automation Level where Application Controllers may reside. The Communications Link is a generic communications element. This element may be a router if the same upper layer protocols are used at the adjacent levels. If not, a gateway may be required. Similar technologies may be applied at the Automation and Field Levels to minimise complexity of the Communications Link. Where possible it is recommended to incorporate router functions rather than gateway functions into the Communications Link between the two lower levels to enhance network integration.

2.1.2 Principal Technologies in Home and Building Automation

During the last decades, fieldbus technology has suffered from continuous investigation and development, at the same time that it has been gaining acceptance. Depending on the sector where the technology is going to be used and on the application to be automated, each particular fieldbus presents different characteristics. These specialisations leads to develop many different fieldbuses that compete on the present market, which has resulted in interoperability problems. Compatibility problems even appear as soon as one tries to connect different applications of the same technology, as Russ explains in his diploma thesis [Rus 01]. Principal protocols to be taken into account on the present market when considering home and building automation area are EIB/KNX, LonTalk® and BACnetTM [Kra 02]. As it is mentioned in [Die 00], particularly up to 1999 the use and comprehension of these technologies in home and building automation area have increased continuously. By using any of these fieldbus protocols both, reduce or large installations can be supported and comparisons just make sense when considering application possibilities.

2.1.2.1 EIB/KNX Protocol

EIB/KNX protocol appeared in 1999 as new European home automation standard that integrates three existing European standards on home and building automation. Konnex or KNX protocol is the result of an alliance between the associations, Batibus Club International (BCI), European Installation Bus Association (EIBA) and European Home Systems Association (EHSA) to create a common home and building networking standard in Europe. Nowadays the Konnex association [Kon 03] supports EIB/KNX, which has a privileged position in relation to European standardisation. According to [Neb 01], the 5 of July 2000 the Konnex association signed up a cooperation agreement with CENELEC, which gave Konnex the right to specify consumers' and industry requirements in the way towards standardisation.

The EIB/KNX protocol implements all layers defined by OSI model but presentation and session layers and it is part of ANSI EIA 776.1 to .5 standard [Kra 02]. Though nowadays EIB/KNX (European Installation Bus/Konnex) supports different transport media such as twister pair, power line, radio frequency, and infrared, EIB appeared as fieldbus where electrical power and data ran through different wires. Therefore, in most installations communication media consist on a 28V DC line that runs parallel to the 230V power-line, with 320mA nominal intensity. As it is specified in different EIB books and manuals such as [Die 00b, Eib 98], the protocol is free topology, which means that the network supports various topologies like line, tree, star and ring, depending on the requirements of the building. 64 devices can be connected to the same line, increasing the number till 256 if repeaters are used. Using bridges 12 lines can be interconnected into a zone, where 15 different zones can be linked. This features and the reasonable price of the components in comparison to other fieldbuses like LonWorks [Mer 02, Sie 02, Sve 02] make EIB/KNX optimal for standard solutions – cost effective, simple an easy to implement.

When configuring an installation using EIB/KNX, some norms have to be considered mostly concerning power supply and maximal allowed line length. For example, individual power supply is required per line with a maximal length of 1000m [Eib 98]. There are also some considerations concerning data-transport wires. In order to avoid or to limit noise, these wires have to be protected from the power line, and a filter is required to avoid disturbances in both directions, from data transport line to power line and from power line to data transport line.

Each EIB device consists on a BCU (bus coupler unit), an application connector and the application itself. BCU is a universal component configured at the time that the installation of the EIB bus takes place, depending on the function to be done by the device. It works as interface between the bus line and the sender or receiver. As it is mentioned in [Kra 02], in order to assure exchangeability, each of the more than 5,000 available EIB/KNX devices is supplied with its relevant parameterisation data stored on a floppy disc which can be used - together with the EIB tool software (ETS) - by the installer for combination.

Additional information about EIB/KNX protocol can be found in various web pages such as [Eib 00, Kon 03] or books and manuals such as ‘EIB handbook system specifications’ or [Die 00b].

2.1.2.2 LonTalk Protocol

The Echelon Coorporation developed LonTalk protocol in California in 1991. The term LonWorks is the registered mark of Echelon to refer to the whole technology around the LonTalk protocol [Cas 01]. LonWorks technology conforms to standards of CENELEC, and it has been recognised as automation standard by the consumers electronics manufacturers association (CEMA) [Cas 01]. LonTalk was adopted as a new standard ANSI/EIA 709 in December 1999. Besides LonTalk is part of the BACnet standard for building control (ANSI/ASHRAE SPC_135) and it is also recognised at the IEEE (IEEE 1473) (train control), AAR (Association of American Railroads) (electro-pneumatic train braking), ECP (Encryption Control Protocol), IFSF (International Forecourt Standards Forum), SEMI (Semiconductor Manufacturing Equipment) and CEN (European Commission for Standardization).

Though the use of this fieldbus technology in home and building automation is widely extended in America, during the last years it is gaining more and more acceptance in Europe. Nowadays and a considerable number of European companies such as Sysmik [Sys 03], Svea [Sve 03], and CeteLab [Cet 03], invest in its development and production mainly for home and building automation.

The intelligence of the nodes is due to a chip, sourced either by Cypress Semiconductor or by Toshiba [Ech 03]. This double patent assures more reasonable prices. Programming language is a variation of C known a Neuron C. Main features of this programming language are the networks variables and the ‘when’ clause, by means of these two parameters tasks are activated depending on events and executed in a cooperative way between devices of the network. In analogy to EIB/KNX, LonWorks is free topology and supports various transport media such as twister pair, power line, IR, radio frequency, coaxial wire, optic fibre, and the original power link. Power link transmits data and power supply (42V DC) all through the same twister pair.

Though LonWorks is designed to meet the requirements of a big range of control applications, referring to home and building automation the protocol had succeeded in bureau buildings, hotels and industries [Cas 01]. Up to now, its imposition at the residential market has failed due to the existence of other technologies such as EIB/KNX in Europe and X10 in America that meet the residential requirements with a more economical inversion.

As it is exposed in [Cas 01], for the point of view of manufacturers of LonWorks products, the biggest mark-up of this technology is that LonWorks is a complete platform, which does not only include a protocol, but also transceivers, interoperability standards and software API. Advantages of the protocol are the lower price of the transceivers, diversity of transport media, large number of already developed products, and the quantity of documentation and development tools. Beside these advantages, the biggest disadvantage of LonWorks technology is the highest price of the development tools.

According to [Kra 02], in analogy to EIB/KNX, LonWorks has future potential on the field level. However, while EIB/KNX is said to be apt for standard simple solutions, LonWorks is well suited for complex field level applications where certain engineer flexibility is needed.

2.1.2.3 BACnet Protocol

BACnet (Building Automation and Control network) is an American national standard (ANSI/ASHRAE 135), a European pre-standard, and an ISO global standard (ISO/DIS 16484-5). BACnet describes a communication protocol for devices used in building automaton and facilities management systems. The standard was developed by a consortium of building management system users and manufacturers under the sponsorship of the American National Standards Institute (ANSI) and the American Society of Heating, Refrigeration and Air-Conditioning Engineers (ASHRAE) at the end of the 80’s. The BACnet standard was first published in 1995 as ANSI/ASHRAE standard 135-1995.

As it is exposed in [Cab 03a] BACnet was designed to provide the communications needs - hardware and software rules - for integrating a variety of building automation and control systems and devices used in commercial buildings independently of their communication protocol. This includes HVAC control, lighting control, access control, fire detection, and vertical transport. BACnet can also be used to interconnect building automation control systems in separate buildings or to link the building control systems to outside service providers such as public utilities.

As it is described in [Cab 03a] the protocol takes an object-oriented approach to modelling a system environment by specifying data structures called objects, properties and services and uses a client-server model for exchanging information. This object-oriented representation contributes to facilitate the application and use of digital control technology in buildings. An extensible object-oriented information model of the information to be exchanged and a common set of application services that are independent of network technology are defined. Several types of LAN options are provided to meet varying cost/performance trade-offs. BACnet provides a way to interconnect these LANs and scale the internetwork up to almost arbitrarily large sizes. It also has wide area networking capability using IP protocols. BACnet defines a set of objects and their standard set of properties, which are used to get information from the object or give information and command to an object. The standard defines 18 standard types of object and identifies 123 different properties.

In order to accommodate installations that require separate networks, BACnet defines a network layer protocol for controlling traffic between networks. Connections between networks take place through routers or gateways. The way to identify the separate networks is to give each a unique ‘"network number’. This, along with a protocol for deciding how and when messages should be passed from one network to the next, is the purpose of the network layer protocol. The term ‘network layer’ comes from the ISO Open Systems Interconnection Basic Reference Model, ISO 7498. The various LAN types are interconnected through appropriate gateways. The standard supports following networking technologies [New 97]:

- Ethernet
- BACnet/IP using the Internet
- Master-Slave Token Passing (MS/TP) using RS-485
- Point-To-Point (PTP) using RS-232 or modem
- ARCNET
- LonTalk

In reference to EIB/KNX, mapping EIB object interworking standards to BACnet objects has been submitted to the International standards committees and ASHRAE for inclusion into the BACnet standard. In Europe there is a large installed base of EIB/Konnex products and defining a way to interconnect these devices with BACnet devices was one of the key features that led to the recent adoption of BACnet as an ISO standard. The document ISO/TC205WG3 BACnet draft addendum d [Ash 03] contains information about how to interconnect BACnet devices with devices that use the EIB/Konnex protocol.

As it is exposed in various documents such as [Haa 97, New 97], Ethernet and ARCNET are used as backbones because of their capability to transfer large amounts of data quickly. On the other hand MS/TP and LonTalk are used as interfaces to field controllers because of the lower installation cost. PTP is used as direct connection points for computers and modems

Principally, as it is described in [Cab 03a], BACnet can be used with any local or wide area networking technology and multiple technologies can be combined into a single system. This enables to use BACnet under a wide range of cost/performance constraints, provides a way to scale economically from very small to very large systems, and provides a way to adopt new networking technologies that do not exist today. BACnet has worldwide support and is maintained by a professional society under rules that provide open access and cannot be dominated by companies with particular commercial interests. No license fees, special hardware, or other special tools are required to implement the protocol. However, in some locations there is limited number of suppliers available.

BACnet was designed to meet cost and performance constraints for commercial building applications. As it is exposed in [Kra 02], since BACnet supports services and functions that are needed for building management (trend/history, time scheduling, back-up/restore, remote management and alarm distribution) and also modern IT and networking technology (e.g. Ethernet and IP) the protocol is suitable for the upper system levels. On the contrary, according to [Cab 03a], although BACnet technology could be used for residential buildings, at today's prices that is still not feasible.

2.1.2.4 Backbone and Internet Connection

As it is defended in [Die 00], fieldbuses give their maximum after global interconnection between systems. This fact makes possible the realisation of functions that require group tasks instead of individual ones obtaining more satisfactory and precise results. Therefore in a house or building one has to talk about different hierarchical networks, supporting different features, and scaled interconnected. At a first level appeared control networks implemented by using technologies like EIB/KNX and LonWorks. Their features make them feasible for relatively easy applications, without requiring large data transport [Kra 02].

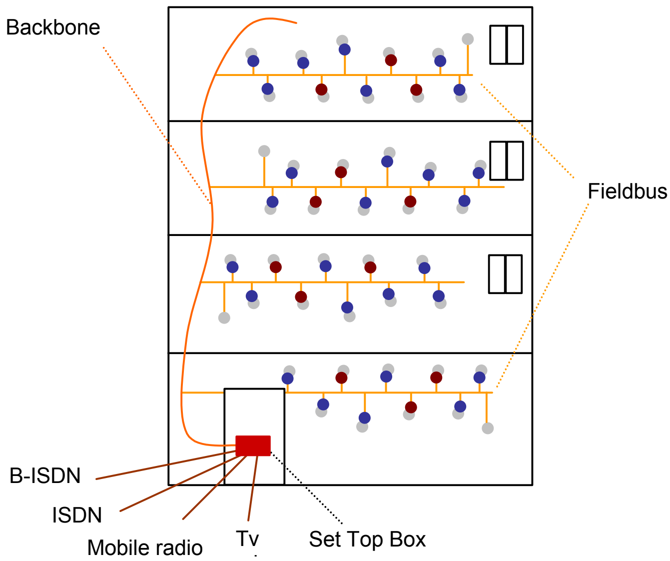


Figure 22 Home and building networking trend

Networks located at this first level are afterwards connected to a higher control network known as backbone. At this second network level, since the amount of data increases in comparison to the first network level, data transfer requirements are higher. At this second level the network has to be capable of joining and directing information coming from first level networks. As example of networks suitable for this second network's level different standards can be considered in Europe such as Ethernet and IEEE 1394 [Die 00, Ski 01].

EIB/KNX has also made some progresses in this direction. EIBnet/IP is a logical extension of the EIB into Internet Protocol at the field level. EIBnet/IP addresses the requirements for WAN and LAN connections between EIB installations in different locations. As it is exposed in [Lan 02], the Internet Protocol serves as a fast backbone and enables remote configuration, diagnostics, operation and control of EIB installations from one central or several distributed locations. Moreover, as a growing number of BACnet based systems use BACnet over IP, EIBnet/IP also simplifies the connection with BACnet.

In addition to the backbone, by means of a device represented in Figure 22 as Set Top Box connection to the outside world is supported. This communication interface (Set Top Box) has to support services between outside and inside, database functions, allowing not only simple integration and maintenance, but also an easy extended automation. By this it is meant that while connecting the fieldbus networks to Internet complete manipulation is expected. In industrial automation this manipulation will be done looking for reducing costs, in private sectors looking for comfort, higher safety and security, what could be grouped into: 'looking for a better life quality'. Though this connection could be materialise individually to every electrical device, so that each one had its own Internet address, according to [Die 00] the idea is economically not feasible at the same time that high energy expenses would be need. Due to efficiency reasons, control systems based on fieldbus technologies connected to Internet appear as solution to the problematic here exposed.

Although networks connection can take place in different ways, the easier present solution is direct connection without making use of the second network level, without a backbone. During the last years, moved by economical interests, some companies such as Merloni [Mer 03] and Elektrolux [Ele 03] started to develop and offer products covering remote network or devices maintenance, which means connection between the fieldbus and Internet. According to [Die 00], future trend is however the use of a backbone between fieldbus and Internet, in order to make possible the transport of large amount of data, requiring less wiring (Figure 22). Some networks determined by CEN to cover this role, such as Profibus FMS (Fieldbus Message Specification) or WorldFIT, present some applicability limitations concerning services that require television or telephone. This consideration drives to think on Ethernet as possible solution (eg. Ethernet running BACnet/IP). IEEE 1394 also appears as technology suitable for covering set requirements, but since possible wiring distances are too short it is not suitable for being use in buildings jet.

2.1.3 Interoperability in Home and Building Automation

One could defend that home and building automation technologies have developed towards open and interoperable solutions. Manufacturers of different fieldbus technologies such as LonWorks and EIB/KNX, upgrade their systems to increase interoperability between their devices. Hence, today's communication protocols such as LonTalk and EIB/KNX make possible that devices from different vendors can communicate to another. However, as it is exposed in woks such as [Rus 01, Rau 99], there are still many cases in which understanding between devices is still missed.

Organisations such as CEN TC247, CEN TC247 WG4, CLC TC65, LONMARK and LNO-D AKII, as well as other committees defend the importance of the layers of the communication models and their future development when talking about interoperability. The existent reference models and reference tools in communication technology, like the OSI-Model, the automations pyramid, definition of profiles, SDL in telecommunications or UML, are not longer sufficient to transcribe complex automation systems in an efficient way and for a reasonable price.

International projects like NOAH (initiated by CENELEC) [Doe 99], national projects like SIIA (LNO Germany, AKII [Lno 03]) or publications at international Conferences like IEEE-WFCS (cf. [Wfc 02]) and IFAC-FeT (cf. [Fet 01]) are concerned with solutions for systems containing different industries. There are other projects such COBA (Connected Open Building Automation) (cf. [Cob 03]) in which companies joint efforts to find the way to change this situation of non-coordination. However the selected methodology falls again to particular solutions. The model presented in this work – Perceptive Awareness Model (PAM) – expects to face the lack of co-ordination at every level by means of supporting a systems' cooperative work instead of just assisting particular cases.

PAM started to be developed in the course of a project named SmartKitchen at the Institute of Computer Technology. The first step of this project was to find solutions just by using current systems and ideas (components on the market and the today's methods to use them). Though, results shown that one had to bear many restrictions by making use of these current solutions [Rus 01]. Furthermore, it demonstrated that a new concept is needed to surpass these boundaries. During the first step of the project, it was concluded that problems were mainly due to the rigid assignment of devices to specific applications and the industry-dependency of the devices. As Russ explains in [Rus 01], the former reason required a great deal of work to extend these applications or to make devices available for applications in different industries, and complicated the maintenance afterwards. They resulted on additional devices, which were not in relation to other components.

These facts show that the limits of today's methods have been reached. The way to handle understanding between industries – in the case of making use of just one communication protocol, e.g. LonTalk – and between different technologies has to be changed entirely. First ideas and sketches of a possible solution are shown in [Die 01, Tam 01, Tam 02, Rus 02]. Due to these works an extensive knowledge base and much experiences in this subject were acquired, which are the bases of the here presented work towards global interoperable and situation-dependent behaviour systems in home and building automation.

2.2 Consciousness and Technology

Many are the disciplines that from hundred years ago till our days have contributed and still contribute to understanding the nature of human consciousness: philosophy, psychology, neuroscience, pharmacology, physics, engineering, artificial intelligence (AI), computer science and mathematics. Over two millennia years ago Aristotle already proposed that thinking proceeded by the basic relations of contiguity, similarity and opposites [Phi 00]. In 1896 Hume talked about the mind as 'nothing but a heap or collection of different perceptions, unified together by certain relations' [Duc 96]. Since in 1950 Alan Turing placed the bases of Artificial Intelligence (AI) [Kur 96] many have been the studies and researches done in this field of technology. Nowadays disciplines like engineering, artificial intelligence (AI) and computer science do not just aim to understand consciousness but try to take profit of these knowledge and, in some cases, even attempt to find its technological analogy.

2.2.1 Basic Theories

In 1951 Hobbes already though on artificial beings when declaring in the introduction to his work Leviathan [Hob 51]: 'Nature (the Art whereby God hath made and governs the World) is by the Art of man, as in many other things, so in this also imitated, that it can make an Artificial Animal'. For seeing life is but a motion of Limbs, the beginning whereof is in some principal part within; why may we not say, that all Automata (engines that move themselves by springs and wheels as both a watch) have an artificial life? For what is the Heart, but a Spring; and the Nerves but so many Strings; and the Joints but so many Wheels, giving motion to the whole Body, such as was intended by the Artificer? Art goes yet further, imitating that Rational and most excellent work of Nature, Man'.

René Descartes was also interested in mechanical explanations of bodily processes and organic life. In 1664 he already argued that human and animal bodies could be mechanically understood as complicated and intricately designed machines [Des 64]. However, in comparison to Hobbes, Descartes talked about an immaterial soul when concerning human beings, which was necessary for Descartes to explain the peculiar capabilities and activities of the human mind.

Focused on understanding human mind, J.G. Taylor wrote 'The Relational Model of the Mind' in 1973. Taylor thought about semantically coded inputs on various working memories competing amongst each other while having their activities supported and amplified by earlier stored episodic representations. Taylor defended that the content of consciousness was determined by use of those parts memories most closely related or most relevant to that input. Beyond other previous models, Taylor argued: 'The conscious content of a mental experience is determined by the evocation and intermingling of suitable past memories evoked (usually unconsciously) by the input giving rise to that experience'. With that, the relations between perceptions are extended to include a range of past experiences related with the present one. In his work Taylor defended that these past memories act as constrains or guides to further experiences and the present behaviour must be related to present consciousness. Thus evidence for the influence of past experiences on present responses is supportive of the relational model of the mind.

The work of Johnson-Laird [Joh 83] is also a significant contribution to understand consciousness. In this work Johnson-Laird proposes that human consciousness arises with a high level processing system that coordinates lower level processes, and argues for the need for much computation underlying behaviour to process in parallel. Johnson-Laird defends that conscious states are states people have access to, can report on to others, and can rely on in conducting their own actions. Though some computer programs had been written trying to translate this feature to technological systems, especially concerning making or advising about decisions as well as evaluate whether these are reasonable, according to Johnson-Laird humans go much further by being capable of using information about their own states to guide their actions. Therefore he presents as requirement for cognitive systems 'to posses a recursive ability to model what is going on them'.

In 1988 B. J. Baars presented 'The Global Workspace Theory' [Bar 88] as psychological theory of consciousness. In his different publications [Bar 88, Bar 97], Baars argues that human cognition is implemented in a multitude of relative small, special purpose processes, almost always un-conscious. He defends that coalitions of such processes find their way into a global workspace and thus into consciousness; the message from this coalition is broadcast to all the unconscious processors in order to recruit other processors to joint in handling the current novel situation, or in solving the current problem.

Following some of these ideas and looking for achieving the artificial similarity to human consciousness new schools were born. The ideas of Aristoteles strongly influenced the associationism schools of psychology in the 18th and 19th century. And inspired from the basic conception of how a brain works appeared the connectionism schools [Med 98], which develop models using processing units that can take on activation to excite or inhibit other units. As it is exposed in [Bec 95] connectionist supporters try to identify functional features of consciousness and show how they might be explicated by means of a connectionist model. Paul Churchland, as example of connectionist theory, identified the following features of consciousness in his work [Chu 94]:

- Consciousness involves short-term memory.
- Consciousness is independent of sensory inputs.
- Consciousness displays attention suitable of being steered.

- Consciousness has the capacity for alternative interpretations of complex or ambiguous data.
- Consciousness disappears in deep sleep.
- Consciousness reappears in dreaming, at last in muted or disjointed form.
- Consciousness harbours the contents of the several basic sensory modalities within a single unified experience.

As connectionist model he proposes a recurrent network in which activation largely flows forward from input units to output units through one or more intermediate layers. Different from previous models Churchland's model incorporates feedback in the processing stream. Churchland defends that feedback is required to influence inputs that are provided later by the processing results of earlier inputs, a kind of short-term memory.

Another connectionist theory is Higher-Order Representation (HOR), which is based on the existence of two kinds of mental states. In [Ros 86] Rosenthal defends that there are phenomenal states (sensory and emotional experiences, like seeing red or smelling coffee), and cognitive states with conceptual content (proposal attitudes like thoughts and desires). As it is described in [Ros 86] this theory defends that a thought - propositional attitude - is conscious if and only if it is the direct object of a representational state of the same mind. HOR presents two versions: Higher-Order Perception (HOP) takes the higher order access to be perception-like (a kind of sense organ dedicated to detecting mental events in one's mind). Higher-Order Thought (HOT) involves a higher order thought about the mental state that is said to become conscious thereby, i.e. a thought becomes conscious only when accompanied by a higher-order thought of the form "I am in mental state x".

According to [Ayd 00], where Aydede and Güzeldere try to reach a step forward in understanding consciousness on the bases of HOR, sensory experiences are supposed to track changes in the environment. They are representations whose primary job is to make available temporally information about their environment. In this publication it is also sugested that sensations are responses to environmental changes; the information value is restricted within a time frame sufficient for the organism to act back on the environment effectively on the basis of such information. While sensory representations are stimulus-driven - vertical information processing, thinking and reasoning are horizontal forms of information processing, which can occur without direct relation with the things being thought. Perception is seen as a vertical process whereby objects of sensation are recognised, i.e. categorised or sorted under concepts. Which particular sensory channels are activated in particular cases is itself a source of information.

On the bases of one or another of these theories many AI researches have concentrated on working out similarities between technological solutions and human consciousness or human behaviour as it is next exposed.

2.2.2 Artificial Consciousness Models

In the last decade many researches have concentrated on implementing the technological analogy to a particular human feature. Conscious agents are an example of such researches, in which humans pretend to equip the systems with selective attention - attend some information received through their senses and ignore much else of it. Another examples are those researches that are focused on the linearity of conscious thoughts. As it is exposed in [Bec 95], these works are based on studies that defend that people are not aware of having multiple thoughts at once but rather of one thought succeeding another. These works have resulted on systems that integrate executive that coordinates and directs the flow of information needed for action.

Parallel to these researches, there are more ambitious ones that have focused on the functions of consciousness. Researches such as [Pol 89, Bal 90, Jov 97, Cma 98, Alb 99, Bis 99, Koc 94, Kit 00] expect a determined behaviour of the system by means of trying to structure the functional elements of consciousness. Due to its conceptual similarity to this dissertation these works are next exposed and analysed following a chronological order.

According to [Bro 97], Pollock presented his Oscar project as the world's first artificial intellect, or artilect. The work of John Pollock [Pol 89] can be related to the Higher-Order Representation (HOR) theory of Rosenthal. It consisted on implementing a robot, named 'Oscar', which was equipped with perceptual sensors allowing him to know about his environment, which would correspond to the operation of human's various sense modalities, and with first- and second-order introspective sensors capturing information about his internal world. The first-order introspective sensors monitored the output of the perceptual sensors making Oscar to respond to his environment according to his goals and reasoning system. The second-order introspective sensor equipped Oscar with the capability of experience feelings.
In the work [Bro 97], where Brockmeier analyses and compares the HOR theory and Pollocks' work, Brockmeier concludes that since Oscar implemented on a serial machine consciousness cannot be reached. His arguments are based on his convincement that phenomenal superposition is a necessary feature of consciousness, and that serial computation cannot produce phenomenal superposition, as he explains in [Bro 97].

Oscar [Pol 89] results on a robot in which information of the environment is collected through some sensors and sent to a CPU, which processes the information following some rules and generates a reaction as result of this processing. Therefore, the first think that differs between Oscar and the pretension of this dissertation concerns the structure of the system. Though Oscar also makes use of sensor to perceive the environment and, after interpreting these inputs, generates an answer, the intelligent of the system just appears at its CPU, which is the only functional unit of the system. Therefore, since the robot mainly consists on a serial machine there is not possibility of differentiating between actions. With that it is meant, that the robot of Pollock supports just one kind of actions. On the contrary, an enlargement of fieldbus systems, as it has been exposed in chapter 1, may result on a structure supporting different kind of actions, similar to human beings [Tam 01]. On one hand actions that can be assimilated to reflexes and basic functions are covered at the fieldbus level, while on the other hand actions that result from complex data processing are ordered at higher control levels.

In [Bal 90] Dan R. Ballard presents the Conjoint Computing model on the basis of insights developed from the study of biological systems such as the specialisation of each of the hemispheres of the brain. The left hemisphere appears to be specialised for those cognitive activities that are associated with symbolic computing. The right hemisphere is specialised for those functions that are more easily achieved using the neural computing paradigm. This statement suggests Ballard a combination of symbolic and neural computing. In such a way the goal of his research is to develop a model for intelligent systems that integrates concepts from numeric and symbolic processing and neural network technologies into a single unified model.

Ballard defends that for a system to exhibit intelligent behaviour it must be:

- Capable of operating in real time
- Able to exploit vast amounts of knowledge
- Tolerant of error, unexpected or wrong input
- Able to use symbols and abstractions
- Able to communicate using natural language
- Able to learn from the environment
- Capable of adaptive, goal-oriented behaviour

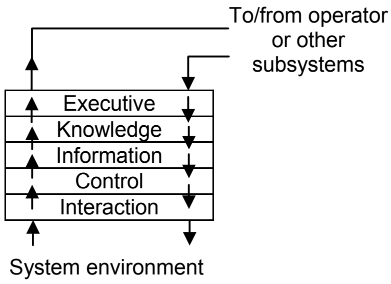


Figure 23 Symple model of Conjoint Computation

As numeric computing Ballard understands the traditional view of computing in which a fixed procedure is used to process data or to develop solutions to scientific or engineering problems. Symbolic computing views computing as a model of various cognitive processes. This view is concerned with symbol manipulation, use of heuristics to restrict search, qualitative models, knowledge representation and inference mechanism. And neural computing views computing as the interconnection of many processing-elements that interact on the local level to achieve a global view of the world. Using the design of human nervous system he proposes a model that is shown in Figure 23.

Though also based on human nervous system, the model of Ballard (model of Conjoint Computation [Bal 90]) focuses on the central nervous system, which is not sufficient to cover the range of work that is expected in this dissertation. Since no reference is made according to the peripheral nervous system, there is no place for fieldbuses in the model of Conjoint Computation. As one of the consequences of none existence of peripheral nervous system in the model of Ballard appears the incapability of the system to differentiate between kinds of actions.

In 1994 Aamodt and Plaza published the document [Aam 94], in which they analyse the ‘Case-Based Reasoning’ (CBR) concept and some CBR implementations. As Aamodt and Plaza say, case-base reasoning is an approach to problem solving and learning from the problem solving results. The CBR concept is also based on principles of nature by analysing the way humans solve problems. CBR is able to utilise the specific knowledge of previously experienced, concrete problem situations, which are called cases. In CBR new problems are solved by means of finding a similar past case. In some implementations such as CASEY [Kot 88] and BOLERO [Lop 93] case-based reasoning is used together with other methods and representations of problem solving, for instance rule-based systems. In these cases the architecture of the CBR system has to determine the interactions and control mode between the CBR method and the other components.

As it is shown in Figure 24 central tasks on CBR are: identify the current problem situation, find a past case similar to the new one, use that case to suggest a solution to the current problem, evaluate the proposed solution, and update the system by learning from this experience. Besides CBR supports learning by means of retaining new experiences after resolving the new problem situation.

The Case Based Reasoning (CBR) work of Aamodt et al [Aam 94], which is also based on principles of nature by analysing the way humans solve problems, is able to utilise the specific knowledge of previously experienced, concrete problem situations, which are called cases. In CBR new problems are solved by means of finding a similar past case, which principally matches with one of the aims of this dissertation. However, the fact that Aamodt does not introduce prioritisation in his concept makes the model weak. A system based on Aamodt's model cannot find the most problematic momentary situation. Moreover, though Aamodt talks about evaluation of the reaction, which could be interpreted as 'preventive technique' and in such a way, be assimilated as the concept of prevention of this dissertation, this evaluation takes place once the reaction has been executed. Therefore, this module does not aim to avoid a dangerous or undesired situation to happen, but informs the system about the result of the reaction as part of a learning process. Differing from some of the works that have been analyses, some of the CBR implementations such as BOLERO [Lop 93] support lower layers – rule based methods, that can be assimilated to reflex and basic control functions, and upper layers that operate as case-based planner

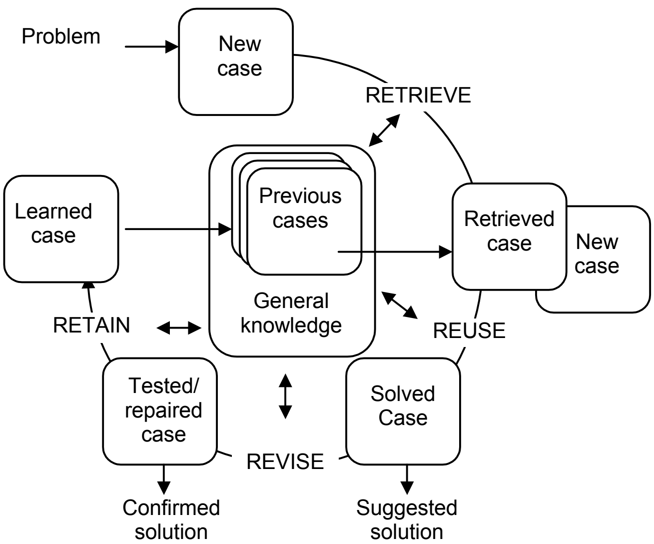


Figure 24 The CBR cycle

In 1997 attending to the neural-physiological basis that present consciousness as thalamus-cortical model and having in mind the works of Francis Crick - the Astonishing Hypothesis, the Scientific Search for Soul, and Bernard J. Baars - the Global Workspace Theory, Emil Jovanov describes his 'Conscious Processing Model' [Jov 97]. Jovanov refers to consciousness, as a complex phenomenon, which arises on a hierarchy of human rhythms and their interaction with the environment. The model is materialised in a real-time distributed, parallel, multiprocessing system with a common bus developed having in mind the anatomy and physiology of the central nervous system. This system supports rhythmic scanning within a set of active modules. Dominant activity is created by the global exchange of information on the system bus and by means of a common electromagnetic field of brain waves. Module priority dynamic is determined by available time slot on system bus and different frequencies of the brain's EM activity may represent different levels of hierarchical processing [Jov 95].

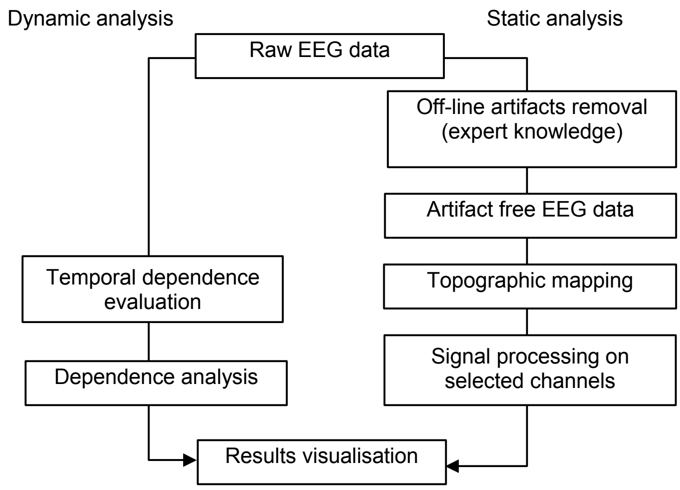


Figure 25 Block diagram of adopted methodology in [Jov 95a]

According to [Guy 76], processing of distinct sensory signals is performed by a specific thalamus-cortical system, which Jovanov models as a single processor on the common system bus. There are dedicated units (modules) and other for general purpose or associative. As it is described in [Jov 97] the system consists on modules (CPUs), permanent memory represented by genetically inherited anatomic organisation (ROM), temporary working memory (RAM) and local connections with neighbour-modules. Associated modules carry dynamic pictures of working space that can be represented as successful copies of working programs or memorised experience. In the case of an already experienced situation, these modules control the activity (automatic action), while in the new situation, modules have to intensively co-operate to modify existing (or create new) programs for this particular situation. Attending to global co-ordination, the system supports it by detection of synchronous activity in different modules and by information exchange between synchronous modules. Opposite to other models such as winner take all neural network models [Xie 01], which consider permanent request for global workspace from different modules and grant it to the most active set of modules, Jovanov's model answers to a serial scan of set of active modules. And in this case every module receives particular time to control the system bus - time slot. Modules' priority changes by increasing the time slots on the system bus or by changing processing performance. The limbic system is also implemented in Jovanov's model making possible the high priority activation of a set of modules necessary for survival.

Differing from the aim of this dissertation, Conscious Processing Model of Jovanov [Jov 97] just focuses on the process towards being conscious of situations. Since in this case behaviour is not considered important concepts of this dissertation such as prevention are irrelevant for Jovanov. Though the model is implemented by means of a real-time distributed, parallel, multiprocessing system with a common bus, the use of time slots to share the control over the bus is one weakness of the system.

Also based on 'The Global Workspace Theory' of Baars appears 'Conscious Mattie'. Conscious Mattie (CMattie) is a software agent that was implemented by Stan Franklin and his colleagues at the Institute of Intelligent Systems at the University of Memphis between 1996 and 1999 [Fra 97, Fra 99]. The software operates under UNIX, sending messages and interpreting them. In CMattie autonomous software agents are equipped with cognitive features, such as multiple sense, perception, short and long term memory, attention, planning, reasoning, problem solving, learning, emotions and multiple drives. The CMattie architecture (Figure 26) is partitioned into two different levels; a more abstract one or high level constructs and lower level or less abstract codelets. High-level constructs such as behaviours overlie collections of codelets. Working memory consists on several distinct workspaces for functions such as perception, composing announcements and learning.

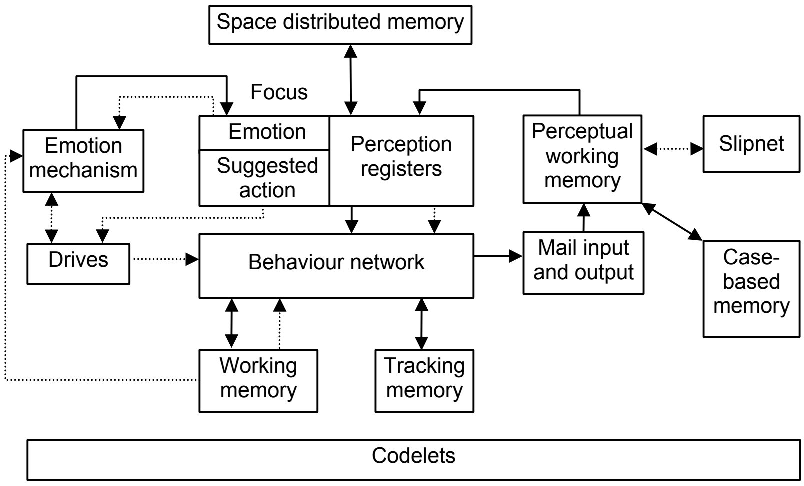


Figure 26 CMattie architecture

According to Baars' theory there are perceptual, conceptual and goal context. The perceptual context provide by a large body of water may help to interpret a white, rectangular cloth as a sail rather than as a bed sheet. The conceptual context of a discussion of money may be interpreted 'Let's go down by the bank?' as something other than an invitation for a walk, a picnic or a swim. Thirst might correspond to a goal context. In the CMattie architecture high levels constructs are identified with their collection of codelets and therefore can be seen as a context. A node type 'perceptual context' becomes active via spreading activation in the slip-net when the node reaches a threshold. Several nodes can be active at once, producing composite perceptual contexts. Conceptual context also resides in the slip-net, as well as in associative memory. Goal contexts are implemented as instantiated behaviours and become active by having preconditions met and by exceeding a time variable threshold. Goal hierarchies are implemented as instantiated behaviours and their associated drives, e.g. hunger drive might give rise to the goal of eating pizza. Consciousness codelets, though always active, act first when problematic situation occurs. Attention appears at the global workspace resulting from perception and from internal monitoring.

A first significance difference between the conceptual ideas of this dissertation and CMattie, the autonomous software agent developed by Franklin et al [Fra 97, Fra 99], concerns the sensing system and consequently the function of perception. Since CMattie just consists on software less care is paid to the input information. CMattie has no sensing system that collects information from the environment, but just a typed message input. Though there are different functional units, which principally match the concept of distribution of tasks persecuted in this dissertation, since they just consist on software the distribution occurs just partially. Besides, among the different functional units of CMattie there is no reference concerning a validation or preventive unit, which does not meet the requirements of a preventative behaviour. There is another relevant difference; since inputs are just typed messages the concept of 'interpretation of the environment' lessens to interpret messages. With that it is means that the domain of work of CMattie is significantly smaller than the domain that this dissertation aims to cover.

J. Albus works on benefit of scientific models of the mind while considering system's emotion and will. In 1996 Albus presents the 'Real-time Control System' (RCS) [Alb 96, Alb99]. In RCS two different kinds of memory are supported – long-term and short-term memory. Nodes (sometimes agents) are part of the control system that processes sensory information, maintains a world model, computes values, and generates behaviour. Each node is a functional block that has to be particularly designed to carry on a specific function such as locomotion or communication. Complex functions are supported by interconnections between nodes.

RCS is a strategy system in which nodes in upper levels in the hierarchy make long range strategic plans, while lower levels behaviour generating modules refine the long term plans into short term tactile plans with detailed activity goals. In this system details of execution are left to subordinates. A task command in RCS is of the form 'Do action on object to achieve a goal x'.

The Real-time Control System (RCS) of J. Albus [Alb96, Alb99] is a 'strategy system' thought for military applications. Consequently the system does not aim to better fieldbuses or to face the problem of lack of understanding between systems that exist in this domain. A particularity of RCS is that upper levels make long range strategic plans, while lower levels refine the long term plans into short term tactile plans, i.e. details of execution are left to subordinates. Such a function may make sense for a 'strategy system', but in case of perceptive awareness automation system (PAAS) it would just make the system more complex. As it is exposed in chapter 1, PAAS requires a functional structure to properly attend to the bottom-top data flow. In the sense that dependency appears in RCS, the top-bottom reaction flow is 'independent' of the functional structure. Though of course, in PAAS the different reactions have to start – be ordered - at different levels: basic actions, for example those equivalent to reflexes, have to be ordered by the lower levels while high control actions have to start at upper levels. Moreover, differing from the work of Albus, whenever the current situation presents no danger at lower levels common automation applications – fieldbus applications - work autonomously. And just in case the system recognises any irregular situation, upper layers take the control and override lower layers' actions. Additionally, the functionality that Albus describes is more concretely task-oriented than the required functionality to reach perceptive awareness (PA). As it has been previously exposed (see section 1.2), PA demands the functions of perception of the global situation, recognition of the global situation and selection and validation of the reaction, while Albus talks about functions such as locomotion. Though, a node of RCS incorporates perception, recognition, reaction selection and validation, since it is task-oriented it is not suitable for facing the global view that PA demands. From a functional point of view, interconnections between the complex nodes of RCS are equivalent to the interconnections between common automation applications that PA demands.

