November 2001 ECN-I--01-005

Computerised Intelligence in Small Scale Home Networks and Beyond

Opportunities for energy efficiency

C.J. Warmer
I.G. Kamphuis

Acknowledgement

The work described in this report was carried out in the ENGINE-program framework of ECN for project-number 76305. The Dutch Ministry of Economic affairs sponsors the framework.

Summary and Conclusions

In this report the possibility of realising smart applications using modern information and communication technologies on the residential scale is investigated from a number of viewpoints.

The first viewpoint, elaborated in chapter 2, is a technological one emphasising the electronic hardware. Various possible processor and network architectures are discussed and connection schemes and protocols are presented to establish a platform for intelligent software applications in residential buildings.

The second viewpoint concerns the software architecture. In chapter 3, using three different types of applications, requirements are defined and functional roles are attributed to individual sub-processes in the software. Apart from functional aspects, such as what computing and memory capacity is necessary for which node in the network, the architectures are also discussed in terms of security and reliability of network traffic. Current developments in standardising this software architecture, especially using residential gateways, are also discussed.

The largest energy saving potential of using intelligent technology in buildings is in the control strategy of the combined air ventilation and heating system. The COMFY system [PAAM2000], which was designed as a software agent application, was developed with this potential in mind. In chapter 4 to 6, this theme is worked out further in the COMFY-2 system. The air and ventilation strategy for a building in this approach is driven by dynamical optimisation of Fanger's composite comfort index, which is not only dependent on air temperature, but also contains an assessment of the radiation temperature and the relative humidity from the environment. Parameters specific to the perceived comfort for individual persons in a building are contained in the index and hence in the optimisation as well. A further feature of the COMFY-2 system is its ability to include varying real-time energy prices in the optimisation and the possibility to model storage of energy in buffers. In chapter 5 a number of simulations of HVAC control strategies are described for different ambient conditions and energy pricing scenarios. The scenarios in chapter 6 take the Dutch normalised "NOVEM Doorzonwoning" as a typical example for estimating the energy saving potential. Strategy calculations are also presented for comfort in utility office buildings.

In chapter 7, the mathematical model, using computationally intensive numerical algorithms, and the inner workings of the different parts of the COMFY-2 system are described.

It can be concluded, that, from a hardware point of view individual components and nodes in the network have reached a mature implementation stage. A consistent interconnection architecture, based on standard Internet technology, is currently evolving. Current pricing levels of the technology at the moment hinder implementation on a wider scale.

From a software-architectural point of view current standardisation efforts are leading to a transparent, portable platform for a number of application types. With the advent of an increasing number of always-on, broadband Internet connections to homes, computational resource demanding energy management applications will be more easy to realise using distributed computing capacity.

From the comfort management strategy calculations a consistent picture of a clear, lifestyle dependent, cost and energy usage reduction benefit of 10-30 percent can be derived. A similar benefit can be found for utility buildings. Although the COMFY model is based on electrical heating – very unusual in the Netherlands – we believe that these conclusions also hold for gas-based heating.

USED ABBREVIATIONS

ADSL	Asymmetric Digital Subscriber Line
ASIC	Application Specific Integrated Circuit
ASP	Application Service Provider
ASP	Active Server Page
COM	Common Object Model
COMFY	COmfort Management For You
CORBA	Common Object Request Broker Architecture
DA	Distribution Automation
DCMS	Distributed Control and Measurement system
DCS	Distributed Control System
DLMS	Device Language Message Specification
DOM	Document Object Model
DSM	Demand Side Management
DU	Distributed Utilities
HVAC	Heating Ventilation and Air Conditioning
EHS	European Home Systems
IEC	International Electrotechnical commission
EPC	Energy Prestatie Coefficient
IP	Internet Protocol
LON	Local Operations Network
OSGi	Open Services Gateway initiative
PALAS	Powerline as an Alternative Local AccesS
PLC/T	Power Line Communication/Transmission
RG	Residential Gateway
RMI	Remote Method Invocation
SCP	Simple Control Protocol
SCP	Sociaal Cultureel Planbureau
SOHO	Small Offices Home Offices
UCA	Utility Communication Architecture
UPNP	Universal plug-and-play
USB	Universal Serial Bus
XML	eXtended Markup Language

CONTENTS

1. INTRODUCTION

In this document the research questions put with respect to the possible application of information and communication technology for improving energy efficiency in buildings are discussed. A major application area, in which ICT can contribute to this goal, is in comfort control in residential and utility buildings. Comfort control has to do with heating/cooling and ventilation; these are known to yield the largest proportion of the energy consumption, especially in existing dwellings.
We developed a prototype comfort management system, COMFY, which optimizes room comfort over a period of time, e.g. one day, in a pre-emptive and intelligent way. In this optimization, besides standard heating (or cooling) COMFY makes use of natural resources like solar irradiation and natural ventilation in order to reach optimal energy and cost efficiency. Through the extended usage of ICT and especially the way Internet technology is immersed in society, applications of intelligent managing of energy, such as COMFY, are coming within reach more and more.

In this report, in chapter 2 a compilation is given of the current status of relevant information and communication technology developments. Chapter 2 is derived from experiences gained from the EU 5th framework PALAS-project. In Chapter 3 the software and hardware architecture of a number of typical sample applications is discussed. In chapter 4 a prototype application and simulation system is discussed: the COMFY-system. Comfort is managed as a multi-parameter function optimised with well-known numerical algorithms on a micro-scale. Chapter 5 and 6 contain a number of scenario simulations in order to study the effect of COMFY control and to quantify potential energy savings. At the end of chapter 6 the residential concept is extended to utility buildings. In the last chapter the complete user manual and technical information for the COMFY-system is contained.

2. ICT CONTEXT FOR ENERGY MANAGEMENT APPLICATIONS

2.1.1.1 Introduction

Setting up a residential application for energy management requires an inventory of all aspects regarding the application. Eventually, all these aspects are reflected in functional requirements for software systems. In the design phase of the development of these software systems, as a first step, processes and business rules, behaviour and control aspects and data structures are modelled in an abstract, implementation independent way. Before implementation a fourth view is added to these three aspects: the architectural view. In the architectural view, hardware and software components and their interfaces are identified. The components and objects identified in this process are mapped onto existing, standardised hard- and software components. The application then consists of a number of processes using these components glued together to the application logic.

In this chapter, the software architecture is discussed in terms of attributes of components and the interfaces between components relevant to ICT-enabled energy management. The software architecture is the bridge between subsequent business-models and processes, the hardware components and the communication channels between the hardware components. Currently, in architectural terms, the emphasis is shifting from PC-centred computing to appliance-centred computing with software being the middle-ware gluing applications together.

In traditional software development methods, as a first step in building a system, a specification, abstracted from the implementation, only looking at functionality of an application is made. In domestic energy management systems, a number of technological developments and existing business models form essential boundary conditions and prerequisites for the architectural work. Therefore, in this document a middle-out approach is followed: a top-down approach from the functionality point-of-view is combined with a bottom-up approach from the technical aspects of energy-related monitoring and control services.

In the architectural discussion an energy management, comfort management application with integrated heating and ventilation control is discussed. Furthermore, a service delivery framework is discussed. From the point-of-view of software technology, designing applications for telemetry and telecontrol pose no insurmountable problems. However, the track record of mass-introduction of successful applications is small. The distributed nature of these applications in terms of security, guaranteed (non) accessibility, safety and (un)interruptability pose software architectural problems. Furthermore, merely designing and implementing one application on a dedicated infrastructure will prove not to be very cost-efficient. There is a large number of stakeholders for in-house and SOHO[1] applications in residential areas. It is more likely that applications with comparable architectures and infrastructures will be offered in service bundles than as mere stand-alone applications. To deliver properly aggregated service bundles, alliances of application providers have to be formed. In this document the factors will be identified, that affect the introduction of service applications and the optimal aggregation in terms of bandwidth and processor requirements by a service bundle provider.

A large stimulus for successful introduction of large scale distributed applications comes from hardware and network innovations and from software standardisation, developments now rapidly progressing. In a subsequent section the focus is on the role of these developments on the software architecture as they form the context of applications in terms of devices, nodes and networks each with their respective interfaces. With respect to the software architecture,

[1] SOHO stands for Small Offices Home Offices as a segment where business is provided from either a small office or a residential home.

software standards elements relevant for remote server application-types will be discussed briefly. The operating environment, as reflected in standards for distributed system operation, is also discussed in this subsection. These pertain to JINI on the micro-network level, COM, CORBA, Java-RMI for inter-application interfaces, XML/DOM for interfacing to a distributed database portal and WWW-wide inter-application communication) and ongoing industry co-ordination activities such as the OSGi (Open Systems Gateway initiative). Furthermore, the software architecture requirements definition will be strongly influenced by the information systems context for last-mile access based services. This context includes software already in use at the utility side. In this respect one might think of customer information and relation management systems, billing, distribution automation systems, electricity load balancing apparatus and so on. Future, renewable energy generation systems will be of a small scale. Distributed operation for optimal utility DA/DSM in "real-time" pricing scenarios, then, will pose a large problem, if information flows across small-scale networks is not possible. Proper matching of scale and application scope of processes and optimal utilisation of information content is an essential prerequisite. A short description of the latter aspects concludes this section.

On a more detailed level, last-mile service applications, from a software architectural point of view, require setting up a hierarchical architecture with a number of distribution levels where processes intercommunicate. As the hierarchical tree is descended, software systems on the service provider level have to be interfaced to software systems on ever-smaller scale systems. This is discussed in the subsection dealing with the communication infrastructure options for such types of applications.

Technically speaking, partitioning and dimensioning of components along the hierarchical tree has implications for object serialisation, persistence and replication. Obviously, no generic architectural framework for all applications can be derived. The way to define the abstract system architecture in this document is by following the industry standard UML (Unified Modelling Language) method, elaborating use cases and associated object models and business rules (object constraints) for a number of archetypal key application-types. These types cover clearly distinguishable categories of applications. This is the subject of the section about the software application model.

Assignment of tasks to a suitable hierarchical network level plays an important and discriminating role in possible service development schemes. E.g. adequate partitioning of tasks between a low-intelligence meter at-home and a medium voltage contractor serving a large number of households can play a key role in cost-effective implementation. Also manageability and versioning of the software and hardware modules in fine-grained networks are discussed and further features the mapping of these architectural considerations onto bundling of service types. These subjects are treated in the last section of this chapter. Finally a number of conclusions are stated.