Among their similarities appears the combination of reflexive and deliberate actions to respectively support either perceptive awareness and or real-time control behaviour. Besides, in both cases the complexity of the real world can be managed by means of attention. However, while Albus says to support both kinds of action at every level of the hierarchy, a bionic solution would consider reflexive actions at low levels while upper levels would concentrate on deliberative actions.

Trying to design systems that recognise problem situations Kockskämpe et al presented the results of their research in [Koc 94]. The authors defended that measurements can often be interpreted only through their contexts, i.e. other measurements, fact that drives to two difficulties when designing a technical solution to recognise situations. On one hand the necessity of considering interactions between components or with the environment and on the other hand the fact that measured data is sometimes incomplete or uncertain. Kockskämper et al centres on dynamic systems, with that they meant that situations are recognised by means of concentrating principally on temporal events.

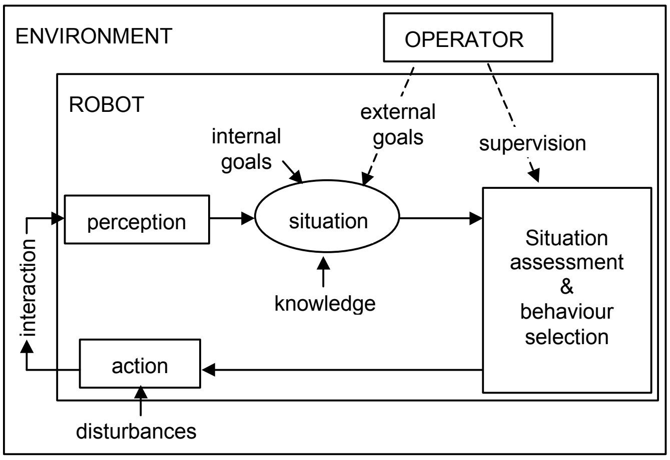


Figure 27 The role of situation as a key concept in the perception-action cycle of a situation-oriented behaviour-based robot

From the point of view of perceptive awareness automation systems the weakness of [Koc 94] is to recognise situations by means of concentrating just on temporal events. In contrast to the case treated in the work of kockskämper, in which just dynamic systems are considered, in PAAS time does not play a special role but, in most cases, appears as additional attribute of determined objects. In such a way temporal events may be prior in some situations while in others they may be even ignored. However when trying to recognise situations in which temporal events are particularly relevant it may be interesting to think about solutions presented in [Koc 94].

The work presented by Bischoff and Graefe [Bis 99] considers the case of a humanoid robot (Figure 27). Bischoff and Graefe defend that a behaviour-based system architecture that relies on understanding of situations for the selection of the behaviour to be executed is required in order to integrate technologies that work on the benefit of a high degree of user-friendliness.

In [Bis 99] the tree major problem areas are presented in order to create a 'Situation-Oriented Behaviour-Based Humanoid Robot':

- Design and integration of the sensors and actuators

- Realisation of the control structure that allows the system to generate useful and goal-directed behaviours
- Development of communication and interaction behaviours to enable the system to communicate intelligently and to display a user friendly and cooperative attitude

An important remark made in [Bis 99] refers to the adaptability of the system to the environment. The key problem of how to choose at each moment the most appropriate behaviour is solved by selecting a behaviour depending on the situation. In [Bis 99] the concept of situation does not only include objects, but also higher-level goals, overall tasks and behavioural abilities of the system. Bischoff and Graefe talk about system's internal image of the actual situation. Attending to this point they said that due to imperfect sensing or imperfect knowledge this image may sometime differ from the true situation.

Figure 28 shows the architecture designed in [Bis 99]. The situation module acts as the core of the whole system and is interfaced via 'skills' in a bi-directional way with all other hardware components – sensors, actuators, knowledge base storage and MMI (man-machine and machine-machine interface) peripherals. The skills have direct access to the components and realise behaviour primitives. They obtain certain information, generate specific outputs or plan a route based on map knowledge [Kno 00]. Skills report to the situation module and the situation module fuses via skills data and information from all system components to make situation assessment and behaviour selection possible. By activating and deactivating skills, a management process within the situation module realises the situation-dependent concatenation of elementary skills that lead to a complex result and elaborate robot behaviour.

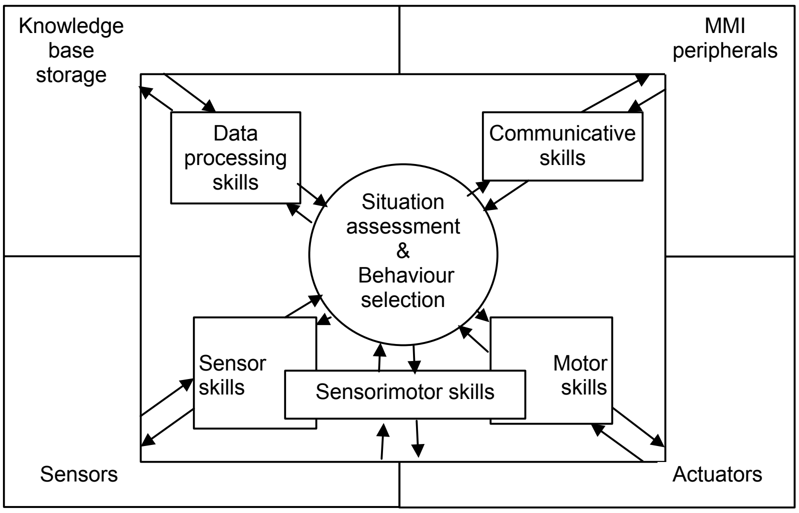


Figure 28 System architecture of a personal robot based on the concepts of situation, behaviour and skills

Bischoff and Graefe make use of the substantive skill due to its following characteristics according to [Pro 95]:

- Skills are acquired through practise or training, they are not innate.
- Skill develops in response to some demand imposed by the task environment on the organism.
- Skills are acquired when the behaviour is highly integrated and well organised.

In such a way Bischoff and Graefe defend that cognitive demands are reduced as skill is acquired, freeing limited mental resources for other activities. Low-level skills are pre-programmed and referred as robot's basic abilities. High-level skills, though also pre-programmed, contain a learning component. They define skill as a goal-directed, well-organised behaviour that is in-built, can be acquired and improved through learning, and is performed with economy of effort. Attending to databases Bischoff and Graefe defend the need of three knowledge representations: attribute topological map – static characteristics of the environment, mission description – defined by the user in more or less detail depending on the abilities of the robot, and behavioural knowledge – required for situation recognition and control.

According to this, a significant conceptual difference between the work of Bischoff and Graefe and a perceptive awareness automation system concerns prevention. The situation-dependent behaviour humanoid robot does not consider prevention at all, which moves the system away from perceptive awareness. On the other hand Bischoff and Graefe pay much attention to the user-interface, less significant for PA. The relationship between a humanoid robot and the user presented in [Bis 99] demands different requirements than the relationship between automation system and user. For example, a humanoid robot has to support movements' relationship with the user. Bischoff and Graefe talked about system's internal image of the actual situation and defend that due to imperfect sensing or imperfect knowledge images may sometime differ from the true situation. Though being conscious about the problem, the developers of the humanoid robot say nothing about integrating some techniques to face the problem such as data redundancy and data validation, which are required to reach perceptive awareness.

Level	Phylogeny	Ontogeny (age)	Consciousness Field	Behaviour
8	Man	4 years	Conception	Linguistic actions
7	Man/ape	2 years	Symbolic representation	Production of tools
6	Ape	18 months	Symbolic images	Use of tools
5	Monkey	1 year	Temporary /spatial relationship of objects	Use of media, mates' motion and geography
4	Quadruped mammal	9 months	Stable emotion to object	Detour, search, pursuit, manipulation of limb
3	Fish	5 months	Temporary emotion to present object	Capture, approach, attack, evade, escape
2	Earthworm	1 month	Valued sensation of pleasure and displeasure	Orientation and position of body and limbs
1	Anemone, jellyfish	0	Memoryless sensation	Reflex action, displacement, feeding

Figure 29 Tran conceptual model of the hierarchical relationship between mental process and behaviour

Among the Japanese researches in this field appears the work of Kitamura [Kit 00]. Kitamura compares his work with the behaviour-based architecture proposed by Brooks in 1991: 'the Subsumption Architecture' (SSA).

According to Kitamura, SSA just employs several fixed reactive behaviour modules almost independent of each other not supporting neither high-level, nor centrally goal-oriented, nor symbolic algorithms. With his Consciousness-Based Architecture (CBA) Kitamura pretends to surpass the deficits that he observed in SSA by means of combining intelligence with low-level behaviours – link reactive behaviours with symbolic ones. Kitamura argues that consciousness is the ability that enables this, and declares that 'human consciousness is certainly what subjectively activates symbolic behaviours while it is objectively visible through action/behaviour'.

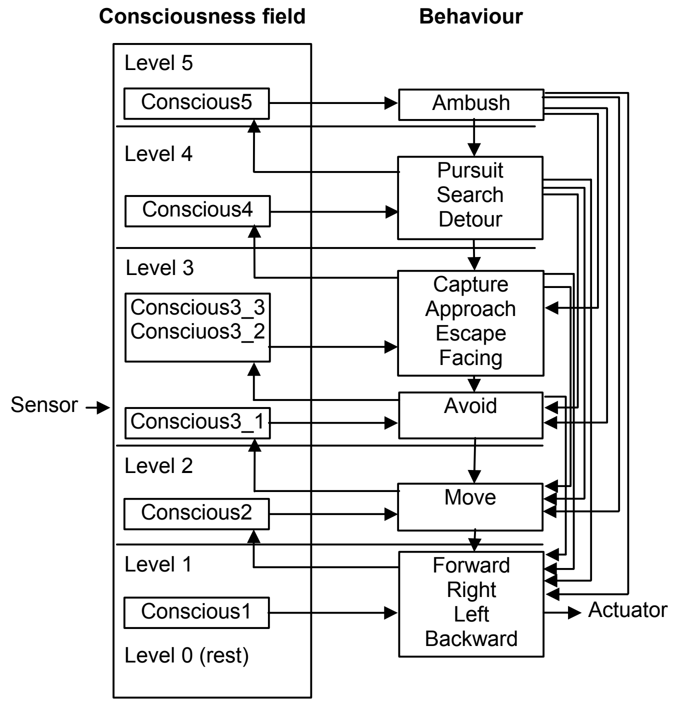


Figure 30 Hierarchy of CBA

Kitamura designed CBA on the bases of Tran's conceptual model (Figure 29). CBA consists on a hierarchical structure of relationships between behaviour and consciousness. Complex tasks are processed with elevation of the level: inhibition of behaviour elevates the level of consciousness and behaviour. In CBA a level of behaviour is chosen at which an emotion-based degree of the consciousness is maximal. Behaviours at the chosen level are selected to maximise the criterion of pleasure. These central functions are linked to the representation of the consciousness, whereas Brooks assumed intelligence without representation [Bro 91]. When a behaviour is inhibited, CBA activates a representation on the next higher level that the level of the inhibited behaviour, while SSA assumes that behaviours are activated without representation.

Although the Consciousness-Based Architecture (CBA) of Kitamura [Kit 00] is also based on concepts like consciousness and behaviour, and the relationship between both, it differs from the concept perceptive awareness in various aspects. While in CBA the level of consciousness increases as soon as behaviours are inhibited, which means that the system does not always reaches the higher consciousness level, perceptive awareness requires continuous consciousness. Continuous consciousness does not entail forbidding reflex actions but assuring a safety environment.

The scope of work is also quite different. CBA is implemented in robots that are expected to execute few specific actions, which are tree-form connected – upper actions are the combination of lower actions. Perceptive awareness is though for automation systems with a brighter domain of action than these robots. Automation systems may support perceptive awareness by means of various lower subsystems that autonomously carry out with determined task and an upper system that supports global consciousness and in case of problem situation leads the lower subsystems.

Apart from the works above mentioned, there are researches such as [Wei 97], which apply the fundamental concept of quantum physics to the neurological process of consciousness arguing that by re-applying classical anatomical and physiological principles and extrapolating to higher levels of neurological functions emerges a new unifying model. Since these works and PAM differ about the conceptual bases of consciousness, analysing them is beyond this work's domain.

Chapter3

Perceptive Awareness Model

There are two requirements which must be identified before defining the perceptive awareness model. On one hand the perceptive awareness model (PAM) has to close the gap in automation theory concerning cooperative work and systems' integration. On the other hand the model has to be suitable as helping tool to implement automation systems that behave by being aware of the global current situation and by reacting in a proper preventative way according to it.

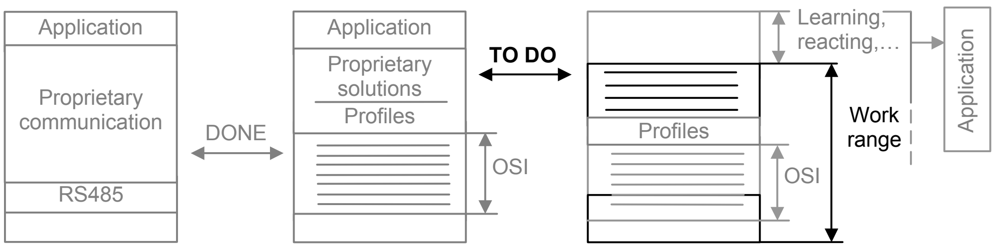


Figure 31 an extension of the current automation model

Concerning the first point and as it is shown in Figure 31, PAM has to be seen as the next harmonious step forward in automation, which entails that the model has to be compatible with the different technologies that are used in home and building automation. The solution is based on some principles of human behaviour (consequently human nervous system), which are analysed and afterwards adapted into this new automation model.

3.1 Scope of PAM

In any scientific endeavour it is necessary to define concepts and state assumptions. Before starting to develop PAM following axioms were made:

- An extension of the automation model is the first step towards teamwork and systems' integration in home and building automation.
- The extension has to be compatible with the conventional OSI model and hence provide impressive enhancements to generalised automation control systems.
- Understanding supports co-operative work between sub-systems, which is required to achieve perceptive awareness.
- The functional elements to reach perceptive awareness are sensory perception, world modelling1, reaction judgement, and behaviour generation.
- The functional elements of PAM are supported by a knowledge database that stores information about the world in the form of symbolic variables and rules.
- The complexity in PAM can be managed through hierarchical layering.
- The complexity of the real world can be managed by means of attention.

At the starting point it is also necessary to define the term 'perceptive awareness' (PA). PA lies in a particular part of consciousness. Considering humans, a preventative reaction takes place once the person is conscious of the current situation [Kan 00g]. This certainty supports the selection of the noun awareness to refer to the new capability of 'preventative behaving' for automation systems. By adding the adjective perceptive the meaning of awareness is limited to the fraction of human consciousness that just considers the state of the current situation. This state consists on one hand on the state of the environment and on the other hand on the state of the components of the system. Self-awareness, which is referred by [Kan 00g] as the answer to the question: 'Who I am?', is not contemplated when referring to perceptive awareness. In such a way, Perceptive awareness concentrates on the part of awareness related to environmental data perception and posterior processing of the data, no considering any other of the complicated aspects of human consciousness exposed for example in [Fra 99].

Perceptive awareness will enable that automation systems to behave in a preventative way in a similar way to humans by means of supporting the functions that emerge from the analysis of human conscious behaviour. As result from this analysis one concludes that human conscious behaviour consists of three main functions (see section 1.3.1), which consequently have to be attended during the design of the perceptive awareness model:

- Perception of the current situation
- Recognition of the current situation
- Selection of the proper preventative response depending on the current situation

Considering the first of these functions, human perception is supported by five senses – multi-source perception. According to the following definition of sense: 'faculty of

[1] The term world modeling refers to the representations of the environment, how the system perceives the world around.

receiving external and internal stimulus through the receptor organs, which transfer the stimulus to the central nervous system [Vox 02], two are the main players of perception: the senses (sense organs and peripheral nerves) and the central nervous system. Analysing humans and using a technical vocabulary, their sensitive organs – receptor organs – act as interfaces between the outside world and the human body. These interfaces deal with collecting environmental data such as visual, acoustic and olfactory data. In such a way humans perceive redundantly the situation from various points of view. Since data redundancy helps to reduce drastically errors, this combination of stimulus contributes to increase the certainty of the results, as it is defended and probed in [Low 00]. Furthermore, according to [Kan 00e], it contributes to accurately perceive the situation by means of data association.

In order to take advantage of the features mentioned above in automation and elaborate their analogy in technology, specific aspects of multi-source perception are analysed and adapted to automation systems. A significant remark reported in [Kan 00e] is that though each sensitive organ carries out with a specific task independently from the other - capturing the situation from different points of view, in a higher processing level data is worked out together. The central nervous system receives the stimulus from the different sense organs and associates them.

Nowadays the analogy between automation systems and human nervous system starts at the level of collecting data from the environment. Data collectors such as different kinds of sensors, cameras and microphones can be assimilated to the receptor organs. Furthermore, fieldbus networks can be assimilated to the peripheral nervous system together with part of the central nervous system (to be exact, the spinal cord). But at this point the analogy finishes. The rest of the central nervous system is missed in current automation systems, which makes the system not suitable to support high processing functions such as data association, in the sense that human brain supports this function.

PAM aims to face this deficiency and enlarge the analogy in coordination with extending the reference model in automation. Thanks to this extension the system will manage to receive collected data and submit the collected data to different processes such as verification, prioritisation and association. As it happens in humans, by supporting those processes system will be capable of perceiving the global situation, instead of limiting the perception to punctual events, as it is the fact in common automation systems.

Once global perception is reached, the faculty of recognition allows the system to give significance to what is being perceived. As it is described in neural-psychological works, recognition of particular parameters and events does not provide much information if there is no integrative action. As it is exposed in [Kan 00g] the recognition of associations is required to understand perceptions, and consequently to reach consciousness. Therefore, order to extend the possibilities of the automation systems and reach perceptive awareness, global perception has to be followed by global recognition.

Lastly and in accordance to the range of this work, there is a third interesting aspect of human nervous system to analyse. Human nervous system supports prevention in front of the recognised situation. As soon as humans are conscious of a situation, they know how to react in front of it. As it is described in [Kan 00], each one of the likely wished goals in front of the recognised situation is related to the proper reaction, which drives to reach the goal. Therefore, since PAM aims to avoid dangerous and undesired situations, during the design of the perceptive awareness model the faculty of preventive reaction will be also integrated in addition to the faculties of global perception and recognition.

Summarising, PAM is an extension of the current automation model and it is designed on the bases of principles from nature. This extension works in favour of a cooperative work between different automation solutions and technologies. Principles from nature support the idea of taking advantage of this cooperative work and make the automation system behave in a preventative way. The combination of the three faculties global perception, global recognition, and preventive reaction will make possible that automation systems present not only a pure reactive behaviour – as it is the fact nowadays, but a preventative behaviour.

3.2 Beginning and Fruition of PAM

With part of the previous contemplation it is possible to start the study and design process of the perceptive awareness model (PAM), which is exposed in [Tam 01, Die 01]. Different aspects of the desired capability (perceptive awareness) are considered along the steps of development, which are presented in the next subsections.

3.2.1 Inter-Industries Function

The first step of the development process of PAM results in an initial rough extension of the automation model, which focuses on cover the lack presented in automation systems concerning teamwork within a determined technology, which is exposed in works such as [Rau 99, Rus 01]. According to these works, a functional analysis of common technologies used in home and building automation leads to the conclusion that though actual solutions are suitable for specific applications in a particular industry, they start presenting some problems when trying to make different industries work together.

In reference to prevention, this first design concentrates on the principle that human preventive behaviour is supported once the person is conscious of the situation. As it is defended in works such as [Kan 00e, Gen 02, Roh 94, Sac 87], humans perceive and recognise the situation by composing a set of different images, e.g. visual, acoustic and olfactory, which are created by the central nervous system from data collected by the sense organs. And after perception, recognition happens as result of comparison processes between the associated image and previous experiences.

Following both statements, the first one related to teamwork and the second one to prevention, the new model has to cover the understanding lack between industries and has to integrate process functions that allow the system to create and understand individual and global images.

To cover the lack of understanding between industries a new layer, which has been called inter-industry layer, is defined (Figure 32). The function of this layer is to support unification of different data formats so that inter-industries communication can take place in an easy way without much additional effort. As soon as the inter-industry layer appears the system management becomes more complicated. From this moment on, the system has to support not only functions that are related to each particular industry – layers two and three (L2 and L3 in Figure 32), but also functions resulting from the association of different industries. Moreover, in reference to prevention, while in common situations the functions supported at layers two and three have to keep on working normally, during an irregular situation these functions may be intervened. Consequently, at the new layer the system captures the global situation as result of associating information from lowered layers, recognises it as result of comparison processes, and is capable to interrupt some functions at layers two and three if necessary, i.e. in case of a problem situation. High decisions are thought to be taken at the reaction layer, which in such as way is responsible for the selection of the strategy of reaction and for the forwarding of the selected reaction to the correspondent actuators.

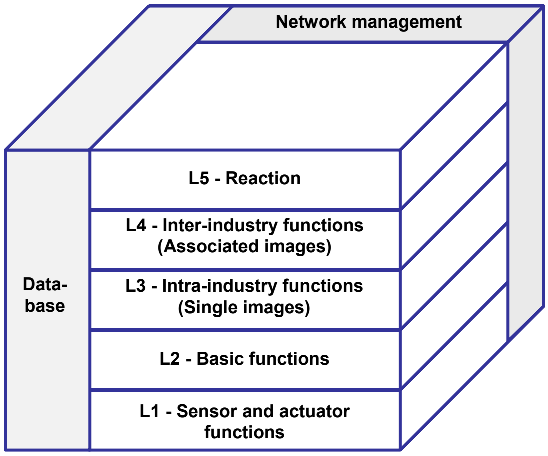


Figure 32 Basic structure of PAM

In order to handle the exposed functions and on the basis of some principles of neurology (see section 1.3.1) the system requires different databases, which have to be accessible by the different layers. On one hand the system needs to have previous experiences, which in correspondence to humans act as long-term memory of the system. According to the works [Hei 98, Sac 87], by analogy to human's long-term memory these perceptions allow the system to recognise what is being perceived to know how to react in front of it. On the other hand the system needs a dynamic database that contains the current value of the different parameters that the system detects, measures or monitors.

As it is shown in Figure 32, at this first step of development PAM is represented in 5 main layers, with additional associated database and system-management units. As it has been described in several publications such as [Tam 01, Die 01], the general system parameterises sensors' and actuators' data while the system management is responsible for regulating the communication within the different layers and the databases of the system. The layers communicate via function access points (FAPs), similar to the service access points (SAPs) of the OSI-model [Loy 01, Leo 00]. Therefore, the functions of PAM can be considered according to the black-box principle: a layer can see neither the layer above nor the one beneath, but the interface between them – the FAPs. As it happens in ISO/OSI [Sta 98], each layer of PAM has to support different services depending on its functionality. Consequently, each layer consists of several sub-layers supporting distinct functions as it is pointed out in Figure 33. Moreover, depending on the direction of the data flowing, of the data transmission (↑ or ↓) different services are supported at each layer.

Analysing Figure 33, in the first layer the different sensors and actuators operate as interfaces between environment and system – as in current automation systems. Common services to be supported at this layer are: prioritisation of the received data, adaptation and parameterisation of data to allow the net to understand with the inside and outside world, and errors identification probing the data consistence, semantic test, as well as data plausibility. Additional, at the sensor branch services such as the capture of data as well as its selection and valuation are also supported.

With this initial sketch of the enlarged model, the philosophy of PAM in front of the philosophy of common automation systems is already shown up layer 2. Opposite to common automation systems, which treat all functions in a same way, PAM presents various work-paths depending on the kind of function. According to this philosophy, the second layer - basic functions layer – is defined to support functions that can be assimilated to reflex actions, in a similar way as they are considered in [Roh 94]. Among these functions appear regulation service, timer service and basic control functions such as turning on the light when in a normal situation somebody comes into a dark room. These basic functions rely upon classical control theory and real-time systems.

Reaction	
Reaction selection Reaction supervision Error identification	
Inter-industry function	
Uniform inter-industry image Prioritisation Error identifiction	Data reduction Uniform inter-industry image Image comparison Prioritisaiton Error identification
Intra-industry function	
Uniform intra-industry image Prioritisation Error identifiction	Data reduction Uniform intra-industry image Image comparison Prioritisaiton Error identification
Basic function	
Regulation Timer Basic control functions Prioritisation Error identification	Regulation Timer Basic control functions Prioritisation Data reduction Error identification
Actuator	**Sensor**
Data parameterisation Data adaptation Error identification Prioritisation	Collect data Data selection Data parameterisation Data adaptation (Multi-)sensor evaluation Error identification Prioritisation
Outside World	

Figure 33 PAM - Detailed structure of main layers

The third layer - intra-industry function layer - works on the level of individual industries. In the area of building automation that could be, for example, demand side management, as it is presented in [Pal 01], or Heating, ventilation and Air Conditioning (HVAC) [Std 96]. At layer three functions lead to the composition and recognition of the images of the current situation attending to each particular industry. For example the image ‘room temperature’ could consist on different input data measured by a network of temperature sensors organised spatially as an array. In this case, it would be possible to extend the idea of an image to the ‘temperature image’ generated by this network of sensors. By suitable data reduction, prioritisation and error identification it is possible to acquire the most significant data from each image of each different industry, and submit this data to association.

Since PAM has to support global data comprehension between industries a common data format is required. Consequently, between the second and the third layer PAM needs an interface that turns data format into system's global data format (symbols). Data flows through this interface from the intra-industry function layer to the inter-industry function layer. Here a global-situation image emerges as result of associating data. Different tasks such as data reduction, prioritisation and error detection are needed in order to assure that the inter-industry image is uniform.

The associated image is compared to predefined images - model scenarios - stored in the database. As result of this process the system recognises what is being perceived. At this point the process reaches layer 5 - reaction layer. At this layer the recognised image is related to proper reactions. Layer 5 is defined to support two main functions. On one hand the selection of the reaction adapted to the current recognised situation. On the other hand the supervision of the selected reaction in order to avoid undesired events to happen, i.e. assure the preventative behaviour. Data resulting from the reaction layer descends and is materialised as actions in the outside world through the actuators of the system.

It is not necessary for data to flow up and down through all five layers of PAM. Depending upon requirements it has to be possible for data to flow up and transfer across and out at any of the five layers. Thus only the indispensable amount of processing is performed.

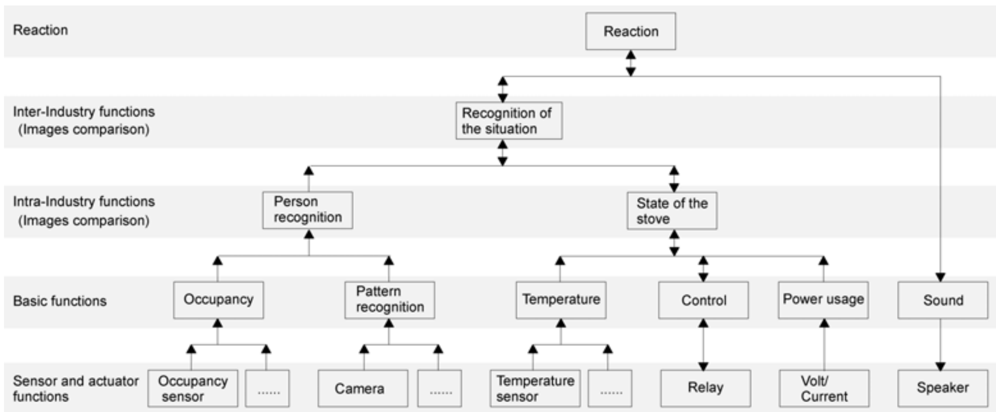


Figure 34 Child-kitchen safety example

Figure 34 helps to explain the way of working of PAM at this first step of the design process. In this example the system has to assure the safety of a child in a kitchen. As it is mentioned in [Tam 01a], top-down analysis of situations aids the design of the automation system. The arrows (either in one direction '↓' and '↑', or in two directions '↨') specify in with direction data transfer occurs. This analysis results on clarifying aspects of execution such as which components are required or which associations have to be done. In this example, the system has to check if there is a child alone in the kitchen and whether that child tries to touch the stove, which is on. If so, the system has to react preventatively by means of switching off the electrical power of the stove and attracting the attention of the child to keep the child away of the stove (in this example through an acoustic alarm). Additionally, the system can notify the adult people about the situation. In order to perceive the environment with high probability of veracity the system would require following data collector systems:

- Occupancy sensors and optical monitoring to detect presence in the room
- Temperature sensor on the cooker
- Ammeter to determine electrical current
- Detector of the position of the relay to know about the state of the stove (on/off)

In a first step data related to the same parameter but collected from different nodes would be together processed, e.g. data coming from several occupancy sensors results on the information of the occupancy's state of the room. At the basic functions layer this information together with pattern recognition information, distance sensors, etc. would allow the system to identify if there is a child in the room and where the child stands. In order to support easy data association and data comparison, input values would be transformed into the common data format of the system (symbols).

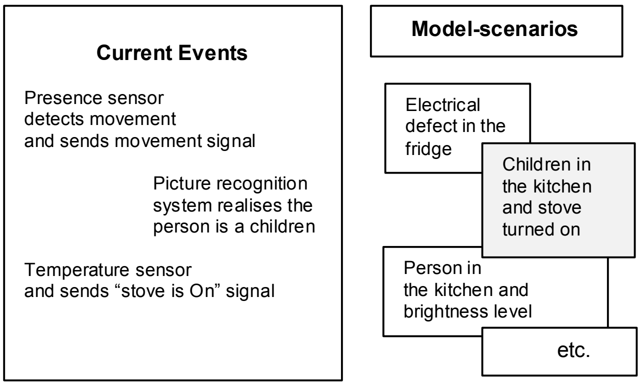


Figure 35 Selection of the model scenario depending on current events

At the intra-industry function layer the system processes together data related to a same particular industry. At this point the system recognises individual images. At layer four the information from the intra-industry function layer is combined into a global image (Figure 35). The system starts a comparison process between this global image (child in the kitchen close to the turned on cooker) and global predefined images at the database – model scenarios. This process results on a recognised situation and consequently on a proper reaction according to it (speaking loud the name of the child to catch his attention, turn off the cooker and notify adults about the situation).

Though this model gives an answer to some of the deficiencies presented by common automation systems, reconsidering the functionality and validity of the design flowed to consider some additional aspects and to some necessary changes.

3.2.2 Layers Resemblance

The second step in the design process of the perceptive awareness model contributes to some structural changes. In the first sketch of PAM, recognition of the global situation and selection of the correspondent proper reaction were represented in two different layers. This structure entailed the conceptual statement that once the system understood the global image it jumped into an upper layer where the reaction was selected and said to execute. But a conceptual analysis of the functions at both layers attempts to locate them in the same level: a reaction is not only a group of actions the system has to execute but a new situation. In such a way, it was decided that the functions of recognition of the global situation and reaction selection had to appear at the same level.

The resulted model is presented in Figure 36. The structure of the model remains the same till the third layer. According to the exposed reasoning, at the fourth layer the recognition of the composed global image appears in the same level than the selection of the corresponding proper reaction. Besides, this second model presents neither system management unit nor databases. Though they are both required they have to be considered in a different plane than that of the main layers.

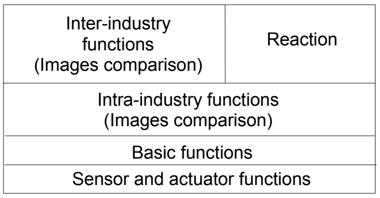


Figure 36 Basic structure of the second defined PAM

Figure 37 shows how data is processed when making use of this second model. A conceptual change is seen once the last layer is reached. At this point reaction is treated as a possible situation, i.e. a situation that just exists in the 'mind' of the system. The system values this possible situation, which results from adding the reaction to the momentary recognised situation, and proceeds executing the reaction just in case this virtual situation entails no danger.

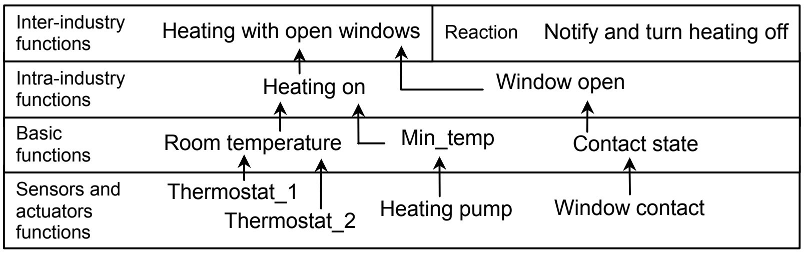


Figure 37: Data processing in the 2nd design of the perceptive awareness model (PAM)

3.2.3 Composing Scenarios

In the third step of the design of PAM some expressions are changed due to conceptual considerations at the same time that a structural change is introduced at the top of the model. These changes are related to the aspects of global situation image and model scenario recognition.

When defining the first and second exposed models it was defended that the system had to recognise the sole current situation up to the previously stored and defined model scenarios. In both cases just one global image was considered. Once more, in this third step principles from nature are use to think on technological solutions; neural science workers talk about recognition of situations in a plural form. In other words and as it is described in [Kan 00e], human nervous system processes input data so that just important data are further processed and combined in different ways. These combinations get significance as soon as they are related to previous experiences. According to this statement the number of recognised model scenarios should not be limited to only one. The global recognised situation has to be understood as a set of stored model scenarios (Figure 38).

Moreover, while analysing and redefining the last layer, a requirement seems to be missed. Since the current situation can be submitted to rapid changes some considerations concerning time have to be done. It does not seem logical to execute a reaction that was selected in a time t1, due to a recognised situation S1, in a later time t2, respectively S2, without any other consideration. Although S2 may be sometimes equal to S1, the worst case has to be considered.

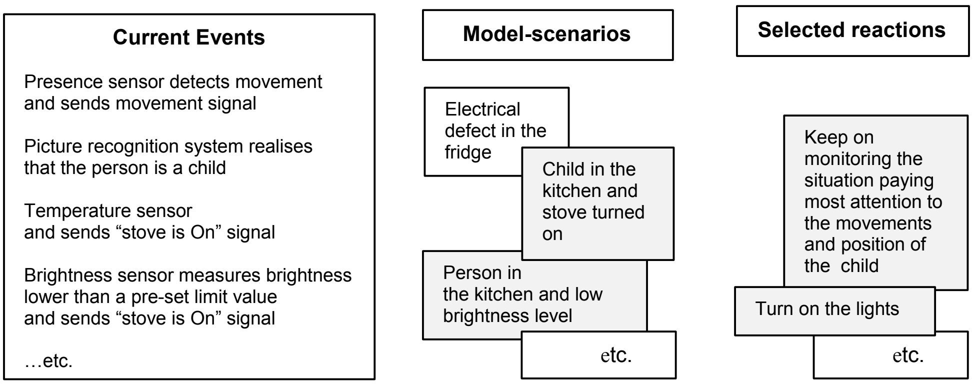


Figure 38 Recognition of model scenarios in the third step of development of PAM

Therefore in unfavourable circumstances executing the selected reaction R1 being S2 the current situation may lead to an undesired situation. Since the automation system has to assure that the reaction will cause no accident, danger, or undesired event, an estimation recurrent process has to be supported at the top of the model in order to meet this need.

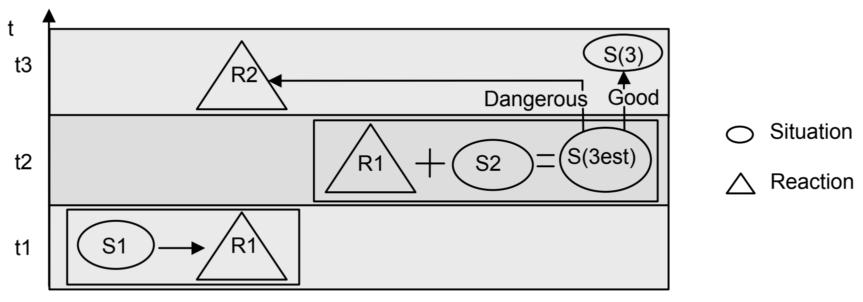


Figure 39 Recurrent reaction-estimation process

A possible solution method is shown in Figure 39, where the new function consists on adding the selected reaction R1 to the ‘in that moment’ recognised situation S2 in order to analyse the new situation S(3est) that would result before executing the reaction R1. The reaction R1 is executed just if the estimated new situation S(3est) results on a normal situation – good situation. In this case the new estimation S3 undertakes the role played by S1. In the opposite case, in which the estimation results on a S(3est) considered as dangerous situation, the reaction R1 is not executed. This second possible state entails that a new reaction R2 has to be chosen and a new estimation process has to start.

Up to this point changes have mostly concerned upper layers of PAM. However, there are still some missed considerations in relation to middle and lower layers, which are faced during the next exposed fourth step of the design process of the PAM.

3.2.4 Sense Organs

As it is referenced and explained in biological and neural science works such as [Kan 00e] (see section 1.3.1), in humans data collection occurs by means of making use of five different sense organs. This multi-source perception system, which is equipped with a high amount of sensors, allows humans to collect an enormous amount of environmental data. There are mainly two benefits of such a multi-source perception system:

- The system supports data redundancy, which contributes to decrease the probability of perceptual errors.
- Data from different sense organs can be associated, which leads to a better understanding of the current experienced situation.

In order to take advantage of such a system in automation, the model has to enable the integration of different sources of perception.

Up to this point of design of PAM, to cover the aspect of data collection just common fieldbus systems like LonWorks or EIB had been considered. However, this limitation entailed restraining the quantity and, specially, the kind of data that could be collected, which would affect the efficiency of the whole perceptive awareness process. Covering this deficit, the fourth step of the development process of PAM refers to the different sources of perception that the automation system has to be capable to integrate.

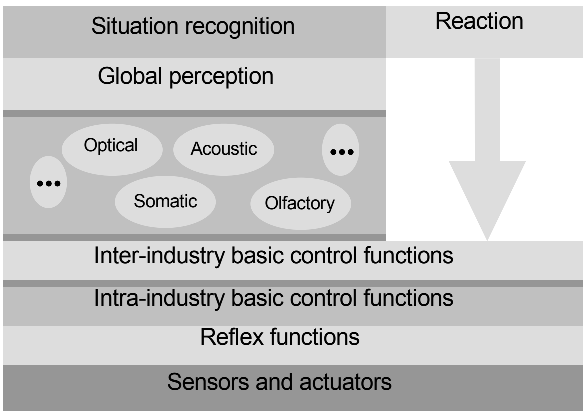


Figure 40 Basic structure of the fourth defined PAM

Figure 40 shows the resulting PAM. A first remark has to be made concerning the functionality of the third and fourth layers. Both layers were defined to cover the lack of understanding within and between industries - particular areas in automation. The prefixes ‘intra-‘ and ‘inter-‘ specify respectively an application that either concerns just one particular area or that requires different areas to work together. But besides, these layers were also considered responsible for the creation of images of the current situation both individual and associative. The analysis of the model leads to the conclusion that in order to support integration of different sources of perception the structure of PAM has to be changed and extended. Since the third and fourth layers concentrate on fieldbus technologies, it does not seem to be reasonable to integrate the function of image creation at them. Images have to be built in an upper layer, which has to be accessible to each one of the technologies that can be utilised to collect data.

As it is defended in neural science books such as [Kan 00], initially input data from the different senses is processed separately and afterwards connections enable data association in various locations in the human brain. According to this statement, the analogy in automation demands new layers in the architecture for automation systems where data has to be first interrelated as individual images and later as associated images.

3.2.5 PAM: Attributes - Objects

In the fifth step some detailed considerations are made concerning the ideas that were developed at the fourth step of the design process, i.e. in reference to the way a situation has to be perceived. Still considering an analysis of the way human perception occurs some additional requirements appeared.