2.2 Hardware and software context and interfaces

In this section a description of software architecture relevant attributes for "energy near" service applications is given and standardised software component attributes forming the context of the technological development of distributed applications are discussed. The scope is taken one segment broader than purely the last mile connection to gain a better insight in the development of applications, that not only have requirements with respect to the last-mile but also to dwellings and buildings as well as to utility company software systems.

2.2.1 Network topology characterisation

A number of functional entities in networks are relevant to PLT-applications.

1. The local network. At this moment, small-scale networks are rapidly emerging [1]. The local network enables local data-exchange between intelligent appliances or computers.
2. The local access node. This node is the portal (i.e a transfer-point) for exchange of selected information between the local network and the outside world. Possible local access nodes are a two-way communication, third generation (intelligent) set-top box, a PC+modem or a dedicated residential gateway.
3. The concentrator node. The concentrator node typically is placed in a transformer station in an urban district. The transformer station enables communication to the fourth level of communication between local access nodes.
4. The utility node. This node is the top-level in the network hierarchy in the scope of the discussion.

With respect to the physical data transfer medium a large number of combinations exist between powerline, RF, cable, fibre optic or a copper telephone line to interconnect individual entities.

2.2.2 Application data transfer characterisation

From the network topology above a number of application types can be discriminated from the point of view of data transfer frequencies and data rates and from control behaviour. In the discussion, a further discriminating point is the extent to which information has to pass all levels in the network topology. Shortly mentioned in the discussion are home automation networks. These sometimes contain devices consisting of networks themselves. These point-to-point communications are concentrated in piconets [2] or micronets, which, in their own, are part of larger networks integrated and accessible as one component. Bluetooth in this respect may be considered as a piconet replacing a USB-cable connection. Piconets together form a functional component/appliance. The distributed application for energy managements may be characterised as a distributely managed, local control and monitoring systems (DCMS). In the network topology these have their control processes and data distributed over the network. Required bandwidth for both directions is low. Real-time constraints are moderate. Software reliability, management and maintenance requirements however are heavy. Applications, standards and tools to produce the applications may be compared to those for large industrial real-time DCSs. Typical required transfer rates are in the order of kB/s. Control network applications place the service delivering companies in the control-loop of critical user processes. In terms of the software architecture this means, that careful consideration has to be given to data and object persistence as function of time, replication mechanisms in case components in the network fail and serialisation mechanisms. This kind of application is most difficult to build, test and manage. Furthermore these applications are very costly, because the expenditure on a per programming statement base for control code is several times larger than for database handling code or output handling code. Other types of applications are the following:

1. Point to point applications (P2P). Although the signals may be sent across the powerline in a broadcast manner, one might imagine the result is a one-to-one connection. These traverse the network topology to a certain level in both directions. One might think of an in-building telephone call, a call within an urban district or a wider area normal telephone call. These types of applications are heavily real-time. Bandwidth requirements in both directions are moderate. Application logic is contained in dedicated small DSP-processors and large telephone exchange systems. Transfer rates are in the order of tens of kB/s. At the moment it is expected, that circuit-switched telephony is going to be replaced by packet-switched telephony. Touring then will become a matter of intelligent IP-socket communication and real-time data transfer.
2. Information exchange applications (IP). These applications are based on the IP-protocol used on the Internet. Data packages are interchanged with low to moderate real-time constraints as information content. Which network components are traversed in the above scheme is arbitrary. Low-bandwidth suffices upstream from the local network; moderate bandwidth is required for the opposite direction from the “content” repository. For distributing data-streams in information networks, the required volume is higher, but the reliability requirements are much lower. Performance also is dependent on connections

outside the network topology. Functional requirements and manageability of IP-applications dictate a mechanism with an emphasis to server or client-side computing. Examples of the first are active server pages (ASPs) and servlets and of the second, applets. Bandwidth requirements for the first category are lower than for the second. The opposite is true for the processor-capability. Distributed systems based on remote-procedure call mechanisms have known difficulties with firewall-protection when programmed in Java.

3. Entertainment/multimedia applications (EMA). These applications require a large bandwidth to the local access node. Large volumes of information are transferred in parallel down the network hierarchy. Upstream volumes are typically very low. Entertainment networks are most demanding in terms of data transfer; the distribution of data however is a minor point because signals are analogously multiplexed.

Given functional requirements hybrid applications may be imagined. For instance, for a pay-TV application e.g. via upstream low-bandwidth PLT, ordering and accounting can be done using a low-speed reliable connection and signal transmission downstream by high bandwidth optical fibre.

2.2.3 Roles of service providers

There are a number of actors engaged in the delivery of services [3] and in the operation of software systems. Deployment of services needs service developers, service brokers, partnering with a service aggregator, an internet service provider, a connection provider and/or a device vendor, and subscribers. The platform and local infrastructure provider, who delivers the hardware infrastructure and basic communication software for the local access node defines IP-address mapping from in-home devices to the gateway. A service/application provider delivers the application functionality. In the business software market, it is expected, that the proportion of applications, that are run centralised by dedicated service providers will increase significantly. The communications service provider (service enabler), that operates communications access for an ASP (application service provider). The role of utilities in this respect may vary. Consumer research [3] shows that utility companies are the preferred providers for bringing packaged services into the homes.

2.2.4 Standards for device interfacing and inter-device communication

In order to combine all these kinds of networks, transparent interfaces are to be defined to establish the common data-structures and procedures. Genericity of the interfaces is the key to interoperability of these different apparatus, applications and networks. Devices have to be connected to other nodes in the network as seamless as possible. A large number of standards for these interfaces and for small-scale network traffic have been defined by the industry.

2.2.5 Device access technology and installation mechanisms

2.2.5.1 IP: the Internet Protocol

Recent miniaturisation efforts from the industry make it possible to have an IP-stack [1] on very limited hardware. In this way the large amount of money invested in Internet communications software is used effectively. Very thin-client applications are promoted in this way, leaving the main responsibility for operation to a connected server. The software to control an IP-node in such a way is connected to attribute-syntax within an HTML context.

2.2.5.2 Jini: device drivers served by devices

Jini is developed to enable spontaneous, dynamic networking for all kinds of devices [4]. Jini is a specification launched by Sun Microsystems. Adoption of Jini by device manufacturers relieves the task of configuring and installing devices in heterogeneous networks. Furthermore, the management and maintenance of complex networks is facilitated, if devices are installed according to this specification. Devices, in the Jini-philosophy, are responsible for dispatching their device driver code and device management parameters to applications instead of having device driver code installed and managed as part of network management. Through this mechanism, the task of building, maintaining and changing a network of users, software and devices is simplified. Devices can be plugged into any network and make their services available to any other device as well as are becoming aware of any services other devices on the network can provide.

From an architectural point of view, Jini devices, once they are installed, can be instantiated and managed as objects, to and from which messages can be received. The Jini infrastructure (the *Jini federation*) consists of

- Look-up service: the database for services available in the federation.
- Discovery protocol: allows discovery of any resource plugged into the federation and registers it's services using the look-up service.
- RMI 1.2 (a method or function call across a network): Remote Method Invocation is a part of the Java Virtual Machine and makes it possible to make use of methods located at a remote machine.
- Distributed security system: a security framework in which an ACL (Access Control List) describes which components are allowed to get access to other components.

Jini can be implemented in as little as 48 kB of base infrastructure code for the virtual Java machine and it's class libraries. Jini is tightly coupled to developments of the Java programming language. From JDK 1.2.2 on, it is supported by the Java development environment. Java does not yet meet the high expectations on delivering transparent inter-process communication primitives for distributed communications.

Although the specification is licensed to over 20.000 customers, examples of use of the Jini specification are coming up very slowly. One of the early adopters is Echelon, which is the owner of LON-technology. Hard disk suppliers such as Quantum incorporate Jini into its hard disks, so that each disk drive can make itself available to all computers on a local network. Madura will launch an ERP package based on the Jini concept as a service over the Internet rather than being sold to customers. Highest pay-off would yield the application of JINI dynamically driving a printer in a PC-network, but presently no examples exist.

2.2.5.3 Universal Plug and Play (UPnP)

As for Jini, UPnP [10] is a standard, defined to make installing devices as general as possible. This mechanism is more classical, as all data and control processes are part of the device driving application. The standard is a follow-up and micro-networked self-exploring extension of the common device driver concept. A device consists of a number of configuration parameters and of a number of specific functions, which are called by applications in a generic scheme. Management and maintenance of device driving code is major focus point for the software architecture. Mechanisms as defined for JINI as an auto-discovery protocol are implemented in the standard.

Universal Plug and Play (UPnP) is an architecture for pervasive peer-to-peer network connectivity of PCs of all form factors (physical card dimensions), intelligent appliances, and wireless devices. It is a distributed, open networking architecture that leverages TCP/IP and the Web to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and everywhere in between. In UpnP device control protocols are developed now for a large number of devices. These device control protocols are the counterparts of the discovery protocol of Jini.

2.2.5.4 LON

Devices having LON-node hardware have a dedicated, proprietary processor handling I/O on one hand and network traffic on the other. LON-devices can form a dynamically configurable network. All LON-devices operate in an advanced peer-to-peer configuration and communication schedule. In the LONMark association a number of companies are governing standard network variable types and node objects.

2.2.6 Local network communicationtechnology and data transfer paths

Currently a large number of technological developments take place and standards are developed to interconnect devices on a small scale. The developments are spread from very low (kB/s) to very high bandwidth (400 MB/s). Especially the development of SOHO's (small offices, home offices) adds to a large increase in small local networks.

2.2.6.1 LONWorks

LONWorks [5] is a network technology built especially for small-scale networks. In most commercial building management systems today, this technology is common. LON abstracts the logical network from the physical network by a communication bus architecture and processors in such a network, which can be addressed from a central place. Connectivity of apparatus and control logic in a LON-network can be programmed and configured dynamically. Main drive for LON was the vast reduction in cabling necessary for experimental installations. LON technology is independent of the physical medium for data transfer. Frequently in-building power line communication is used. In some residential gateways, LON-technology is used extensively.

2.2.6.2 X10/EHS

X10 is the standard for low bandwidth, home automation communication. As a physical transport medium for signals the powerline is possible. EHS is the result of a number of projects conducted by the European Home Systems Association [5]. In architectural terms, X10 and EHS can be seen as bus-structures allowing the individual addressing of network components and appliances. EHS-networks can interface to an EHS-modem central in the network to communicate to peripheral networks.