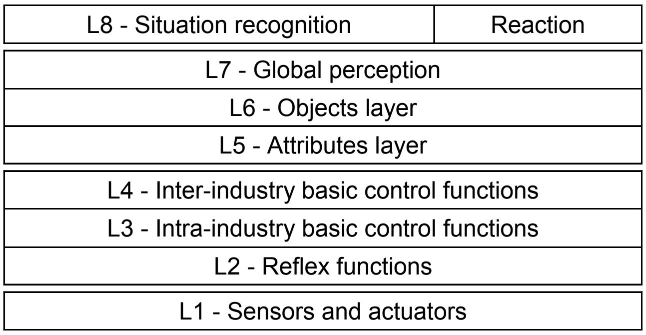


Figure 41 Basic structure of the fifth defined PAM

As it had been mentioned, humans do not just perceive individual data points but specific compositions, which allow humans to recognise what they perceive through single data points. In order to enable this in automation systems, some kind of data ordering process is needed to make possible the association of data coming from different sources of perception.

While thinking on adding this exposed function into the perceptive awareness model some conceptual connections to object oriented programming are required. As it is exposed in [Kur 97], object oriented programming talks about creating objects that show particular characteristics. Making some kind of analogy to human perception the individual characteristics exposed in object oriented programming are analogue to the single perceived data points, at the same time that each object can be functionally equalled to the complexes that humans perceive.

In Figure 41 the resulted PAM is represented. Layers three and fourth (L3 and L4 in Figure 41) assure respectively understanding within a particular industry, e.g. lighting, and between industries in a specific technology, e.g. LonWorks. Going a step forward, the fifth and sixth layers (L5 and L6 in Figure 41) work on benefit of associative work. At these levels of processing data has to be equally treated and combined relaying on the conceptual consistence of the association and completely independent of the data source.

The first action towards this equal treatment occurs between layer four and five (L4 and L5 in Figure 41). Data formats are first translated into the system's global format (symbols) so that afterwards, data belonging to a same measured or monitored parameter can be submitted to data validation and reduced to an only one value, which receives the label of attribute, in an easier way. Consequently, the state of an object is known up to the state of its attributes (significant parameters of an object).

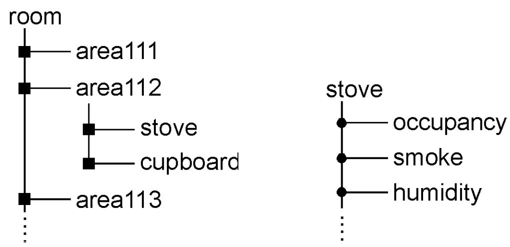


Figure 42 Objects and attributes

Each object is a meaningful entity such as a spatial area, a household electrical appliance or a person in the monitored space where the automation system has been installed and can contained another objects (Figure 42).

3.2.6 Dynamic Module

Up to this point PAM has been designed considering moment data like state on/off. In such a continuous monitoring, collected data answers to the transient changing particular state of the monitored or measured variables. However, the momentary value of the different variables may not always be enough. There may be scenarios where the action of change is a decisive event. In other words, the dynamism of the system has also to be considered.

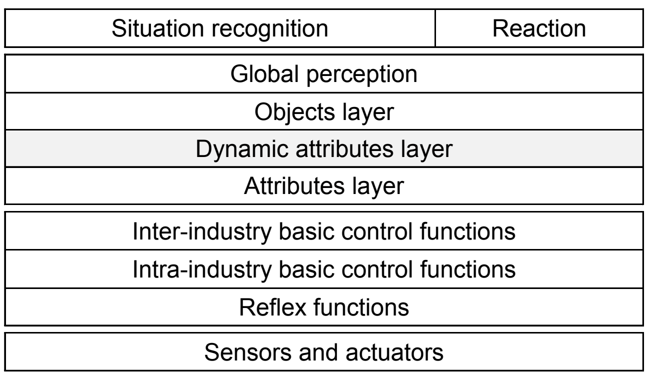


Figure 43 Dynamic Attributes Layer

At this sixth step of the design process of PAM in order to meet this requirement a new layer is defined above the existing attributes layer. This new layer is responsible of the creation of dynamic attributes from the static ones. While integrating this new function parameters such as movement's directions, which are defined from two consequent positions, can be taken into account by the automation system.

3.3 End Result

Figure 44 shows a rough structure of the end design of the perceptive awareness model. A brief analysis of the model shows that the first three layers could also respond to a common automation system in building and home automation. At this level, data processing is based on simple basic control rules, which produce one result or another just depending on particular inputs. During this work this comportment is named pure-reactive behaviour and it is defended that current common automation systems follow this pure-reactive behaviour philosophy.

Though common automation systems are appropriated for simple common applications such as turning on the light when somebody comes into a dark room, they are not suitable for more complex applications. For example, as soon as somebody comes into a dark room the presence sensors detect the person. Automatically the system turns the light on without taking into consideration any other variable but the luminosity value of the room and the sudden presence of somebody entering. In case there was gas at the room and some defect in the electrical circuit these events would not be appreciated by the system, and therefore, by means of using a common automation system, the reaction of the system could result in an explosion. In order to avoid such situations and extending the capabilities of common automation systems the upper layers of PAM are defined.

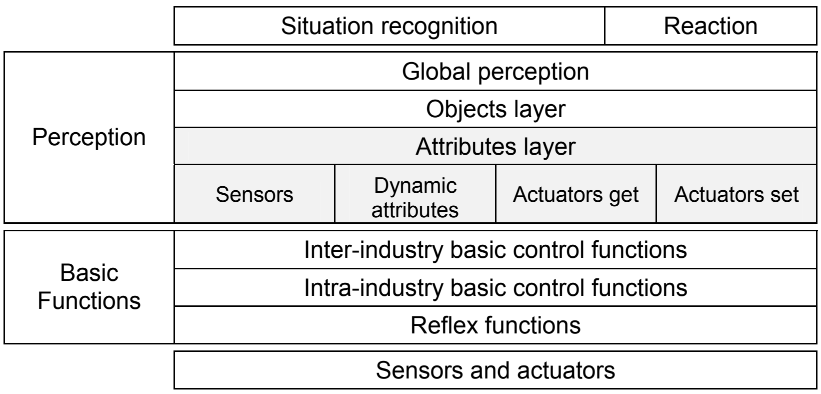


Figure 44 Structure of the PAM

These layers have to meet the demands of following higher functions, which are required to reach the preventative behaviour in automation systems:

- Take into account the global situation
- Consider and evaluate the possible side effects of the reaction before embarking upon it
- Once the evaluation has been decide if the resulted action should be executed or if a different one has to be evaluated

These higher functions require activities like data unification, data validation by means of for example comparison techniques, attention by means of methods such as data prioritisation, perception of complexes, recognition of global scenarios and selection of the proper preventative reaction according to the current recognised situation, as it is represented in Figure 45. In the next sub-sections a detailed description of the end-result model is presented.

3.3.1 Environment-System Interface Layer

According to its functionality, the first layer of PAM acts as the interface between the outside world and the system itself. Through this first layer functions between the environment and the automation system are carried out in both directions. While lots of sensors enable the system to collect a large amount of data about the environment, the system reacts on the environment by making use of its actuators.

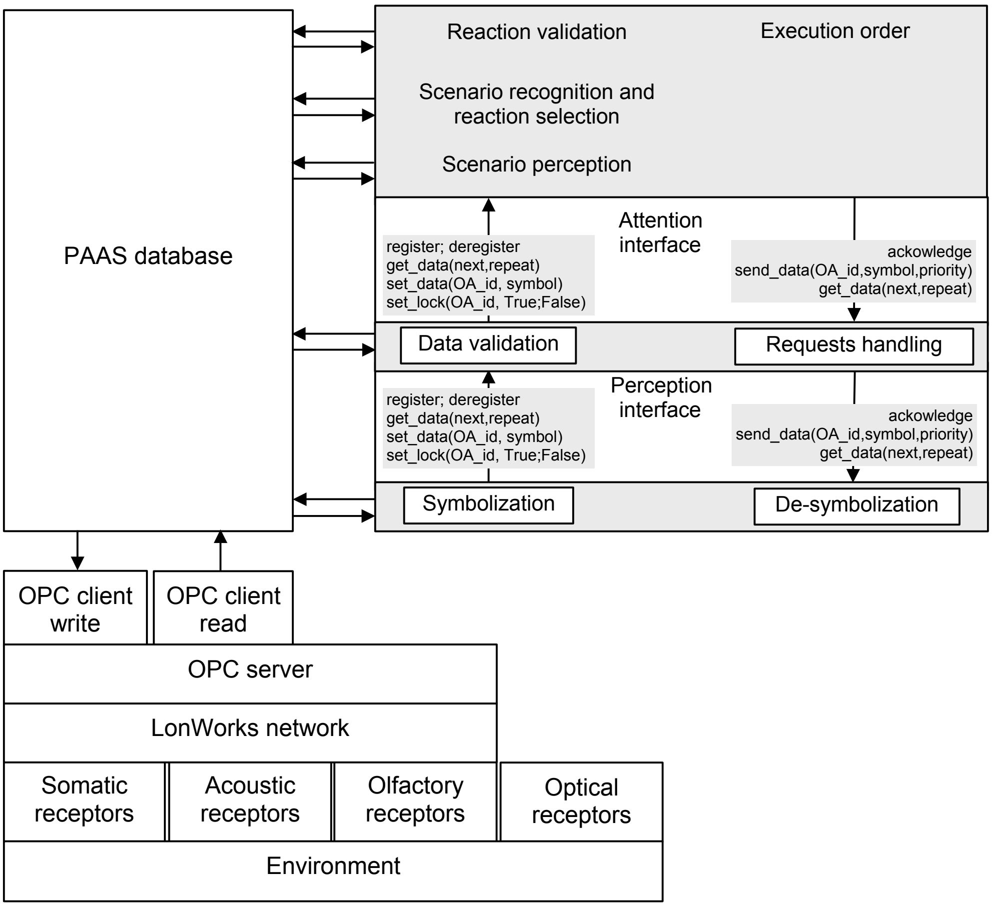


Figure 45 Structure example of a perceptive awareness automation system (PAAS)

Sensors – Diverse sensors such as temperature sensors, cameras and microphones are required to cover the various faces of perception. Moreover sensors present particular peculiarities depending on the technology they are related to. While from the physical point of view visual and acoustic data grabbing is supported by the simple installation of cameras and microphones, a more costly task is required in order to capture data related to other sectors such as olfactory and touch. In these cases a huge amount of sensors attending to different parameters like pressure or temperature can be used. However, when referring to the software that is needed to let the system knows the state of the environment the situation changes. In this case, visual and acoustic perceptions require much more complex software than olfactory and somatic perception. Since characteristics concerning data transmission change depending on data type this variety of sensors entails distinct technologies.

Actuators - Nowadays, due to growing use of fieldbus technologies in automation, most automation systems influence the environment by making use of actuators that belong to a fieldbus network. However no constrain has to be made at this point. In such a way, PAM has to be designed to enable the automation system to integrate different technologies that are suitable for acting on the environment. Moreover, though fieldbus systems can influence the environment in many different ways thanks to the enormous amount of already developed actuators, they are not suitable for every kind of reaction. For example, as long as the acoustic reaction just consists on an alarm this can be implemented by simply utilising a relay of the fieldbus network to open/close the electrical supply circuit of the alarm. Nevertheless, in case the acoustical signal does not just consist on an alarm but on some words a different and more appropriate technology should be used. Once more different technologies are needed.

3.3.2 Basic Control Functions Layers

At the second layer begins the data processing. This means that this layer is already related to system behaviour. There are different behaving levels depending on the complexity of the action behind. At this second layer the behaviour of the system has to be assimilated to reflexes and basic control functions in human beings.

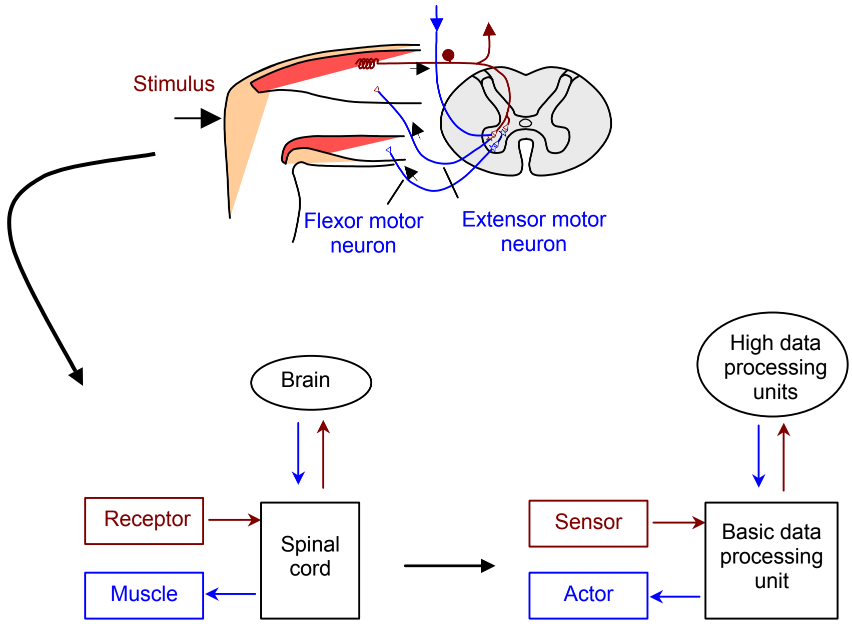


Figure 46 Reflexive actions: a) human reflex, b) PAAS reflex

Reflex Actions - Reflexive movements as the one shown in Figure 46 are involuntary coordinated patterns of muscle contractions and relaxation elicited by peripheral stimuli. As it is exposed in [Kan 00], if external conditions remain the same, a given stimulus will elicit the same response time after time. However, the intensity and the local sign on reflexes can be modulated by mechanisms that switch the patterns of connections of afferent fibres to spinal interneurons and motor neurones depending on the context of the behaviour.

Some biological reflexive movements are related to protecting the species, and in a similar way the reflex functions layer is defined in PAM to support these kinds of protective reactions in a perceptive awareness automation system (PAAS). Consequently defensive actions like visual or acoustic alarm when gas is detected, which have to be ordered without requiring any ostentatious processing and without delay, are supported at this second layer

Figure 46 shows the similarity between reflexive human movement and PAAS reflex action. The input received by PAAS plays the role of the stimulus in the reflexive human action. This signal flows into the system to the second layer. At this layer the signal is processed in a similar way that the signal produced by the stimulus is processed at the spinal cord. Once processing is completed the answer signal flows back into the first layer resulting on of some output – activation of actor(s) – alike as the extension and contraction of some muscles.

Though reflexes are not considered voluntary actions, it does not mean that they are out of control. In fact human reflexive movements are not immutable and as it is defended in [Kan 00f] the reflex patterns produced through spinal circuits can be converted from one set of movements to another by signals from higher levels of the nervous system. In the same way reflex actions in perceptive awareness automation systems (PAAS) have to be suitable for being controlled by the higher layers. This control is possible thanks to different kinds of connections between upper and lower layers (as it is represented in Figure 46), which work for the benefit of proper system behaviour at any time.

Basic Control Actions - PAM presents different reaction levels that can be related to different human conscious behaviours. At the third and fourth layer PAM deals with specific conscious actions.

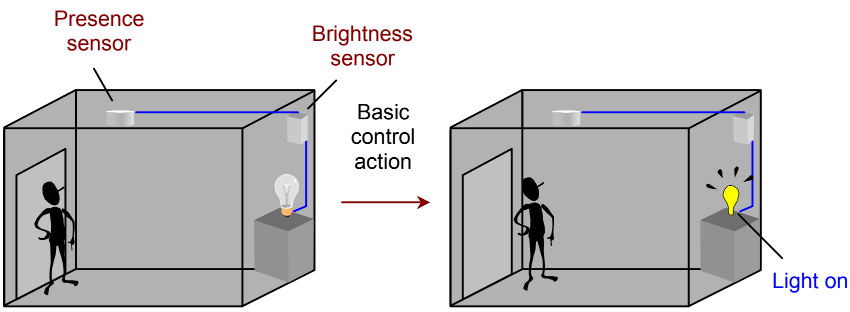


Figure 47 A basic control action in a PAAS

These layers are defined to cover automation applications in a compatible way to common automation systems. In comparison to the reflex actions these applications do not depend on a specific event but on a combination of particular events. Taking as example the light control system one sees that in order to properly react to the light situation, the system does not just requires the input coming from the presence sensor but also, at least, a second input coming from the brightness sensor (Figure 47). The basic light controller reacts depending on the values received from these two sensors. The controller delivers either the output that makes the actuator turn the lamp on, or the output that makes the actuator turn the lamp off.

When analysing these actions one realises that though the basic controllers have to know the values of particular inputs and though some control rules have to be established, the resulting behaviour is still far away from a preventative behaviour. The analogy to this behaviour is also found in biological system. In [Kan 00f] it is explained how a cat behaves with its cervical cord severed. The animal, if provided with body support, can walk on a moving treadmill and bring its paw around an obstacle after hitting it. But it cannot lift its forelimb before impact with an obstacle, as an intact animal does, because this movement requires control of the limbs using visual information. This anticipatory control requires intervention by the cortex – higher data processing unit.

Therefore, at this level of processing, action-decision units do not take care of any other variable of environment that does not appear as parameter of the control rules, which drive the behaviour of the controllers. Therefore, although at this third layer the system is not completely blind in front of the environment, it neither pays attention to the global situation but just to fixed events. This kind of control loop actions does not entail large data processing and is the working base for most common automated systems nowadays as it is defended in [Die 01, Tam 01].

At the third layer PAM works on the level of individual specific industries. Some organisations like LonMark Interoperability Association [Lma 02] have defined tasks groups such as heating, ventilation and air conditioning (HVAC), lighting, security, building automation systems (BAS), and sunblind. The definition of these different areas of application is performed in order to cover properly the various markets. Each task group defines profiles to assure the interoperability at its particular industry. Though by defining profiles it is pretended to make easier the communication between different devices from different vendors, there are some limitations. As it is described in [Rau 99] and presented through several application examples in [Rus 01] in most cases co-ordination is not supported between industries, which involves communications' disadvantages. Experiences in the area of automation using fieldbus technologies confirm that data from one industry is in some cases demanded by a different one, as it has been exposed in section 3.2.1. The fourth layer is defined in PAM in order to support understanding between industries. This understanding entails, for example, co-ordination between the various profiles defined by the different 'task group' when talking about LonWorks technology.

3.3.3 Perception Layers

The previously exposed layers do not differ much from the structure that would come out when analysing common automation systems implemented by means of a fieldbus technology. At the perception layers the high data processing activity, which contributes to the system's preventative behaviour, starts.

Attributes' Layer – As it has been previously discussed, in order to cover the different faces of perception the system requires of various kinds of sensors. Since characteristics concerning data transmission change depending on data type this variety of sensors entails distinct technologies. Consequently, for each source of perception data adaptation is required to convert the different data formats to a system's global data format to make easier the groups' data processing.

As it is exposed in [Kan 00e], this action is also supported by human nervous system. Since there are different kinds of sensory information, and the nervous system has to process them together all input data is submitted to a conversion process known as transduction, which consists on specific mechanisms depending on the kind of input signal to translate the stimulus.

Once validation has taken place, at this layer data is organised into different groups depending not only on the kind of information that each particular data represents, for example: temperature, occupancy, power, etc, but also on its location at the monitored room. In such a way the system observes the situation through, what could be called, individual images (Figure 48).

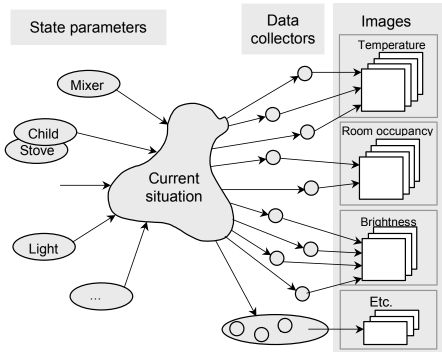


Figure 48 Attributes' image composition

Objects' Layer – Similar to the associations that are mentioned by neural-psychological works when describing human perception, layer five is defined for presenting the data, which at the previous is organised layer into functional groups (attributes), into entity groups (objects).

Among these studies, those concerning human optical sense system such as [Kan 00e] defend that in a scene the different objects are separated and that those objects of interest are distinguished from the background. The individual signals produced by these objects have to be grouped in order to be useful to the system, as it is shown in Figure 49.

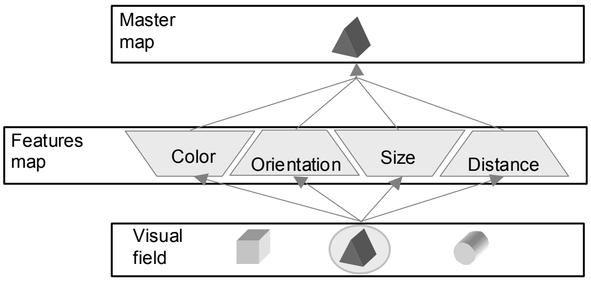


Figure 49 Humans visual processing

The objects layer of PAM is defined to work for the benefit of such associations and complexes' perceptions. Under the term entity one has to understand each one of the different objects of the environment where the system is located. For example, since the whole system is confined to a room, the monitored space is divided into various physical parts called 'areas'. Each of these areas is an object and shows determined area-features such as temperature, brightness, occupancy, humidity, water and smoke (Figure 50).

Continuing the analysis of human perception, it is known that a detailed perception of the whole environment does not take place but attention is paid to particular objects or events at each situation [Kan 00e] (Figure 49). Therefore, additional to the presentation of data into different entities, layer five is responsible for events' prioritisation too. Similar to human beings, the system has to give different priority grades to the various happenings so that the most important events are first considered, and gradually the less relevant are taken into account. As result the system perceives the current situation through objects perception.

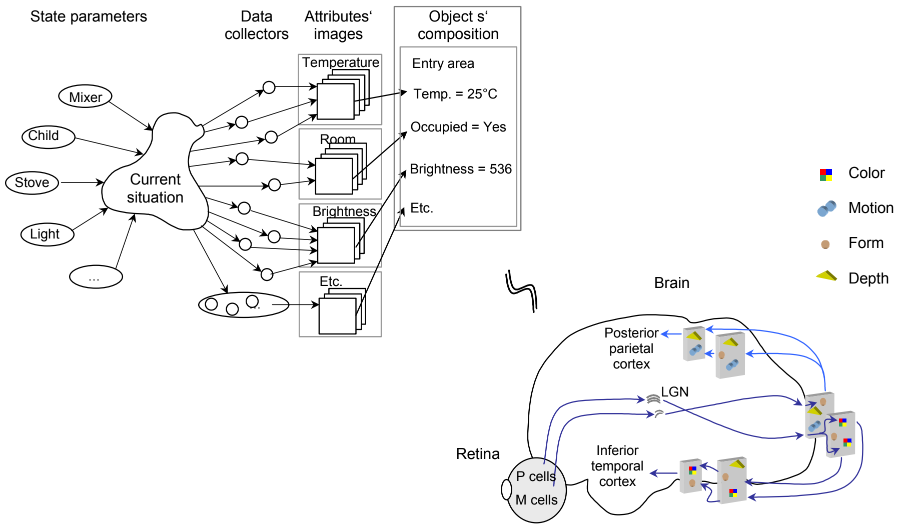


Figure 50 a) Attributes' composition in the PAM; b) human visual features association

3.3.4 Recognition Layer

The 'situations' recognition layer' is defined in order to cover the function of recognising the situation. In humans two requirements have to be meet to support recognition. On one hand previous experiences are needed, otherwise it would be not possible to talk about recognition of something that it is unknown. On the other hand, as it is defended in [Smi 99] recognition occurs as result of comparisons between the present situation and the previous experiences.

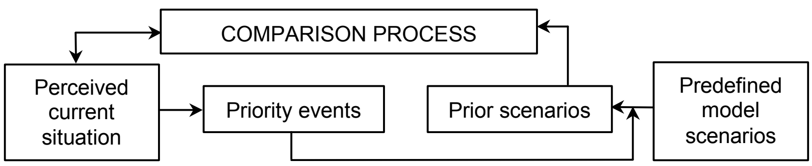


Figure 51 Comparison process in the PAM

In PAM the analogy to human previous experiences are predefined model scenarios. These predefined model scenarios have to be specified for each particular perceptive awareness automation system (PAAS). Consequently, the set of model scenarios plays the role of long-term memory of the PAAS. However, though there is a tight connection between these memories and the 'situations' recognition layer', the creation and location of these scenarios does not directly involve the layer itself but the database of the system, topic which will be discussed lately in section 3.3.9 Database. In order to recognise what the system perceives, layer six has to carry out with comparison processes. These comparison processes consist on contrasting the current perceived information about the state of particular attributes of specific objects to the state of these variables at the model scenarios. Recognition is reached as soon as the perceived current situation matches a model scenario, as it is represented in Figure 51.

Though the process mainly consists on the tasks above exposed, some important aspect, like prioritisation, have also to be supported at this layer. By means of prioritisation the system recognises first those situations that are considered more important, working out less-significant situations after important ones have been treated. As result the system is conscious of the current situation.

3.3.5 Reaction Layer

The reaction layer is responsible for assuring the preventative reaction of the system. In humans the reactions to concrete previous experiences are kept in mind in connection to the past experiences, which influences human behaviour as it is mentioned in [Jac 97]. This fact allows humans to know which are the proper reactions in front of the different situations they face. Translating this ability to automation systems means that proper reactions to each model scenario have to be stored and related to the scenario, so that the system knows what to do in front of each recognised situation.

Additionally, and in order to support prevention the system has also to assure that the chosen reaction entails no undesired side effects. PAM's prevention method consists on theoretically adding the selected reaction to the recognised situation (see section 3.2.3).

3.3.6 System Management Unit

Apart from the presented layers PAM integrates a system management unit. A main task of the management unit is to control the data flowing through the different layers.

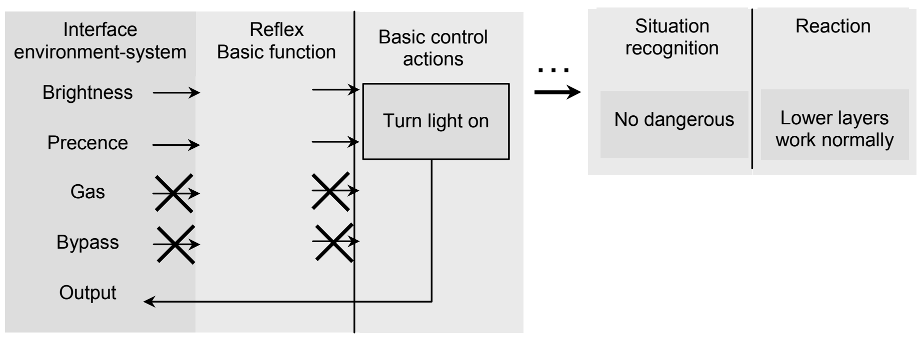


Figure 52 Data flowing example A

In such a way, the management unit is responsible for the transfer of data between layers as well as for the reaction handling, i.e. which layer orders the proper reaction to the current situation at each moment.

The system management has to determine how data has to fluctuate from one layer into another and even when the fluctuations have to take place through all layers and when they have to spring from one layer to a non-successive one. Since the system management unit decides from which layer the system influences the outside world, this unit is responsible of the behaviour of the system.

To clarify this last point one can consider the situation when somebody comes into a dark room. As it is represented in Figure 52, in a normal case the automation system reacts by turning on the light (reaction that comes from the third layer ‘intra-industries layer’).

In case there was some inflammable gas in the air and the power cable was damage at some point, the presented reaction could result on an explosion. PAAS has to avoid such things happening. This means that once the system has recognised the inflammable gas and that the power cable is damage, a signal from the upper layer prevent the third layer from reacting, which is shown in Figure 53.

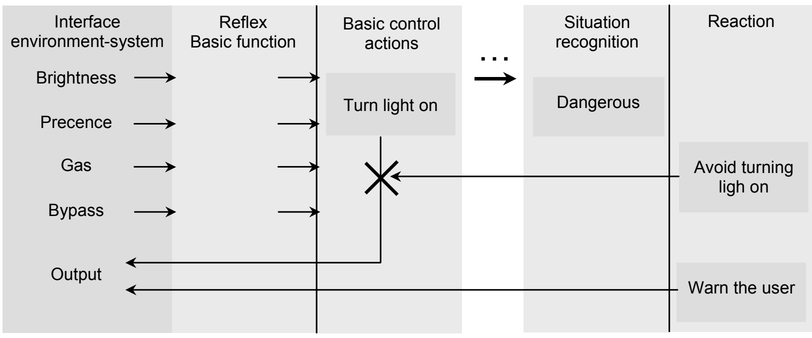


Figure 53 Data flowing example B

In addition to this preventive action the system warns the user about the situation. Of course, there are many normal situations in which reactions come from any of the basic control function layers. In these cases, though no reaction is expected to come from upper layers, it is obvious that the preventative process has to keep on running. In such a way, as soon as any strange event occurs, the system is on the mood to detect it. Supporting these functions the system is always ready to react appropriately.

3.3.7 Database

At the section 3.3.4 an introduction has been already given concerning predefinition and storage of scenarios, which play the role of long-term memory of the system.

According to [Kan 00i], human memory consists on three different kinds of memory: sensory memory, short-term memory and long-term memory. Sensory memory acts as buffer for stimuli that are received through the senses and a sensorial memory exist for each sensory channel. Short-term memory is required for temporary recall of information. Long-term memory is intended for storage of information over a long time. Information passes from sensorial memory to short-term memory by attention, filtering the stimuli and just attending the most interesting ones at a given time.

In a similar way to humans PAM deals separately with three kinds of memory. Sensorial memory can be assimilated to a momentary recording of information. This memory is not-interpreted by the system. Information only lasts a few seconds before being replaced by a new measured or detected value. Short-term memory is required for encoding and recalling of explicit knowledge. It is defended that short-term memory requires of an attention control system, with limited capacity that actively focuses perception on specific events. This attention control system is related to cognitive processes that determine the available information that will be used and the one that will be ignored. This means that the flow of information in the system is interactive.

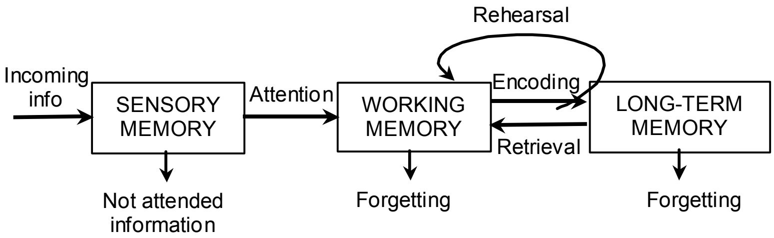


Figure 54 Model of the human information processing system

As described in [Thi 03], sensorial information streams bottom-up to more central stages at the same time that attention, driven by mental representations, modulates (top-driven) sensorial information. Figure 54[2] shows a reference model of how information flows through the different levels of human memory and which are the facts – functions - that are required to go from one level into the following. As it is graphically described in Figure 54 not attended information does not reach higher levels and information at higher levels can also be forgotten.

Considering human long-term memory, it is known that this memory is created, develops and is dynamically modified all through human life [Geo 03]. The analogy in automation systems would involve learning processes. Although conscious of the limitations that it entails, this first version of PAM does not cover dynamic artificial learning but works with a dynamic long-term memory, a predefined model scenario database is created on the bases of user's needs and wishes. Since the proper behaviour of the system depends on the information stored in the databases. These have to be suitable for easy access and updating in order to assure proper running of the system.

3.3.8 Data Processing

An example of the data flow in PAM is shown in Figure 56 at the last page of this chapter. Part of the lower division of the figure, which represented in hell grey, describes a model that could be used as reference for common automation systems. In this lower part, collecting data from the environment is performed through different kind of sensors and while some sensors such as smoke sensors directly act on the actuator when activated (i.e. sensor and actuator are physically connected) there are some others that are suitable for executing basic control functions. These second kind of sensor, which can be called smart sensors, are equipped with microprocessors and communicate to the actuator through the fieldbus without requiring direct physical connection, e.g. multiple sensor to control light such as Helio Multi-Sensor from the company Philips [Hel 03]. In this case both, sensor and actuator, are components of the same fieldbus system. There is a third kind of sensor such as brightness sensors (either smart or not) that sends the measured data to a basic function control unit. The control unit, while considering the values from different sensors processes the information and acts on particular actuators.

[2] Not attended information refers to the information humans do not pay attention to. In relation to forgetting: it is know that humans forget, however it is still unknown how this process happens.

But the designed perceptive awareness model (PAM) enables the integration in the automation system not only of common fieldbuses such as EIB/KNX and LonWorks, but also of different technologies related to data collection such as pattern recognition systems and sound recognition systems. Considering, for example, visual pattern recognition the camera plays the role of sensor, which captures images that arrive to the image-processing unit. Integrated as part of a perceptive awareness automation system (PAAS) the image-processing unit would extract the relevant information from the received picture and forward it to upper processing control levels.

All data, independently from the sensor, has to reach the upper part of the model (upper part of Figure 56). Due to today's automation technology, among other requirements this action needs a data access interface to enable data transfer from the 'peripheral system databases' (e.g. LonWorks Network Service (LNS) database when referring to the LonWorks network) to the database of the PAAS. As it is explained in detailed in chapter 4, OPC (Object Linking and Embedding (OLE) for process control) appears as an example of a possible solution when talking about fieldbus systems.

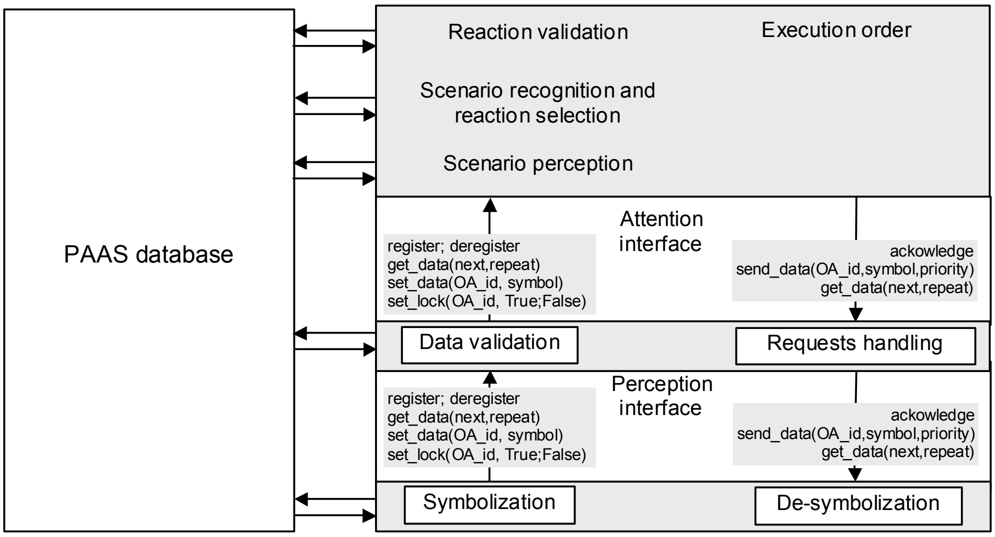


Figure 55 Structure example of the upper processing layers of PAM

Besides data access, data format is another aspect to consider. A global data format is required in order to enable an easy and efficient higher data process. The example presented in Figure 56 shows a possible solution to give equal format to different inputs that refer to the same parameter 'distance'. A distance sensor – component of the fieldbus network - measures the variable 'distance'; at the same time the system receives information concerning this parameter from the visual data source (pattern recognition system). The distance sensor measures distance in cm while visual pattern recognition supports a range format. Symbols are used in order to achieve general format. The 5 cm that are measured by the distance sensor are considered a 'too close' distance and related to the numerical symbol '1', on the other hand information extracted from pattern recognition is also transformed into such numerical symbolisation – in this example it also transmits the value '1'. Obviously, the symbol code has to be communally defined for all data sources. Once communal data format is supported the process of data validation, which works on benefit of a more reliable further data processing, can be easily worked out. The demanded modularity of PAM requires of an interface, which has been called perception interface, between the functions symbolisation (equal data format) and validation (Figure 55).

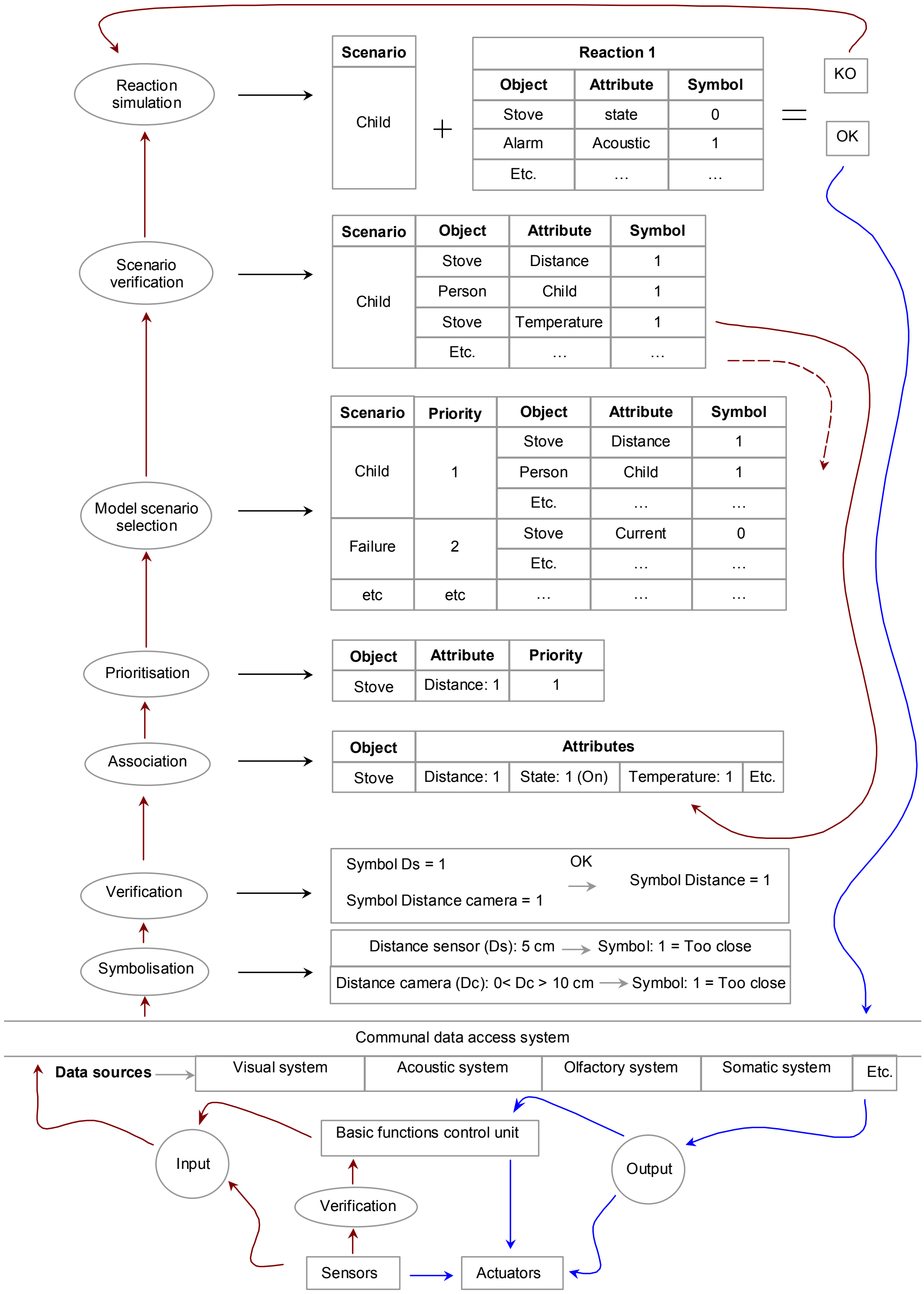


Figure 56 PAM data processing

Data values that result from the validation process are sent through the attention interface (see Figure 55) to a higher level where verified data is submitted to different processes of association and comparison. There are two main kinds of association process. First association consists on grouping parameters belonging to the same entity. In Figure 55 an association is presented in regard to the object ‘stove’. The standing of the object ‘stove’ is known from the status of its attributes, i.e. distance, temperature, state, etc. Second association entails grouping different objects into a common significant context, i.e. into scenarios.

Additionally, a specific status of a particular attribute of an object can be signed with a priority label. In the previous example priority has been given to the attribute distance when its status is ‘1’. In those cases in which an attribute has been assigned with a priority label, the system searches the current situation between the model scenarios (past experiences of the system) in which the particular parameter appears. Model scenarios are also prioritised, so that the system takes this prioritisation into account while going through them during the searching process. Since not every parameter is prioritised, the selected model scenario has to be verified. At this point verification consists on comparing the symbol values of those parameters that also play a role in the selected scenario, though without having a priority label, to the current symbols of these attributes at the short-term memory of the system. Once a scenario has been verified a correspondent proper reaction is delivered, which the system validates before embarking upon its execution. Depending on the effects of the reaction the next step followed by the system is either telling some selected actuators to do a specific task, which is part of the top-down information flow of PAM, or select a new reaction that better met the current situation.

Since there are no presumptions made as to whether processing is carried out in hardware or software this approach presents an ideal tool for designing systems. Once designed, the choice of implementation strategy can be tailored to use suitable technologies to meet time constraints or re-use existing modules to reduce cost and uncertainty.

Chapter 4

The Function of Perception in PAM

In order to expose the principles of nature that have been taken as reference to design the function of perception of the perceptive awareness model (PAM) in the first part of this chapter human perception is briefly exposed. Afterwards, in the second part of the chapter the function of perception of PAM is designed in base to the principles exposed in the first part.

4.1 Humans Perception and Technical Analogies

Over the last years researches and studies in the field of neural science have contributed to acquire knowledge about the way human perceive. Among the various outcomes, following points and statements, which have been extracted from [Kan 00e], are taken as reference to design the function of perception of the perceptive awareness model (PAM):

- Important elements of human perception are:
  - Various sensory receptors and the stimuli they respond to,
  - The pathways that carry information from the receptors to the cerebral cortex.
- Stimuli are submitted to a signal unification process called transduction. Sensory receptors transform a stimulus into electrical energy, thus establishing a common signalling mechanism in all sensory systems.
- Human perceptions are not direct copies of the world around us, but the brain constructs representations of external events and objects.
- The functional and anatomical organisation of human sensory processing networks is hierarchical.

In reference to the first of these statements and as it is mentioned in neural science books such as [Lyd 99, Kan 00] humans have four kinds of receptors that are sensitive to different physical energies - mechanical, chemical, thermal and electromagnetic. This fact entails that each one of these four kinds of receptors require a specific mechanism (called transduction) to translate the stimulus into electrical energy - a common signalling mechanism. Since these sensory receptors are the structures that enable the initial communication contact with the external world they can be seen as the first units in human perception process: they are the first cells in each sensory pathway. Sensors can be seen as the technical analogy to human sensory receptors. Starting form temperature sensors as analogues to human receptors sensible to temperature, until video cameras playing the role of eyes of a system.

When encoding information the various sensory systems behave in different particular ways. For example, taste depends greatly on receptor specificity while the differentiation of sounds depends mostly on pattern coding. However, though the various sensory systems differ in their modes of perception, all them share three common steps:

- A physical stimulus
- A set of events transforming the stimulus into nerve impulses
- A response to this signal in the form of a perception or conscious experience.

Figure 57 shows the process of transduction mentioned in the second statement. As it is explained in [Kan 00e], in response to the stimuli, for example a single photon of light that stimulates particular receptors in the eye, individual sensory cells generate rapid changes in membrane potential, which produces neural signalling. This signalling depends on the ability of nerve cells to respond to these small stimuli by producing rapid changes in the electrical potential difference across nerve cells. In such a way, information is carried by electrical signals that are produced by temporary changes in the current flow. These changes are mediated by heterogeneous ion-channels, i.e. different types of channels in different parts of the nervous system can carry out specific signalling tasks.

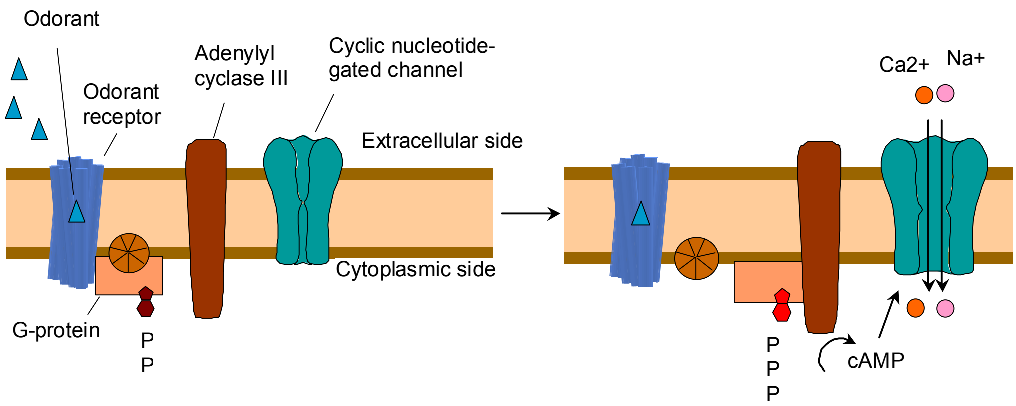


Figure 57 Olfactory signal transduction. Binding of an odorant to an odorant receptor causes the receptor to interact with a G protein, which stimulates adenylyl cyclase III. The resultant increase in cAMP opens cyclis nucleotide-gated cation channels, leading to

The analogy in technology can be found while analysing fieldbus. On one hand, the different fieldbus sensors transform the collected 'impulse' into digital signal. On the other hand, fieldbus systems make use of different channels depending on the kind of signal that has to be transported.

In the sensory pathways neurones link receptors at the periphery with the spinal cord, brain stem, thalamus, and cerebral cortex.

The functional design of neurones attending to signal transfer is determined by two principal parameters. First maximum computer power of the nervous system - neurones must be small so that large number of them can fit into the brain and spinal cord, and second maximise ability to respond to changes in environment - neurones must perform signalling rapidly.

However the materials from which neurones are made constrain these design objectives. On one hand nerve cell membrane is very thin and surrounded by a conducting medium, which produces a very high capacitance slowing down the conduction of voltage signals that must flow through poor conductors. On the other hand the ion channel that gives rise to the resting potential also degrades the signalling function of the neuron making the cell leaky. These facts contribute to limit the distance that a signal can travel passively. In order to compensate these physical constrains human nervous system makes use of voltage-gate channels. In this case there is also a correspondence between fieldbuses and human nervous system. In a similar way that human nervous system uses voltage-gate channels to improve the quality of the signal and enable longer transmission distances, fieldbus systems makes use of repeaters in fieldbus systems [Die 00b, Loy 01].

In reference to the third statement, and as it is described in [Kan 00e], human perceptions differ from the physical properties of stimuli because the nervous system operates by extracting only certain pieces of information from each stimulus, while ignoring others. Afterwards, the brain interprets this extracted information in the context of its intrinsic structure and previous experiences. For example, humans encounter chemical compounds floating in the air or water and experience them as smells and tastes. Colours, tones, smells and tastes are mental creations constructed by the brain out of sensory experiences. In such a way they do not exist outside the brain. Does a waterfall make a sound if no one is near to hear it? Sound occurs only when pressure waves from the waterfall are perceived by the brain of a living being.

Considering this point no analogy is found between today's automation systems and human nervous system. Of course, fieldbus systems do not attend to all information of the environment. However, this fact is not due to an attention process but to the reduce number of sensors that the systems integrate.

In relation to these representations, human brain organises object's essential properties letting humans handle objects appropriately. The complex qualities of sounds, visual images, shapes, tastes, and odours require the activation of many receptors acting in parallel, each one signalling a particular stimulus attribute. Afterwards, as the signals converge on processing centres in the central nervous system the messages of individual sensors are integrated into an associative perception. With that sensory perception derives from conveying of sensory information from simultaneously activated receptors where the information is processed in parallel pathways before it is combined in the highest centres of the cerebral cortex.

Considering the fourth statement Kandel et al report in their work [Kan 00e] that sensory information in the central nervous system is processed in stages, in the sequential relay nuclei of the spinal cord, brain stem, thalamus, and cerebral cortex. Each of these stations brings together sensory inputs from adjacent receptors and transforms the information to emphasise the most relevant signals. Consequently it can be said that the pathways of human nervous system have a serial organisation: receptors project to first-order neurones in the central nervous system, which in turn project to second and higher order neurones resulting on a distinct functional hierarchy. Once more, no analogy is found between automation systems and human nervous system at this point. At most, fieldbus system associate a few inputs by sending them to a control unit, which depending on fixed rules processes these inputs and generates a output.

During the following section the function of perception of PAM is presented. This function has been designed having in mind the mentioned deficits of automation systems perception in comparison to human perception and trying to develop solutions to cover these deficits.

4.2 Automation System Perception

Since there is no reason to limit the faculty of perception to the area that is being occupied by the perceiver the perceptive awareness model (PAM) is developed to enable not only local perception but also remote perception. While the term ‘local perception’ is defined to refer to data perceived from the environment around the system, the term ‘remote perception’ is defined to indicate data that the system perceives out of the controlled environment such as orders send by the user through Internet.

4.2.1 Local Perception

Although the way contemporary automation systems perceive differs from human perception there are some first similarities that hold the idea of extending automation systems perception on the basis of human perception. Sensors are the analogy in automation systems to human sensory receptors. Stimuli are equivalent to changes in the monitored and/or measured parameters the automation system attends to, and the connectors between sensors and control units, i.e. the transport media, are the analogy to the pathways from sensory receptors to cerebral cortex. Furthermore, the analogy even reaches the way of doing of the sensorial units, which constantly listen to their surroundings in both cases humans and automation systems.

At the sensor or receptor level, the main difference concerns the range of perception and redundant data. By making use of particular kinds of sensors, which are appropriate for common technologies used in home and building automation (HBA) such as fieldbuses, automation systems cannot support all aspects of perception. In order to cover this deficit, the perceptive awareness model (PAM) is designed to support the integration of various kinds of sensory techniques to embrace the different domains of perception:

- Cameras to cover visual perception
- Microphones that work as voice and sound receptors
- Gas sensors to perceive smells
- A combination of temperature, pressure, humidity, and distance sensors, ammeter, voltmeter, etc. to support somatic perception

A second difference refers to the quantity of receptors. While human body is equipped with billions of receptors [Kan 00e], until now only a minimum number of sensors have been placed in common automation systems. Redundancy and perception from different sources were simply not cost effective. However, this may change in the next years with the advent of very low cost and low power wireless ad-hoc sensor networks [Rab 00, Sun 01], that are capable of organising themselves aggregating data and providing much more reliable information about the environment. Since, as it is explain in [Mah 02], these sensor networks can be integrated into fieldbus existing systems, it will be possible to use all these information as input for perceptive awareness automation systems (PAAS).

In reference to signal transfer, there are works such as [Jov 97] that discuss the equipment that is needed to transmit data from the data collectors to the inside of the system. Nowadays, considering home and building automation (HBA) common technologies used in this field such as LonWorks and EIB/KNX are not suitable for transferring every kind of data. Consequently, in a similar way that human nervous system supports different ion channel to transport distinct signals, different transport media are required in HBA. For example, for video applications the system needs proper technologies such as IEEE 1394 in order to meet the requirements of the application, as it is the case presented in [Ski 01].

The main disadvantage of using distinct technologies to integrate various kinds of data receptors is related to data format. Different technologies may represent collected data in different formats. Consequently, similar to the action of signal translation – transduction - executed by sensory receptors in biological systems PAM integrates data unification functions to face the problematic of data format discrepancy.

In this search towards the analogy between human perception and the perception function of perceptive awareness automation systems (PAAS), from the four statements pointed out at the beginning of the previous section concerning human perception two points are still opened. One refers to the declaration that human perception is not a direct copy of the world around us, but the brain constructs representations of external events and object. The other one considers the assertion that the functional and anatomical organisation of human sensory processing networks is hierarchical.

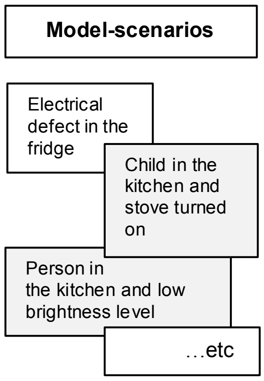


Figure 58 Representations of the world for a PAAS

Considering the first of these statements, present common automation systems do not show any analogy to human perception. These systems just consider particular parameters of the environment and react to them depending on fixed predefined rules. PAM approaches this point through introducing the concepts of objects' image and model scenarios' image (see section 3.2.5) as representations that are meaningful to the automation system. In such a way a PAAS perceives the world through the defined objects and by means of composing objects the system creates representations of the world around based on predefined model scenarios, as it is represented in Figure 58.

In reference to the hierarchical organisation of human sensory processing networks, few similarities can be pointed out between present automation systems' structure and human perception hierarchy.

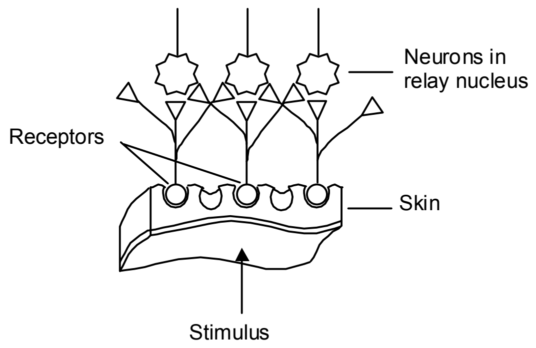


Figure 59 First steps of the staged information processing in human nervous system

As it is shown in Figure 60, the sensors of the automation system can be placed in a first level in analogy to the receptors shown in Figure 59. In a second level one can locate the individual data points, which represent the single signals that are sent by each sensory cells. And on a third and last level – 'action associative' data points - one can place those signals that in an automation system are processed at a same unit in a similar functional level to the neurones in relay nucleus that are represented in Figure 59. With this structure today's technology in automation manages to reach at most the functionality of human nervous system from sense organs to spinal cord.

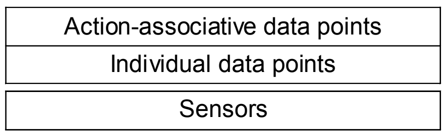


Figure 60 Hierarchy of common automation systems' perception

Extending this analogy between automation systems and humans, PAM supports additional levels such as the 'attributes layer', the 'objects layer' and the 'scenarios layer' (Figure 62). At these levels different higher order associations as well as data contrast are in turn supported between the single signals that are sent by the different sensors. But in order to enable proper data associations some requirements have to be met.

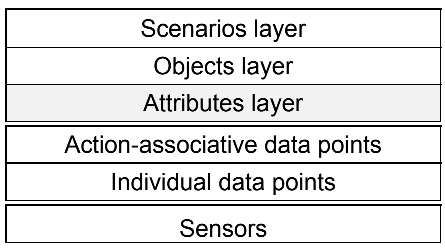


Figure 61 Hierarchy of PAAS perception

For example, in his work [Cia 93] Ciaccio makes some considerations concerning the data that has to be associated to enable recognition of objects. In this work, Ciaccio defends that to obtain successfully results in pattern processing, patterns have to be defined through their salient features. From this declaration one can conclude that the salient features of an object are the elements to associate in order to recognise the object. Based on this statement objects and scenarios used by any PAAS have to be described by pointing out their most important characteristics.

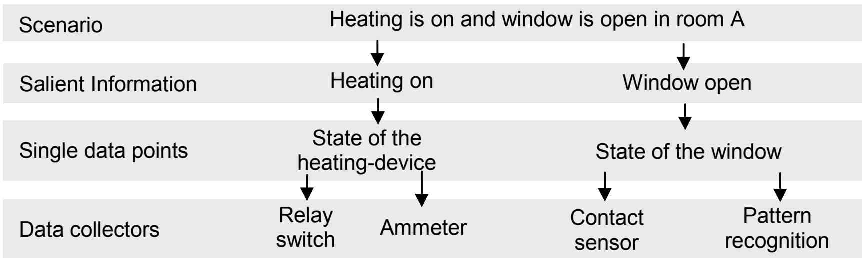


Figure 62 Top-down analysis of a scenario

There are numerous ways to represent patterns as a grouping of features, which Ciacci subdivides into statistical characteristics or syntactic descriptions. In this first end result of PAM it has been decided that when considering object definition, object salient features will be selected depending on those parameters of the object that can be measured by the system. On the other hand, a top-down analysis of the scenario will be followed to enumerate scenarios' salient features.

As it is shown in Figure 62, a top-down analysis results on a set of variables, whose momentary values have to be suitable for being measured through the perceptive technologies that are integrated in the automation system.

4.2.2 Remote Perception

Until now, technology has made it possible to transfer different kinds of data, sound and pictures over long distances. These technologies enable human remote perception function, which concerns the senses of hearing and sight, for example through television retransmission. Since, as it has been previously mentioned, the perceptive awareness model (PAM) is designed to enable automation systems to integrate both hearing and sight senses their remote perception function could be designed on the basis of human remote perception. However, though visual and acoustic remote perception are highly appreciated by the human society, since this research focuses on extending automation systems possibilities and not on designing an artificial analogy to humans, the integration of such remote perceptions is out of the domain of this work. In this approach automation system remote perception mainly concerns to perceive users' wishes in the distance, e.g. through Internet.

As it is shown in Figure 63, it has been decided that the remote user accesses the system at the 'objects layer'. In this connection the user can either ask for some variables or even change the value of some parameters. Since at this level of data processing data has already been validated by the perceptive awareness automation system (PAAS), connections to the 'objects layer' avoid that the user accesses to not unified or even wrong data. Besides, through accessing the 'objects layer' the user receives better understandable information of the different components at the monitored space due to the group associations. For example, instead of receiving information concerning temperature as 'temperature sensor = 24°C', the user would received ' window area-temperature = 24°C'.

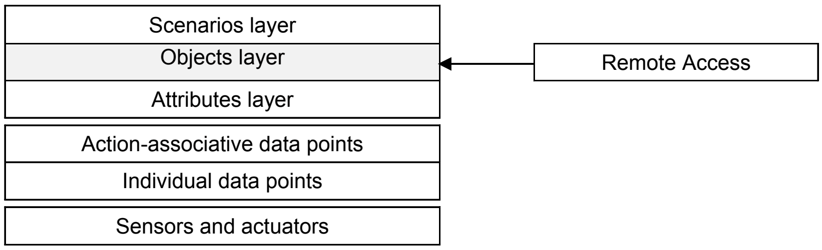


Figure 63 Remote access for remote perception

A second aspect to work out in the function of remote perception of any PAAS concerns data flowing. If the value that has been changed by the user would override the value that the system receives from the lower layers, layers above the 'objects layer' would process this data before the data would be reflected in the environment. As example can be considered the case in which a user wants to activate the heating from a remote location. If the change is executed in the store location occupied by the variable coming from the environment the upper layers of the system process information that it is not real at that time (in this case that the heating is 'on'). When everything works properly this may entail no problem and even may save time. If not executed the system would process wrong information and consequently would not behave according to the real situation. In such a way, it is convenient to avoid the processing of a 'requested change' before this change has been executed. Consequently, in case the user accesses the object layer to ask for a change it is required to differentiate between attributes' values depending on the direction of the data flow ('bottom-up' as input from the environment or 'top-down' either as input from the GUI or as order from upper layers).

A first solution to face this problem could be to create additional databases for actuators' variables. A second possibility could be to have just one database in which each actuators' variable occupies two locations depending on its origin – environment or upper layer/user interface, and difference them, for example introducing a 'locations label' (Figure 64).

ObjectAttribute	Symbol	Location
heating	on	user
heating	off	lower layer

Figure 64 Location label to differentiate between attributes

In this second possibility at the attributes layer variables would be stored taken into account not only their type - static sensor variable, dynamic variable or actuator variable – but also their origin, i.e. if the value comes from the environment, from upper layers or from user's interface.

4.3 PAAS Perception Process

In order to be capable of perceiving the global situation the perceptive awareness model (PAM) has to enable cooperative work between the different technologies that are required to integrate the various kinds of data collectors.

As it has been already mentioned, using various sensory technologies results on data format discrepancy, which makes a cooperative work between technologies more difficult. Consequently, PAM has to firstly support those functions that allow unifying data formats. These data format unification functions are the first steps towards the differentiation between PAAS perception process and common automation system perception processes. Once data format unification is reached PAAS perception requires additional functions such as data comparison for validation or data association.

4.3.1 Modularity

In order to manage the complexity of the perception process different functional blocks have to be defined. This modularity, which provides a more understandable global view of the perception process, demands the definition of proper interfaces and the correspondent communication functions - communication protocols - between modules. Since in some cases the different modules may require similar functions, for example to read received data, the definition of a uniform communication protocol, instead of individual ones, seems to be more convenient. The advantages of a uniform protocol are:

- Interchangeability: A uniform protocol enables the interchange of functional block whenever these blocks are developed according to the defined interfaces. In the global concept functional blocks have to be seen as 'black boxes'; the way to face each individual task is irrelevant in a global view of the process.

- Uniformity: In the case that a functional module had not been implemented, or had been turned off, it would be convenient that data could be forwarded to the next functional block without being submitted to any change. A uniform protocol would support this data forward requiring neither data interpretation nor adaptation to a different interface.

- Reusability: Once the interface of a functional block has been defined, and in case that a uniform protocol was used, the defined interface could be adapted to the rest of interfaces. In such a way it would be possible to reuse already programmed parts.

- Maintainability: In case of failure between functional units, a uniform protocol would make possible to submit the functional block to a revision process, which could be compatible to the whole system.

To reach such a uniform protocol interfaces have to be globally defined to support all required communication tasks. However, this global definition does not force every block to implement all communication functions. Each block would be equipped with just the indispensable communication functions to reach its targets.

4.3.2 Adapting Individual Perception Systems - Communal Data Access

In order to enable cooperative work between the different technologies that are integrated in the automation system, PAM has to fulfil following requirements

- Provide easy data access

- Solve the discrepancy of data formats that results from making use of different technologies

Each one of these requirements involves the implementation of particular functions and interfaces, which are shown in Figure 65 and which will be discussed during the following subsections.

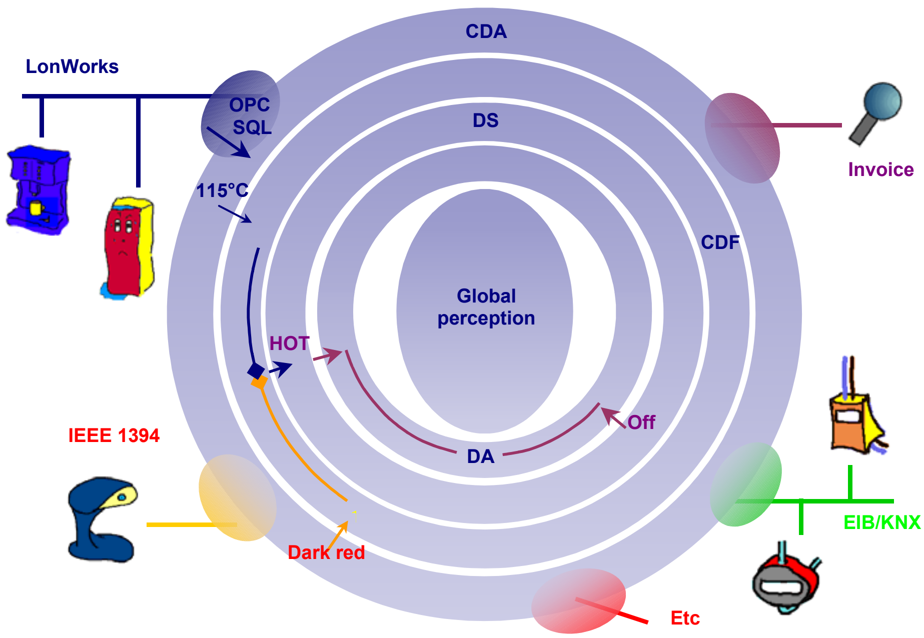


Figure 65 Functions demanded by PAM perception process: CDA: communal data access; CDF: communal data format; DS: data significance; DA data association

- **Enable Data Access - Communal Data Access**

Easy data access entails to place collected data in a communal location so that upper processing-units can easily access collected data for further processing in the perceptive awareness model (PAM). Nowadays, in order to enable data complex processing input data has to be transferred from the store locations of the different source technologies. E.g.: the LNS database in case of LonWorks, to PAM communal database.

The integration of an OPC (OLE for process control) server [Iwa 02] and the proper client application to move data from the fieldbus database into a communal location, e.g. MySQL database, is a possible solution when using fieldbus technologies such as LonWorks, EIB/KNX and BACnet (Figure 66).

Since easy data access has to be enabled for every sensory technology that the automation system supports, at this level each one of the integrated technologies has to be analysed to find out its particular requirements to transfer data from its own database to the selected PAM database.

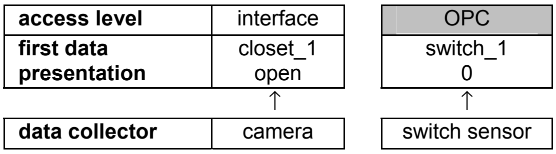


Figure 66: Enable data access

The position of this function in PAM can be seen in Figure 67. As it is represented in this picture data access is the first function that is needed to enable data processing by PAM upper layer.

- **Equalising Data Formats through Symbolisation - Communal Data Format**

Works such as [Hsu 85] have demonstrated that working with symbols increases the efficiency of numerical computations. As defined in [Daw 02] symbolisation involves transformation of measured values into symbols, which are processed by the system to extract information about the process, in the case of PAM about the situation. Symbols are treated as transforms for the original data that retain much of the important information for the system. Therefore, in order to implement each adaptation interface a detailed study of the format and variable types supported by the correspondent technology is demanded. A second requirement is to know which information the upper processing-units demand.

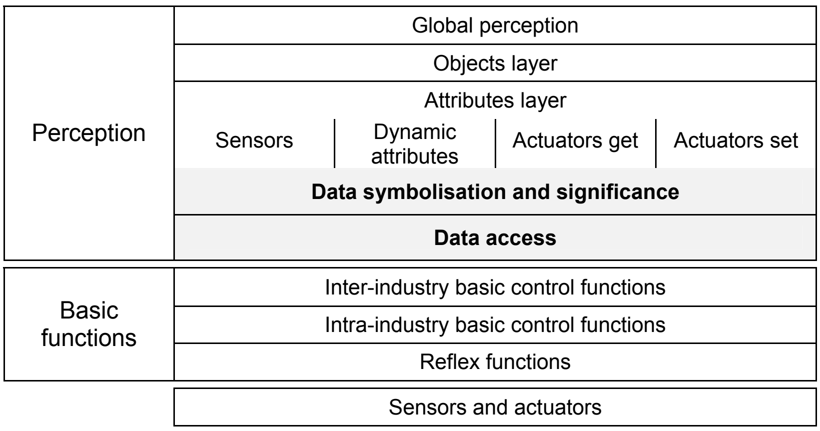


Figure 67 Location of the function of symbolisation and data significance in PAM

Nowadays, as it is described in [Daw 02], techniques of symbolisation are being used for efficiency in various fields such as astrophysics/geophysics, biology and medicine, fluid flow, chemistry, mechanical systems, artificial intelligence, control and communication, and data mining, classification and rule discovery. In the area of intelligent data processing several works such as [Bal 90, Tak 98, Sim 99] have dealt with symbolisation during the last years. In his work [Bal 90] Dan R. Ballard defends that for a system to exhibit intelligent behaviour it must be able to use symbols and abstractions. For Ballard symbolic computing views computing as a model of various cognitive processes. This view is concerned with symbol manipulation, use of heuristic to restrict search, qualitative models, knowledge representation, and inference mechanisms. In his work, Ballard models a visual process as a three-stage process of detection, pattern recognition and interpretation. The first process is dominated by numeric-algorithmic computing, the second process requires converting numeric or pixel data into a symbol by means of assigning a label, and the third process combines the various labelled symbols with contextual symbols. A similar division can be useful in the implementation of PAAS.

Before implementing any symbolisation it is also useful to know which aspects require special attention. For example, in conventional studies of navigation by autonomous robots the symbols that are easily understandable for humans are used to represent the state space. The state space represented by such symbols does not agree in many cases with the space constructed by the characteristics of the robots and the environment, which causes the 'symbol grounding problem' [Har 90]. In [Tak 98] Takeuchi faces this problem by means of proposing a structure to translate heuristic values into grounded symbols and vice versa for an autonomous robot (Figure 68). While user indicates the route through heuristic symbols, the robot navigates using grounded symbols. In the framework, the grounded symbols can be used for symbolic processing through the translation.

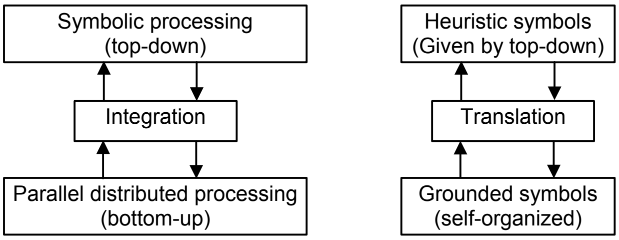


Figure 68 Framework for integration of symbolic processing and parallel distributed processing

Symbolisation in PAM is used for two main reasons. On one hand, as it is defended by works such as [Hsu 85] symbolisation contributes to increase the efficiency of data processing while making easier functions such as data comparison, data association and data validation. On the other hand, and as it is exposed in the following subsection, symbolisation is required to bring system's perception closer to human perception and, furthermore it contributes to a better communication between user and system.

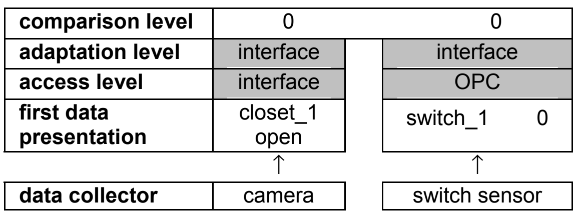


Figure 69: Adaptation of data formats to support comparison

As it is shown in table 2, above the access level a data adaptation interface level is defined. This adaptation interface level is responsible for the unification of data formats into symbols. As an end result of the adaptation level data referring to the same monitor and/or measured parameter, but coming from different sources of perception, appears in the same format according to the specify automation system's global format. As function required for PAM higher data processing, this function is positioned after the data access function in Figure 67 and in the example presented in Figure 69.

Due to its complexity data symbolisation in PAM has been the theme of research of the diploma thesis of Markus Falkner: 'Symbolisierung für effiziente Verarbeitung von Sensordata' [Fal 03]. The use of integer numerical symbols in PAM to equalise data formats have been chosen on the basis of results shown in the work of Falkner. In this work comparison tests are presented using different symbols such as integer, float, character, decimal, real and text. Results show that integer numerical symbols require less time for comparison operation than the rest of tested symbols.

- **Turn Data into Significant Information – Data Significance**

The second reason to use symbols can be understood while analysing how humans perceive temperature. In this analysis one observes that the sensations that temperature produces in humans are not related to particular temperature grades. In other words, humans recognise levels of temperature but not an absolute value. Since the automation system lives together with the user and operates for the user's benefit, it is convenient that both system and user perceive events in a similar way. As example, one can consider the temperature in a room.

The exact numeric temperature value that is measured by the temperature sensor has to be related to significant information. Figure 70 shows how significance is given to the attribute temperature by means of translating numerical measured values to meaningful sensitive values. Additionally, a numerical symbol is associated to each one of the temperature levels in order to make further data processing easier and consequently more efficient.

Temperature				
range (°C)		meaning	value	priority
from	to			
-50	-5	freezing	1	1
-5	10	cold	2	-
10	20	mild	3	-
20	32	warm	4	-
32	50	hot	5	1

Figure 70: Conversion from temperature values into significant information

The function of turning data into significant values has been positioned in the same level as the function of symbolisation, as it can be seen in Figure 67. This is due to the existent one-to-one relation between symbolised values and significant values. An example of this relation can be seen in Figure 70.

4.3.3 Data Comparison – Data Redundancy

Data redundancy does not only benefit biological systems, as it is described in section 1.2.1. In the same way, and as it has been already demonstrated in different studies such as [Mar 99], data redundancy improves the reliability and robustness of artificial systems. In PAASs data redundancy can be supported through the integration of various data collectors taking care of the same variable. These data collectors could either belong to the same technology such as EIB/KNX, IEEE1394 or LonWorks or to different technologies that the automation system could support.

Since benefits of redundancy do not appear in redundancy itself but when redundant data is compared, in order to take advantage of data redundancy PAM has to support data comparison. This function, which consists on contrasting redundant data in order to validate measured values, results on a most reliable single value, which reduces errors of perception.

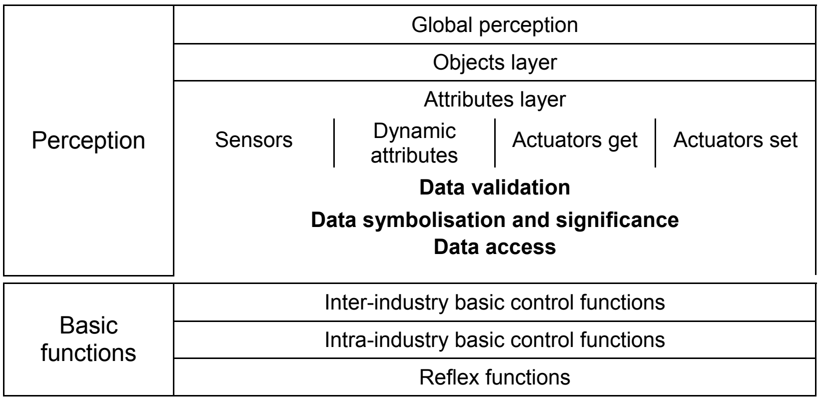


Figure 71 Location of the function of data validation in PAM

Though data validation should take place before any other process starts, since the model is designed to be compatible to the already existent technological solutions in the field of automation (for example, automation systems for common applications like light control, HVAC, etc.) PAM integrates data validation at its high processing part, as it is shown in Figure 71. In such a way, today's application functions that take place either at the second, at the third or at the fourth layer of PAM can take advantage of data redundancy by means of some kind of feedback from upper layers.

The comparison function has to be designed having in mind the following statements:

- Since comparison can only take place once the values to be compared are known a solution has to be found so that the system is aware of these values.
- Comparison of values from different sources concerning the same variable or parameter is to happen as soon as a change occurs in any of these values, independently of a change of value or none change of value concerning data sent by the other sources.

The first statement can be met by means of using a dynamic global database in continuous communication with the different 'receptor technologies' that are integrated in the automation system. In this way, upper processing units could acquire current measure values by accessing this dynamic global database. The second statement entails that the system has to support a function that is activated as soon as a change occurs, so that the system can react immediately to every change.

4.3.4 Data Association

Also based on principles from nature, PAM integrates the function of data association. Though in human perception each sense organ works independently concerning the function of data collection, there is a subsequent and unconscious association of single perceptions that enables the person to perceive entities, as it is defended in [Wei 97]. In accordance to this declaration, data association consists on combining collected data in benefit of a more accurate perception of the environment, which enables perception of complexes from single data values.

The perceptive awareness model (PAM) supports data association at two different levels. First association function is placed between attributes layer and objects layer, while second association function is placed between objects layer and scenarios layer (see section 3.2.5). During the first association function PAM makes use of the terms attribute and object. As Figure 72 shows, objects are composed by attributes and can contain several other objects too, similar to [Roy 95]. The hierarchy of objects and attributes represents a detailed description of the environment including information about locations.

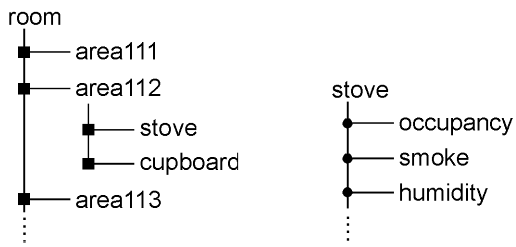


Figure 72 Objects and their hierarchy

Since PAM considers the possibility of integrating redundant sensors, the term ‘sense’ is introduced to distinguish between the different measured values referring to the same parameter. In such a way the hierarchy object-attribute (oa) is enlarged to object-attribute-sense (oas). This term ‘oas’ is used to process data at upper processing layers. For example, in case that the attribute temperature of the object fridge measured by a LonWorks temperature sensor was identified as ‘id_oas = 15’ the measured value of this attribute would be consequently seen by upper layers as current measured value (‘id_symbol’) of the ‘id_oas = 15’.

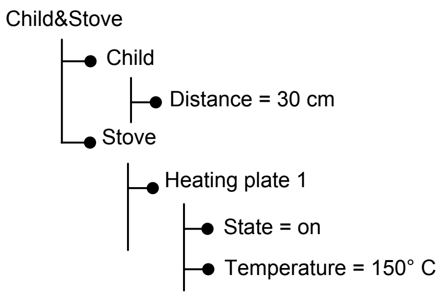


Figure 73 Scenario structure

The second kind of association also requires a predefinition, in this case, of model scenarios. Therefore, in the same way that each object consists on a group of determined attributes, each model scenarios consist on a group of particular objects in a particular state, as it is shown in Figure 73 with the model scenario ‘Child&Stove’. In this model scenario salient objects are the child and the stove. But, even when both objects appear together in a situation is does not mean that it correspond to the model scenario. The model scenario corresponds not only to an association of particular objects but to an association of particular objects in a determined state. In the example of Figure 73 it is represented with, for example, the state ‘on’ of the ‘heating plate 1’ of the stove and the distance from the child to the stove ‘distance = 30 cm’.

4.3.5 Storage of Data

Beside the different functional blocks, the storage and organisation of data is a second important aspect of the data processing. There are many ways to support an electronic storage of data and the posterior access to this stored data. Data can be stored in the local memory of a computer, it can be piled in text-data in one or several computers or it can be stored in complex database applications. Requirements of each application are the criteria to select one or another of the different data storage forms. A perceptive awareness automation system sets following requirements:

- Global availability: Data has not only to be suitable for local access. Several modules at different computers have to be able to access the data.
- Performance: Almost every module accesses and manipulates stored data. This continuous and multiple access and manipulation of stored data is an important factor in relation to the processing speed. Efficient and rapid reading and writing are more than convenient.
- Easy handling: Among other things database has to enable easy placement of new information, uncomplicated data access, and simple data removal.
- Error tolerance: Even if the data writing does not come to a successful end due to any problem such as network connection failure, stored data has to remain consistent and usable.
- Space requirements: Since the system manages lots of information, it is convenient to store data in an as most efficient and space-saving way as possible.
- History: Chronological data storage is convenient to enable a sequential progressive monitor of events.

A database system appears as possible data storage solution that meets all these requirements. Databases have been used during the last five decades for those applications that require and efficient processing and storage of lots of data [Heu 97]. These storage forms enable synchronous access of several processes. Data inconsistency is avoided by means of encapsulated transactions. A decisive criterion is the standardised interfaces and correspondent functions that have been defined to access database For example in case of SQL database, on one hand there are functions that allow executing basic functions directly over the data structure such as the creation and removal of tables. These functions are grouped under the term 'data definition language' (DDL). On the other hand those functions related to data manipulation, data request, and data storage are grouped under the term 'structure query language' (SQL), and defined as SQL-functions.

Chapter 5

Implementation of PAM Perception System

This chapter consists on two main parts. The first part focuses on the different technologies that have been implemented in the SmartKitchen lab at the Institute of Computer Technology (TU Vienna) to meet the requirements of the function of perception in perceptive awareness automation systems:

- Somatic and olfactory perception
- Acoustic perception
- Visual perception
- Remote perception

The second part covers the implementation of upper layers and interfaces required by the function of perception:

- Perception and attention interface
- Storage of data
- Transformation layer
- Validation layer

5.1 Perception Systems

The four different senses of the SmartKitchen perceptive awareness automation system are presented in the following subsections. Each subsection starts with a description of some relevant aspects for PAAS of the corresponding human sense, which are taken as reference during the implementation of the specific perception system.

5.1.1 Somatic Perception

As it is defended in several neuroscience and behavioural books such as [Kan 00, Lyd 99] human somatic perception is related to information provided by receptors distributed throughout the body, i.e. touch, proprioception, pain and temperature information. Making the analogy between biological systems and automation systems the term automation system somatic perception is understood as the part of perception of the system that is responsible for different parameters such as temperature, humidity, and pressure. Since, as it is exposed in [Tam 01a], this kind of factors are common measured variables in today's automation fieldbus systems, it is reasonable to implement the perceptive awareness automation system somatic perception making use of these sort of technologies.