2.2.6.3 IEEE-1394

For high band-width applications a bus architecture, IEEE 1394/Ilink, has been developed to facilitate high-bandwidth interconnectivity of entertainment and home theatre applications. The 400 Mb/s databus twisted pair and low-connection cost enables high volume data-transfers if broadband has passed the access node. At lower bandwidth the Apple advocated FireWire standard connects devices without wires. The IEEE 802.11 (Ethernet) standard has been extended to bring network technology to the homes and has a wireless version.

2.2.6.4 HiperLAN-2

These standards utilise the 4.0 GHz-band to wirelessly interconnect high-speed data-transfer devices in a computer network setting.

2.2.6.5 Bluetooth

Bluetooth is a recent development lead by Ericsson [2]. The standard uses wireless technology in 2.4 GHz-band to realise 750 kB/s point-to-point connections. In the topology instantaneous piconets are set-up with one master and up to 8 slaves within a reach of 10 meters. Problems for in-home use of Bluetooth enabled devices may come from microwave oven radiation, which may emit in the same frequency band.

From an architectural point of view, piconets present a one-point addressable subsystem in an application. As such, they can be modelled as separate entities in the system. Prime target of Bluetooth is to replace the USB-bus cabling of PC's to peripheral apparatus with wireless communication.

2.2.6.6 HomePNA

The Home phoneline network alliance uses existing copperwire to connect the local access node to the individual devices. The ADSL and XSDL standards, currently available, support high data-transfer rates; up to real-time video.

2.2.6.7 The HomePlug association

In this association research activities are set-up and co-ordinated to make the in-house powerline suitable for high-bandwidth applications. Data transfer speeds obtained are in the order of 2 Mbit/s; just below the limit for digital video.

2.2.6.8 SCP, the simple control protocol

Microsoft advocates this protocol. Simple Control Protocol (SCP) is a lightweight, royalty-free networking technology optimised for devices with very limited memory and processing power and for networks with low bandwidth, such as power-line carrier (PLC) networks. Devices that stand to benefit from SCP include lights, home security devices, home automation devices, and other small appliances that are not capable of supporting TCP/IP networking or that connect to the home network through a low speed powerline carrier medium.
Microsoft, General Electric and other industry leaders are combining their home control networking technology focus to converge existing home control initiatives into SCP, thus enabling a global standard for lightweight home control networking. Originating from Microsoft SCP is tightly coupled to uPnP. The first implementation of SCP will be embedded in an inexpensive power-line-networking chip next year. Domosys, ITRAN Communications Ltd., and Mitsubishi Electric Corp. are all actively developing SCP-enabled PLC chips which manufacturers can use in the development of smart appliances and home control products.

2.2.6.9 HAVI

For inter-device communication for audio and video devices a software standard describing the interface, message format and the protocol for audio and video devices has been defined by industry leaders in the multimedia branch. The standard is geared towards high-bandwidth transmissions.

2.2.7 The services gateway

The services gateway is the central point by which part of the local network is controlled and served. The services gateway may be a residential gateway but this does not necessarily have to be so. Essentially, a modem+telephone is a primitive form of a service gateway. The services gateway is attached to a wide services network to connect service providers to internal clients. RG's terminate all networks and enable multiple home targeted services. RG's are to be discriminated on the complexity dimension of applications and on the number of networks interconnected. RG's may be only enabling one service specific service and one specific network. At the other extreme a whole-house gateway terminates all external networks and enables all services.

2.2.7.1 Existing devices.

The residential gateway forms the entrance point from the last-mile to the buildings or homes. From a software architectural point of view the residential gateway has a definitive role in synchronising and controlling applications implemented by several service providers. The physical implementation of the residential gateway has a number of forms. For metering applications one might imagine an intelligent meter. For streaming multimedia at the moment a third generation set-topbox with two-way communication possibilities can be imagined. More universal residential gateway concepts are under development now. Devices consist of a central processor, an amount of memory and access channels to several networks. In that sense they mimic a conventional computer. The processors mostly have an embedded real-time operating system. Operating systems range from very dedicated and functionality tailored ROM-able ones up to general purpose Linux with real time extensions. Programming the devices can be done by uploading cross-compiler generated code, a purpose designed command language or through access through an IP-stack.

2.2.7.2 CoActive connector

For control applications, CoActive systems has a gateway on the market in which the system has an Internet-address and software to communicate to the inside and outside world using the IP-protocol. The IP-messages are transmitted through 10BaseT Ethernet or modem. Internally the connection is established through an EIA-709 connection via LONWorks. The IOConnect Architecture in the Coactive connector avoids the server bottleneck at the centre of other architectures. Instead of requiring that all control data be gathered into a single server, and then forwarded across this server's network connection in a fixed way, the IOConnect Architecture allows you to connect multiple control subsystems to the IP network in a distributed fashion according to logical and physical requirements. The subsystem connection is accomplished using a compact embedded connectivity device, rather than a PC. The IOConnect Architecture makes the IP infrastructure available to all electronic devices. There are applications for automated meter reading using Sensel. The latter company [6] uses CoActive systems to do meter reading and offering feedback using the Internet. In a field trial on the Isle of Gotland in Sweden, the Vattenfall spin-off, Sensel, positions itself as an infrastructure provider using the Co-Active systems technology. Several thousands of households are currently equipped with intelligent residential gateways to allow application service providers to install applications.

2.2.7.3 i-LON

LONWorks markets the I-LON technology. In an I-LON box a bridge between a LON- and an IP-network is established. In this way, systems may be remotely controlled from every place having an Internet connection. I-LON devices are managed through a dedicated operating system interface with a number of DOS-like commands with 1 Mbyte flashdisk as a background storage. I-LON devices are addressable in a HTML-environment using a networkvariable type. With a special tag in the HTML syntax input and output attributes of devices may be set. Using the tag, the mapping of networkvariables to displayable fields in forms is made. Using forms these may be linked to fields in forms, that are transmitted using the get-form attribute. The I-LON box has a WEBserver, that mediates the IP-traffic and makes the conversion to LON-network topology. Appropriate mapping schemes exist between logical and physical network components and to build hierarchical data and network structures and security mechanisms. Using NAT (Network Adress Translation) the individual in-home IP-addresses may be translated to one central address for communication with the provider. The mapping scheme adheres to the LONMark conventions for configuring types of devices. In this way the WWW can be safely connected to small local LON or IP peer-to-peer networks. LON-networks are not addressable across firewalls. CGI-like processing takes care of sending the whole content of a form from one place to the other. Network variables may be exchanged on a get and on a by polling base. Using this I-LON technology in Italy 27 million homes are equipped with an intelligent meter by ENEL in the coming year. Services implemented include meter-reading and load management applications. In Italy especially exceeding a threshold value leads to more expensive tariffs. In the future more value-added services are foreseen.

More advanced control logic for applications may be obtained by Java applet or servlet programming. The latter will be the method of choice for large centralised control applications. Currently however the iLON-box does not support the Java because the JVM is not supported.

2.2.7.4 E-box.

Ericsson targets its strategy not only on home networks but also on home applications. Essentially, the added value is the extension of the context of their wireless devices to a central interconnectivity box in the home. As a part of this, an E-box was developed as a solution for a residential gateway. From a software architectural point of view, the accessibility and interfaces of this kind of apparatus are most important. The E-box software is based on a multi-processing environment in the Unix operating system. An E-box can be connected to a WEBPad, which allows a portable in-home screen for establishing an Internet connection.

2.2.7.5 Enikia

In the Enikia [7] perspective in-home powerline communication is combined with last-mile access to transparently enable Internet access directly to the device/appliance level. Via pervasive computing an information appliance network based on the Internet a platform to exploit services is set-up.

2.2.7.6 Cisco

Cisco Internet Home Gateway 2000 series is a family of residential gateway devices that enable multi-services delivery to the home over high-speed, always-on broadband connections using DSL-technology. The product family features easy home networking and self-configuration technology. The product family is open to service providers.

2.2.7.7 Aladn, emWare

These gateway-types are typically home automation centred control servers with outside connections and proprietary operating system architectures. For communication standard low-speed protocols are used and the primary application is in home control systems.

2.2.8 The Open Services Gateway initiative (OSGi)

This initiative has been started by a number of companies including IBM. The Initiative is now gaining broad support from the industry. OSGi is creating open specifications for the delivery of multiple services over wide-area networks to local networks and devices.

In the Services Gateway, OSGi [8] is the platform for communication based service applications. The SG can enable, consolidate and manage voice, data, Internet and multimedia communications to and from the home or office. It also functions as a service enabler for a range of high value services such as energy management, home automation and security, device control and maintenance, etc. The scope of services includes

- Communication Services: point to point communication for customer, inter-PC networks and appliances.
- Energy Services: Automated Meter Reading, load management and comfort management.
- Home Automation Services: more flexibility, bundling with other residential services.
- Security Services: id.
- Remote Home Healthcare Services: special services for elderly and disabled people.

Aim is to integrate parts of the OSGi software specification in products and applications and map these to set-top boxes, cable modems, routers, residential gateways, alarm systems, energy management systems, consumer electronics, PCs, and so on.

As a software specification, OSGi is platform and communication medium independent. It supports multiple local network technologies whether wired or wireless. It also supports multiple device access technologies such as UPnP and Jini. OSGi is strongly supporting Java as an implementation platform. At the moment a completely documented Java object model is available.

From an architectural point of view, OSGi has an embedded server, attached to the wide-area network, that connects external service providers to internal clients. OSGi includes APIs for service life cycle management, internal service dependency management, data management, device access and management, client access, resource management and security. In May 2000 release 1 of the OSGi-specification has been issued. Objects in OSGi are described and implemented in the Java programming language. Thereby OSGi anticipates on expected developments in the Java arena like JavaOS's for appliances, embedded Java, personal Java and Jini. On the other hand, HTML, XML and other Internet-related technologies are supported. The transmission medium for program code and data for services is the Java .jar archive format. In the OSGi model the service provider provides access to services by downloading software on the gateway as a .jar-file, the deployment item in Java. In the OSGi-specification a service aggregator can also be discriminated, that provides a set of bundled - compatible - services. Furthermore a role is attributed to a gateway operator, that manages and maintains service gateways and its services. In many cases the gateway operator will also be the gateway owner, retailer, installer and/or hardware maintainer. A provider comparable to an "Internet Service Provider" provides the necessary communications over a wide-area network.

Entities in an OSGi scope are:

- Service management. OSGi defines API's for creating and running services in a multiprocessor, heterogeneous network. A multiple provider environment is also defined.
- Device-access management.
- Logging of events.