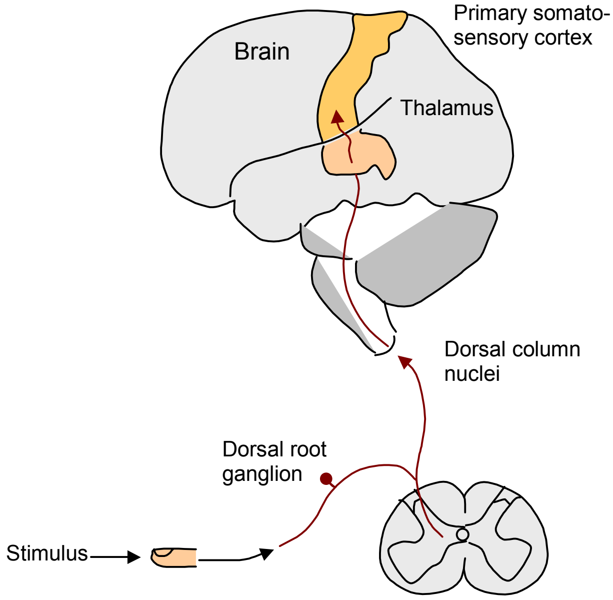


Figure 74 Example of transmission of somatic perception

In this research LonWorks technology has been chosen for the implementation of the somatic perception of the system. As it is defended in several publications such as [Kra 02, Die 00], nowadays in the field of home and building automation LonWorks appears as one of the most accepted technologies together with other fieldbuses such as EIB/KNX and BACnet.

Since the possibilities of this technology are not just limited to detect somatic parameters such as temperature and pressure, but it is also suitable to perceive events such as presence in a room, and to measure variables such as brightness, current and power consumption, or water consumption, in this implementation the term automation system somatic perception is extended to those parameters that can be detected, measured or monitored through fieldbus technology, in this case through LonWorks. Obviously these parameters depend on the commercial sensors that exist on the market like the ones produced by companies such as Sysmik [Sys 03], Svea [Sve 03], and Cetelab [Cet 03] in the case of LonWorks fieldbus technology.

- **Human Somatic Perception**

As defended in [Lyd 99, Kan 00] the different modalities of somatic perception (touch, proprioception, pain, and temperature) are supported through different system receptors and pathways. The dorsal root ganglion neuron transforms the stimulus into an electrical signal - nervous system global signal – through the mechanism of transduction (see section 4.1), and transmits the encoded stimulus information to the central nervous system, as it is shown in Figure 74.

Different types of somatic information are transmitted by distinct sensory neurons, and convey in parallel pathways to the primary somatosensory cortex, where information is combined into a somatic unify perception. Sensory information from the limbs and trunk is conveyed to the thalamus and cerebral cortex by two different ascending pathways:

- Touch and proprioception are transmitted by large axons with fast conduction velocities to the dorsal horn of the spinal cord and then to the brain stem and thalamus.
- Pain and temperature sense reach first the most superficial layers of the spinal dorsal horn and are conveyed directly to the thalamus through multisynaptic networks.

In such a way, somatic perception occurs in a modularised functional way, where each one of the different specialised units is responsible for the perception of a determined characteristic of the object. Consequently, when somatically perceiving an object, information about an object such as size, shape, temperature, texture and mass, is fragmented by peripheral sensors and integrated, afterwards by the brain.

5.1.1.1 PAAS Somatic Perceptions

Nowadays fieldbus automation systems reach a level of functionality that can be in most cases compared to the functionality of the nervous system from receptors to spinal cord (see section 3.3.2). In similarity to the modularised human somatic perception, this work aims to design a layered perceptive awareness automation system somatic function.

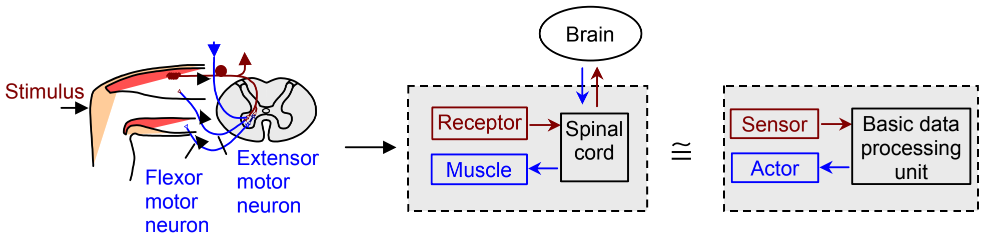


Figure 75 Reflex action and its analogy in automation systems

As it has been exposed in section 3.3.2 one can say that today's automation fieldbus systems cover two levels of actions when comparing their way of doing to human behaviour. The first level of action can be assimilated to reflexes and as automation application example case one can consider a smoke detector that, as soon as is activated through smoke detection, responds through acoustical alarm and red light (Figure 75). The second level of action can be somehow related to basic human functions such as turning the TV sound louder when the person can understand almost nothing. In this case as example of fieldbus application one can think on an indoors-light control unit that depending on the brightness level and occupancy or not of the room turns the light on or off, action shown in Figure 76.

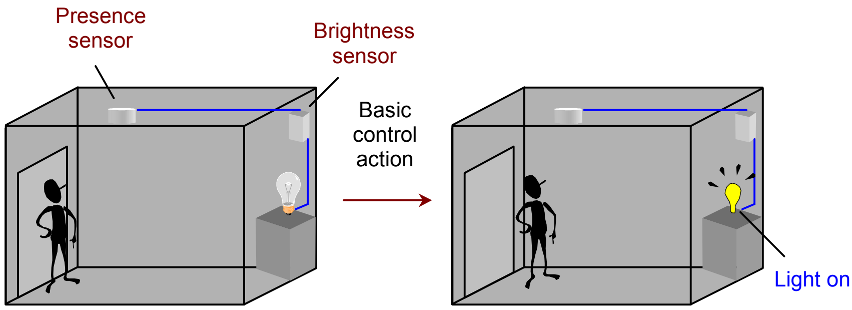


Figure 76 Common function in automation systems

However, unlike to common automation systems, in humans even in case of a reflex action the input signal is forwarded to upper processing units of the nervous system; Arrows that go from the spinal cord to the brain in Figure 75 represent it. The way of doing of human nervous system enables the person to be continuously aware of every relevant input of the current experience situation, and consequently adapt his response to it, as it can be interpreted from different works such as [Sac 97, Kan 00, Lyd 99].

Since the design of the perceptive awareness model (PAM) has been performed to extend the possibilities of automation systems and reach a system behaviour similar to human conscious one, the extension of the functionality of fieldbus systems is reached by means of joining the somatic perception of the lower layers of the automation system to implemented upper layers of PAM.

Table 1 LonWorks sensors and actuator in the SmartKitchen PAAS

9 contact sensors to detect the state open/close of entry door, windows, and cupboards	12 relays to control the state on/off of electrical devices such as lamps, cooker, sunblind, and microwave
1 humidity sensor above the cooker to detect evaporation	1 dimmer to regulate the light atmosphere in the kitchen
5 power meters to measure power consumption of dishwasher, cooker, coffee machine, microwave, and fridge	8 temperature sensors to measure room temperature, the temperature close to electrical devices like cooker and fridge, and the temperature of the lamps
3 brightness sensors to detect indoors and outdoors brightness level	1 distance sensor to measure distance between the cooker an a person
3 smoke detectors	1 switch for the sunblind
4 electromagnetic valves to control the state open/close of the pipes of the sink (cold and hot water), the dishwasher and the coffee machine	4 water counters to measure cold and hot water consumption, dishwasher and coffee machine water consumption
8 water sensors to detect water leakage and water accessing through the window	8 IR-sensors to detect occupancy at different areas of the kitchen

5.1.1.2 Implementation and Integration in the Perceptive Awareness Automation System

The implementation of the perceptive awareness automation system somatic perception function starts by selecting the variables of the environment that are interesting for the automation system to detect, measure or monitor.

Since, in a first moment, the implementation of this perception function is planned to use LonWorks commercial solutions, the selection of variables entails searching the LonWorks sensors and measure devices that exist on the market and that meet the somatic perceptive requirements of the system. Products from companies such as Cetelab, Sysmik, and Svea, which can be found in [Sys 03, Sve 03, Cet 03], have been integrated as part of the somatic perception of SmartKitchen (Table 1, Figure 77).

In addition to the previously enumerated devices and in order to install a LonWorks network in the system a PCLTA (personal computer Lontalk adapter) card supporting FTT10 (free topology transceiver 10) transceiver from the company Echelon has been required to communicate the LonWorks network and the PC for installation and commission of devices. Moreover, since the selected LonWorks devices require different communication channels, to be precise FTT10, RS485 and TP-1250, two routers are needed to enable communication between the three subnetworks.

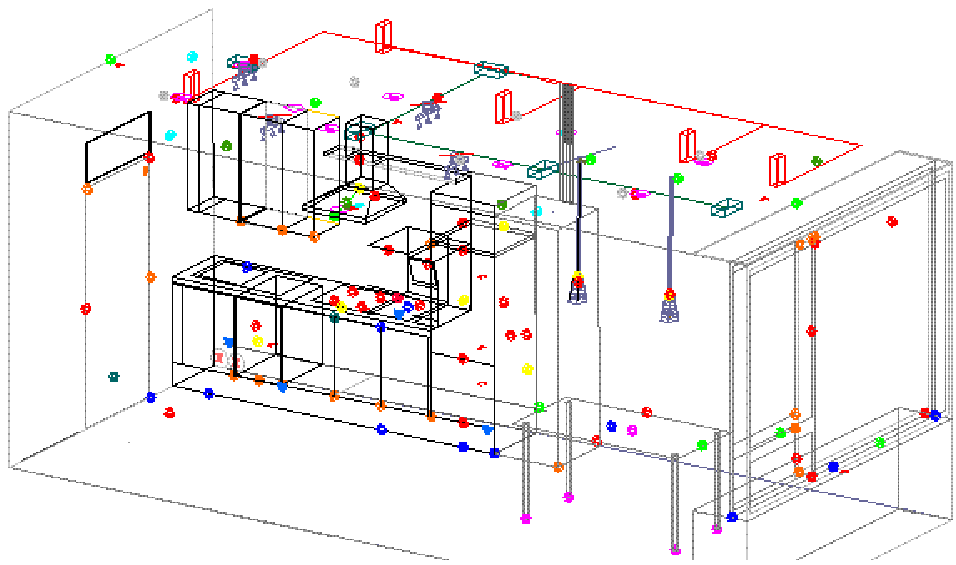


Figure 77 View of the SmartKitchen and the required cabling for the LonWorks network

The transmission cable is a cable Cat5, which runs through different parts of the kitchen in a predesigned way to reach each one of the desired data point at the same time that allows easy future extensions of the LonWorks network. Cable Cat5 has been chosen following recommendations found in the 'LonWorks Installation Handbook' [Lon 02]. In this book, it is declared that free topology specification states the use of cable that conforms to the TIA-568A (Telecommunications Industry Association Standard 568A) specifications of category 5.

In reference to cable installation, some considerations have also to be made in reference to power supply. Since selected devices require either 12V DC, or 24V DC, or 230V AC, in order to supply the different components three different power nets have been installed all through the upper part of the kitchen and close to the different home electrical appliances such as fridge, dishwasher and cooker.

Figure 77 shows the SmartKitchen room and some of the cabling required for the LonWorks network. The red lines represent the 230V AC power lines, while green lines at the higher part of the room represent the bus channels FTT10 and RS485, which run parallel. The small coloured points correspond to the different sensors and actuators such as temperature sensors (red points), water sensors (blue points), etc., while the boxes symbolise respectively 230V AC switches (red boxes) and bus hubs (green boxes). This distribution enables to connect the large number of components in an easy way, and benefits future extensions.

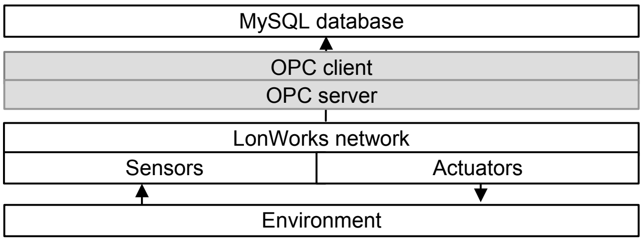


Figure 78 Structure required for the integration of somatic perception in SmartKitchen

In order to enable the perceptive awareness automation system to take advantage of the somatic perceived values an interface is required to transfer result data from the own LonWorks database (LNS database) to the database of the perceptive awareness automation system (PAAS). In this case, where a fieldbus is used, OPC appears as possible standard solution. Since there are commercial OPC servers for LonWorks such as the one developed by the company Newron System [New 03], in order to enable data transfer the automation system just needs the additional proper OPC client application software. Both together, OPC server and client make possible to transmit LonWorks input data to the database of the perceptive awareness automation system, in this case a MySQL database.

Figure 78 shows the structure of the function of somatic perception in the SmartKitchen according to the elements that are required to implement this function before reaching the higher processing layers of the model. A detailed structure is shown by Figure 79, which contains not only lower required components but shows as well interfaces and upper processing layers.

5.1.2 Olfactory Perception

The sense of smell allows humans to perceive and give significance to many gas molecules that arrive to the nasal cavity. Though usually this perception just drives to generate a like or dislike feeling in humans, for example when smelling a rose, in some cases it helps to rapidly perceive an event such as burning.

Though there are commercial solutions like the ones developed by companies such as Alpha M.O.S. [Alp 03], Cyrano Science [Cir 03], Electronic Sensor Technology [Ele 03], and MIL-RAM Technology [Mil 03] that recognise smells their high prices make the integration of such solutions in a common automation system unfeasible.

Therefore, in this first implementation of a perceptive awareness automation system, the importance of olfactory perception for SmartKitchen is related to the case in which the perception allows the system to be aware of determined events like burning. In such a way, since the implementation of this function comprises equipping the system with gas sensors, and fieldbus technologies such as LonWorks and EIB/KNX offer such sensors, it has been decided that the integration of olfactory perception in SmartKitchen occurs through the LonWorks network (see section 4.1.2).

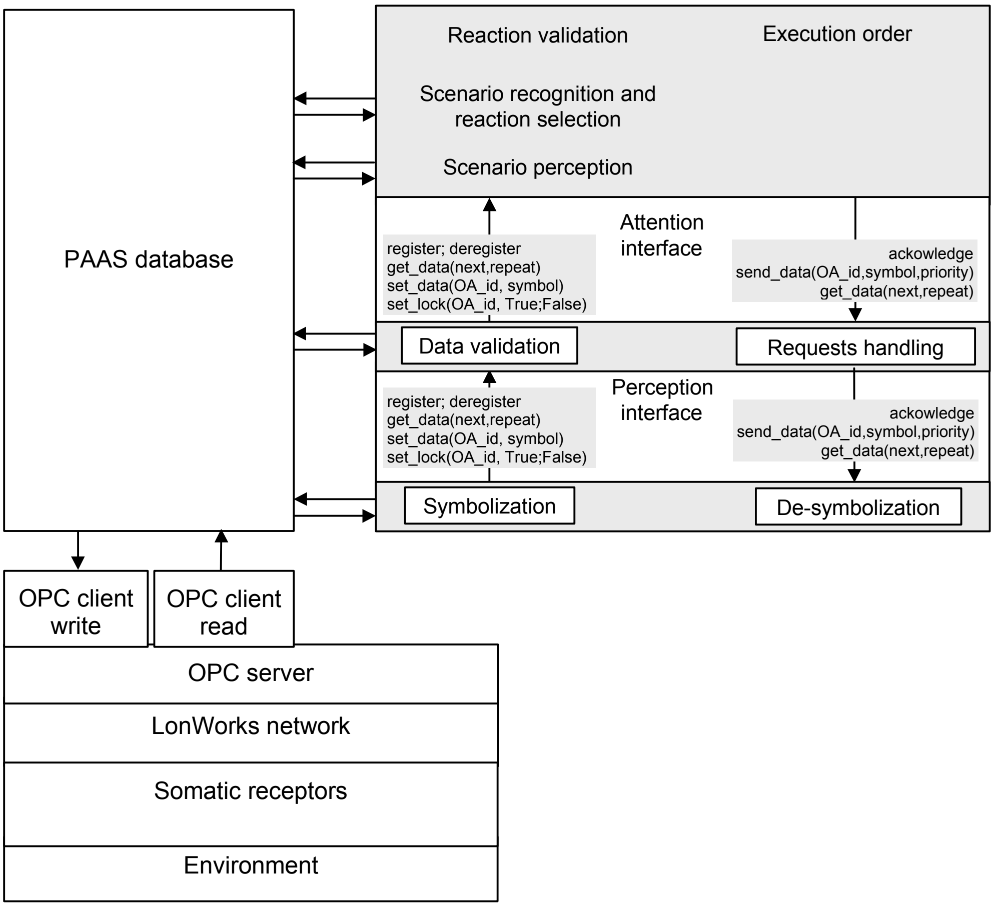


Figure 79 Structure example of a PAAS

5.1.3 Acoustic Perception

The sense of hearing covers an important part of perception. Since sound waves are transmitted through the air from the emission source to human ears, hearing allows humans to perceive events even if they occur out of the visual domain.

In the same way that acoustical perception plays a role in human behaviour, by means of enabling humans to perceive determined events that are out of the domain of other sense organs or by reinforcing the perception of some events that are also perceived by other senses in a different form, the automation system can also benefit from perceiving the acoustic signals of the environment.

- **Human Sense of Hearing**

Human hearing begins when the cochlea, the snail-shaped receptor organ of the inner ear, transform sound energy into electrical signals - nervous system global signal – through the mechanism of transduction (see section 4.1). As it is described in [Kan 00e], mechanical energy flows through the middle ear to the cochlea and makes the basilar membrane vibrate. The basilar membrane acts as mechanical analyser of sound frequency thanks to an array of hair cells that detect the different frequency components of the stimulus. The signal is transduce and encode into electrical signal, which flow through eight-nerve fibres from the cochlea to the cochlear nuclei, continuing to the brain stem, which among other functions is responsible of localise sound sources and which forwards auditory information to the cerebral cortex. Here, distinct areas analyse sound to detect the complex patterns.

5.1.3.1 PAAS Acoustic Functions and Software Solutions

Current technology makes it possible to create artificial cochlear like the ones developed by the companies 'The Chicago Otology Group - Cochlear Corporation' [Coc 03] and 'Advanced Bionics Corporation' [Adv 03] (in most cases to face deafness). However, the expectations of this work are focus on developing a hearing sense for automation systems not in physical equivalence to human hearing but in functional likeness.

In a similar way to human hearing, in order to enable the perceptive awareness automation system to acoustically perceive the situation not only sound receptors (cochlea in biological systems and microphones in automation systems) are required. In addition, functional units have to be integrated to make the system capable to understand the signals that microphones collect, as it is the case of the basilar membrane, nerve fibres, the cochlear nuclei, the brain stem, and several areas in the cerebral cortex, when considering human hearing.

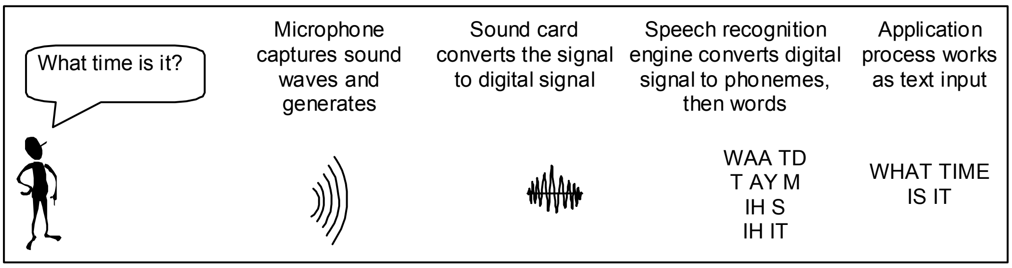


Figure 80 Speech recognition

According to [Spe 03], in the mid- to late 1990s, personal computers started to become powerful enough to make it possible for people to speak to them and for the computers to speak back. Nowadays, there are several commercial solutions in the field of acoustic perception and recognition such as [Dra 03, Sca 03] that have been developed on the basis of acoustic pattern recognition. Being outstanding in from of most of these commercial solutions, which focus on speech recognition, Voice Extreme [Sen 03] from the company Sensory enables to integrate different acoustic functions at a reasonable price. Making use of this technology the perceptive awareness automation systems is equipped with following functions:

- Speaker verification to identify a person on the basis of acoustic passwords
- Acoustic control to control some components of the system via acoustic commands
- Sound recognition to detect specific events such as flowing water or glass break
- Acoustic reaction to enable for example acoustic notifications and greetings form the system to the users

The implementation of these functions is based on two main techniques: speech recognition and speech synthesis. Speech recognition, also known as speech-to-text, involves capturing and digitising sound waves, which are converted to basic language units or phonemes. In such a way, words are constructing from phonemes, and they are contextually analysed to ensure correct selection between words that sound alike, as it s the case of the words write and right. An example of this technique is shown in Figure 80.

Speech Synthesis, also called text-to-speech, is understood as the process of converting text into spoken language, as it is shown in Figure 81. In this case words are broken into phonemes, analysing for special handling of particular text such as numbers, and punctuation, and generating the digital audio for playback.

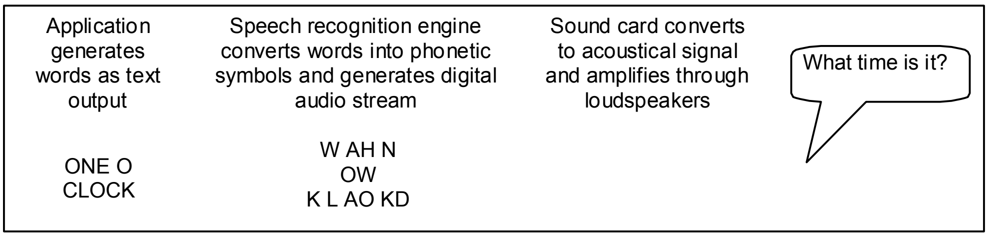


Figure 81 Speech synthesis

Speech recognition can be speaker dependent or independent. In the first case, the user has to go through training exercises, which in Voice Extreme entail that each recognition word has to be trained twice by the user to create voice templates. In the case of speaker independent speech recognition no end – user training is required. This function is designed for specific languages and sets of words, called 'weight files' in Voice Extreme. These 'weight files' are offered by the company and they just have to be downloaded into the application memory.

5.1.3.2 Implementation and Integration in the Perceptive Awareness Automation System

When it comes to implement the previously enumerated acoustic functions in the SmartKitchen two possible solutions are considered: a centralised and a distributed solution, both shown in Figure 82. The centralised solution demands the voice extreme modules to be connected to a central processor unit and consequently it mainly entails software work.

The distributed solution entails one intelligent sensor per function to be implemented. These sensors contain a microprocessor, which processes the acoustic signals according to the implemented function. The advantage of the centralised solution is that sensors can be very simple, just simple microphones.

Among the disadvantages of this solution stand out:

- Since the central processing unit has to solve a lot of arithmetic problems it has to be quite powerful.
- If this central unit fails, every acoustic function fails.
- The processing of the acoustic signals of one sensor can block the central unit for the other sensors.

Since the distributed solution does not need a central processing unit none of these disadvantages appear:

- Processing is distributed between the various microprocessors.
- In case that one intelligent sensor fails the others still work, which entails that the rest of acoustic functions are still active.
- The intelligent sensors cannot block each other.

The disadvantage is that more complex sensors are needed.

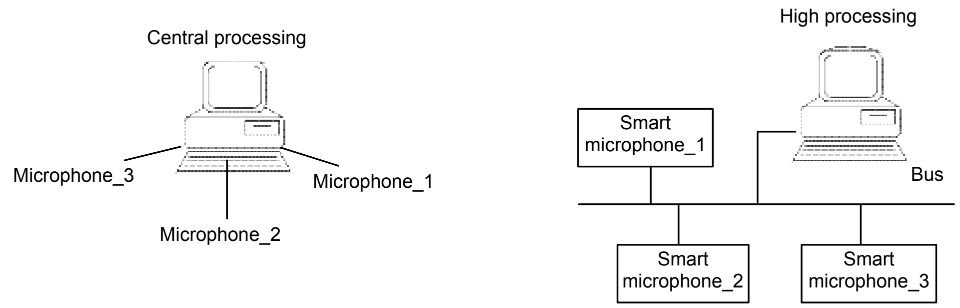


Figure 82 Centralise and distributed solution for the acoustic perception of SmartKitchen PAAS

For the implementation of the acoustic perception function in the SmartKitchen perceptive awareness automation system the distributed solution has been chosen making use of the LonWorks fieldbus as distributed network. Consequently, the required intelligent sensors consist on a voice extreme module, a microphone, a loudspeaker and the components, which are necessary for the connection to the LonWorks network. Each voice extreme module contains a central speech processor, 64KB ROM for the 'voice extreme interpreter' and the 'speech technology code', 2MB Flash RAM for storing the applications and the module connector pins.

- **Speaker Verification Function**

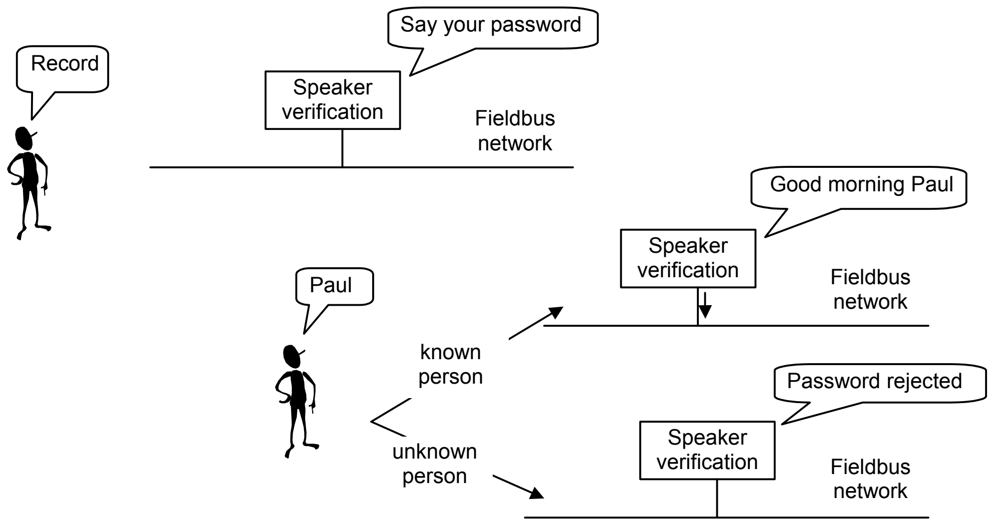


Figure 83 Speaker verification function

In order to identify a person, an acoustic password must be trained twice by the user to create the voice template. This training action starts when the system recognises the speaker independent keyword ‘record’, and requests the user acoustically to say a personal password for recording. After recording the password, the module sends the message to the LonWorks network that a new user password has been stored and switches the operating mode back to continuous listening.

The recognition of the person once the module recognises the speaker independent keyword ‘password’ requests the user acoustically to say his password. The module compares this password with those that have been previously stored as password template. In case that the password matches one of the templates, the module sends a message to the LonWorks network to notify which user has been identified, acoustically welcomes the user in the kitchen, and sends a message to the acoustic control module, as activation signal. If the password does not match any of the templates, the system acoustically reacts saying ‘password rejected’, and sends a message to the network that an unidentified person is in the kitchen. A graphical example is shown in Figure 83.

- **Acoustic Control Function**

The first test of this function, as part of the acoustic perception of the SmartKitchen perceptive awareness automation system is performed to control light and radio in the kitchen. This module is in a standby mode until the speaker verification module activates it (as it has been previously explain by means of sending a message). After activation the module waits for one of both speaker independent keywords either ‘activate’ or ‘panic’.

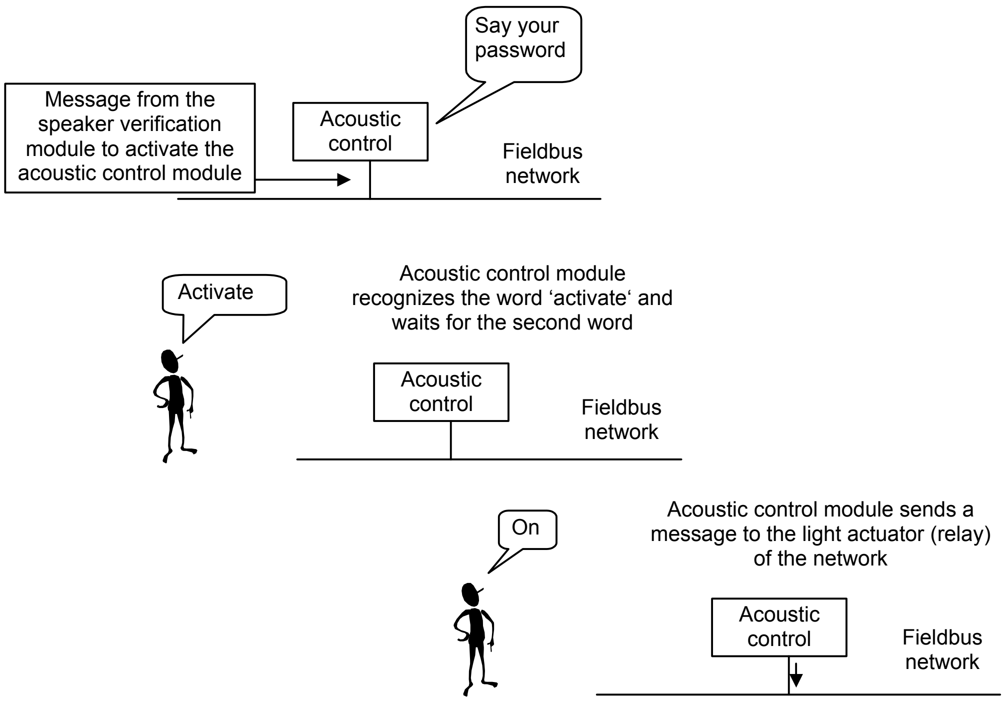


Figure 84 Acoustic control function

The keywords ‘activate’ or ‘panic’ have been chosen instead of light and radio because in the free ‘weight files’ that are included in the voice extreme software development kit the words light and radio do not appear. And the creation of an own ‘weight file’ is not possible right now with the software that Sensory Inc. provides in the Voice Extreme package. Consequently for SmartKitchen the keyword 'activate' is understood as 'light' and the keyword 'panic' as 'radio'. If the module recognises one of these two words, it waits for an acoustic command represented by the key words 'on' and 'off'. Depending on the combination of the first and the second keywords, the light or the radio are said to switch on or off by means of sending the corresponding message to the network. A graphical example is shown in Figure 84.

- **Sound Recognition Function**

The sound recognition function is responsible for recognising a specific sound in the kitchen. For the sake of simplicity this first implementation is limited to let the automation system know if water is running. In order to enable the module to recognise this sound, a sample has to be recorded first. As soon as not-identified person is in the kitchen the module goes into the word-spotting mode, which means that the module does not expect a time of silence after the sound that has to be recognised, as it happens in the other modes. After recognition of the sound sample, the module notifies the LonWorks network that water is running. A graphical example is shown in Figure 85.

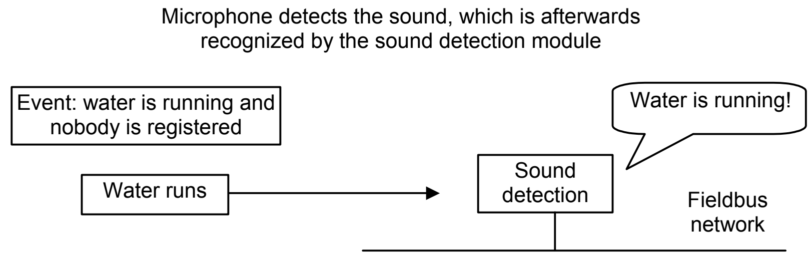


Figure 85 Sound recognition function

- **Acoustic Reaction Function**

As it has been indirectly mentioned when describing the other three acoustic functions that the perceptive awareness automation system integrates, this function does not required an additional voice extreme module but it is supported by each of the modules already described.

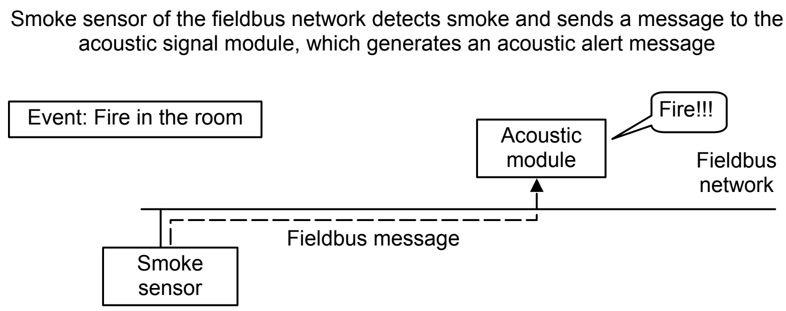


Figure 86 Acoustic reaction function

Acoustic reaction is based on synthesis technology and is implemented using a tool from Sensory Inc. to create the sentences to be said by the system. An application example is represented in Figure 86. In this example, the smoke sensor has detected smoke and sends a message to an acoustic module notifying the event. As soon as the acoustic module receives the message, it generates the acoustic alarm 'fire!'.

To integrate the acoustic applications in the perceptive awareness automation system, it has been necessary to create one LonWorks node per voice extreme module, which needs the creation of a motherboard (Figure 87).

The resulting smart acoustic modules support communication to the environment, i.e. hearing and speaking, by means of a microphone and a loudspeaker. Obviously, the modules require additional power supply, a service button to register the module in the LonWorks network during installation and commission, and a reset button to enable application changes.

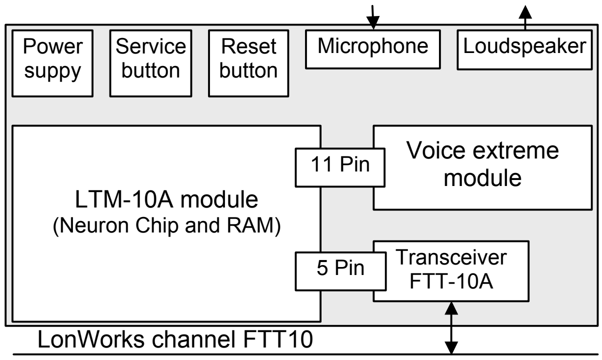


Figure 87 Architecture of a smart acoustic module

With this implementation solution, since the acoustic functions have been integrated into the LonWorks network, no additional interface is required. The resulting structure of the function of acoustic perception in the SmartKitchen perceptive awareness automation system according to the components that are required to implement this function is shown in Figure 88.

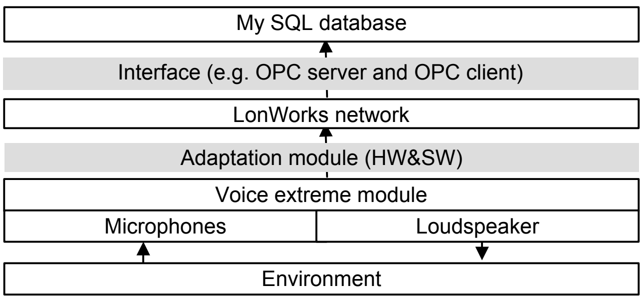


Figure 88 Structure required for the integration of acoustic perception in SmartKitchen

5.1.4 Visual Perception

'Vision is the process of discovering from images what is present in the visual world and where it is' (David Marr in his book Vision: A Computational Investigation into the Human Representation and Processing of Visual Information [Mar 82].

Humans perceive lots of information through their sense of sight. As exposed in [Kan 00], in many cases data coming from any of the other sense organs is used as additional data, which either corroborates visual data or adds some trait to it. However, in automation systems the sense of sight plays a lower role. Though there are successful bionic researches and projects such as the ones mentioned in publications such as [Hub 03, Hub 02] focused on developing artificial iris, the possibilities of existent commercial pattern recognition solutions are limited. These limitations entail for example recognising faces under determined constrains such as a maximal deviation grade between the perceived face and the face template [Int 03a], perceiving well defined object such as a cup or well-differenced colours such as red [Die 01], recognising obstacles under light constrains [Hub 03a], or tracking objects with well differentiated contour [Vin 01].

- **Human Sense of Sight**

The process of visual perception begins in the retina, which converts light into an electrical signal - nervous system global signal – through the mechanism of transduction (see section 4.1). Through the optic nerve this signal flows from the retina to the midbrain or thalamus and from here to the primary visual cortex, which is organised into functional modules that process visual information from a particular region of the visual field.

Studies about human optical sense such as the ones done by Anne Treisman and her colleagues [Tre 77, Tre 86] and [Jul 84] defend that in a scene the different objects are separated, and that those objects of interest are distinguished from the background. Besides, they say that in this process much of the sensory information is filtered out through selective attention to limit the amount of information that reaches the higher centre of processing in the brain.

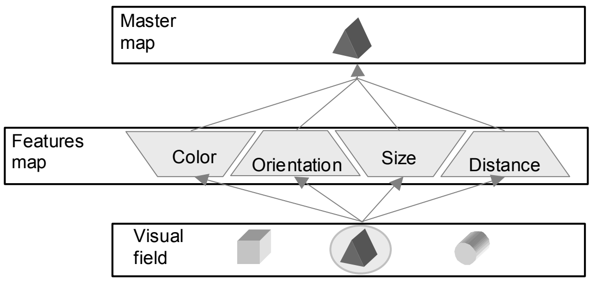


Figure 89 Humans visual processing

According to this statement, and as it is exposed in [Kan 00e], gestalt psychologists argue that the visual system accomplishes the organisation of contextual interactions by processing sensory information into a master map. This information is located in different feature maps about shape, colour, distance, and movement of objects according to computational rules that are inherent in the system in distinct parts of the brain (Figure 89). This theory is supported by studies such as the ones made by Sigmund Freud at the end of S.XIX [Fre 91], as it is mentioned in [Kan 00e], or the ones exposed in [Sac 87]. These works prove that the inability of certain patients to recognise specific features of the visual world is due to cortical defects that affect the ability to combine components of visual impressions into a meaningful pattern. In such a way, results of these studies support the theory of requirement of feature and master maps to perceive the environment.

5.1.4.1 PAAS Visual Functions and Software Solutions

During the last years, some pattern recognition researches such as the ones that are taking place at the Institute of Flexible Automation (Infa) at Vienna University of Technology [Vin 02] or computer vision projects done at the research centre Advanced Computer Vision have developed technological solutions that can be partially assimilated to biological solutions. In similarity to the principle that human visual perception works by distinguishing objects of interest from the background researches such as the one exposed in [Hub 03a], in which objects are detected on structure 3D modelled backgrounds, are being successfully developed.

SmartKitchen, as perceptive awareness automation system, has to support optical communication between automation system, environment and user. The system has to be able to perceive the situation visually and to generate visual reactions such as red light to notify a dangerous event or adaptation of the light atmosphere depending on user's wishes and on the situation. Attending to visual perception, in this first implementation of a perceptive awareness automation system, the system integrates following functions:

- Persons perception
- Objects perception
- Colours perception

A face recognition demo program (HMM-Demo) offered by Intel is used to cover the function of person perception. This program is based on the Hidden Markov Models (HMM) [Fin 00, Gla 00]. C++ and C pattern recognition source codes are accessible at the OpenCv [Int 00] web page of the company Intel [Int 03a].

The cooperation with the Institute of Flexible Automation (Infa) makes possible to take advantage of solutions implemented at this institute to cover the functions of objects and colours perception. Due to the complexity of pattern recognition it has been necessary to limit the domain of perception to particular objects and colours. To be exact, in a first moment, the system aims to recognise a cup located at the coffee machine and its content, and the red colour of a heating plate of the cooker to perceive that the heating plate is on.

5.1.4.2 Implementation and Integration in the Perceptive Awareness Automation System

As it has already mentioned (see section 4.2.1), due to the features of the signal that has to be transferred the automation system needs of specific technology to cover visual perception. Nowadays USB and IEEE 1394 are suitable technologies to meet the requirements. The advantages of IEEE 1394 over of USB according to [Apl 03, Sel 00, Mor 02, Koe 01], which next enumerated, have been the criterion to choose the IEEE 1394 technology in the implementation of this first perceptive awareness automation system:

- IEEE 1394 enables isochronous data transfer, which enables to send pictures from different cameras at the same. In such a way the process unit can together process received pictures on real-time [Fir 03], without the need of waiting for the other pictures to be received, as it happens with the USB due to its incapability of isochronous data transfer. First figure in Figure 90 shows a first possible connection (USB 2.0 (a)) between cameras and PC using USB 2.0. In this case each camera is connected to a PC, which can process received images. Since image transport is not supported between PC the only possibility to joint information from both cameras is through the pattern recognition software. The second picture in Figure 90 represents a second USB 2.0 possible connection. In this case a bandwidth problem appears. Since USB 2.0 does not support synchronous data transfer images have to be send one after the other. The third picture represents a connection between two cameras and PCs using IEEE 1394.

In this case, images collected by the cameras can be either sent to the same computer at the same time or two different computers, which can interchange pictures.

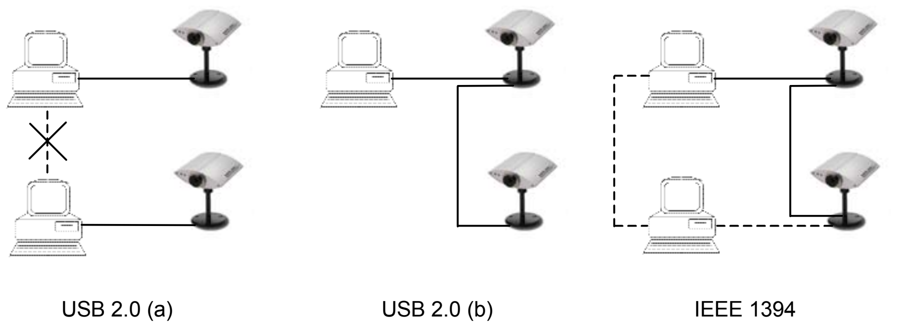


Figure 90 USB 2.0 versus IEEE 1394: connections

- IEEE 1394 enables transmission rates of 400 Mbit/s. FireWire 800, the implementation of the IEEE 1394b Standard, which has been defined for 3200 Mbit/s, enables transmission rates of 800 Mbit/s, versus the 12 Mbit/s of USB 1.1 and 480 Mbit/s of USB 2.0.
- FireWire 800 can transfer data across 100 meter cables versus the 5 meter of USB 2.0.
- IEEE 1394 supports a direct peer-to-peer connection between two components.

In such a way to implement the function of visual perception in the SmartKitchen perceptive awareness automation system optical data is captured by means of four IEEE 1394 cameras, which are located at the four upper corners of the room - SmartKitchen lab, as it is shown in Figure 91, so that the whole space is submitted to automatic visual monitoring.

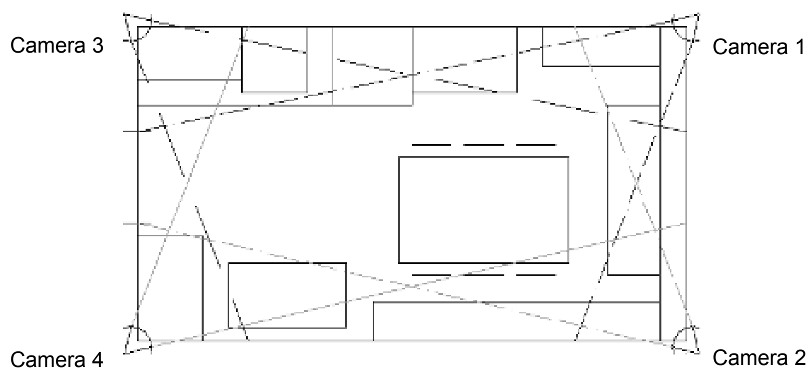


Figure 91 Domain area of the four cameras

The cameras work in synchronised mode with that it is meant that the four cameras send captured images to the process unit at the same time. There are different possibilities to work with the collected images. A first possibility would be that the process unit associate results from the pattern recognition process of the four images. A second possibility would be to combine the four images into a global one and submit this global image to pattern recognition.

The cameras are configured to send 10 images per second. Since relevant changes in the kitchen may entail few seconds, this restriction does not involve any disadvantage for the automation system and makes the pattern recognition process more manageable. Each image has a size of 640x480 pixels, and 8bit for the colour (256 colours). As it is shown in Equation 1, at least 98Mbit have to be transported per second. The common IEEE 1394, which in isochronous mode enables approx 80% of its 400 Mbit/s, i.e. 320 Mbit/s [Fir 03], is suitable for the exposed application.

Pixel (High)*Pixel (Width)*(Images/second)*(Colour bits)*(Nr. Cameras) = required Bits

(640 pixels (High))*(480 pixels (width))*(10 images pro second)*(8 colous bits)*(4 cameras) = 98304000 bits ≅ 98 Mbits

Equation 1

Pattern recognition software (see section 1.1.2) is also installed at the mentioned processing unit and operates on the global image to either detect the cup, the red colour of the heating plate, the light atmosphere or the persons that are at the kitchen in a determined moment.

In order to enable the perceptive awareness automation system to take advantage of the patter recognition function an interface is required to transfer results from the own pattern recognition application to the database of the system. Though several solutions could be developed, both in form of hardware or software, the implemented one consists on software developed at the Institute for Flexible Automation.

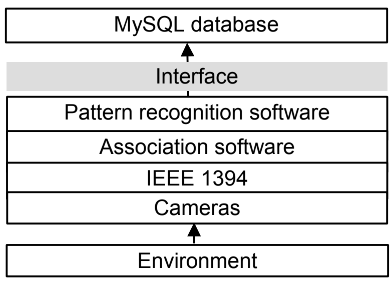


Figure 92 Structure required for the integration of visual perception in the SmartKitchen

This software solution sends result data to the database of the perceptive awareness automation system, in this case a MySQL database. Figure 92 shows the structure of the function of visual perception in the SmartKitchen perceptive awareness automation system according to the components that are required to implement this function.

5.1.5 Remote Perception

Remote perception requires technological solutions such as television, radio and Internet to enable the transfer of information from the remote place where the event is taking place or is being generated, or where the information stands, to the place where the receiver is located. In humans, once this action has occurred the person perceives the information making use of either his sense of sight, in case of optical information, or his sense of hearing, in case of acoustical information.

Nowadays, as defended in several publications such as [Die 00], the tendency of automation systems is towards a connection to the outside world, which means that in automation systems the aspect of remote perception (e.g. through Internet) is also considered. This communication has to support services and functions between the outside and the automation system allowing not only easy maintenance, but also complete manipulation looking for reducing costs, in the case of industrial automation, and looking for comfort, higher safety and security, i.e. 'a better quality of life', in private sectors. Though this connection has been in some cases materialise individually to each electrical device, e.g. the Screenfridge of Elektrolux [Scr 03], according to [Die 00] due to efficiency reasons, control systems based on fieldbus technologies connected to Internet appear as reasonable solution.

5.1.5.1 PAAS Remote Perception and Possible Solutions

Since SmartKitchen focuses in the field of home and building automation the remote perception function concentrates on perceiving the requests of the remote user. Obviously, this connection flows directly to let the remote user know about the state of the SmartKitchen and its different components, too.

Some considerations have to be taken concerning access security using Internet to control functions of the automation system. Attending to this point, in order to support secure multi user control the system has to know about who can access the system and which are the system control conditions of this user. A possible solution is that the system acquires this knowledge by means of storing the names and passwords of the different users together with their rights over the system during the configuration phase of the remote perception function. This configuration phase entails a first registration of the users.

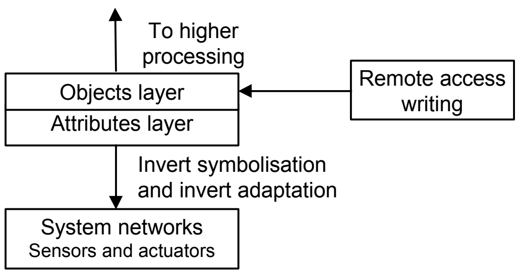


Figure 93 Bottom-up data flow due to remote access

A second aspect to consider is synchronisation in relation to remote monitoring of data values. Since remote perception has to be suitable to be opened to more than one user at the same time, in case one user decides to change the value or state of any automated parameter of the system (e.g. turn the heating on) this wished action is not appreciated by a second user who is also remotely monitoring the system at that time. This situation can be avoid by equipping the function of remote perception with some kind of lock-system of those variables that are wanted to change by any user to let the rest of the users know that some remote control action is taking place at that moment on the correspondent variable.

A third important point in remote perception concerns the aspect of data values synchronisation in relation to the bottom-up data flow processing. If changes ordered by remote users are first reflected in the database of the perceptive awareness automation system, as it is shown in Figure 93, higher processing might start working with these new values before changes have actually taken place.

5.1.5.2 Implementation and Integration in the Perceptive Awareness Automation System

In order to allow communication between the perceptive awareness automation system (PAAS) and the remote users through Internet a graphical user interface (GUI) has been created under the premises of design simplicity, understanding and handling simplicity using 'active server page' (ASP) [Asp 00], which is suitable for accessing databases.

Figure 94 shows one of the pages of the GUI to help the user to navigate through the automation system either for monitoring, for administration, or for control. The navigation menu of this GUI describes all the action that the user is allowed to do according to his rights, which depend on user's role. With the term 'out of order elements' the system refers to those elements that at that moment are not working properly and consequently require some revision or maintenance. Since wishes of change of a parameter of defected components may not flow to a satisfactory result this information is offered by the GUI to avoid inefficiency of this function. By using the GUI of the perceptive awareness automation system any authorised user can send a request to change the status of determined parameters of the system.

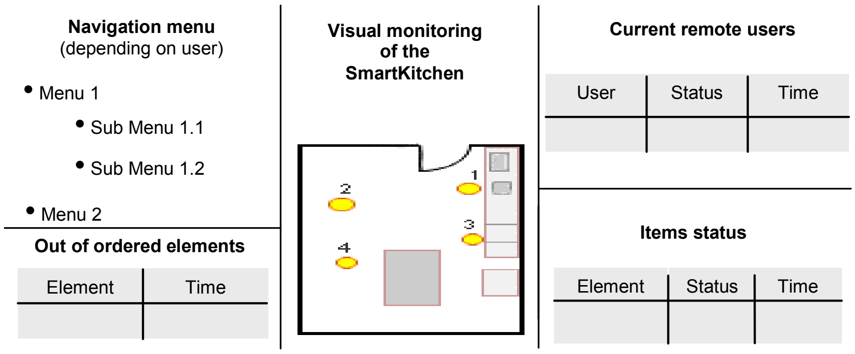


Figure 94 GUI SmartKitchen

In answer to the aspect of access security the GUI presents a registration page during the configuration phase. Through this page the user is asked to introduce his username and password, which are automatically stored in a 'registered users' database, together with the rights of the user (Figure 95). Each time a person tries to access the system remotely, the system asks for his username and password and verifies these entries in the 'registered users' database, allowing access just in case of a 'full match' of the data.

The problem of synchronisation of monitored data values exposed in the previous section has been faced by means of a 'lock variables' function. This function avoids a second access to a variable that is being changed by any user, and adds the variable to an information table, where the label 'locked' appeared beside the variable name together with the name of the user that is changing the variable, the start time of the control operation, and the time the operation has to be at the latest executed – 'Execute before' column, as it is shown in Figure 96. This last information is used to avoid accumulation of changes that after a certain time are unnecessary.

Username	Password	Right
Peter	xxxxxxx	Complete control
Anna	xxxxxxxx	Heating_3 control
....	....	....

Figure 95 Registered users database

Since the system might received several changes from different users in a short time interval, and the execution of each change may take its time, requested changes are stored in the 'variables' change queue' in case that they cannot be at once executed. The entry is deleted from this database as soon as the system starts to process this change or as soon as the 'execute before' time is reached.

Variable	User	Locked	Start time	Execute before
Heating_3	Anna	Yes	17.03.03 17:21	17.03.03 18:30
Dishwasher	Peter	Yes	17.03.03.......17:22	17.03.03.....21:00
....		....	....	

Figure 96 Locked variables database

According to the structure of the perceptive awareness model (PAM) adapted to the SmartKitchen perceptive awareness automation system the monitoring of values through the GUI occurs by reading the values of those variables that the particular user can access from the 'unify data' table of the database of the SmartKitchen PAAS. Due to the modularity of the system, the GUI should access data not directly but through the perception layer, and the communication with this layer should occur through the attention interface. In this case, users' requested-changes are also communicated through the attention interface to the perception layer. This layer handles the execution of requested-changes made by both, user and upper layers of the perceptive awareness automation system (PAAS). If the lower high processing layers of the system are free a requested-change is executed at once. Otherwise if lower layers are busy with any other task at this perception layer the requested change is written in the variables' change queue' table, where it waits for execution or till the 'execute before' time is reached.

Though this indirect data access matches the modularity of PAM, the implementation at the SmartKitchen PAAS follows a different resolution focused on application functionality. The implementation centres on enabling two communications: first communication between GUI and SmartKitchen PAAS databases, and second communication between GUI and the LonWorks network. The first communication allows data monitoring while the second communication permits users to control determined devices of the LonWorks networks.

In this second communication, in order to materialise the execution of requested changes an OPC client application is required to enable the transfer of requested changes to the LonWorks network database (LNS) (Figure 98).

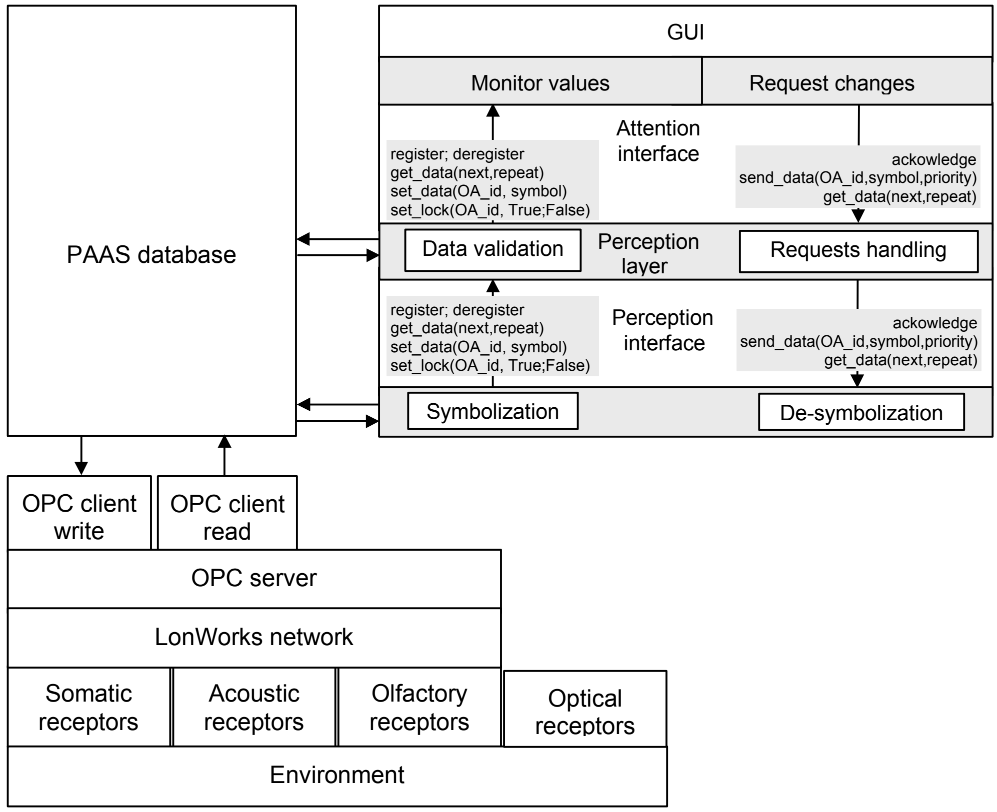


Figure 97 Remote perception structure in SmartKitchen PAAS

As soon as a change occurs in the database of the fieldbus network, a message is automatically sent to the correspondent component (actuator) of the network requesting the execution of the wished change.

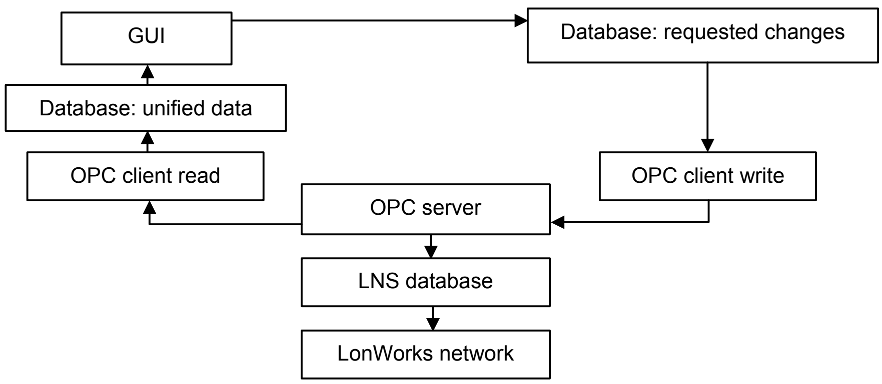


Figure 98 GUI connections to execute users' requested data changes

5.2 Global Perception

Once individual perception systems have been implemented and in order to enable global perception perception's upper layers and interfaces have to be implemented.

These upper layers and interfaces constitute the first part of the high processing of the SmartKitchen perceptive awareness automation system (PAAS) and support data flow in both directions bottom-up and top-down.
On one hand, after collected data has been placed at the global database of the SmartKitchen PAAS data is processed at the transformation and validation layers during the bottom-up data flow. On the other hand, the execution of actions is handled during the top-down data flow at the validation layer and inversely transformed at the transformation layer, so that the correspondent actuator can execute the requested change. Changes are requested to the validation layer by either the reactions layer or by the remote user layer.
Communication between the transformation layer and the validation layer occurs through the perception interface while validation layer communicates to the above representation layer through the attention interface. Communication between different layers is supported using sockets.
The implementation of the upper layer of the SmartKitchen PAAS has been done using distinct programming languages, to be exact Java and Delphi. However, since both languages support sockets the communication between layers is established without any difficulty.

5.2.1 Perception and Attention Interfaces

Perception and attention interfaces enable data flow between transformation layer and validation layer and between validation layer and representation and remote access layers respectively. In the first interface socket communication is defined so that the transformation layer plays the role of server and the validation layer appears as client. In the second case the validation layer is the server and the representation and remote access layer the clients.

Bottom-up data flow happens each time a new device is integrated as part of the automation system or as soon as a change is perceived in any of the detected, measured or monitored parameters. Top-down data flow corresponds to actions the system has to execute. These actions can either correspond to system's reactions or to remote users' wishes. Following functions have been defined to enable both kinds of communication.

- Register(type): In order to enable accessing the interface of a functions block it is indispensable to be registered at this interface. Through this interface both writing and reading clients can communicate to the transformation layer. The kind of client is established through the label 'type', e.g. register(write). Although several 'write' clients can register, the system supports the registration of just one 'read' client. This restriction is required because the 'read' client corresponds to the next upper functional blocks of the PAAS.
- Deregister: This function is called when a client wants to break the communication. The execution of this function results on ending the data transmission and closing the data transmission channel between server and client.
- Get_data(type): In order to ask for data layers make use of the function get_data(type). In this case 'type' refers to either 'next' or 'repeat'. 'Next' is used when the layer asks for a new data, while 'repeat' covers the case in which the layer solicits the last sent data. In such a way the usual data-request order will be get_data(next) while get_data(repeat) will be used in case of data processing failure or loss of data.

- Send_data(value): This function appears as answer to the function get_data() in the top-down data flow. The parameter 'value' consists on one or several information, which are separated using comas, depending on requirements of the client.
- Set_data(): This function is executed as answer to the function get_data() in the bottom-up data flow, i.e. when the server solicits a get_data() to a client. Once more the parameter 'value' refers to the transferred information, and in case of more than one value they are separated using comas.
- Set_lock(value): This function is received by the validation layer either coming from layers above or from the remote perception layer. This function is used as part of the maintenance of the system and to handle multiple remote users accessing the system at the same time. In the first case this function is called as soon as the system detects a failure in any of its components. In the second case set_lock(value) is activated as soon as a client receives the control over a determined parameter. The parameter 'value' notifies which component has to be locked, who locks it, when, and for how long.

5.2.2 Transformation Layer

The transformation layer is responsible for two main tasks. On one hand the layer receives the values that are collected from the different sources of perception in their original format and turns them into unified symbols. On the other hand the layer receives symbolised data from upper layers and turns it into the data form demanded by the actuators. Before embarking upon the first action data has to be presented in an object-attribute-sense ('oas') combination, which allows the system to perceive the situation as composition of its several objects.

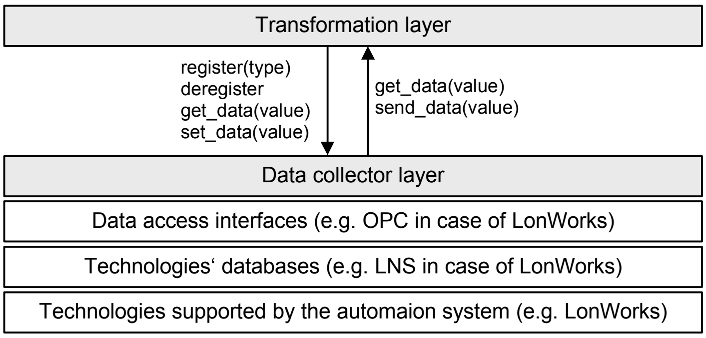


Figure 99 Layers of the transformation

Events' prioritisation takes place at this layer too. In that way symbolised data values are forwarded together with a priority label. Since upper layers process the data with higher priority first, this priority label constitutes the attention method of the SmartKitchen PAAS. Six different priority labels have been defined:

- Priority 1 is the highest priority. It will be set in case that the event could lead to a dangerous situation. As example one can consider the case of a smoke sensor. In case the sent symbolised value 'id_symbol' of the 'id_oas' corresponding to the smoke sensor is 'yes' the priority label related to this 'id_oas' is automatically set to 1.

- Priority 2 is set whenever a situation can be dangerous for the person, e.g. when a child is alone in the kitchen and the stove is on.
- Priority 3 is used in case unusual measured values, which could mean or lead to damage of a machine.
- Priority 4 is set in case collected data lead to a reaction already in lower layers, i.e. in case of reflex action.
- Priority 5 is used when the data notifies the activation of a machine.
- Priority 6 is set whenever a change occurs. In such a way priority 6 has been defined as the lowest priority.

The transformation of data formats has been implemented using two different layers – ‘data collector layer and transformation layer’ (Figure 99), which support socket communication using following orders, which have been described in section 1.2.1: register(type), deregister, get_data(value), sen_data(value), set_data(value).

The main task of the transformation function, data symbolisation, is implemented in several steps and requires different tables to reach the final data transformation, as it is exposed in the following subsection. But this is not the only functionality of the created tables. Beside their use during the data transformation function, these tables are required to support the handling of data as attributes of particular objects. The different tables, which represent part of the database of SmartKitchen PAAS, have been created using SQL language [Din 94].

- **Database Implementation**

A first ‘items’ table is created to define each object, attribute, sense or scenario, which can be related to an automation system (Figure 100). This table is needed for the eventual implementation of a semantic searching. This semantic search is though to enable a future learning process. This learning process will be based on reasonable connections between components of the system depending on different aspects such as the functionality of the components. The ‘Items table’ is enlarged as soon as a new object is installed in the automation system or a new model scenario is defined,

Items table		
Name	**Data type**	**Comment**
ID	INT	Autovalue
Label	VARCHAR(20)	Object description
Meaning	CHAR(1)	Defines item-type

Figure 100 SmartKitchen table database 'Items'

A second table called ‘Objects’ contains the objects that are relevant in the room or area where the perceptive awareness automation system is implemented. In the case of SmartKitchen PAAS this table contains objects such as the stove, the fridge, and the cupboards, which are relevant in the kitchen of the Institute of Computer Technology (ICT) at the TU Vienna. Represented in Figure 101 as ‘ID_item’, a connection between the ‘Items table’ and the ‘Objects table’ has been defined to enable the system to support the eventual semantic searching function demanded by a future learning process. The enlargement of this table occurs as soon as a new object appears either as component of the automation system or as relevant object in the kitchen.

Objects table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_item	INT/FK	Refers to an ID at the 'Items table'
Label	VARCHAR(20)	Defines the object

Figure 101 SmartKitchen table database 'Objects'

The detectable, measurable or suitable for monitoring parameters of the different objects are stored as attributes at the 'Attributes table' (Figure 102). For example, in reference to the object stove appear several attributes can be defined such as temperature and state (on/off) of each one of the heating plates, and power consumption.

Attributes table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_item	INT/FK	Refers to an ID at the 'Items table'
Label	VARCHAR(20)	Defines the attribute

Figure 102 SmartKitchen table database 'Attributes'

Once more a connection is established between the 'Items table', which is required for the eventual semantic searching. Differing from the 'Items table' and the 'Objects table', the 'Attributes table' is enlarged not in case of new objects but in case of new attributes. Since in some cases new objects may not introduce new attributes this table demands less enlargements.

As it has been mentioned in chapter 3 in reference to the theme data redundancy, each attribute can be measured, detected or monitored through different perception sources like LonWorks devices and IEEE 1394 cameras (the so called 'senses'). Each time a new perception source technology is introduced in the automation system this new information is stored at the 'Senses table' (Figure 103). The row 'reftable' shown in Figure 103 refers to a table at the 'data collector layer'. This table contains the measured, detected or monitored values of the correspondent perception source technology.

Senses table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_item	INT/FK	Refers to an ID at the 'Items table'
Label	VARCHAR(20)	Defines the attribute
reftable	VARCHAR(20)	SQL Table where the sense stores the data

Figure 103 SmartKitchen table database 'Senses'

Detected, measured and monitored values are transformed into uniform symbols, which are defined and stored at the 'Symbols table' as 'label' (Figure 104). The words hot, warm, mild and cold are an example of symbols when thinking on temperature. Brightness values can be transformed into symbols such as dark and bright. Moreover each symbol label is related to unique numerical ID in benefit of the efficiency of the processing.

Symbol table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_item	INT/FK	Refers to an ID at the 'Items table'
Label	VARCHAR(20)	Defines the symbol

Figure 104 SmartKitchen table database 'Symbols'

Table 'ObjectAttributes' (Figure 105) is defined to make more manageable the association object-attribute and to avoid the need of continuous storing of objects' and attributes' identification's information separately.

ObjectAttributes table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_object	INT/FK	Refers to a ID at the 'objects table'
ID_attribute	INT/FK	Refers to a ID at the 'attributes table'

Figure 105 SmartKitchen table database 'ObjectAttributes'

All possible combinations 'object-attribute' are stored in this table. Obviously as soon as a new object joint the automation system this table is automatically enlarged with the new information concerning this new object and its respective attributes. An additional table is defined to store all possible combinations 'object-attribute-sense'.

For example, in case of SmartKitchen PAAS possible combinations concerning one heating plate of the stove would be 'heating plate-temperature-LonWorks network' and 'heating plate-temperature-IEE 1394'. In the first case the attribute temperature of the object heating plate is measured through a temperature sensor of the LonWorks fieldbus network. In the second case the value of the same attribute of the same object is collected making use of patter recognition applications, where image transport is supported using IEEE 1394. This table has been called 'ObjectAttributeSenses table' and it is shown in Figure 106.

ObjectAttributeSenses table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_objectattribute	INT/FK	Refers to a ID at the 'objectattributes table'
ID_sense	INT/FK	Refers to a ID at the 'senses table'

Figure 106 SmartKitchen table database 'ObjectAttributeSenses'

Particular attributes can be related to determined symbols depending on the kind of attributes they are. For example, a temperature attribute can be related to symbols such as hot, warm, mild and cold while a brightness attribute can be related to the symbols like dark and bright. In benefit of efficiency the system does not process the symbols as words but making use of their related unique numbers. These associations are stored in the table called ‘AttributeSymbols table’ (Figure 107).

AttributeSymbols table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_attribute	INT/FK	Refers to a ID at the ‘attributes table’
ID_symbol	INT/FK	Refers to a ID at the ‘symbols table’
sequence	INT	Sets the logical sequence between related symbols

Figure 107 SmartKitchen table database 'AttributeSymbols'

The parameter ‘sequence’ that appears in this table is needed to establish the logical progressive succession of the different symbols that are related to a same attribute. For example, considering the attribute temperature if the label sequence of the symbol chilled was 1, the label sequence of the symbol cold would be 2, of the symbol mild would be 3, of the symbol warm would be 4, and of the symbol hot would be 5.

After appearing in one of the reference tables of the global database depending on the sense (see label ‘reftable’ in ‘Sense table’ (Figure 101)), collected data are placed in the ‘data table’ (Figure 108). In this table each collected value is stored together with the detection’s time label ‘TStamp’ and set in one-to-one unique relation to one ‘object-attribute-sense’ combination. The ‘TStamp’ label shows at what time and date the value has been measured.

Data table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_objectattributesense	INT/FK	Refers to a ID at the ‘attributes table’
TStamp	TIMESTAMP	Refers to a ID at the ‘symbols table’
Label	VARCHAR(20)	Data description

Figure 108 SmartKitchen table database 'Data'

In this implementation a possible hierarchical organisation of the existing objects is also considered. Such an organisation enables objects such as stove and each one of its heating plates to be set in relation, or to place ‘device objects’ by relating them with ‘area objects’.

ChildDefinition table		
Name	**Data type**	**Comment**
ID	INT/PK	Autovalue
ID_parentobject	INT/FK	Refers to a ID at the ‘attributes table’
ID_childobject	INT/FK	Refers to a ID at the ‘attributes table’

Figure 109 SmartKitchen table database 'ChildDefinition'

This hierarchy is stored in the table called ‘ChildDefinition’ (Figure 109). In this example the stove would appear as ‘parentobject’, while each of the heating plates would be defined as ‘childobject’ in four different relations.

- **Data Transformation**

All previously defined tables are needed to enable the transformation of data and to make possible the organisation of the data into the different relevant objects of the automation system. However, the transformation itself takes place thanks to the table called ‘transformations table’. Through this table, which is shown in (Figure 110), a connection is established between each particular combination object-attribute-sense, and the possible symbol values that this combination supports. The meaning of each symbol depending on the combination object-attribute-sense is also defined in this table through the ‘minvalue’ and the ‘maxvalue’ labels.

In the case of the temperature of the heating plate measured by a LonWorks sensor the first label of the ‘transformations table’ would be the ‘ID_objectattributesense’ that reflects the combination ‘‘heating plate-temperature’-‘LonWorks network’’ according to the ‘objectattributesense table’ (Figure 106). Afterwards, for each one of the possible ‘ID_symbol’ for the attribute ‘temperature’, when the LonWorks sensor measures the attribute, a ‘minvalue’ and a ‘maxvalue’ have to be set.

Transformations table		
Name	**Data type**	**Comment**
ID_objectattributesense	INT/FK	Refers to a ID at the ‘objectattributesenses table’
ID_symbol	INT/FK	Refers to a ID at the ‘symbols table’
minvalue	FLOAT	Lower limit for the correspondent symbol
maxvalue	FLOAT	Higher limit for the correspondent symbol
priority	INT	Priority of the combination objectattribute-symbol

Figure 110 SmartKitchen table database 'Transformations'

This dependency between senses and attributes is due to the possible difference of measurement ranges depending on sense. For example, if a LonWorks sensor measures the attribute temperature, the range of measurement can go from -15 to 250º C. In this case, according to the table ‘AttributeSymbols’ (Figure 107), possible ‘ID_symbols’ would be those related to the words ‘chilly’, ‘cold’, ‘mild’, ‘warm’, and ‘hot’. However, if the attribute is measured through pattern recognition the system can just differentiate between ‘cold’ and ‘hot’ depending on the detected colour of the heating plate (respectively black or red). Again with the exposed example, taking the word ‘hot’ the label ‘ID_symbol’ of the ‘transformation table’ would be the number that corresponds to the word ‘hot’, according to the ‘Symbols table’ (Figure 104).

In this case 100ºC could be ‘minvalue’ while 250ºC could be the ‘maxvalue’ of the symbol ‘hot’, when being measured by the LonWorks sensor. The label ‘priority’ would be set to one of the priority values that have been exposed at the beginning of the section 1.2.2.

5.2.3 Validation Layer

As it is shown in Figure 111 validation layer is placed once data transformation has been reached.

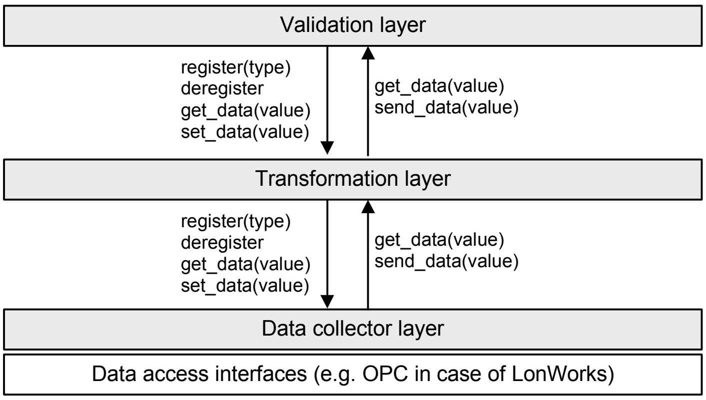


Figure 111 Location of the validation layer in PAM upper processing

This layer is responsible for two main tasks. On one hand it has to take advantage of redundancy during the bottom-up data flow. This task consists on supporting a comparison function, which has to be executed as soon as the automation system detects a change. On the other hand the layer has to manage the system's reactions in a top-down data flow. With that it is meant that this layer receives orders of execution from the reaction layer and wishes of the remote users from the graphical user interface (GUI) and has to handle them. Attending to the function of data comparison, following solutions are contemplate:

- **Global Table**

The global table consists of a database in which all collected values, after being turned into symbols, are stored.

As soon as a change occurs in any of these symbols, comparison begins by selecting all symbols (plus additional required information related to them) to be compared from this global table, i.e. all symbols referring to the same variable but coming from different sources of perception are selected and placed in the dynamic database. Afterwards selected values are individually compared, one to one.

The result of each single comparison, that is a specific symbol in case both compared values are compatible or zero in case of compared values' disagreement, is stored into the dynamic comparison results table. Finished the one-to-one comparison process results are grouped depending on id_symbol and stored into the results counter table. As an end result the system selects the symbol that appears at the dynamic comparison results table most frequently. An example of how this function works is shown in Figure 112. In this case three different sensors (id_oase 1, 2, and 3) measures the object-attribute number 23 (id_oa = 23).

As soon as one of these sensors (e.g. id_oasen 1) measures a new value of this id_oa 23 the comparison function is activated. At first the function searches in the global table all id_oasen that also take care of the id_oa 23 and locate them, together with additional information required for the comparison process itself, into the dynamic table. Once this action is finished starts the one-to one comparison process. Results of each one-to one comparison are sent to the comparison results table and after being grouped into the results counter the system selects as end result the id_symbol that differing from zero appears at most.

New measured value ⇒ Global table

⇓

Dynamic table						
id_oa	id_oasen	id_symbol	sequence	max_symb	priority	order
23	1	19	3	5	1	2
23	3	17	1	6	0	3
23	2	18	2	4	1	1

⇓

Comparison function

Comparison results			
id_oasen_1	id_oasen_2	Id_symbol	priority
1	2	0	1
1	3	1	1
2	3	0	1

⇒

Results counter		
comparison	id_symbol	priority
0	1	1
1	3	1

⇓

End result		
id_oa	id_symbol	priority
23	19	1

Figure 112 Comparison example with global database table

Though the description of this method is quite simple some considerations have to be taken before starting to implement the exposed concept. Since each source of perception has its particular scale symbols do not fit one to one. To understand the problem one could think, for example, of the variable temperature of a heating-plate of the stove. Considering the optical perception system, the received symbol coming from this system could be either hot or cold depending on the colour of the specific plate: red or black. On the other hand the symbol coming from a fieldbus network attending to the same variable, i.e. the symbol that is related to the temperature value but that is measured by the temperature sensor of the fieldbus network, can get not only the values hot or cold but also cool, tepid and warm.

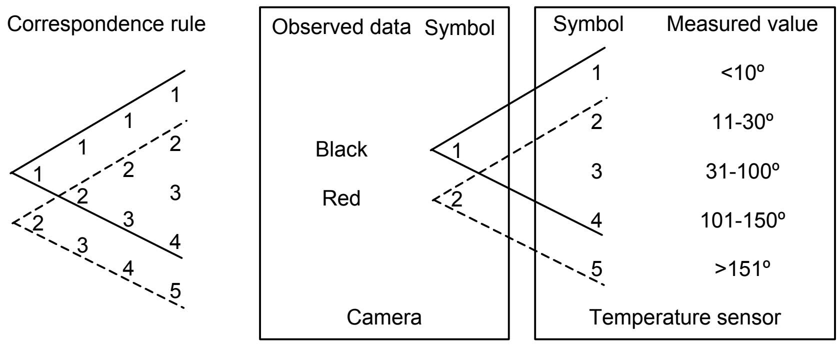


Figure 113 Correspondence rule between symbols of different receptors

Figure 113 shows the correspondence rule that is being used in the comparison process. On the right side of the picture two data receptors are represented. Attending to the left side hand one - the camera - just two possible values can be detected, either black, corresponding to the numerical symbol 1, or red equivalent to the numerical symbol 2. On the other hand, the temperature sensor differentiates between five temperature ranges, with their correspondent numerical symbol. In order to assure correct comparison, the same criterion has to be followed for every receptor during the assignment of numerical symbols. In this example the followed criteria is to assign the lower symbol to the input related to lower temperature. The rest of the inputs are related in ascending progression depending on the temperature range that they cover to a numerical symbol in ascending sequence.

To come along with this problem one-to-one comparisons are supported. This comparison process needs two values per input data source: the maximum number of possible symbols and the corresponding position number of the current symbol in the list of possible symbols of the referred input data source.

For example, when considering optical perception the maximum number of symbols concerning the temperature of a heating plate of the stove would be two, either cold or hot. In this case the position number of the current symbol would be one, in case the current input symbol was cold or two in case it was hot. On the other hand, considering the input from the fieldbus network referring to the same measured variable the maximum number of possible symbols would be five while position numbers would be: one for cold, two for cool, three for mild, four for warm and five for hot. Summarising, compared values requires to consider not only the simple symbol but also the maximum number of possible symbols.

When comparing, current symbol position value, represented in Figure 114 as Sp, plays the role of divisor while maximum number of possible values, represented in Figure 114 as M, appears as dividend. Considering a single one-to-one comparison, the comparison is positive if the result of the division with bigger dividend - higher maximal number of possible symbols - is between the results of the divisions with smaller dividends - smaller number of possible symbols, having as divisors the position number of the current symbol of the corresponding input data source plus one and minus one (provided that the current symbol position number is neither one nor the maximum possible position number).

Otherwise the comparison is considered negative. If comparison is considered positive the selected symbol, which is to be stored in the 'dynamic results' table, is the one coming from the data source with maximal number of possible symbols. In case the comparison is negative the new input data in the dynamic table is zero. This way of selecting the symbol that has to be stored in the 'dynamic results' table assures a finer end result, since the data source with higher number of possible symbols is the finer data source.

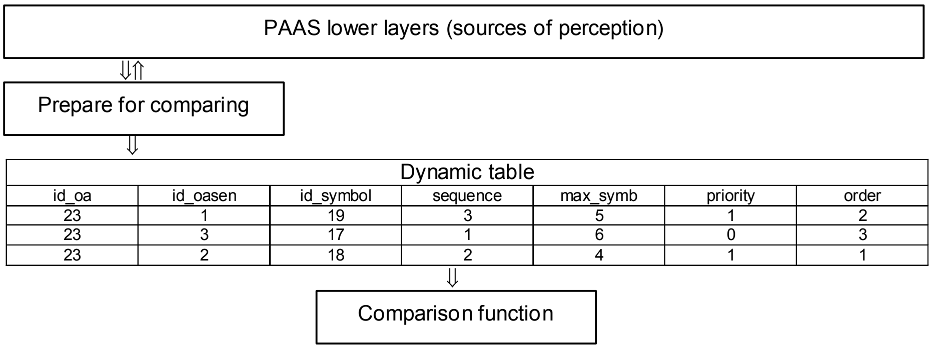
$$\frac{\text{Mmin}}{\text{Sp min} + 1} < \frac{\text{Mmax}}{\text{Sp max}} < \frac{\text{Mmin}}{\text{Sp min} - 1}$$

Figure 114 Comparison formula

Since global database table solution requires a large database where every current symbol from every measured variable has to be stored two other possibilities have been considered. One-to-one comparisons occur however in the same way that described.

- **Specific Dynamic Database**

The specific dynamic database solution consists on a dynamic table where, opposite to the global database table solution, symbol values to be compared are stored at the time that comparison is required, i.e. as soon as a symbol changes. The way this table is generated by asking the corresponding sources, which is the current value measured by each of them concerning the variable related to the perceived changed, and translating these values into symbols.



id_oa	id_oasen	id_symbol	sequence	max_symb	priority	order
23	1	19	3	5	1	2
23	3	17	1	6	0	3
23	2	18	2	4	1	1

Figure 115 Comparison process with dynamic table

For example, in case the symbol related to the variable temperature of one of the heating plates of the stove coming from the optical data source changes the system stores the new value, after being turned into a symbol, into the dynamic database table together with information required by the comparison function such as the number of possible symbols related to the measured variable and the optical data source. Since all current measured symbol values are related to this particular variable and are needed for comparison, the system has to ask every other data sources, which also takes care of this variable, which are the values that they are detecting at that time. In the current example the system would search for the symbol value detected by the fieldbus network and the number of possible symbols for the determined variable when measured by the fieldbus network source. As end result of this process the dynamic database is filled in with the current symbols, and additional required information, from the corresponding different data sources, as it is shown in Figure 115.