Follow-on activities within OSGi include HTTP Services, Client Access Device interaction schemes, configuration data utilities and persistent data services linking to large database systems and legacy software through transparent interfaces.
The service framework provides a convenient context for developers to design applications. Services may be contained in bundles. OSGi provides primitives for live-insertion of services into aggregates called service bundles, life-cycle support of services and containment of devices in device bundles. In the following architectural discussion OSGi is one of the key mapping mechanisms

With respect to the device access specification, OSGi is a software layer complementary to Jini, in the sense that it can use the Jini infrastructure to define the device interfaces. OSGi has primitives for automatic detection of attached and detached devices. A mechanism has been designed for device driver search in a network. Devices can be configured dynamically and may be plugged in an executing set of applications. Furthermore, the object interaction vehicle is the Java language. The packaging concepts and expressiveness of this language are also used to define the detailed specification [8]. Native programming code, code not executable by a virtual machine, is encapsulated into JNI, the Java Native Interface. The way native programming code, different from Java-code, is integrated and deployed in the different components of the environment is defined in the standard.

In abstraction level, the OSGi specification resembles the CORBA and DCOM- standards for object interaction in distributed networks. A casting to the operational environment with distributed intelligence and data in heterogeneous size and processing power processors and different speed networks is made.
The key entities in the Framework are:

- Services - Classes that perform certain functionality written the interface and the implementation separated. Service providers have to write software to implement the interface functions. Essential for the design and implementation of the service functions are their live-update, reliability and stability attributes. Services have a well-defined level of operation as reflected in various states of service-processes.
- Bundles - The functional and deployment unit for shipping services. A bundle consists of a number of programs deployed as a Java-archive, .jar, file. Bundles may interact with each other through marshalling via a registration and publishing mechanism similar to common operating systems environments.
- Bundle contexts - The execution environments of the bundles in the Framework. The bundle context governs the execution of bundles and services in a heterogeneous system environment.

OSGi gives a number of concepts to discuss service gateway applications in a common way, be it from the hardware or from the software point of view. The further developments within the

OSGi consortium are of utmost importance to companies developing hardware and software for applications. In the further discussion connection is sought.

2.2.9 Integration through XML/DOM

Several software vendors are strongly advocating XML at the moment to tackle problems with distributed data management and industry standardisation of information models in schemas. In the Microsoft .NET-architecture [11] the standard plays an important role for application development. All various components are integrated to the XML meta-language. XML features data management aspects for very large-scale applications, XML has self-describing mechanisms for device access. Inter-application mapping using XML is facilitated through the Document Object Model standard. In this standard, distributed objects are made understandable in a transparent way across a network.
SOAP (Simple Object Access Protocol) is a lightweight communication protocol designed to let COM or CORBA objects communicate. SOAP is a lightweight protocol for exchange of information in a decentralised, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data-types, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols (COM, CORBA). With the SOAP protocol methods can be invoked through the Internet. SOAP codifies the existing practice of using XML and HTTP as a method invocation mechanism. The SOAP specification mandates a small number of HTTP headers that facilitate firewall/proxy filtering. The SOAP specification also mandates an XML vocabulary that is used for representing method parameters and signature, return values, and exceptions. SOAP has been submitted to the W3C for the formation of a working group.

2.2.10 Utility service end

Utilities are trying to discriminate themselves by getting into the E-commerce business. For instance, in the USA, Kansas City Power and Light [12] gives customers the possibility to access customised information. ENRON, together with IBM and America-On-Line recently established the New Power Company [13] to deliver services partly as described above to residential and small customers.

For accounting, billing and customer relationship management, systems operate batch-wise using standard third or fourth generation business software. These systems are loosely coupled to systems governing the primary process, delivery of electricity, gas or water, which operate in a real-time environment. Small local intelligent systems near electricity producers or consumers operating in concert with above two types increase the complexity of interaction.

With energy saving and renewable generation in mind the near-future the trend is from large, centralised power generation units with a easy production forecast to small, distributed generation units, the production of which is more difficult to predict. An “energy-farm” may be thought of as to consist of a cluster of mini- and micro-CHP (Combined Heat Power) units, solar energy and wind-turbines. These “energy-farms” do not utilise a large distribution infrastructure but can pose net-balancing problems on several time-scales (from milliseconds to hours). Management of these small individual energy generating and optimally using components requires the design of advanced distributed information systems and new algorithms for distribution automation.

Components in such next generation systems perform:

- Data-acquisition van essential parameters with several sampling intervals.
- Storage of operational parameters in a distributed database. In the database the history, the long and short-term use of data, is an important attribute.

With the advent of small scale, distributed computing, the following new service applications are possible for energy delivery systems:

- (A)DSM. Advanced, fine-grained, demand side management specific to energy farms.
- AMR. Automated meter reading and trending on a small scale. Aggregation of production figures and determination of forecasts to optimise performance. Small sizes of production units require solid procedures for accountable registration.
- Tampering and outage detection. Algorithms can be implemented on small processor hardware to detect temporal anomalies in consumption patterns. This may give valuable information about the misuse or defects of the distribution network.
- Monitoring of the quality of power (harmonics, dips, spikes, probability of failure) delivered. Depending on the use of power, a customer may require different qualities.
- Predictive maintenance. Viewing the behaviour of selected, sampled parameters on different time-scales may allow the detection of upcoming defects in equipment before these defects cause damage.
- (A)DA. Advanced Distribution Automation. Systems now are under human supervised real time control via SCADA-software. The future network topology will change yielding several clusters of individually optimised production islands. Given a set of distributed computers in a suitable network topology, distribution resource planning with real-time pricing constraints can become a reality.

The topology of the distribution and demand side management gives utilities a competitive advantage to set up value added services. For exerting these services a connection to local home/building automation networks is necessary. Utilities in this way may be in the position to build up a more extensive customer relation as energy knowledge experts. Derived information of metering and seasonal trends may lead to advise for improving efficiency. Especially if the readings of electricity, gas and heat delivered are analysed, efficient partitioning between these carriers may improve overall efficiency. Preventive monitoring techniques used in the distribution process may also be delivered to customers. For larger customers, in real-time tariff situations, an increase in information is an essential prerequisite for contract management, strategy implementation and cost-effective prediction of future trends in generation and consumption.
For large customers in a liberalised market, next day tariffs are to be given on a quarter of an hour basis from the demand and the supply side. This stresses the need for increasing metering and control intelligence.

For tele-metering and tele-control a number of standards are or are being defined by the IEC. Some notable examples are discussed below.

2.2.10.1 IEEE61334

In this protocol the message formats are defined in a DLMS-specification (Device Language Message Specification. To this standard a large number of standard metering devices are adhering at the moment.

2.2.10.2 IEC60870

The IEC 60870 standard describes a protocol for remote metering and control applications. It incorporates an object model and the interfaces between communicating parties on both sides of the powerline. The communication mechanism defined in the standard, however, has a poor performance. Especially setting up a large number of connections takes a lot of time.

2.2.10.3 UCA

The Utility Communication Architecture (UCA) is an ensemble of open protocols and standards that helps eliminate the extra costs, redundancy and inconvenience associated with using system-specific, or proprietary, communications interfaces each time new equipment is connected to automation systems. UCA makes use of a selection of international data communication standards for the complete range of electric utility needs. It facilitates distribution automation and has interfaces to Supervisory Control and Data Acquisition (SCADA). In the standard an attempt is made to seamlessly connect all of their key facilities, from the control centre to the customer's meter, along the electronic communications highway. UCA is based upon open standards. UCA offers interconnectivity between equipment from different manufacturers and interoperability between the databases used to exchange high-speed, real-time data in utility operations. In other words, it allows utilities to "plug and play" equipment from different vendors over the same data network.

2.2.10.4 IEEE P1547

A major development in energy distribution and the topology of distribution networks is the distributed utility concept. In this concept, local generation of energy is favoured. In the distribution network from the substation-level local energy generators like PV, fuel cells and CHP feed in into the grid. The way to balance load and generation is more complicated than in the hierarchical case: each individual generator has another form (Voltage, frequency, AC/DC) in which the energy is generated. To guarantee net stability and power quality hardware and software for control has to operate in a concerted fashion. In the IEEE P1547 standard, that is being defined now, requirements and protocols are defined for control system hardware and software. This standard provides a uniform standard for interconnection of distributed resources with electric power systems. It provides requirements relevant to the performance, operation, testing, safety considerations, and maintenance of the interconnection. More information can be found on http://www.manta.ieee.org/groups/scc21/1547/#top . In view of recent power shortages and infrastructure insufficiencies in California, distributed generation scenarios, also with ample usage of small-scale renewables are developed in the US.

2.2.11 Summary

In essence the following scheme illustrates the standardisation efforts of the application types, that are further discussed in this chapter:

Type	Local network	Local access node	Concentrator/ router node	Utility node
DCMS	LON, X10, EHS, HomePNA, HomePlug, SCP, IEC60870	OSGi, DCOM, CORBA, RMI		IEC80870, XML/DOM, UCA, IEEE P1547
PP	DECT, Blue Tooth	Home PNA		
Internet	IP, HTTP	IP	IP	IP, XML/DOM, SOAP
EMA	HAVI, IEEE1394, HiperLAN2, IEEE802.11	OSGi		

Table 2.1 Hardware and software standards

Essentially, at the moment the number of technical hardware and software standards is sufficient to build powerful applications. With the OSGi-standard a first conceptual standard is becoming accessible. There have to be more of these conceptual standards and implementations of existing standards like COM and CORBA to facilitate large scale layered application development. Furthermore, standardisation further on the hierarchy, especially on the concentrator node is essentially lacking. This concentrator node would have a discrete advantage for last-mile access applications in allowing RG's to be relatively cheap, "thin" clients served by a "thick" concentrator node server. This not being the case, the concentrator node only has a transparent routing and data transfer function with emphasis in application logic on the utility node and RG processing power.

2.3 References:

[1] D. Estrin, R. Govindan, J Heidemann, "*Embedding the Internet*", Communications of the ACM, May 2000.
[2] Bluetooth specification can be found on www.bluetooth.com.
[3] Proceedings the second Home Networks European congress. 23-24 May London, 2000. Especially the articles of Parks Associates contained therein.
[4] www.jini.org. The central place for obtaining the specification.
[5] LON is extensively described at www.echelon.com. EHS can be reached at www.ehs.org. EHS-devices are the results of a number of EU-projects conducted in the ESPRIT-framework.
[6] Sensel is a firm installing residential gateway technology from Co-Active systems. The role of Sensel is providing the infrastructure. References are www.coactive.com and www.sensel.com.
[7] Enikia is a firm recently established for operating services through gateways. On www.enikia.com detailed information can be found.
[8] The Open Service Gateway initiative is described on www.osgi.org.
[9] SOAP (http://www.w3.org/TR/SOAP) gives a description of the proposed standard
[10] UPNP2000. www.unpn.org The main source of information Universal plug-and-play forum.
[11] Various authors. Microsoft Developer Network Journal. September 2000. Microsoft Press.
[12] This service can be found at http://www.kcpl.com.
[13] The company presents itself at http:/www.newpower.com.