The main advantage presented by the specific dynamic database solution comparing to the global database table solution is the no requirement of a large database where every current symbol is stored. On the other hand, since the system has to ask every current value related to the specific variable each time a change occurs in any of the current symbols, and translate them into symbols, comparison process has to wait until the specific dynamic database has been filled in. Therefore the main problem presented by this solution is the time that is required in order, not only to access the data stored in the dynamic database, time that is also required if implementing the solution that has been first exposed, but also the required time to create the specific dynamic table whenever a current symbol changes.

- **Instantaneous Data Array**

The last proposed solution to come along with the comparison process is to improve some weak points presented either the first solution, global database table, or the second, specific dynamic database. In such a way this third solution is designed under the premise:

- No global database, as it is already defended by the solution specific dynamic database
- No dynamic database but an array

The instantaneous data array solution consists of a dynamic array composed of data that is required for the one-to-one comparison process. Therefore data referring not only to current symbols, which attend the same variable but come from different sources of perception, but also maximal number of symbols that available in each case depending on the source of perception are information that contained in the instantaneous data array.

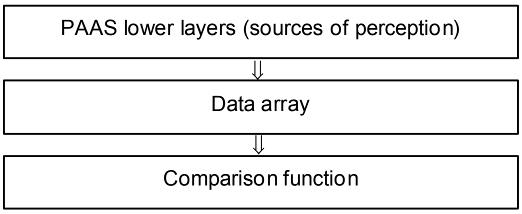


Figure 116 Comparison process with data array

The way this array is generated is similar to the way the dynamic table is generated in the specific dynamic database solution. As soon as the system detects a change in any current symbol, symbol values related to the affected variable, but coming from different sources of perception that also take care of this variable, are searched and stored as components of the instantaneous data array together with data required for comparison such as the maximal number of possible symbols.

In this case the comparison function receives the instantaneous data array, i.e. comparison function receives directly all needed data: current symbol values from every source that takes care of the specific variable object of comparison and the maximal number of possible symbols in each case. Since the function does not have to look for data in any table but receives the information directly the advantage of this solution, in comparison to the specific dynamic database solution, is less required time to acquire data to be compared. However, since data has to be extracted from the instantaneous data array to come along with the comparison process the process requires additional actions, which are unnecessary either when implementing the global database solution or the specific dynamic database and which bears additional required time.

Chapter 6

Results

This chapter is subdivided in two sections. During the first section individual results of each one of the perception systems that have been integrated in the SmartKitchen lab at the Institute of Computer Technology (TU Vienna) to meet the requirements of the function of perception for perceptive awareness automation systems (PAAS), which entails:

- Somatic and olfactory perception using LonWorks fieldbus technology
- Acoustic perception using Voice Extreme technology adapted to LonWorks fieldbus technology
- Visual perception using IEEE 1394 bus, OpenCV from Intel and proprietary pattern recognition software
- Remote perception through Internet

The second part concentrates on the function of global perception, i.e. global perception processing, which is subdivided into the main tasks of data symbolisation, data comparison and data association.

6.1 Individual Perception Systems

Since each one of the perception systems that have been integrated in the SmartKitchen perceptive awareness automation system (PAAS) presents its particularities attending not only to the own implementation's requirements but also to requirements for its adaptation into the SmartKitchen PAAS, results are presented and discussed individually in following subsections.

- **Somatic and Olfactory Perception**

As it has been described in section 5.1 and section 5.2, the functions of somatic and olfactory perception have been implemented using LonWorks fieldbus technology.

Though in most cases the registration and configuration of the sensors in the LonWorks network using the LonMaker configuration tool from Echelon were done without any difficulty, in a first phase appeared some communication interferences between the communication channels RS485 and FTT10. Connection errors were almost unavoidable in an installation, where each one of the two mentioned channels made use of two different wires in a common 8 threads flat cable. In order to diminish the probability of connection errors a new installation was done making use of separate twisted pair cables. At the same time the electrical installation (230 V AC, 12 V Dc and 24 V DC) was redefined to increase the flexibility of the system and to make the installation of devices much simple.

Independent of communication interferences unexpected problems of understanding also emerged during the configuration phase. In this case the problem was due to few 'LonWorks compatible' sensors such as Helio multisensor from the company Philips [Hel 03] that had been developed without following common rules in reference to definition of network variables. These sensors could only work as part of a subsystem where all components were developed by the same company, not supporting the normal communication between them and the rest of devices of the network. Since the integration of these sensors in the network lead to no functionality it was decided not to use these sensors.

Although these sensors were the most problematic, there are other ones that also present some disadvantages. In this case the weakness refers to the power supply and concerns the smoke sensors from the company Arigo. The fact that a 9 V battery supplies these devices decreases their grade of availability.

In reference to the integration of the LonWorks network as part of the SmartKitchen perceptive awareness automation system, the task of data transfer from the LNS database to the PAM database making use of the OPC as interface did not entail much complication. However, due to the large quantity of defined LonWorks variables much work was required concerning data symbolisation (see section 6.3).

- **Acoustic Perception**

As it has been described in section 3.2 the acoustic perception has been implemented following a distributed system philosophy, by means of adapting acoustic modules to the LonWorks distributed networks (Figure 117).

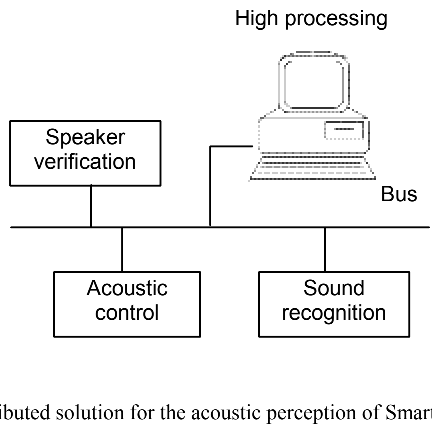


Figure 117 Distributed solution for the acoustic perception of SmartKitchen PAAS

In such a distributed solution the limited processing capacity of the developed acoustic modules could be seen as a disadvantage. However the fact that this first acoustic perception system has been developed to cover the functions of speaker verification, acoustic control, sound recognition, and acoustic reaction not for every event but in determined circumstances reduces the required processing capacity of the modules.

One of the main benefits of the implemented solution is the direct communication between systems, which enables fast reaction of the system in front of determined situations. For example, in case that there was an intruder in the room the acoustic perception system would detect it and would automatically communicate this intrusion to specific devices of the LonWorks network. These devices would automatically execute predefined reactions such as blockade the restrict access to specified parts of the room and turn on the alarm.

In order to know the rate of success of each one of the acoustic modules, these have been submitted to a test phase. The acoustic control module has been tested using seven different voices and four different words: ‘active’, ‘panic’, ‘ein’ and ‘aus’. The system has always recognised each one of these four words while being spoken by five of the seven voices. In the case that the other two voices, the system was not capable to recognise any of the words. The reason of this unsuccessful result was the bad quality of the recorded words, which were hardly recognisable even for a person.

In reference to the function of personalised password recognition tests were always successful. In this case four different passwords, which entails four different voices (persons), were recorded and stored as password templates. During the test phase the system recognised each one of the passwords once the correspondent person pronounced them, while ignoring passwords that were pronounced by a different person than the one that recorded the template.

The sound recognition module was tested for the running water sound. In this case recognition success depended on the quantity of running water. In case of enough running water recognition tests were always successful. However, for low water flows the system could hardly recognise the running water sound.

- **Visual Perception**

First implementations concerning visual perception have been limited to integrate frontal face pattern recognition in the SmartKitchen PAAS. The face pattern recognition software that has been installed (see section 5.1.4) has been submitted to different tests to obtain its effectiveness depending on the deviation grade.

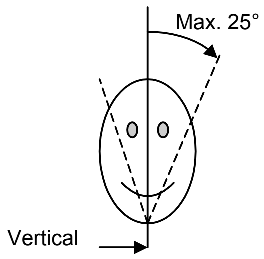


Figure 118 Success rate for face detection depending on the inclination angle of the face

The pattern recognition process consists on two main tasks. During the first task the system detects a face, while during the second task the system recognises the detected face.

First tests have been performed in reference to face detection. In this case results have shown that the system is capable of detecting faces with a higher deviation grade of +- 25° from the vertical, as it is shown in Figure 118. In those cases in which the deviation angle was less than 25 degrees the system not only detected the faces but in most cases, it also recognised them.

Focusing on the function of recognition, in order to test the success rate of the system a man's face without beard was stored as template. Recognition tests were also performed on man with two-day beard growth. In this case the system was not capable of recognising the face.

- **Remote Perception**

As it is shown in Figure 119 in this first implementation of the remote perception function the graphical user interface (GUI) communicates to the SmartKitchen perceptive awareness automation system (PAAS) by means of two different connections. On one hand the GUI is connected to the OPC client application called 'values monitoring', on the other hand the GUI is connected to the OPC client application called 'changes handling'.

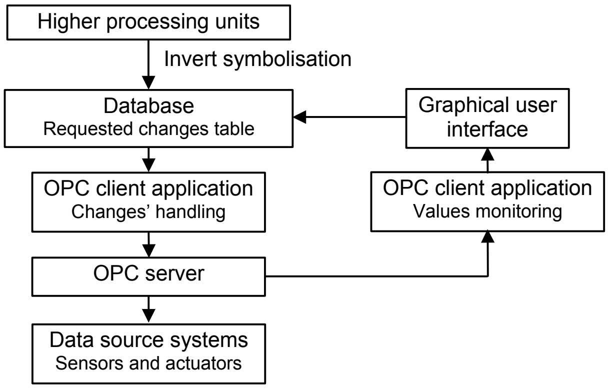


Figure 119 Diagram for GUI communication

The first connection is required to support remote data monitoring. Depending on user's rights the 'values' monitoring' client application asks the OPC server for the values of determined parameters of the environment such as the temperature of the refrigerator.

The second connection enables the parameters of the actuators of the data source systems, for example the brightness level of a lamp to be changed. In this case, both remote users' requested changes and higher processing units' requested changes are stored into the database of the system ('requested changes table' in Figure 119) together with a priority label. Depending on the priority of the requested changes the OPC client application 'changes handling' asks the OPC server to make a change or another. Once the OPC server receives a change order it changes the value of the correspondent parameter in the database of the correspondent data source system, which is automatically executed.

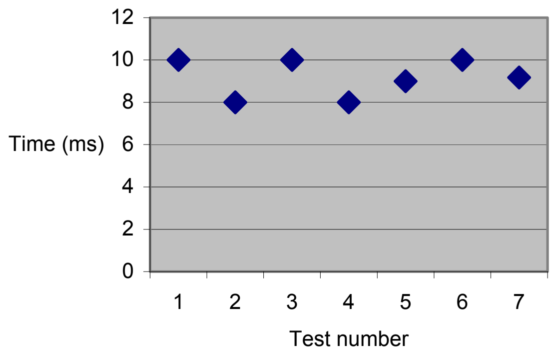
Remote perception has been tested independently of the higher processing units of the SmartKitchen. In this case, the system has needed approximately 170 ms to execute a requested change.

6.2 Global Perception

The required data processing to reach global perception mainly entails the higher functions of data symbolisation, data validation, and data association working on continuous modus.

- **Data Format Adaptation - Symbolisation Function**

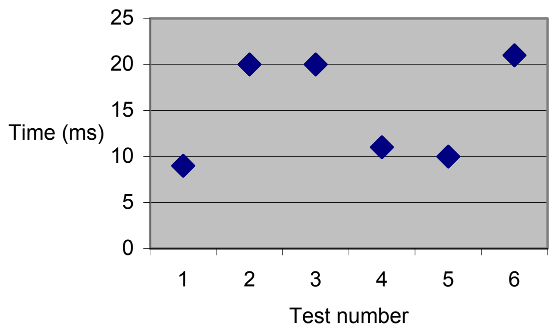
Several tests, which have been focused on the time that the transformation function needs to symbolise values, have resulted on two average times of approximately 192 ms and 189 ms. In the first case test were performed considering the order get_data(next). In this case the data collector layer waits for the order get_data(next) from the transformation layer before sending any value. In the second case, the data collector layer sends values without waiting for any order from the transformation layer. Comparing both average times: 192 ms and 189 ms, one concludes that waiting for the word get_data(next) entails an average delay of 3 ms.



Average =9,16 ms

Figure 120 Time required to symbolised 1 data value

A second test has been performed considering the symbolisation function itself. In this case it has been measured the time that the symbolisation function needs to symbolise 1, 2, and 3 data values respectively. Results obtained from this test, which are shown in the diagrams of Figure 120, Figure 121, Figure 122, have shown that the efficiency of the function increases with the number of values that are waiting for being symbolised.



Average =13 ms

Figure 121 Time required to symbolised 2 data values

In the first case (Figure 120) the process needs approximately 9 ms just to symbolised one data value. Analysing Figure 121 one observes that the system needs less time per data value to be symbolised. To be exact this second test results on an average value of 13 ms to symbolise 2 data values, i.e. 6,5 ms per data value. The tendency to this higher efficiency is corroborated with Figure 120. In this case, the process symbolises data values in an average time of approximately 15 ms, that means that the process needs approximately 5 ms pre data value to be symbolised.

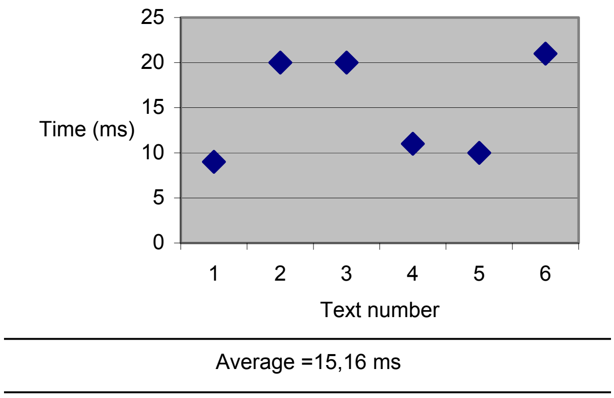


Figure 122 Time required to symbolised 3 data values

The reason of this increase of efficiency depending on the number of symbolised data values can be partly due to the time that the system requires to start symbolisation process. Based on the average results of these three tests one could conclude that the system requires about 6 ms to start the process plus 3 ms to symbolise one data value. However, analysing individual results in the second and third case (two and three data values to symbolise respectively) one observes a big discrepancy between the resulted time values. According to the diagrams of Figure 121 and Figure 122, while in some cases the system needs less than 5 ms to symbolise one data value in some others 10 ms are required. This time discrepancy can be due on one hand to the processing load of the computer, which can be busy with other tasks, and on the other hand to the database access load, since more processes can request accessing the database at the same time.

- **Data Validation - Comparison Function**

The function of data validation principally consists on the comparison function, which has been presented in section 4.3.2. In addition to the time required by the comparison function, the process needs a time to realise the change of the value of an attribute and a time to access and get the additional data that is needed for the comparison function.

The diagram shown in Figure 123 represents the time in milliseconds that is required by the function of comparison depending on the number of values to compare, i.e. depending on the number of redundant values. Since redundant values have to be compared one to another the number of comparisons that the process has to execute respond to the formula '$\sum(j*(j-1))$ from j=(2,n)' where n represents the number of redundant values.

As it was expected, the diagram shows that the higher the number of redundant values, the bigger the required time to execute the comparison function. An interesting result is obtained in that case that the system just receives one data element, i.e. in case there is no data redundancy. As in is represented in the first column of the diagram, even in this case the system needs approximately 15 ms to execute the function. Since in this case there is no comparison at all, the obtained result represents the time that the process needs to access determined information related to the particular data such as the identification number 'id_oa', which the system needs for upper processing, from one table at the database of the system, plus the time that the process needs to insert this data to a second table at the database.

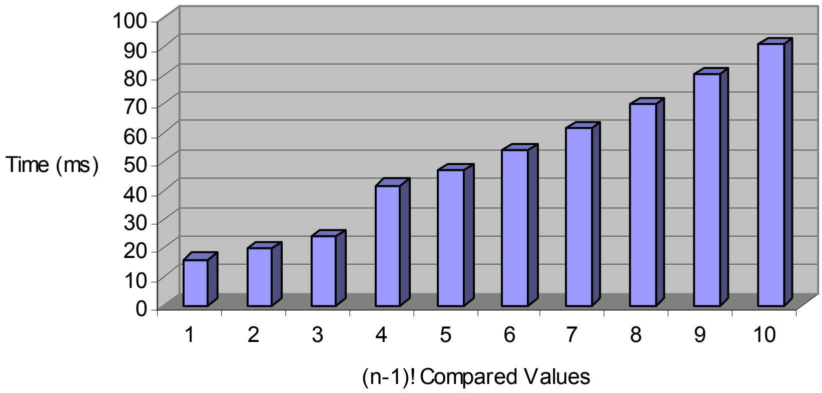


Figure 123 Comparison function

The diagram shown in Figure 124, corresponds to the function of data comparison once this function has been integrated in the system. This integration entails that the function needs of an additional time to execute the data transfer between layers, to search for the redundant data in the database, and to collect the different information related to the transferred current measured data value and to the correspondent redundant data that is needed for the comparison function to execute. Though the diagram also shows a temporal growth at the time that the number of redundant values augments, this temporal function does not present a linear growth according to the number of compared values.

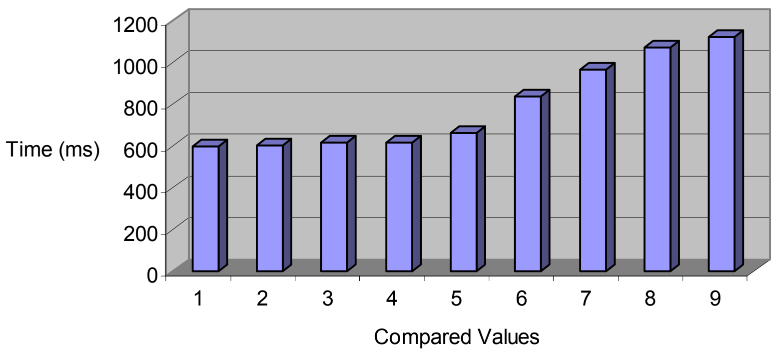
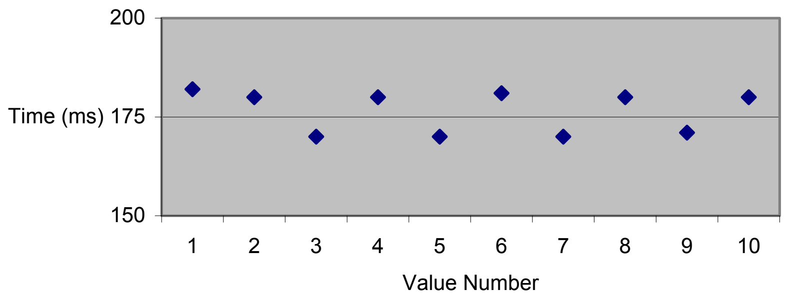


Figure 124 Data transfer and comparison

Comparing Figure 123 and Figure 124 one concludes that in reference to the function of data comparison, the system spends most of the time executing secondary but necessary tasks such as data transfer between layers and accessing the database, while less time is needed to execute the comparison function itself.

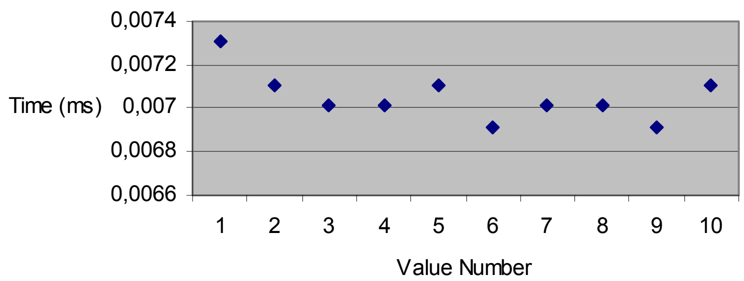
The diagrams in Figure 125 and Figure 126 represent the time that the process needs to transfer data from the perception layer to the adaptation layer in both cases, when the adaptation layer asks for a new value ‘get_data(next)’ and when the layer asks for the last sent value ‘get_data(repeat)’. In both cases ten different measurements have been done in order to obtain a better average of the time that is required for these operations to happen. Comparing both diagrams one can clearly see that the function of ‘get_data(next)’ needs much more time to execute than the function of ‘get_data(repeat)’, to be exact: time(get_data(next)) ≅ 25000*time(get_data(repeat)).



Average value: = 176,4 ms

Figure 125 ‘Up-botton’ data transfer with order get_data(next)

This time difference is due to the additional time that the function ‘get_data(next)’ requires to get the different information referring to the new data. In case of the function ‘get_data(repeat)’, the process already knows this information, which remain stored in different variables of the function untill it is replaced by the information of a new data as soon as the function ‘get_data(next)’ is called.



Average value: = 0.00705 ms

Figure 126 ‘Up-botton’ data transfer with order get_data(repeat)

The time that the process needs to transfer data from the perception layer to the representation layer (Figure 127) in both cases, when the representation layer ask for a new value 'get_data(next)' and when the layer ask for the last sent value 'get_data(repeat)' is represented by the diagrams in Figure 128 and Figure 129.

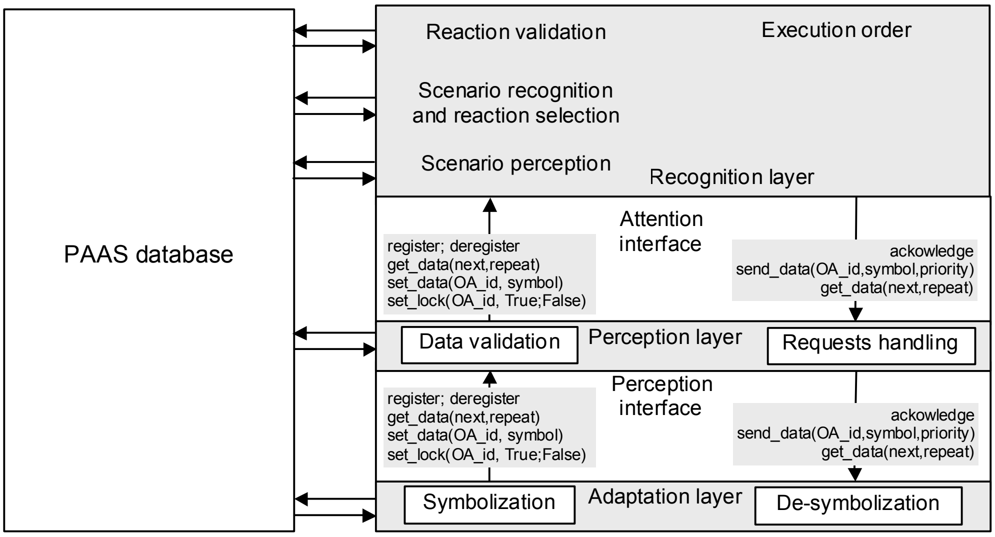


Figure 127 Structure of the upper layers of the SmartKitchen PAAS

In the same way as it has been measured the time that is required by the 'up-bottom' data transfer between the perception layer and the adaptation layer, time measurements for 'bottom-up' data transfer between the perception layer and the recognition layer have been done in ten different tests.

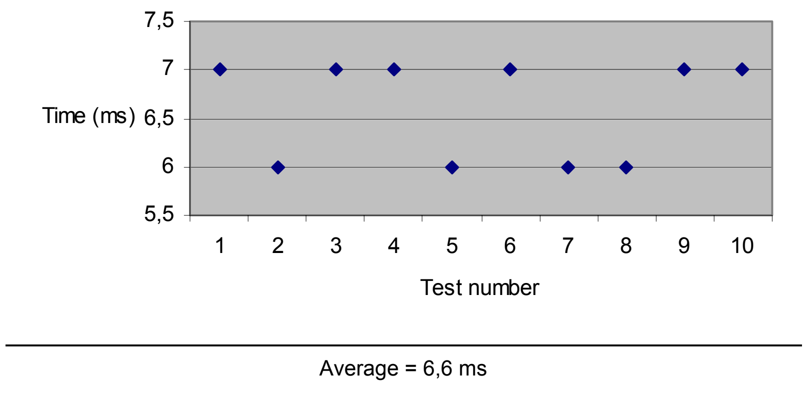
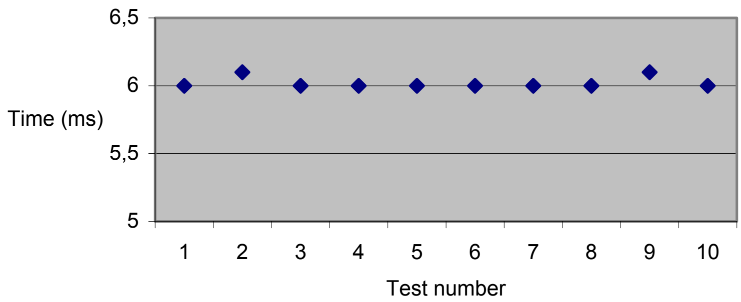


Figure 128 Bottom-up data transfer with order get_data(next)

Comparing both diagrams (Figure 128, Figure 129) one observes that the function of 'get_data(next)' needs more time to execute than the function 'get_data(repeat)'. However the time difference is much smaller than the one observed per transfer data from the perception layer to the attention layer: 'time(get_data(next)) ≅ 1,1*time(get_data(repeat))'. In this case the time difference is due to the action of 'delete(record)' that has to be executed by the function 'get_data(next)' but not by the function 'get_data(repeat)'.

The time difference between data transfer from the perception layer to the adaptation layer (Figure 125, Figure 126) and data transfer from the perception layer to the representation layer (Figure 128, Figure 129) is due to the quantity of data to transfer. While in the first case seven different values pre transfer data are required in the second case just two values are moved.

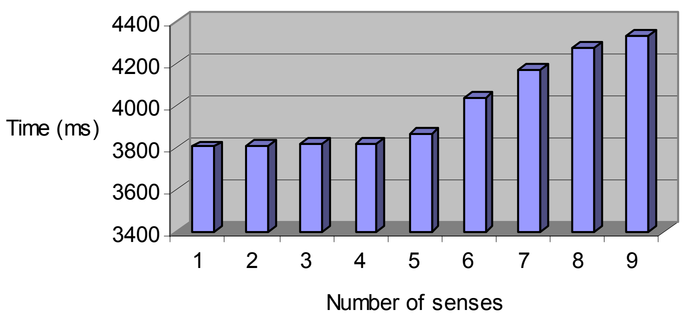
Tests in reference to the recognition process and the reaction process appear at the work done in the PhD thesis of G. Russ [Rus 03]. For these tests, ten different models scenarios have been defined and stored in the database of the system. The average values are 150 ms to represent the global situation as an association of objects in a particular state, 2700 ms to recognise the represented global situation, and 120 ms to select a proper reaction according to the recognised situation and send the correspondent orders to lower processing layers.



Average = 6,02 ms

Figure 129 Bottom-up data transfer with order get_data(repeat)

Summarising, in case of normal processing, the system needs an average time of approximately 4 s to complete the data processing as it is shown in the diagram of Figure 130, which results from adding the different required times. Data processing starts as soon as a change is detected in the environment by any of the receptor source technologies and finishes when the reaction selected by the upper processing layers is sent to upper lower layers.



Average = 3993 ms

Figure 130 Processing time depending on the number of redundant data

With that it is meant that in the case of a dangerous or problematic situation the preventative behaviour of the system requires of approximately 4 s to respond. Obviously, in normal situations, in which reactions are ordered at lower layers, the system seems to behave as a common automation system. In this case, though prevention keeps on running, the system reacts by means of either reflex actions such as turning off the light when the room is not occupied or basic actions such as turning on the light when somebody comes into the dark room.

Testing the preventative reaction of the system has been done simulating the execution of the reaction by means of activating a red light. This simulation has been necessary since nowadays common automation system's applications (e.g. LonWorks light control) are not suitable for being overridden by an external system as upper layers of the SmartKitchen perceptive awareness automation system.

Chapter 7

Conclusion and Further Work

During the first section of this chapter the result of the perceptive awareness model (PAM) is firstly examined. Afterwards some comments are made about features and possibilities of this PAM end design and its implementation as SmartKitchen perceptive awareness automation system. The chapter concludes by presenting suggestions to improve the functionality and effectiveness of both model and practical implementation and new targets to reach in the future.

7.1 Conclusion

The designed perceptive awareness model (PAM) enables a cooperative working between different technological solutions that can benefit automation systems. Since the model is suitable not only for common fieldbus systems but also for different technical solutions that related to data collection, it contributes to most reliable and multifaceted automation systems, as it has been discussed in section 1.2 and section 3.3.

In such a way, on one hand and from the technical point of view PAM is the step forward towards systems' cooperation in automation, covering the deficiencies described in section 1.1.3 and section 2.1.3. On the other hand and from the user's benefit point of view PAM is the way to more secure and safe automation systems.

7.1.1 Perceptive Awareness Model (PAM)

In order to analyse and discuss the end results of the design process of the perceptive awareness model is it necessary to go through the axioms that were made at the beginning and examine if the end design meet these predefined expectations.

Axiom 1. *'An extension of the automation model is the first step towards a teamwork and systems' integration in home and building automation'.*

Though most of the high processing layers of the model work with comparisons and associations of input data from different technologies (as example see chapter 5) this statement is already fulfilled at the first layer of the higher processing part of PAM. The layer accomplishes the requirements of this first statement by means of supporting the function of unification of data formats, as it is shown in Figure 131.

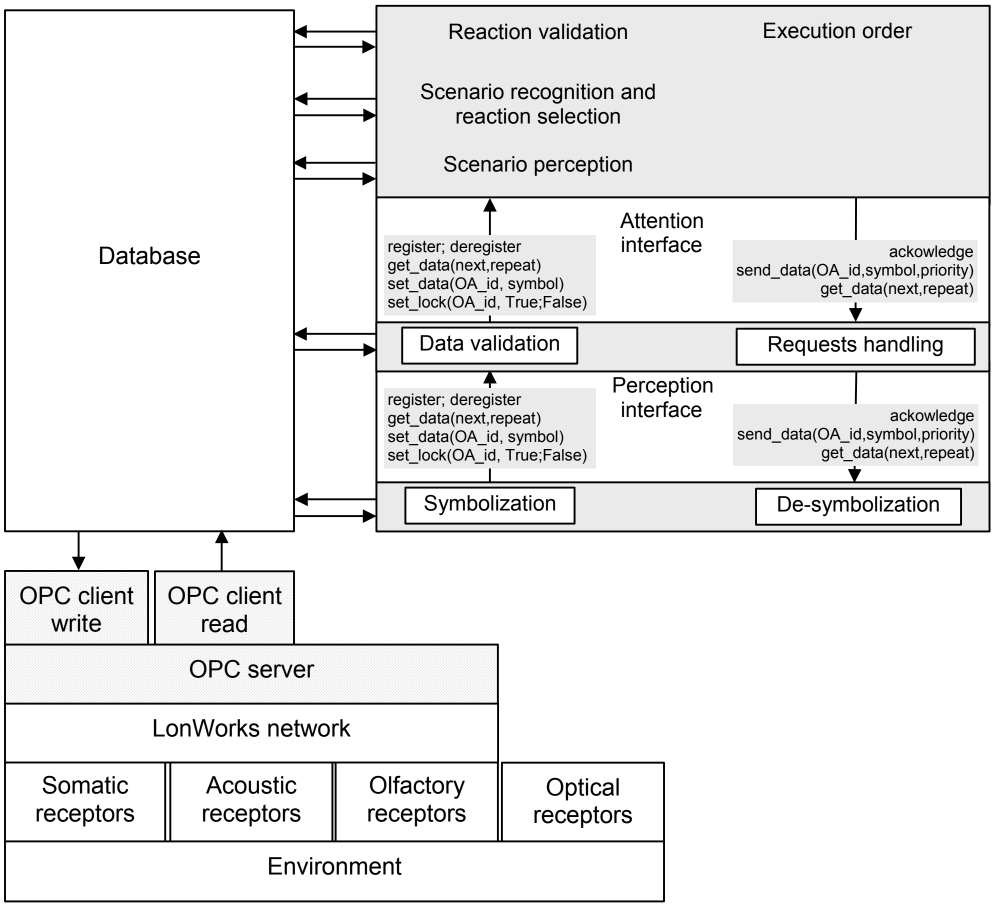


Figure 131 Structure of an example of PAAS

Axiom 2. *'The extension has to be compatible with the conventional OSI model and hence provide impressive enhancements to generalised automation control systems'.*

In the same way that this second axiom is related to the first axiom, implementations required to fulfil them are also related. As it is shown in Figure 131 the first layer of the higher processing part of PAM accesses PAM global database, which entails that collected data has to be placed here. Since the new model has to be suitable for existent technologies based on the OSI model an additional layer is required to place collected data, which is usually stored in the database of the correspondent technology, into this database. In Figure 131 this layer has been particularly defined for LonWorks fieldbus technology. In this case data transfer is supported through OPC (OLE for process control), which can be used as data transfer interface for fieldbuses.

Axiom 3. *'Understanding aids co-operative work between sub-systems, which is required to achieve perceptive awareness'.*

Results of the way of working, i.e. of behaving, of the implemented SmartKitchen perceptive awareness automation system prove this axiom (see section 6.3).

Axiom 4. *'The functional elements to reach perceptive awareness are sensory perception, world modelling, reaction judgement, and behaviour generation'.*

The whole PAM is designed on the bases of these four functions. Data collection (first part of sensory perception) is supported at the first layer of PAM. Data transfer layer (OPC in Figure 131) enables to place all collected data into PAM database, which makes easier further processing that lead to world modelling. Reaction judgement occurs at the higher layer and is followed by higher behaviour generation. The model also supports lower behaviour generation at lower layers (see section 3.3.2).

Axiom 5. *'The functional elements are supported by a knowledge database that stores information about the world in the form of symbolic variables and rule'.*

As it is shown in Figure 131, the final perceptive awareness model (PAM) is equipped with a database that is open to both parts of the model (low and high data processing parts). This database contains different information such as not-unified collected data, symbolised data (attributes) grouped in form of objects, and model scenarios as association of determined objects in particular states.

Axiom 6. *'The complexity of processing can be managed through hierarchical layering'.*

The structure represented in Figure 131, corroborates the modular architecture of the designed perceptive awareness model (PAM).

Axiom 7. *'The complexity of the real world can be managed by means of attention'.*

This axiom affects the management of each perceptive awareness automation system. The aspect of attention is covered in PAM through the definition of priority labels. During the configuration phase different priority labels are given to determined events such as fire detection, water leakage and child alone.

7.1.2 SmartKitchen Perceptive Awareness Automation System

Considering the implementation of PAM, SmartKitchen PAAS has shown that in order to let the system support a preventive reaction some changes are required at lower actuator-levels. In order to let the system react in a preventative way reflex actions and basic control functions have to be suitable for overriding. However, nowadays, common automation technologies like the ones that have been used in SmartKitchen PAAS are not designed to enable this override.

Low effectiveness is also found in today's commercial visual and acoustic pattern recognition systems, which may improve in the future. In both cases recognition capabilities are limited to few specific things. For example in the case of visual pattern recognition to recognise the colour red of a heating plate of the stove (see section 5.1.4), and in the case of acoustic pattern recognition to recognise the word 'password' (see section 5.1.3).

In reference to face recognition most existent commercial solutions, like the one used in SmartKitchen PAAS, are limited to frontal face recognition, which also restricts the efficiency of the system.

Considering olfactory recognition in meaning of recognising odours and smells, commercial systems (i.e. solutions for a reasonable price) seem to be far less effective compared to visual and acoustic systems (see section 5.1.2).

In reference to required response time, results have shown that most of the measured time is required to access the database. However, since there are no presumptions made about the indispensable need of a database to store data, but just about some recommended features of the data storage of a perceptive awareness automation system (PAAS) (see section 4.3.4), different solutions can be implemented in the future.

Another important point of the implementation is that of environment's description. Since a PAAS is thought to process data as attributes of particular objects, which represent the environment, there is also a need to find a way to support the description of the environment through its different objects. The design of such a tool requires of a first detailed analysis of attributes that are relevant in the different possible environments.

Furthermore, measured values of the several attributes have to be related to particular symbols to enable efficient higher data processing as it has been exposed in section 4.3.2). Besides, symbol coherence is demanded between values of a same attribute, which are detected by different technologies (e.g. LonWorks and IEEE 1394 camera). In order to meet these requirements another environment configuration tool is needed. As it is commented in [Fal 03], using Fuzzy logic during the transformation phase is also a possibility to consider in the future to obtain more precise values.

Since the designed PAM does not integrate learning, it is also needed to find a way to let the system know about its 'past experiences'. With that it is meant that a tool is needed to define scenarios up to the described environment, i.e. up to the described objects and specific values of their respective attributes.

Summarising, on the bases of the designed perceptive awareness model (PAM) perceptive awareness automation systems are needed of environment configuration and past experiences configuration

7.2 Further Work

This work, together with the dissertation of Gerhard Russ [Rus 03], is the first step towards turning the dream of perceptive awareness automation systems into reality. Further steps will be done concerning both the design of the perceptive awareness model (PAM) and the implementation of the SmartKitchen perceptive awareness automation system (PAAS).

Attending to processing time requirements and according to the executed tests, one observes that the SmartKitchen PAAS spends most of the processing time accessing the database, and reading and writing data from and to this data storage. In such a way, it seems reasonable to test the processing time rate of future implementations using another kind of data storage solutions or trying to improve the accuracy of the different applications that access the database.

Another aspect of the implementation to be next improved corresponds to face recognition. A first idea is to integrate not only frontal but also profile face recognition By means of this extension it is expected a higher success rate of the face recognition application.

In reference to the implementation of the system not only does the response time need to be improved but extensions have to be made. For example, as it has been exposed in the previous section two environment's description tools are required. The first tool has to aid to describe the environment through its objects, which consists on particular attributes as measurable, detectable or suitable for monitoring data points. The second tool will be responsible for the description of all attributes of the correspondent environment (as detectable, measurable or suitable for monitoring parameters of any of the existent objects) in means of a relation between input values and symbols. In reference to this second tool, and as it has been briefly mentioned in the previous section, in order to work with more precise values data symbolisation could be done using Fuzzy logic rules. Christian Lechner is undertaking first researches in this direction in his diploma thesis: 'Intelligent Software for Data Processing in a Profibus based Distributed Control System'. Though Lechner works with Profibus, since the SmartKitchen as PAAS enables the integration of every technology suitable for perceiving the environment, as it is the case of Profibus, the integration of his work as part of the PAAS at the Institute of Computer Technology (TU Vienna) may entail no problem.

Attending to the design of the perceptive awareness model, the integration of new capabilities in future extensions are further steps to work out. Among these capabilities one can think in functions such as learning and forgetting. Attending to the first function, first thoughts have been done considering the possibility of integrating an events-relation function to enable a kind of learning.

Due to the openness of the designed perceptive awareness model, perceptive awareness automation system can be integrated in different environments. Private houses, schools, residences, hospitals, hotels, office buildings, shopping centres, etc. can take advantage of such automation systems. The description of the environment, as well as the definition of predefined scenarios, has to be done according to the functional wish. For example, in case of integrating a perceptive awareness automation system in a hospital, it would be convenient to predefine scenarios concerning patience behaviour in order to allow the system to detect irregular comportments that can lead to a dangerous situation such as heart attack.

Though the capability of perceptive awareness enables automation systems to present a preventive behaviour, proper descriptions not only of the environments but also of the predefined scenarios are crucial factors to achieve the expected behaving of the systems.

Abbreviations

AC	Alternating Current
AI	Artificial Intelligence
ANSI	American National Standards Institute
API	Application Programming Interface
ARCNET	Attached Resource Computer Network
ASHRAE	American Society of Heating, Refrigeration and Air-Conditioning Engineers
ASP	Active Server Page
BACnet	Building Automation Control network
BAS	Building Automation Systems (
BCI	Batibus Club International
BCU	Bus Coupler Unit
CA	Central Arithmetical unit
CAMAC	Computer Automated Measurement And Control
CBA	Consciousness-Based Architecture
CBR	Case-Based Reasoning
CCITT	Comité Consultatif International Téléphonique et Télégraphique
CDA	Communal Data Access
CDF	Communal Data Format
CEN	European Commission for Standardization
CENELEC	European Commission for Electrotechnical Standardization
CMattie	Conscious Mattie
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CU	Central Control unit
CV	Computer Vision
DA	Data Association
DC	Direct Current
DDC	Direct Digital Control
DDL	Data Definition Language
DS	Data Significance
EDVAC	Electronic Discrete Variable Automatic Computer
EEPROM	Electrical Erasable Programmable Read Only Memory
EHS	European Home System
EHSA	European Home Systems Association
EIB	European Installation Bus
EIBA	European Installation Bus Association
ENIAC	Electronic Numerical Integrator and Computer
FAP	Function Access Point
FTT	Free Topology Transceiver

GPIB	General Purpose Interface Buss
GUI	Graphical User Interface
HOP	Higher-Order Perception
HOR	Higher-Order Representation
HOT	Higher-Order Thought
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HVCA	Heating Ventilation and Air Condition
IBM	International Business Machinery
ICU	Intelligent Control Unit
ID	Identification
IEC	International Engineering Consortium
IEEE	Institute of Electrical and Electronic Engineers
IO	Input/output
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International organization for standardization
ITU	International Telecommunication Union
JTC	Joint Technical Committee
KNX	Konnex
LAN	Local Area network
LNA	Local Area Network
LNO	LonWorks Nutzer Organisation
LNS	LonWorks Network Services
LON	Local Operating Networks
M	Maximal possible value
M	Memory
MAP	Manufacturing Automation Protocol
MIL STD	Militar Standard
MMI	Man-Machine Interface - Machine-Machine Interface
MMS	Manufacturing Message Specification
MS	Master-Slave
NOAH	Network Oriented Application Harmonisation
NV	Netzwerkvariable
ObIS	Object Interface Specification
OLE	Object Linking and Embedding
OPC	OLE for process control
OSI	Open Systems Interconnections
PA	Perceptive Awareness
PAAS	Perceptive Awareness Automation System
PAM	Perceptive Awareness Model
PC	Personal Computer
PCLTA	PC LonTalk Adapter

PL	Power line
PLC	Programmable logic controller
PTP	Point-To-Point
R	Reaction
RACKS	Reusable Application Interface for Communicating - Real Time Kernels
RAM	Random-Access Memory
RCS	Real-time Control System
ROM	Read Only Memory
RS	Recommended Standard
S	Situation
SA	Smart Actuator
SAP	Service Access Point
SC	Subcommittee
SCPT	Standard Configuration Property Type
SDL	Specification and Description Language
SDL/GR	Graphical Representation
SDL/PR	Phrase Representation
SNVT	Standard Network Variable Type
Sp	Symbol Position
sS	Smart Sensor
SSA	Subsumption Architecture
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TP	Twisted Pair
TP	Token Passing
TU	Technical University
UML	Unified Modeling Language
USB	Universal Serial Bus
WAN	Wide Area Network
WG	Working Group

Bibliography

[Aam 94] Aamodt, A. and Plaza, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press, Vol. 7: 1, pp.39-59, 1994

[Alb 96] Albus, J.S.: The Engineering of Mind. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior: From Animals to Animats 4, Cape Code, MA, September 1996

[Alb 99] Albus, J.S.: 4-D/RCS Reference Model Architecture for Unmanned Ground Vehicles. Proceedings of the SPIE Aero Sense Technical Conference 3693, Orland, FL, April 1999

[Asf 00] Asfour, T., Berns, K., Dillmann, R.: The Humanoid Robot ARMAR: Design and Control; Proceedings of the First IEEE-RAS International Conference on Humanoid Robots; HUMANOIDS 2000, Boston, MA, USA, 2000

[Ash 03] BACnet – A Data Communications Protocol for Building Automaton and Control Networks. BSR/ASHRAE Addendum d to ANSI/ASHRAE Standard 135-2001. ASHRAE 2003

[Ast 88] Astrom, K. J., Wittenmark, B.: Sistemas Controlados por Computador. Ed. Paraninfo, 1988

[Ayd 00] Aydede, M., Güzeldere, G.: Consciousness, Intentionality and Intelligence: Some Foundational Issues for Artificial Intelligence. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 12, pp. 263-277, July 2000

[Bal 90] Ballard, D. R.: Conjoint Computing: Integrating Numeric, Symbolic and Neural Computing. Reticular Systems Inc. IEEE, 1990

[Bar 87] Barsalou, L.W. The instability of graded structure: Implications for the nature of concepts. In U. Neisser (Ed.), Concepts and conceptual development: Ecological and intellectual factors in categorization (pp. 101-140). Cambridge: Cambridge University Press, 1987

[Bar 88] Baars, B. J.: A Cognitive Theory of Consciousness. Cambridge University Press, 1988

[Bar 97] Baars, B. J.: In the Theater of Consciousness. Oxford University Press, 1997

[Bec 95] Bechtel, W.: Consciousness: Perspectives from Symbolic and Connectionist. AI. Neuropsychologia, 33, 1075-1086, 1995

[Ber 90] La Berge, D.: A Network Simulation of Thalamic Circuit Operations in Selective Attention. Journal of Cognitive Neuroscience, 2, 358-373, 1990

[Bis 99] Bischoff, R. and Graefe, V.: Integrating Vision, Touch and Natural

Language in the Control of a Situation-Oriented Behaviour-Based Humanoid Robot. IEEE International Conference on Systems, Man and Cybernetics. Tokyo, October 1999

[Boo 65] Booth, A. D.: Automatic Digital Calculators. Third Edition, Butterworth & Co. Publishers Ltd., pp. 9-24, 1965

[Bro 97] Brockmeier, S.: Computational Architecture and the Creation of Consciousness. The Dualist, Undergraduate Journal of Philosophy, vol. 4, 1997

[Bur 99] Burns, R.W.: 57 (history of technology) meeting report. Engineering Science and Education Journal, Vol. 8 Issue: 5, pp. 208-208,1999

[But 02] Butler, J.: BACnet and Ethernet: partners for building automation, 2002

[Car 96] Carley , L. R., Gielen G. G. E, Rutenbar, R. A., Sansen, W. M. C.: Synthesis tools for mixed-signal ICs: progress on frontend and backend strategies. Annual ACM IEEE Design Automation Conference, Proceedings of the 33rd annual conference on Design automation conference, Las Vegas, Nevada, United States, 1996

[Cas 01] Casas Nebra, R.: Buses domóticos. Una perspectiva del corazón de la domótica actual. Montajes e Instalaciones, October 2001

[Chu 94] Churchland, P. M.: The Engine of Reason. MIT Press, Cambridge, MA, 1994

[Cia 93] Ciaccio, E. J., Dunn, S. M., Akay; M.: Biosignal Pattern Recognition and Interpretation Systems. Part 2: Methods for Feature Extraction and Selection. IEEE Engineering in Medicine and Biology, Dec 1993

[Cma 98] Ramamurthy, U., Bogner, M., Franklin, S.: "Conscious Learning In An Adaptive Software Agent." Proc. of The Second Asia Pacific Conference on Simulated Evolution and Learning. Canberra, Australia, Nov. 1998

[Com 90] Computing Before Computers. Edited by William Aspray. Iowa state university press, ISBN 0-8138-0047-1, 1990

[Cra 02] Craton, E., Robin, D.: Information Model: The Key to Integration. Automated Logic. AutoamtedBuildings.com., January 2002

[Dav 77] Davis, R. M.: Evolution of Computers and Computing. Science 195, pp. 1096-1102, 1977

[Daw 02] Daw, C. S., Finney C. E. A., Tracy E. R.: A review of symbolic analysis of experimental data. Submitted to Review of Scientific Instruments, 2002

[Dec 00] Deckel, A. W.: Getting the Big Picture: Higher Cortical Brain Functions. 2000

[Del 01] Delorme, D.: Modeling driver cognition. International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design. Colorado, 2001

[Des 64] Descartes, R.: Treasite on Man. In The Philosophical Writings of Descartes. Vol. 1, Cambridge University Press, 1664

[Des 66] Desmonde, W.H., Berkling, K.J.: The Zuse Z3. German Predecessor of the MARK I. 1966

[Die 00] Dietrich, D.: Evolution potentials for fieldbus systems. Proceedings of the WFCS2000, pp. 145-146, 2000

[Die 00b] Dietrich, D., Kastner, W., Sauter, T.: EIB – Gebäudebussystem. Hüthig, 2000

[Die 01] Dietrich, D., Russ, G., Tamarit, C., Koller, G., Ponweiser, W., Vinczen, M.: Modellierung des technischen Wahrnehmungsbewusstsein für den Bereich Home Automation. e&i.Vol.:11, pp. 545-555, November 2001

[Die 84] Dietrich, D.: Die Natur als Vorbild. Elektronik 19, pp. 81-86, 1984

[Die 98] Dietrich, D., D. Loy and H.-J. Schweinzer LON-Technologie: Verteilte Systeme in der Anwendung. Hüthig, Heidelberg, Germany 1998. English edition Open Control Networks: LonWorks /EIA709 Technology; Kluwer Academic Publisher, 2001

[Die 99] Dietrich, D.: Biologische Systeme – Automatisierung. PNO Generalversammlung, Wien, 1999

[Din 94] Din, A. I.: Structured Query Language (SQL): A Practical Introduction. ISBN 1-85554-357-5, 1994

[Doe 99] Döbrich, U., Noury, P.: ESPRIT Project NOAH – Introduction. In Fieldbus Technology. Systems Integration, Networking, and Engineering. Proceedings of the Fieldbus Conference FeT, Magdeburg, Germany, Springer Verlag, pp. 414-422, 1999

[Duc 96] Duch, W.: On simplifying brain functions. Proceedings of the Second Conference on Neural Networks and their applications, Orle Gniazdo, 30.IV-4.V.1996, pp. 118-124, 1996

[Ear 02] Hurd, C. C.: Electronic Data-Processing Machines. Presented at A.I.Ch.E. Biloxi Meeting, vol. 49, pp. 49-54, 1953

[Eib 94] EIB Journal. October 1994

[Eib 98] The EIB System for Home & Building Electronics. The EIB Handbook Series. Marc Goosens, EIBA, 1998

[Fal 03] Falkner, M.: Symbolisierung für effiziente Verarbeitung von Sensordata. Diploma thesis, Institute of Computer Technology, Vienna University of Technology, 2003

[Fal 95] Fallner, A.: Der Körper des Menschen, Einführung in Bau und Funktion, 12. Auflage, Thieme, 1995

[Fet 01] Proceedings form the 4th International Federation of Automatic Control

IFAC Conference, Nancy, France, ISBN 0-08-044003 7, 2001

[Fin 00] Finesso , L.: The Complexity of Hidden Markov Models. ERCIM News No.40, January 2000

[Fis 02] Fischer, P.: Analyse und Bewertung von Kommunikationssystemen in der Gebaudeautomation. Dissertation, TU Wien, 2002

[Fod 87] Fodor, J. A.: Psychosematics: The Problem of Meaning in the Philosophy of Mind. Cambridge MIT Press, 1987

[Fod 90] Fodor, J. A.: A theory of Content and Other Essays, Cambridge MIT Press, 1990

[Fol 86] Foley, J.S., Weon-Yoon, Y.: The current status of MAP. Proceedings of the ACM SIGCOMM conference on Communications architectures & protocol, Stowe, Vermont, United States, 1986

[Fra 97] Franklin S.: Global Workspace Agent. Journal of Consciousness Studies. 4, pp. 322-334, 1997

[Fra 99] Franklin S., Graesser A.: A Software Agent Model of Consciousness. Consciousness and Cognition 8, pp. 285–305, 1999

[Fre 91] Freud, S.: On Aphasia 1891. London and New York, 1953

[Gen 02] Genetics and human behaviour - the ethical concept. Nuffield council on bioethics, ISBN 1 904384 03 X, 2002

[Ger 97] Schckhuber, G., McCarthy, O.: Distributed fieldbus and control netwrok systems. Computer & Control Engineering Journal, Feb. 1997

[Ghg 97] Gürtler, G.H.: Fieldbus Standardisation, the European Approach and Experiences. Feldbussytem in Forschung, Entwicklung und Anwendung, Springer, pp. 2-11, 1997

[Gil 97] Giloi, W. K.: Konrad Zuse Plankalkül: The First High-Level ‘non von Newmann’ Programming Language. IEEE Annals of the History of Computing, Vol. 19, No. 2, 1997

[Gir 95] McGirt, F., Wilkerson, J. F.: An Object Oriented Software Bus. Invited paper, Ninth Conference on Real-Time Computer Applications in Nuclear, Particle and Plasma Physics, Michigan State University, East Lansing, MI, May 1995

[Gla 00] Le Gland, F.: Statistics of Hidden Markov Models. ERCIM News No.40, January 2000

[Gol 01] Goldberg, K., Chen, B.: Collaborative Control of Robot Motion: Robustness to Error. IEEE/RSJ International Conference on Robots and Systems, October 2001

[Gol 77] Goldstine, H. H.: A Brief History of the Computer. Proceedings of the American Philosophical Society, vol. 121, no. 5, pp. 339-345, 1977

[Gol 95] Goldstein, E.B.: Wahrnehmungspsychologie, Spektrum Akademischer Verlag Heidelberg, 1995

[Guy 76] Guyton, A. C.: Text Book of Medical Physiology, W. B. Saunders Company, 1976

[Haa 97] Haakenstad, L. H.: How to Specify BACnet-Based Systems. Alerton Technologies, Inc. 1997

[Har 90] Harnad, S.: The Symbol Grounding Problem. Physica D, vol.42, pp. 335-346, 1990

[Hei 98] Heinze, H.J. Bewusstsein und Gehirn aus Gene, Neurone Qubits & Co, Gesellschaft deutscher Naturforscher und Ärzte. Hrsg. Detlev Ganten, Berlin, 1998

[Hes 99] ISO/IEC JTC 1 SC 25 WG 1 N 844A: Interconnection of Information Technology Equipment Home Electronic System. Berlin, August 1999

[Heu 97] Heuer, A., Saake, G.: Datenbanken: Konzepte und Sprachen. International Thomson Publishing, 1997

[Hob 51] Hobbes, T.: Leviathan. Penguin Books. Reprinted in 1986

[Hoy 03] Hoyle, M. A.: Computers: From the Past to the Present. Abacus: A Primitive Calculator, 2003

[Hsu 85] Hsu, A. T., Marshall, A. G., Ricca, T. L.: Anal. Chem. Acta 178, 27, 1985

[Hub 02] Huber, R., Nowak, C., Spatzek, B., Schreiber, D.: Adaptive Aperture Control for Image Acquisition. IEEE Workshop on Applications of Computer Vision, Orland, December 2002

[Hub 03] Huber, R., Nowak, C., Spatzek, B., Schreiber, D.: Adaptive Aperture Control for Image Enhancement. IEEE CAMP 2003 International Workshop on Computer Architecture for Machine Perception. New Orleans, USA, May 2003

[Hub 03a] Huber, R., Biber, J., Nowak, C., Spatzek, B.: Recognition of Obstacles on Structured 3D Background. 3rd International Conference on Computer Vision, Graz, Austria, 2003

[Hur 53] Hurd, C. C.: Electronic Data-Processing Machines. Presented at A.I.Ch.E. Biloxi Meeting, vol. 49, pp. 49-54, 1953

[Ica 02] Educational Encyclopedia Icarito. Characteristics of II Revolution. 2002

[Iee 90] Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY, 1990

[Ilg 00a] Ilg, W.; Albiez, J.; Berns, K.; Dillmann, R.: Learning of Posture Control Mechanisms for the Four-legged Walking Machine BISAM; Proceedings of the ICARV 2000, Singapore, Malaysia, 2000

[Ilg 00b] Ilg, W.; Albiez, J.; Dillmann, R.: Adaptive Posture Control of a Four-legged Walking Machine using some Principles of Mammalian Locomotion; Proceedings of the International Symposium on Adaptive Motion of Animals and Machines; Montreal, Canada, 2000

[Int 01] Intel Corporation: Open Source Computer Vision Library - Reference Manual. Order Number 123456-001, 2001

[Isd 00] Internetworking Technologies Handbook. Chapter 12 Integrated Services Digital Networks (ISDN), Cisco Press. ISBN-1-58705-001-3, 2000

[Iwa 02] Iwanitz, F., Lange, J.: OPC book: 'OPC - Fundamentals, Implementation and Application'. New revised and extended edition, Hüthig Fachverlag ISBN 3-7785-2883-1, 2002

[Jac 89] Jacoby LL, Whitehouse K.: An illusion of memory: false recognition influenced by unconscious perception. J. Exp. Psychol.: Gen. 118:126-35, 1989

[Jac 97] Jaczynski, M.: A framework for the management of past experiences with time-extended situations. 6th ACM Conference on information and Knowledge Management (CIKM'97),Las Vegas, Nov. 1997

[Joh 83] Johnson-Laird, P. N.: Mental models: Towards a cognitive science of language, inference, and consciousness. Cambridge, MA: Harvard University Press., 1983

[Jov 95] Jovanov, E., Rakovic, D., Radivojevic, V., Kusic, D.: Band power envelope analysis – a new method in quantitative EEG. Proc. 17 Ann. Conf. IEEE EMBS 1995

[Jov 97] Jovanov, E.: A Model of Consciousness: an Engineering Approach. Brain and Consciousness, Proceedings of the ECPD Symposium (Belgrade, Yugoslavia), pp. 291-295, September 1997

[Jul 84] Julesz, B.: Towards an axiomatic theory of preattentive vision. Dynamic Aspects of Neocortical Function. GM Edelman, WE Gall, WM Cowan (eds), pp. 585-612, NY, 1984

[Kah 86] Kahneman, D., & Miller, D.T. :Norm Theory: Comparing reality to its alternatives. Psychological Review, 93, 136-153, 1986

[Kah 93] Kahn, M.: I2C Specification information. Programming the I2C Interface. When intelligent devices need to communicate. Philips Semiconductors, 1993

[Kan 00] Kandel E. R., Schwarz J. H., Jessell T. M.: Principles of Neural Science. McGraw-Hill, 2000

[Kan 00a] Kandel E. R., Schwarz J. H., Jessell T. M.: Principles of Neural Science. Part I The Neurobiology of Behaviour. McGraw-Hill, 2000

[Kan 00d] Kandel E. R., Schwarz J. H., Jessell T. M.: Principles of Neural Science. Part IV The Neural Basis of Cognition. McGraw-Hill, 2000

[Kan 00e] Kandel E. R., Schwarz J. H., Jessell T. M.: Principles of Neural Science. Part V Perception. McGraw-Hill, 2000

[Kan 00f] Kandel, E., R., Schwarz, J. H., Jessell T. M.: Principles of Neural Science. Part VI Movement, 2000

[Kan 00g] Kandel E. R., Schwarz J. H., Jessell T. M.: Principles of Neural Science. Part VII Arousal, Emotion, and Behavioural Homeostasis. McGraw-Hill, 2000

[Kan 00i] Kandel E. R., Schwarz J. H., Jessell T. M.: Principles of Neural Science. Part IX Language, Though, Mood, and Learning, and Memory, McGraw-Hill, 2000

[Ker 00] Kandel, E. R.: Banquet Speech. Speech at the Nobel Banquet, December 10, 2000.

[Kit 00] Kitamura, T., Tahara, T., and Asami K. I.: 'How can a robot have consciousness?' Advanced Robotics, Vol. 14, No. 4, pp. 263-75. 2000

[Kno 00] Knowledge Harvesting Glossary Version 4.8. Knowledge Harvesting Workshop, Knowledge Harvesting Inc. 2000

[Koc 94] Kockskämpe, S., Neumann, B., Schick, M.: Extending Process Monitoring by Event Recognition. Intelligent Systems Engineering, Conference Publication No 395, Sep.1994

[Kon 89] Koton, P.: Using Experience in learning and Problem Solving. Massachusetts Institute of Technology. PhD. dissertation, Oct 1989

[Kot 88] Koton, P: Reasoning about Evidence in Causal Explanations. AAAI, pp. 256-263, 1988

[Kra 02] Kranz , H. R., Gisler, O.: Standardization & IT Technology will shape the BACS Industry. Siemens Building Technologies Ltd, Jan. 2002

[Kun 00] Kunsch, K.: Der Mensch in Zahlen, Datensammlung in Tabellen, 2. Auflage, Spektrum Akademischer Verlag Heidelberg, 2000

[Kur 96] Kurzweil, R.: The Age of Intelligent Machines 'Passing the Turing Test'. Kurzweil Technologies 1996

[Kur 97] Kurata, D.: Learn the Fundamentals of OOP. Intermediate, Programming with Class, Visual Basic Programmer's Journal, pp.81-83, June 1997

[Lan 02] Langels, H. J., Goossens, M.: Internet Protocol extends to EIB. 2002

[Leo 00] Leon-Garcia, A., Widjaja, I.: Communication Networks, Fundamental Concepts and Key Architectures. McGraw Hill Series in Computer Science, 2000

[Lin 96] Jacobson, L.: Cognition, perception, and experience in the virtual environment: do you see what I see? International Conference on Computer Graphics and Interactive Techniques. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996

[Lli 91] Llinas, R. R., Pare, D.: On Dreaming and Wakefulness. Neuroscience 44, 521-535, 1991

[Lli 92] Llinas, R., Ribary, U.: Induced Rhythms in the Brain. Eds Basar, E. & Bullock, T.H., pp. 147-154, Birkhäuser, Boston, 1992

[Lon 02] LON Nutzer Organisation e.V. The LonMark interoperability Association: LonWorks Installation Handbook.VDE Verlag GMBH. ISBN 3 8007 2687 4, 2002

[Lop 93] Lopez, B, Plaza, E.: Case-based planning for medical diagnosis. Methodologies for Intelligent Systems: 7th International Symposium, ISMIS '93, pp. 96-105, 1993

[Low 00] Low, B., Ostrom, E., Simon, C., Wilson, J.: Redundancy and Diversity in Governing and Managing common-pool Resources.8th Biennal conference of the International Association for the Study of common Property (IASCP), Indiana University, Bloomington, Indiana, 2000

[Loy 01] Dietrich, D., D. Loy and H.-J. Schweinzer (1998). LON-Technologie: Verteilte Systeme in der Anwendung. Hüthig, Heidelberg, Germany. English edition: Open Control Networks. LonWorks(R)/EIA709 Technology. Kluwer Academic Publisher, 2001

[Lyd 99] Lydic, R., Baghdoyan, H. A.: Handbook of Behavioural State Control. Cellular and Molecular Mechanisms. CRC Press LLC. ISBN 0 8593 3151 X, 1999

[Lyo 91] Lyons, D., Hendriks, A., Mehta, S.: Achieving robustness by casting planning as adaptation of a reactive system. In IEEE conference on Robotics and Automation, 1991

[Mah 02] Mahlknecht, S.: Virtual Wired Control Networks: A Wireless Approach with Bluetooth, Proc. Of IEEE Africon, pp. 269-272 Vol1, September 2002

[Mar 82] Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. New York: W. H. Freeman and Company, 1982

[Mar 99] Marino, P., Dominguez, A., Poza F., Nogueira, J.: Formal Description Software for WorldFIP Industrial Fieldbus; in Fieldbus Technology – System Integration, Networking, and Engineering. Dietrich, D., P. Neumann and H. Schweinzer. Springer Wien/New York, 1999

[Med 98] Medler, D. A.: A Brief History of Connectionism. Neural Computing Surveys 1, pp. 61-101, 1998

[Mer 02] Merten, Systemlösungen für die Elktrotechnik. Instabus EIB, Catalogue 2002

[Ned 02] History of Energy. The NEED Project. Primary Energy Infobook, 2002

[Öno 01] ÖNORM EN 13306: Begriffe der Instandhaltung: Normengruppe A und M: Ausgabe 2001

[Ott 02] Ottensen, L.E.: Evolutionary Psychology: an emerging integrative perspective within the science and practice of psychology. The Human Nature Review, vol.2, pp. 17-61, ISSN 1476-1084, 2002

[Pal 01] Palensky, P.: Distributed Reactive Energy Management. Dissertation, Institute of Computer Technology, Vienna University of Technology, 2001

[Pen 96] Eniac. February 14, 1946: The birth of the information age. PENNPRINTOUT, Vol. 12:4, 1996

[Per 02] Perry, B. D.: Brain Structure and Function. Basics of Organization I. Interdisciplinary Education Series, 2002

[Per 99] Perrin, B.: The Art and Science of RS-485. Circuit Cellar Online, 1999

[Pol 89] Pollock, J. P.: OSCAR: A General Theory of Rationality, JETAI Journal of Experimental and Theoretical Artificial Intelligence, 1989

[Pos 90] Posner, M. I., Petersen, S. E.: The attention system of the human brain. Annual Review of Neuroscience, 13, pp. 25-42,1990

[Pro 95] Proctor, R.W., Dutta, A.: Skill acquisition and human performance. Advanced Psychology Text Series, Thousand Oaks, 1995

[Qua 97] Quade, J.: Ein einheitliches Programmierinterface für Feldbusapplikation. Feldbustechnik in Forschung, Entwicklung und Anwendung, Springer, pp. 12-18, 1997

[Rab 00] Rabaey, J., M. Ammer, M. J., da Silva, J. L. J., Patel, D., and Roundy, S.: Picoradio supports ad hoc ultra-low power wireless networking. IEEE Computer Magazine, pp. 42-48, July 2000

[Rat 01] Ratko, R.: Integrated Management Middleware for Building-and Home-Automation Systems. Dissertation. Institut of Computer Technology, Vienna University of Technology, 2001

[Rau 99] Rauscher, T.: Object Oriented user Layer for LonSpeed. Diploma Thesis. Institut of Computer Technology, Vienna University of Technology, 1999

[Rie 96] Riera Guasp, M., Roger Folch, J., Roldan Porta, C.: Technologia Electrica. Introduccion a las instalaciones electricas. Departamento de Ingenieria Electrica, Valencia University of Technology, Servicio de Publicaciones SPUPV-96.015, 1996

[Roh 94] Rohen, J.W.:Funktionelle Anatomie des Nervensystems. Schattauer. Stuttgart-New York, 1994

[Ros 86] Rosenthal, D. M.: Two Concepts of Consciousness. Philosophical Studies 49, pp. 329-359, 1986

[Roy 95] Royer, V.: Hierarchical Correspondence between Physical Situations and Action Models. The 1995 International Joint Conference on AI, Quebec, Canada, 1995

[Rus 01] Russ, G.: Interoperabilität standardisierter LON-Komponenten im Home-Bereich. Diploma Thesis, Institute of Computer Technology, TU Vienna, 2001

[Rus 02] Russ G., Dietrich D., Tamarit C.: Situation Dependent Behavior in Building Automation. Workshop Proceedings of EurAsia-ICT 2002 Advances in Information and Communication Technology, Shiraz, Iran, pp. 319-323, 2002

[Rus 03] Russ, G.: Situation-dependent behaviour in building automation. Dissertation, Institute of Computer Technology, TU Vienna, 2003

[Sac 87] Sacks, O. The man who mistook his wife for a hat. Summit Books, Simon & Schuster, New York, 1987

[Sad 96] Sadiku, M.N.O.; Obiozor, C.N.: Evolution of computer systems. Frontiers in Education Conference, 1996. FIE '96. 26th Annual Conference, Proceedings of, Vol. 3, pp. 1472-1474, 1996

[Sag 01] Sagiv, N., Bentin S.: Structural encoding of human and schematic faces: Holistic and part-based processes. Journal of Cognitive Neuroscience 13(7), pp. 937-951, 2001.

[Sbs 98] SBS Avionics Technologies: An Interpretation of the MIL-STD- 1553B. Doc: 1553 Interpretation, version 2.2, May 1998

[Sch 80] Schiebel, A.B.: Anatomical and physiological substrates of arousal: view from the bridge. In: The Reticular Formation Revisited, ed. Hobson, J.A. and Brazier, M.A.B. New York, Raven Press, 1980

[Sch 88] Schweir, U., Saupe, D.: Funktions- und Konstruktionsprinzipien der Programmgesteuerten mechanischen Rechenmaschine Z1. Arbeitspapiere der GMD, Gesellschaft für Mathematik und Datenverarbeitung mbh., 1988

[Sch 97] Schickhuber, G., McCarthy, O.: Distributed fieldbus and control network systems. Computing and Control Engineering Journal, Feb 1997

[Shi 98] Shimek, R.: Without a Backbone. Sponges – pump-filter modules through space and time. Aquarium Frontiers and Fancy Publications Inc., 1998

[Sie 02] Siemens, Elektroinstallation von A bis Z, Gamma Gebäudetechnik. Katalog 2002

[Sim 99] Simon, S., Kestler, H. A., Baune, A., Schwenker, F., Palm, G.: Object classification with simple visual attention and a hierarchical neural network for sub-symbolic-symbolic coupling. 0-7803-5806-6/99, pp. 244-249, IEEE 1999

[Ski 01] Skiczuk, P.: Porject IST 1999-10622 InHoMNet: Final Public Report

2001

[Std 96] Building Management Products and Systems for HVAC Applications. Draft of European Prestandard, CEN/TC247, July 1996

[Ste 00] Steininger,A.: The Testing of Fault Tolerant Computers Habilitation. Institut für technische Informatik, Vienna University of Technology, 2000

[Sun 01] Sun, J. Z.: Mobile ad hoc networking: an essential technology for pervasive computing Proceedings. ICII, International Conferences on Info-tech and Info-net, Beijing, vol.3, pp. 316 –321, 2001

[Suw 01] Suwannik, W., Chongstitvatana, P.: Improving the robustness of evolved robot arm control programs with multiple configurations, 2nd Asian Symposium on Industrial Automation and Robotics, Bangkok, Thailand, pp.87-90, 2001

[Sve 02] Svea Building Control Systems. Der LON Katalog 2002

[Tak 98] Takeuchi I. Furuhashi T.: A Proposal of Architecture for Intelligent Systems with Manipulative Grounded Symbols. Second International Conference on Knowledge-Based Intelligent Systems, Adelaide, Australia, April 1998

[Tam 01] Tamarit, C., Dietrich, D., Dimond, K., Russ, G.: A definition and a model of a Perceptive Awareness System (PAS). FeT'2001 Proceedings, 4^{th} IFAC International Conference on Fieldbus Systems and their Applications, Nancy, France, November 2001

[Tam 01a] Tamarit C: Automatizacion de la Vivienda a traves de la Technologia LonWorks. Diploma thesis, ETSII, TU Valencia, 2001

[Tam 02] Tamarit C., Russ, G.: Unification of Perception Sources for Perceptive Awareness Automatic Systems. Africon'02 Proceedings, George, South Africa, October 2002

[Tau 63] Taub, A. H.: John von Neumann. Collected Works. Vol. 5, Design of computers, theory of automata and numerical analysis, General and logical theory of automata. Pergamon Press, pp. 215-235, London, 1963

[Tlc 99] Thomesse, J.P., Leon Chavez, M.: Main Paradigms as a Basis for Current Fieldbus Concepts. Fieldbus Technology, Springer, pp.2-15, 1999

[Tre 77] Treisman, A., Sykes, M., Gelade, G.: Selective attention stimulus integration. S Dornie (ed). Attention and Performance VI, pp. 333-361. Hilldale, NJ: Lawrence Erlbaum, 1977

[Tre 86] Treisman, A.: Features and Objects in Visual Processing. Sci Amm 255(5), pp. 114-125, 1986

[Veg 98] Vega Martinez, A., Cerezo Sanchez, J. M.: Automatización de Procesos Industriales. Departamento de Ingeniería Electrónica y Automática. Universidad de Las Palmas de Gran Canaria. 1998

[Vin 01] Vincze, M., Ayromlou, M., Ponweiser, W., Zillich, M.: Edge Projected

Integration of Image and Model Cues for Robust Model-Based Object Tracking. International Journal of Robotics Research 20(7), pp. 533-552, 2001

[Vox 02] VOX Dictionary, 2002

[Wei 97] Weinberg, I.R., van Wyk, C.P.: 'An integrative Neurological Model of Consciousness: the case for quantum-determinism'. Transformation Strategies, Articles, 1997

[Wfc 02] Proceedings of the IEEE International Workshop on Factory Communication Systems, Mälardalen University, Sweden, ISBN 0-7803-7586-6, 2002

[Whi 89] Whitaker, J.: Milestones in the Evolution of Technology. Broadcast Engineering magazine, 1989

[Wit 85] Witherspoon, D., Allan, L. G.: The effects of a prior presentation on temporal judgments in a perceptual identification task. Memory and Cognition, 13, 103-111, 1985

[X25 00] Internetworking Technologies Handbook. Chapter 17 X.25, Cisco Press. ISBN-1-58705-001-3, 2000

[Xie 01] Xie, X., Hahnloser, R., Seung, S.: Learning Winner-Take-All competition between Groups of Neurons in Lateral Inhibitory. Networks. Proceedings NIPS'2000, vol. 13, MIT Press, 2001

[Zus 99] Zuse, H.: John von Neumann's Computer Concepts versus Konrad Zuse's Ideas and the Machines Z1 and Z3. May 1999

Web Pages

[Adv 03] Advanced Bionics Corporation

http://www.advancedbionics.com/aboutus/mission.html

[Aeg 02] AEG Allgemeine Elektricitäts-Gesellschaft. History

http://toasters.de/history/hist_c01.html

[Alp 03] Alpha M.O.S.

http://www.alpha-mos.com

[Apl 03] Apple – FireWire

http://www.apple.com/firewire/

[Asp 00] LearnAsp.com - ASP.net and ASP Classic Tutorials by Charles Carroll http://www.activeserverpages.com

[Bac 00] BACnet Website

http://www.bacnet.org

[Cab 03a] CABA Continental Automated Buildings Association. BACnet

http://www.caba.org/standard/bacnet.html

[Cab 03b] CABA Continental Automated Buildings Association

http://www.caba.org

[Can 02] Canadian Psychological Association: Psychology works for Cognitive Disorders and Dementia, 2002

http://www.cpa.ca/factsheets/cognitive.htm

[Cet 03] Cetelab

www.cetelab.se

[Cir 03] Cyrano Sciences Inc.

http://cyranosciences.com

[Cob 03] COBA

http://www.coba-group.com

[Coc 03] The Chicago Otology Group - Cochlear Corporation

http://www.chicagoear.com/im-n24.html

[Con 01] Consulting-Specifying Engineer: A Brief History of Building-

Automation Interoperability. 2001

www.csemag.com/esec/Article_63135.htm

[Cou 95] Coutinho, J, Martin , S., Samata , G., Tapley, S., Wilkin, D.: Department of Electrical and Computer Engineering: The History of Fieldbus.

http://dali.ece.curtin.edu.au/~clive/Fieldbus/history.htm

[Dom 02] Dominguez, A.: Energías. Breve historia de la energía. Naturaleza Educativa

http://www.iespana.es/natureduca/energ_historia.htm

[Dra 03] Dragon. Century Eloquence

http://www.voicerecognition.com

[Ech 00] Echelon Corporation

http://www.echelon.com

[Ech 03] Echelon Corporation – Neuron Chip

http://www.echelon.com/products/lonworks/neuron.htm

[Eib 00] EIBA: European Installation Bus Association

http://www.eiba.com

[Ele 03] Electronic Sensor Technology - Portable Vapor detectors

http://www.estcal.com

[Fir 03] FireWire 800 Next-generation high-speed serial interface. Apple Computer 2003

www.apple.com/firewire/pdf/FireWire_Tech_Brief-a.pdf

[Geo 03] Georgia Institute of Technology. College of Computing. Atlanta

http://www.cc.gatech.edu/classes/cs6751_97_winter/Topics/human-cap/memory.html

[Hel 03] Helio Building Control, Philips

http://www.controls4lighting.com/lightingcontrol/information/applications/helio/helio.asp

[Hof 03] Hoffmann, T.: Summary of Popular Protocol Standards, Johnson Controls, Controls Group, 2003

http://www.jci.com/metasys/articles/article4.htm

[Iec 03] International Engineering Consortium (IEC)

http://www.iec.org

[Ins 03] Instituto de Fisiología Celular, UNAM: A Brief Introduction to the Brain.

http://ifcsun1.ifisiol.unam.mx/Brain/segunda.htm

[Int 00] Intel

http:/www.intel.com

[Int 03a] Intel Face Recognition and OpenCv

http:/www.intel.com/research/mrl/research/media-visual.htm

[Iso 00] ISO - International Organization for Standardization Website

http://www.iso.org

[Koe 01] Koehler, P.: Fire Wire versus USB. 2001

http://www.chips.navy.mil/archives/01_winter/firewire.htm

[Kon 03] Konnex Association

http://www.konnex.org

[Lma 02] LonMark concerning the term Task Groups

http://www.lonmark.org/about/taskgrp.htm

[Mab 00] Marquez, M., Brusco, L.I. Funciones cognitivas

http://www.bmd.com.ar/Consulta_Neuro00.htm

[Mer 03] Merloni

http://www.merloni.it

[Mil 03] MIL-RAM Technology, Inc.

http://www.mil-ram.com

[Mod 00] Modesti, M. R.: Controladores de logica programable (PLC). 15 Jun. 2000

http://www.frc.utn.edu.ar/institucional/industrial/catedras/sistemasinteligentes/UT3/UNI5200.pdf

[Mor 02] Morgan, R. A.: USB 2.0 versus FireWire. Bare Feats. 2002

http://www.barefeats.com/fire26.html

[New 03] Newron System

www.newron-system.com

[New 97] Envenergy BACnet white papers

http://ww.envenergy.com/pdf/bacnet_white_paper2

[Ope 03] OpenCv source. Intel

http://www.intel.com/research/mrl/research/opencv

[Out 02] Outsourcing Integral G.A. Process Automation

www.oiga.cl/plc.pdf

[Phi 00] The Internet Encyclopedia of Philosophy. Aristotle

http://www.utm.edu/research/iep/a/aristotl.htm

[Roj 99] Rojas, R.: Z1, Z2, Z3 and Z4. Konrad Zuse, Internet Archive, 1999

http://www.zib.de/zuse/English_Version

[Sca 03] VoiceRequest

http://www.scansoft.com/voicerequest

[Scr 03] Screenfridge from Electrolux

http://www.electrolux.com/screenfridge/start.htm

[Sel 00] Sellers, D.: USB 2.0 versus FireWire. MacCentral Online, 2000

http://maccentral.macworld.com/news/0004/26.usb.shtml

[Sen 03] Sensory

http://www.sensoryinc.com

[Smi 99] Smith, A: The olfactory process and its effect on human behavior. Neurobiology and Behavior, Second Web Reports, 1999

http://serendip.brynmawr.edu/bb/neuro/neuro99/web2/Smith.html

[Spe 03] Microsoft Speech Technologies

http://www.microsoft.com/speech/evaluation/techover

[Sta 98] Stallings, W.: The Origins of OSI

WilliamStallings.com/Extras/OSI.html

[Sve 03] Svea, Building Conrol Systems

www.svea.de

[Sys 03] Sysmik

www.sysmik.de

[Thi 03] Human behaviour. Cognitive processes. © ThinkQuest Team 26618

http://library.thinkquest.org/26618/en-5.2.2=sensory%20memory.htm?tqskip1=1&tqtime=0718

[Tra 02] La Revolucion de los Transportes. Revolución Industrial e Industrializacion.

http://www.fortunecity.es/imaginapoder/humanidades/587/industrializacion.htm

[Tuf 02] Tufts: Development of Intelligence. Trustees of Tufts College

http://www.tufts.edu/as/wright_center/cosmic_evolution/docs/text/text_cult_6.html

CURRICULUM VITAE

Clara Tamarit Fuertes
February 18th, 1976
Valencia, Spain

Education

1994-1998	Mechanical Engineering (1st - 7th Semester) UPV - Valencia University of Technology
1998-1999	Electrical Engineering (7th - 8th Semester) UPV - Valencia University of Technology
1999-2000	Electrical Engineering (9 th - 10 th Semester) Vienna University of Technology, ERASMUS Program
06.2000-12.2000	Diploma Thesis: '*Automation of a House using LonWorks Technology'* Institute of Computer Technology – Vienna UT
21.01.2001	Graduation: Master of Science (Dipl.-Ing.) UPV - Valencia University of Technology

Training

08.1997-09.1997	Electrical signal treatment programming using Hewlett Packard visual engineer environment (HP VEE). HTU Dresden, Germany
07.1999-09.1999	Computational simulation of atomic orbits Helsinki University of Technology, Finland
06.2000-12.2000	Local operating network design and installation using LonWorks technology Institute of Computer Technology – Vienna UT, Austria

Work Experience

03.2001-03.2003	Assistant at the Institute of Computer Technology – Vienna UT. Project SmartKitchen: Improve the behaviour of automation systems, increased comfort, greater security and safety, energy efficiency and support of communication facilities