3. MODELS FOR DISTRIBUTED CONTROL AND MONITORING; SYSTEMS ARCHITECTURE FOR ENERGY-NEAR SERVICES

3.1 Introduction

As we saw previously, the service infrastructure consists of at least four levels,: the devices at the end-nodes, a local access node through which the home is coupled to the outside world, a concentrator or base station, which connects to the communication backbone, and the utility node, which delivers the services from the utility or service provider.

A number of requirements can be summed up for smart devices in a last-mile<> home network. A device should have a unique id to allow addressing in large hierarchical networks. A physical and logical mapping mechanism should exist to enable proper context definition. Furthermore a device should be reasonably priced and standards used in the device hard- and software architecture should not be proprietary and have the support of many vendors; it should be self-configuring, live-insertable in the network and have self-diagnostic features and have non-volatile memory. In this way, smart devices can be seen as appliances performing services on a small scale: a smart device has responsibility for a special task. Other components in the network can call upon the device to perform this task.

Traditionally, the local access node to networks is dedicated to one application. At this moment, in-home networking and automation is emerging at the market, in which the access node is implemented using a residential gateway [1, 2]. This gateway provides local intelligence, is able to connect different Internet access networks to multiple types of in-home networks and can deliver different e-services over one single connection. Although this gateway can serve as an 'all-purpose' access node we will have to take into account that a home will have more than one local access node.

E.g. the base station, also called in literature the concentrator, for PLT-applications is normally located at the transformer substation. A main task for the base station will be to act as a multiplexer for requests from the top end of the network and as an intermediate for information transfer from the lower part of the network. The way the multiplexer station is embedded in the power distribution network, determines the functionality of powerline value-added services. In the USA, a very limited number of households are connected to one base station. In Europe, the situation differs per country, but the number of connections generally is very much larger. This poses a definite advantage for local applications with residential area scope.

From the software architecture point of view, the OSGi separates utility node in two roles, the service provider role and the gateway keeper role. The first party is responsible for the content of the services, the second party is responsible for the way these services are set up and rolled out in the network.

We therefore enhance the four levels mentioned above to five levels, as outlined in figure 3.1, and which will be elaborated further in the following paragraphs:

- devices,
- local access node,
- base station,
- service provider,
- gateway keeper.

These infrastructure levels must communicate using standardised interfacing such as described in previously. Powerline services then can be deployed on the above infrastructure in such a way that decisions can be taken at a logical level, thus relieving the rest of the network. The main responsibility of a software architecture is to deploy tasks at the right level and to assure that information needed to fulfil these tasks is communicated to this level. The infrastructure structure requirement, as stated above, complies with the OSGi model. In some applications two or more levels will coincide with one network node.

In the context, mentioned above, the following elements have to be considered for our model:

1. The user or customer. Application types have been discussed in Chapter 2. It should become clear at which level the user can operate on the network. The user interfaces should be simple yet adequate. Both browsers and touch screens can deliver this functionality.
2. The legacy systems at the level of the service provider and/or the gateway keeper. For obvious reasons the powerline service systems should be linked with BackOffice systems in areas as finance or accounting, customer relationship management, marketing and sales, etc. These systems can be loosely coupled.

The user should not be aware of the underlying technical details and network infrastructure. All objects and components should be real world and be transparent.

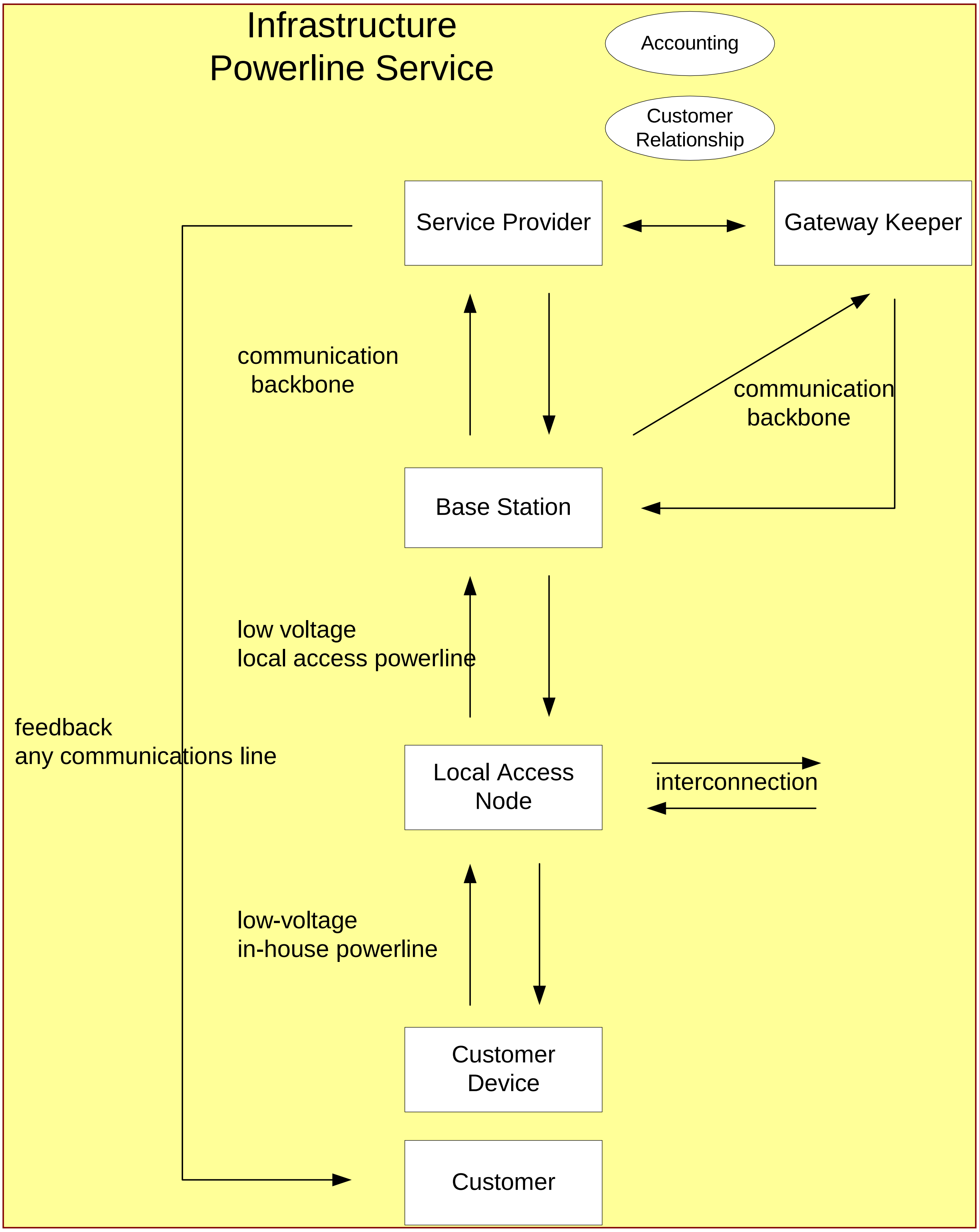


Figure 3.1 Infrastructure Powerline Services

3.2 Devices, architecture and value creation

Devices are the basic elements in the information structure, as they either provide information to be handled in the service or they receive information, which will be handled internally by the device.

Enikia [2] differentiates three types of devices:

1. An **appliance** creates value primarily through its physical processing features. Therefore, the embedded intelligence features will likely focus on device control and feedback data relating to the device's mechanical functions such as power consumption, maintenance status, or performance data. Examples are a commodity meter, which provides usage statistics, and a washing machine, which washes clothes.
2. The fundamental value of **computers** is using embedded intelligence to process digital content.
3. In between lie **electronic devices** that rely on both embedded intelligence and mechanical functions to add value (for example, CD players or printers).

Appliances, electronics, and computers will all have embedded intelligence and communications abilities. These can be seen as the service(s) of a device, which are available for other nodes in the network. Note that the term 'service' in this context is not quite the same as the 'service' in PALAS terms. On request a device can deliver information, or it can perform a task.

3.2.1 The role of access nodes

The access node is a first level of aggregation, which acts as a buffer between in-house information and applications and the outside world. This access node might provides local intelligence, might be able to connect different Internet access networks to multiple types of in-home networks and might deliver different e-services over one single connection. Note that a home may be supplied with more than one access node, each used for different applications. One can regard even a modem as an access node. Its only intelligence is to transfer data from a computer to the telecommunication network and vice versa. An intelligent access node, such as the residential gateway, can be used as a layer where control is based at a low point in the network. If we can deploy decisions to be taken at this level, we might prevent extensive data traffic to and from the base station and the service provider. Also the service provider will not be extensibly busy for individual customers, but can concentrate on integrating tasks.

An important discriminating point for applications utilising the intelligence contained in access nodes is the extent to which control actions may be performed autonomously. As an example consider an entrance security application. Many home automation applications can be working autonomously at the gateway. This also holds for energy management in the house based on e.g. fixed pricing and / or availability of local energy supply and energy buffers.

3.2.2 Base Station

Powerline is especially equipped to serve as a medium for communication in-house and at the last mile. The concentrator serves at the buffer between the powerline last mile services and the (telecommunications) backbone, thus opening up the Internet-based services for the in-house applications. Communication between the concentrator and the gateway will be provided using the powerline. Communication between the concentrator and the service provider may use any communications infrastructure, such as ADSL, cable, wireless, etc.

Example: direct telecommunications applications can use the concentrator level to set up direct connection to another party without involving a central host of the service provider. Connection information can be gathered real time either at the gateway level or at the concentrator level.

3.2.3 Service Provider

The service provider is responsible for the content of the service. Each provider will also be interested it in the usage of its services. Therefore this information has to be collected at the service host and stored for later handling. Typically, processes at this level are accounting and billing and customer relationship management.

3.2.4 Gateway Keeper

The gateway keeper is primarily concerned with operation and maintenance of the hardware infrastructure and basic communication software. For powerline applications it is assumed that the utility serves as the gateway keeper. Note that the gateway keeper can also act as a service provider. However, conceptually we keep these two tasks separated.
All operational, management and maintenance issues around services and gateway configuration will be dealt with by the utility.

3.3 Modeling metering and energy near Services

3.3.1 UML Modelling

In section 2 we have characterised HVAC-applications as a typical DCMS-application: a Distributed local Control and Monitoring System.

In this chapter we will describe some typical service examples for this application type and elaborate these into object diagrams according to the Unified Modelling Language (UML) standard. UML is an industry standard for developing object models and is described thoroughly in Rumbaugh [3], Jacobson [4] and Booch [5].

UML starts with describing a system by so called use cases, in which, in clear language, examples of operational parts of the system are worked out.
The next task is to identify the object classes in the model from the description of these use cases. A class can be described by its attributes and operations. Attributes are properties of each class object, e.g. the name of a person. Operations are the responsibilities of the class objects. It will be clear that the meter is responsible for supplying some kind of usage figures.

Typical classes, which will be identified, are:

- Information carriers, such as: Meter, Electric Appliance, ...
- Information handlers, such as: Registry, Billing or Accounting, ...
- User interface devices: Screen, Audio, Telephone, ...
- Involved Parties: Customer, Utility, Service Provider, ...
- Topology: relations between involved parties.

In the UML object model the basic data and functions of a system are determined. In a detailed UML design also the complete behavioural model is determined. In this document we will only discuss this aspect for a number of typical applications as far as it reveals the main characteristics of these applications.

Since services will be working in a distributed environment special attention has to be paid to the deployment of the responsibilities (operations) on the network topology. Deployment can be described as the configuration of all nodes in a system and the distribution of all components,

objects and operations on these nodes. In chapter 4 we have already indicated a likely network configuration for powerline services. From here we can deploy the object model on this network.
References for deployment are Orfali [6] and CORBA (see Ben-Natan [7]). The latter is a standard for object oriented component system interfacing and is closely related to the standards described in chapter 2.

3.3.2 Distributed local Control and Monitoring Systems

In the DCMS network the control processes and data are distributed over the network. The system is interested in a service delivered by any of these nodes in the network. Examples of this type of application are automatic meter reading and monitoring of appliances. In this paragraph we will elaborate on the first example.
The communication network uses infrastructure such as BlueTooth, USB, PLC on a narrow bandwidth up to 100 kBps.

3.3.2.1 Use Case: Automatic Meter Reading

Automatic Meter Reading (AMR) can be described as the remote collection of consumption data from customers' utility meters. AMR provides electric, gas and water utilities with the opportunity to streamline metering, billing, and collection activities and to enhance service to customers and gain a competitive advantage (AMRA definition). AMR also provides detection of tampering and/or energy theft.

Step 1: Description of the use case

The metering service can be described as follows:

Each <u>customer</u> has one or more <u>meters</u> which provide information on the <u>usage</u> of any <u>commodity</u> (electricity, gas, and water). These usage data have to be *registered* such that they reflect the price fluctuations over time of the commodity. Pricing can be based on fixed prices for fixed time intervals (e.g. day vs. night <u>tariff</u>) or it can be market dependent, in which case <u>prices</u> can fluctuate every hour or so. The information needed in automatic meter reading is the <u>amount of usage</u> in every <u>price period</u>.
At certain <u>intervals</u> the registered usage data will be *read out* by the relevant <u>utility company</u>. The read information is *stored* for later reference.
At certain <u>periods</u> in time the utility company *sends out* a bill to the customer, who *pays* either electronically or by normal <u>bank transfer</u>. The bill is based on the commodity usage and the <u>prevailing prices</u> at the time of use.
The customer can *review* his <u>usage pattern</u> by visualisation of the usage <u>history</u>. Also the <u>cost of usage</u> can be *visualised* on the basis of prevailing prices.
The utility can *analyse* the data on tampering and/or energy theft. How this is done is not subject of study in this case.
Another part of the AMR service is maintenance, including adding and deleting new meters, etc. Standards such as Jini, UPnP and OSGi can be used to automate this task.

Step 2: Identifying objects and attributes

From the description we can derive the following objects in AMR:

- <u>Meter</u>: A meter measures usage, e.g. of a commodity. A meter usually is a kind of counting device.
- <u>Utility</u>: The commodity provider.
- <u>Customer</u>: Has an agreement with the utility on delivery of one or more commodities.
- <u>Commodity</u>: Electricity, gas and water are mentioned as commodities, which can be metered. A commodity is characterised by a certain amount of usage over a time period.
- <u>Tariff</u>: The cost per unit associated with the commodity; tariff is time-dependent.
- <u>Clock</u>: Recognition of intervals and periods requires data to be read on a regular time-base. A clock will provide us with timing.

– Usage data: Usage data consist of periodic readouts of the meter counter. The readout period depends on the tariff changes of the commodity. Each change in tariff determines a new readout period.

– Accounting System: Metering information is used for billing. Therefore we need an accounting system to handle this billing. In the accounting system the billing frequency and banking information is included.
The usage data are the input in the accounting system. Output is an account view to be sent to the customer.

– Information System: The user should be able to view the metering information. Therefore we need an information system to handle this user feedback. The usage data are the input in the information system system. Output is a usage view, to be shown to the customer.

NB. We will not consider the preparation of the bill as a part of the service system. This will be handled by the accounting system. However, on-line viewing of the account can provide added value. We can enhance this to a special service for "e-Billing".

Step 3: Identifying operations

The following operations can be identified in the above description of AMR:

– register usage: Commodity usage over time has to be registered by the meter. The residential gateway can be used as a buffer to store these data for a longer period.

– read out meters: Either the provider periodically reads meter usage data from its customers' gateways or the gateways send usage data to the provider host. The frequency depends on the buffer capacity of the gateway.
In order to reduce communication traffic the concentrator may act as an intermediate by assembling the data from its gateways and sending these to the provider host.

– store usage: The collected usage data, including the time of usage or the valued price, have to be stored into the accounting system of the service provider.

– visualise usage: A customer should be able to view his or her own usage statistics, including prices. Standard Internet facilities can be used.

– get price: The price of a commodity within a given period has to be retrieved.

– prepare bill: On the basis of usage and prices a bill has to be prepared for a user, either on request by the user or periodically by the service provider.

– send out bill: The prepared bill has to be sent out to the customer. This can be done electronically or traditionally by postal service.

– pay bill: The user should pay the received bill.

– detect failure: The residential gateway can detect any meter deviation and notify either the service provider or the gateway keeper.

– analyse usage: The usage patterns may be analysed in order to detect any misbehaviour or to inform and advise the customer or to compare usage with standard patterns, etc.

When we think about the remote metering service operational and maintenance tasks must not be forgotten. We must be able to add or remove customers, meters and appliances, detect meter failure, handle operational errors, synchronise the service system, etc. Also we must be able to set commodity price levels or cost rises.

Step 4: deployment of tasks

A typical way of deployment of the metering service is to provide the homes with a local access node, e.g. residential gateway. Information is then gathered by the utility by addressing this access node for each customer through last mile access, using the base station as an intermediate. The access node takes care of the in-home information using the home network.

The access node and the base station are used to buffer usage data. For visualisation of these usage data information will be gathered not only from the service provider (where the history data are archived), but also from the access node or the base station, since the most recent usage data will not always be known at the service provider.

In figure 3.2 the AMR schedule and deployment of tasks is outlined.

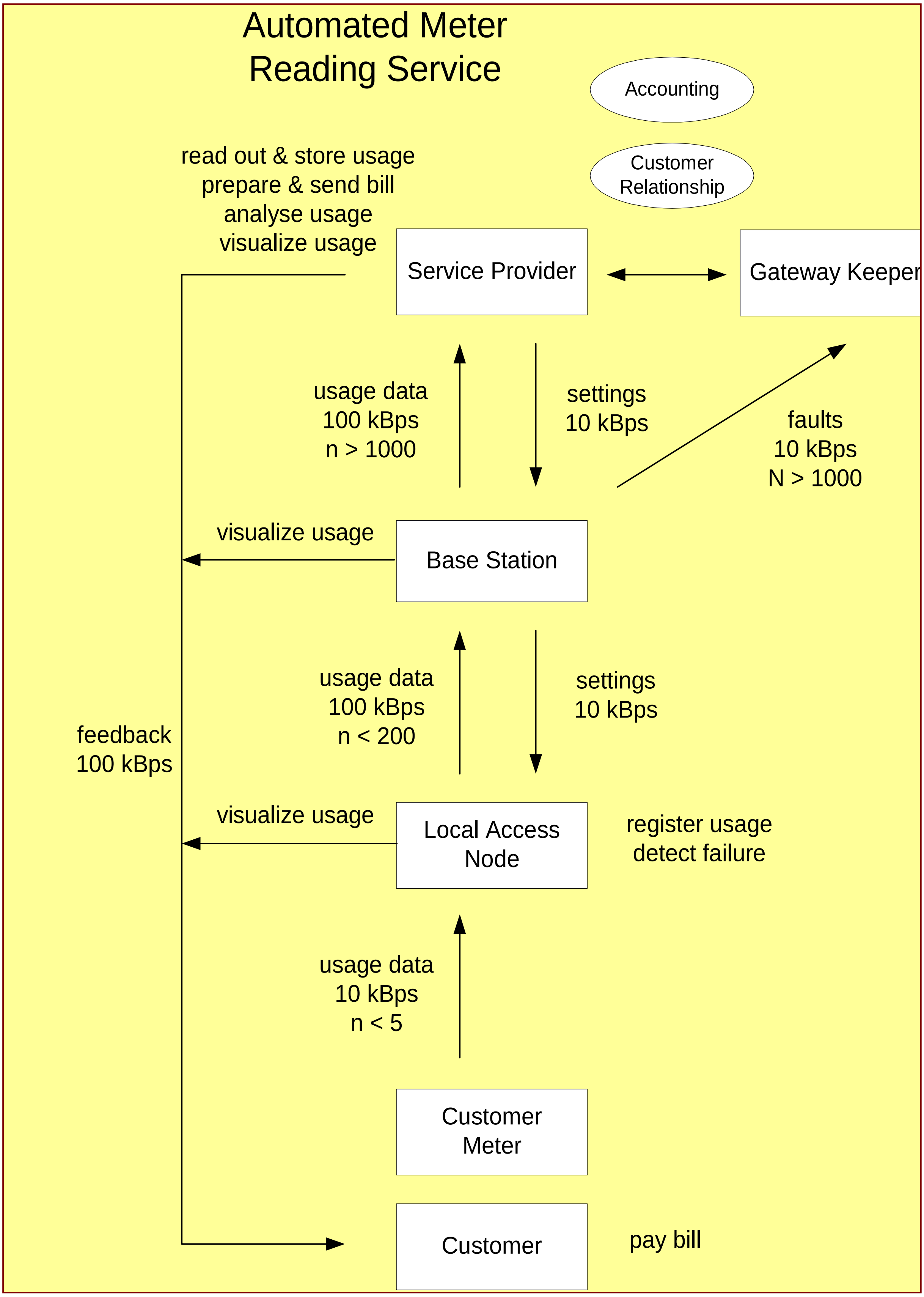


Figure 3.2. Powerline Automatic Meter Reading Schedule

3.3.2.2 Expansion to Monitoring and Control

Meter reading as described above is concerned with meters and usage figures, needed for billing. We can also see meter reading as a special case of monitoring and control, in which a device can exchange any kind of information with the outside world and its state influenced by the system.
We then arrive at the following object generalisation:

- Device — A device delivers information the user or utility is interested in. A device has embedded intelligence. Communication ability enables the device to share information with the outside world. This outside world can also influence / control the state / operation of the device.

Devices are not strictly reserved to in-home usage, but might also be placed at outer sources (e.g. current regional climate data, Internet based market data). Contrary to in-home devices, which are usually owned by a customer, these devices share information among a group of customers. Therefore we introduce the SharedDevice class:

- SharedDevice — A shared device is defined as a device, which is shared by more customers.

Devices, by their embedded intelligence, can be seen as services in the sense of Jini / OSGi. A meter is a service, which delivers usage data on request; a device also delivers e.g. state or value-based information and will listen to commands to control it. Together these services can form a service bundle for monitoring and control.

In the next section an example has been worked out of a comfort management system. This example has been taken from the TRAFO project [8].

3.3.2.3 UML Object Diagram

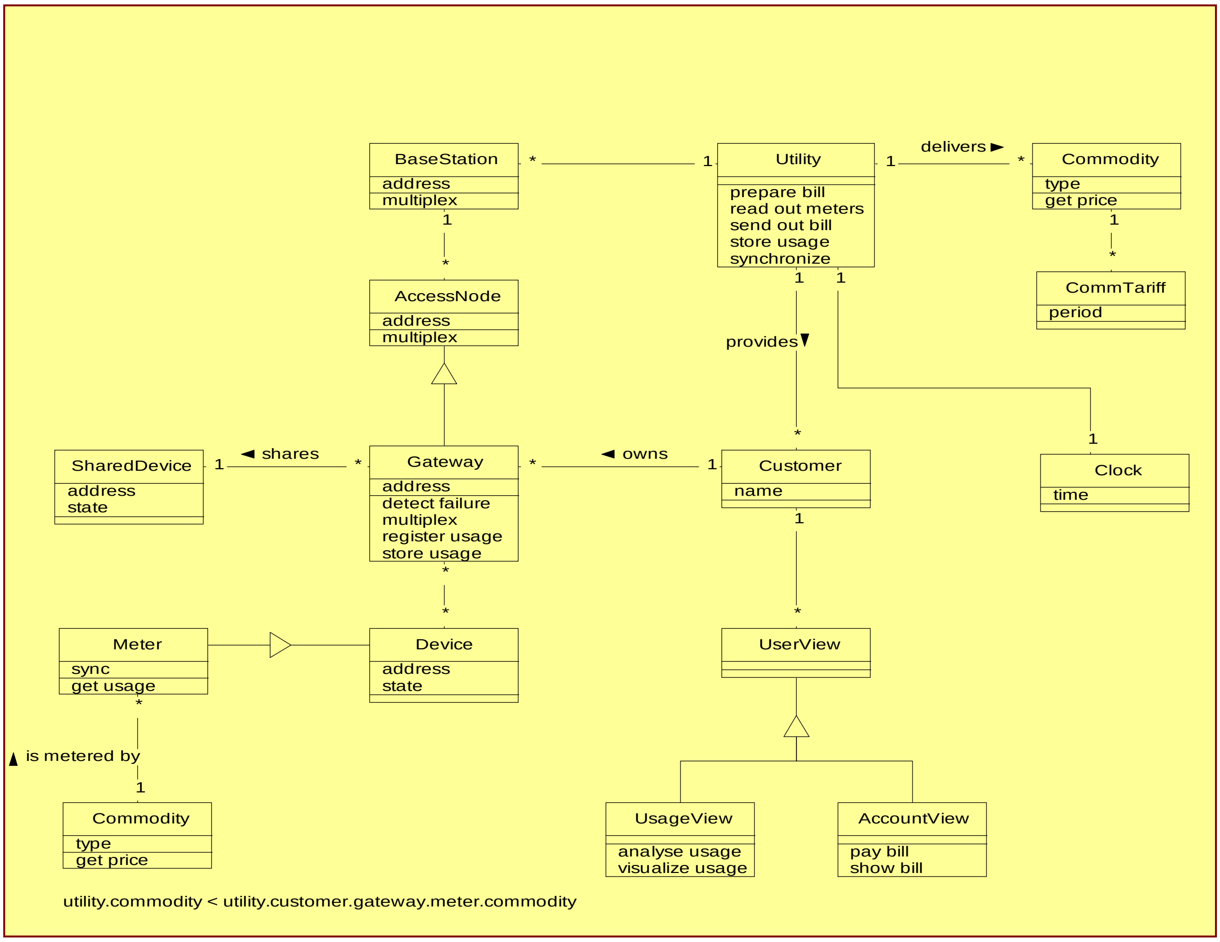


Figure 3.3. UML Object Diagram Automatic Meter Reading

Although Automatic Meter Reading seemingly is a simple application, a lot of stress is laid upon the communication channels and the network nodes. System failure at any level should not lead to loss of data. Therefore metering information objects have to be persistent by data replication at all levels in the infrastructure. This complicates software maintenance.
Since timing is essential in AMR (especially since commodity pricing is time dependent) synchronisation also is essential on the network.
In a control network less emphasis is laid on communication. Local intelligence is important, and therefore more processor capacity and larger memory requirements are required. Since control networks often need a lot of information, the control granularity of the (local) network becomes finer.

3.3.2.4 Concerted actions in an Automation and Control Network

In a control network we have different types of devices working together to perform a defined task. The control network is based on the following procedure:

- collecting information from a number of sources,
- making any decision based on this information,
- communicate the decision to the involved devices and sources.

This procedure can have a semi-continuous character, in which the control is continuous over time, or it can be a single decision in time, potentially followed by some action elsewhere in the network. Also control can function as an autonomous process or it can be triggered by an event in the network.

Part of these control systems can function locally. Added value for powerline services is the connection to the outside world. A main issue for a good software architecture is the distribution of the control logic in the network.

Examples of such control networks are:

- Presence detection: a sensor can be used to registrate the event. The control system acts by e.g. switching on the lights or turning on the heat control.
  Presence detection can function purely in-house. If it is extended to security, a warning can be issued to the outside world, e.g. the security service provider.
- Performance monitoring: power quality control, outage detection, appliance diagnostics, etc. The control system monitors the network for any special event. On occurrence either an action can be taken or a warning can be issued at some level in the network.
  Performance monitoring can function mainly in-house. In case of an event it can transcend to the service provider.
- Heat control: on the basis of presence in rooms, expected climate conditions (temperature, wind, sunshine) and even expected energy price a heat control system can continuously optimise the indoor climate in a room or building.
  Comfort management can function in-house or near-house. However, information on fluctuation of energy prices, which will be retrieved form the outside market, enhance the value of the system since it can be used to cut costs.
- Load management: given a number of tasks a load control system can distribute these tasks over time such that within certain constraints peak energy usage can be avoided.
  Load management in-house does not add much value. Peak avoidance might be a main topic for the utility if cumulative effects over a large number of households can be taken into account.

3.3.2.5 Use Case: Comfort Management

Step 1: Description of the use case

The following example is taken from [8]. In this paper a complete comfort control system has been worked out. A modification of this system is discussed elsewhere in this document. At this level we are not concerned with the logics of the control itself, but we will concentrate on the main tasks in the control network.

Comfort management can be described as follows:

A customer can *define* personal profile for thermal comfort and air quality in the rooms of his/her house, based on expected presence, type of activity, etc.
The comfort management system *controls* a number of devices, which can be used to influence the indoor climate. These or other devices can also be used to *measure* the different aspects of the indoor climate and the environment, which influences the indoor climate.
Each setting of a control device during a certain period in time requires an amount of energy, which can be obtained from local sources (e.g. solar power, solar radiation, heat buffer) or has to be obtained from an energy market. Energy usage from the market is coupled with a market price for each period.
The task of the comfort management system is to provide in each room comfort within the customer-set ranges at a minimum price from the customer point of view. In this task information is used from expected market prices and weather forecasts.

Step 2: Identifying objects and attributes

From the description we can derive the following objects in comfort management:

– Customer	The communication provider.
– Customer profile	Time-dependent settings of required comfort by the user, given either as a time-series of values or as a time-series of ranges on comfort aspects.
– Indoor climate	A possibly multi-dimensional vector giving the quality of the comfort in a room. Thermal comfort and air quality are both aspects of the indoor climate.
– Room / House	We have to model the customer premise in which each room might have different characteristics and different possibilities for comfort control.
– Device	In the above description we have two types of devices: control devices can be instructed to perform a task; measurement devices give information needed by the control system.
– Period	
– Energy source	Different energy sources can be used to actuate different control devices. Energy sources can have limited capacity and time-dependent pricing.
– Market	The market is the place where energy is supplied. In many cases the utility serves as the market place for the customer. Also the power exchange might serve as the energy market.
– Market price	At the energy market energy will be supplied at a price per period principle. Note that the energy market can be considered at the same level as a device: on request the market can deliver information about price.

Step 3: Identifying operations

The following operations can be identified in the above description of comfort management:

– set user profile	the user can define a required comfort profile and permitted deviations over a period in time (typically 24 hours).
– get value	from a number of in-house and shared devices information on current climate must be gathered which serves as input for the comfort control:
– control comfort	given the current settings of a number of comfort devices and the expected market energy prices and the weather forecasts a scenario is determined to deliver comfort within given profile bounds at optimum (minimal) cost.
– set device	on the basis of the output of the comfort control the setting of each comfort influencing device must be adapted.
– get prices	expected market prices for energy over the decision period (24 hours) must be retrieved.
– get forecast	expected climate conditions over the decision period (24 hours) must be retrieved.

Step 4: Deployment of tasks
Comfort management is a service, which is focussed on one house. Most of the information needed, can be obtained from the house or its direct environment. Optimisation by price requests pricing information from the market to be available.

The logical level for comfort control therefore is the access node to the home. A weather 'station' can be set up at a base station to obtain climate information. The connection to the outside world is used to obtain weather forecasts and market price info. If prices are fixed by contract with a utility these prices might be made available at the access node.

A detailed design is described in [8]. Characteristic in this design is the high level of control needed at a low level in the infrastructure, the Local Access Node. Therefore the access node requires larger processor capacity and memory. Communication to the service provider is less essential. Most information is gathered locally in the house or near the house. Pricing information and weather forecasts have to be retrieved from a central level. In the comfort management system example multiple autonomous agents achieve this.

3.4 Architectural implementation issues.

3.4.1 Partitioning of functionality

In the previous chapter a number of application types enabled by last-mile access were modelled. Next step is deriving a partitioning scheme for the objects discussed there. Classically, in the hierarchical model three partitions are to be discriminated:

- Peripheral equipment. Sensor, actuator.
- Communication. LAN, MAN, WAN.
- Data processing

When viewing in this way, the largest number of interfaces is generated in communication. A customer communication interface has to interact with the power line carrier, the local, metropolitan and wide area network, the utility side telecommunications interface and the data processing packages through several layers. In order to enable two-side communication of control signals, very complex mechanisms have to be designed. Therefore delegation of control and distribution tasks to suitable concentrator subsystems in the chain are necessary.

Apart from the large number of different interests of parties involved in realising applications and the lack of standardisation to “open” hard- and software and the cost-consequences thereof, these issues form the largest problem for realising applications. Current developments in software and hardware technology lead to an increasing processing power and bandwidth for small scale, powerful dedicated systems. In this respect there is less a need for **big** bandwidth, but for **smart** bandwidth. The last statement stresses the fact that, for most powerline applications, efficient and reliable usage of limited bandwidth between powerful processors is more important than a high bandwidth between “dumb” processors. Important in this respect is the point at which data is converted into information and control directives.

3.4.2 Required management and configuration facilities

3.4.2.1 Object persistence

In current software design methods nowadays, data (attributes) and procedures and behavioural aspects are contained in objects. Depending on the functionality of applications and for security reasons, the attributes of objects and the state information have to be saved at the proper level. In case of outage of the application or system, procedures have to be designed to restore the object’s state to the one before the crash.

For a “simple application” as meter reading this already poses problems. A crash can occur during read-out using the sensors, during transfer to the gateway or during operations further on to the legacy, utility accounting system. How higher hierarchically the metering “intelligence” is in the network, the more complicated is the restoring algorithm. Therefore metering information objects should be persistent at the lowest hierarchical level. From the software maintenance point of view, metering applications should be as much as possible thin-client. Embedded processors should have high MTBF-figures and should be able to store status information with high reliability. Metering read-out should be preferably realised using "servlets" wrapped to legacy metering and billing applications and databases.

In more complex control applications like multi-parameter optimising comfort management, more complex software is implemented on larger processor capacities. Memory requirements will be larger. Information contained in objects can be downloaded at regular intervals from a larger computer system in the network. Again the emphasis in reliability and computing power is on the RG. Loosely coupled feedback (analysis of measured data and adaptation of control parameters) on the performance of the algorithms from larger computer resources higher in the network hierarchy will be a major benefit but not essential prerequisite. Sustained operation in the absence of larger computing systems is essential.

For the other application types like P2P, IP and EMA the emphasis for secured object persistence clearly is on the server side. The control complexity as can be derived from the UMLK-models is lower.

3.4.2.2 Replication mechanisms, serialisation and versioning

If object information is needed at other levels in the application hierarchy, data and state information have to be transferred and replicated across the network. For successful execution of remote procedure calls between objects residing at different nodes of the network, a serialisation mechanism transfers all necessary information to the correct nodes. A number of standards facilitate this mechanism. Currently COM and CORBA are the mainstream standards for remote procedure calls, but the needed resources are heavy and are geared to larger processor-capacities than present in current small-scale apparatus. In SOAP and DOM, a WWW based document object model, a connection to these standards is made also incorporating XML. In Java, a mechanism for serialisation using the RMI standard is also contained. This technology however, especially concerning real-time Java, is in the development phase and has a lack of support of the leaders in the field of real-time software development. Furthermore, firewalls pose large difficulties for transparent Java-RMI.

From a software maintenance point of view, the home network and gateway appliances will have a long usage cycle. Maintenance of hardware and software versioning therefore is important for proper operation of applications. Current software standards have a versioning mechanism to check if objects interact with each other are generated with the same compiler products and support libraries. This mechanism will be mandatory for DCMS applications.

3.4.2.3 Multi-agent architectures

Applications, in which a large number of information-sources have to be monitored or inspected, benefit from multi-agent architectures and technology. The scope of applications for agents therefore is geared to surveillance systems, scheduling of events and meetings in residential areas and for calendar applications. Agents have the advantage of an auto-replication mechanism. Furthermore, they are autonomous in bargaining with other agents for a resource using market algorithms [9]. In this way optimisation problems are easier to solve in a computer network environment. Especially for large distributed systems the agent abstraction has its benefits. The control behaviour of these systems is hardly codeable in conventional ways because the large number of possible states of the system as a whole. The optimisation of the system operating with distributed processors in a concerted way, e.g. for energy management, is hardly solvable by central, analytical algorithms and procedures as well. Market and auction algorithms implemented in a large number of autonomous processes in a distributed processor network are the only way for successful implementation.

3.4.2.4 Control timing implementation and synchronisation

A large layered control-network has delicate timing. Frequent time synchronisation and propagation mechanisms are necessary to guarantee secure parallel operation. Again, the

DCMS-type of applications is most important in this respect. For the other types of applications the responsibility can be laid on the server side.

3.5 Conclusions

In view of the above architecture requirements the perspective of single residential gateway apparatus covering the four application types discussed above is difficult to imagine. Distributed control and management applications do stand too far apart in architecture requirements. The prime combination potential in gateway functionality combination lies in telephony and Internet application types especially for packet-switched telephony and in telephony and multimedia applications, if the DSL-bandwidth can be extended to support real-time video. However, the three order of magnitude difference between telecommunication company and cable communication fares remains a large barrier. Therefore, hybrid solutions, in which several gateway apparatus are involved in combinations, are most likely. Integration in the application then is on a higher level on the network application software. This conclusion follows from the software architecture perspective, where only reduction of the complexity of interfaces and control behaviour and transparency counts. Many stakeholders being involved, a multi-gateway, multi-service software architecture, from a business point of view, imposes a constraint on the business modelling process.

The possibility of attributing a role to the transformer station hard- and software as a residential **area** gateway is an important difference compared to other last-mile access technologies. In case of control applications, if such a gateway could replicate the functionality to a large number of small size, cheap DCMS-gateways in dwellings a definite advantage can be imagined. To a lesser extent this also holds for IP and telephony applications. However, the number of standards and software components for residential area gateways is scarce.

	DCMS	**Point-to-point**	**Internet**	**Multimedia (3rd generation Set-top box)**
Preferred messaging	Client/thin server, asynchronous	Peer-to-peer, synchronous	Thin client/server, asynchronous	Thin client/server, synchronous
Software deployment, maintenance	PROM, Download of applications	PROM	Download	Once
Local Processing power	High	Low	Medium	Low
Local Memory	Low	Low	High	Low
Central processing power	Low	Medium	High	Medium
Control complexity	Large	Medium	Medium	Low
Autonomous Intelligence	High	Low	Medium	Low
Upstream bandwidth	Low	Moderate	Moderate	Low
Downstream bandwidth	Low	Moderate	High	Very High
Response time	Low	High	Moderate	High
Always-on connection of communication channel	Low	Low	High	Moderate

Table 3.1. Comparison of application type attributes.

Table 3.1 gives an indication of the importance of certain aspects of application types on residential gateways discussed before.

DCMS need to be very reliable and should perform autonomously. Control is fine-grained, exception handling is very delicate and processing power requirements need to be laid out on handling complex transition state logic and real-time handling of combinations of events. Most of the application logic resides on a locally processing RG, whose bandwidth requirements are very limited.
Point-to-point communications are based on synchronous peer-to-peer communication. Processing power requirements are limited; real-time software with dedicated DSP's (Digital Signal Processors) will take care of coding and decoding.
Internet traffic uses its processing power mainly for dynamic graphical visualisation and for bulk data transfer. Control is coarse-grained and exception handling not complicated.
For broadband, digital multimedia applications third generation set-top boxes with local intelligence have very limited, but very fast local processing power in FPGA's or ASIC's.
In an architectural sense imagining an RG handling all four kinds of applications cannot be foreseen. The diverse requirements for residential gateway technology will most likely lead parallel application driven implementation of several gateways. As stated earlier, hybrid solutions, from the resemblance in requirements, are only between Internet applications and DCMS-applications.

3.6 References

[1] Coactive Networks - *The business case for residential gateway deployments - delivering a new world of Internet services*. Coactive Networks Brochure, 2000 - http://www.coactive.com/media/busmodbro.pdf.

[2] Enikia - *The Information Economy Derivative Markets Model: A Technology Value Chain for the Digital Economy* - http://www.enikia.com/download/iedmm.pdf.

[3] James Rumbaugh, Grady Booch, Ivar Jacobson - *Unified Modeling Language Reference Manual*. Addison-Wesley, 1998.

[4] Ivar Jacobson, Grady Booch, James Rumbaugh - *The Objectory Software Development Process*. Addison-Wesley, 1998.

[5] Grady Booch, James Rumbaugh, Ivar Jacobson - *Unified Modeling Language User Guide*. Addison-Wesley, 1998.

[6] Robert Orfali, Dan Harkey, Jeri Edwards - *The Essential Distributed Objects Survival Guide*. John Wiley & Sons, 1996.

[7] Ron Ben-Natan - *CORBA, a Guide to Common Object Request Broker Architecture*. McGraw-Hill, 1995.

[8] Boertjes, E. *COMFY, Comfort Management System, design document*. Free University Amsterdam, December 1999.

[9] P. Novais, L. Brito, J. Neves. *Agreement in Virtual Marketplaces with CBR supported negotiation*. Proceedings of the Fifth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology. Manchester, 2000.

4. THE COMFY COMFORT MANAGEMENT SYSTEM

4.1 Introduction

In 1999 a first prototype of an agent-based system for comfort management was developed. The system, called COMFY, aims at the optimisation of comfort in a residential building based on given user preferences during the day on both comfort and the costs the user is willing to pay for it. The prototype has been developed in the work described below. With this demo prototype a number of scenarios have been studied in order to get a good impression of the potentials of the COMFY comfort management model for energy efficiency.

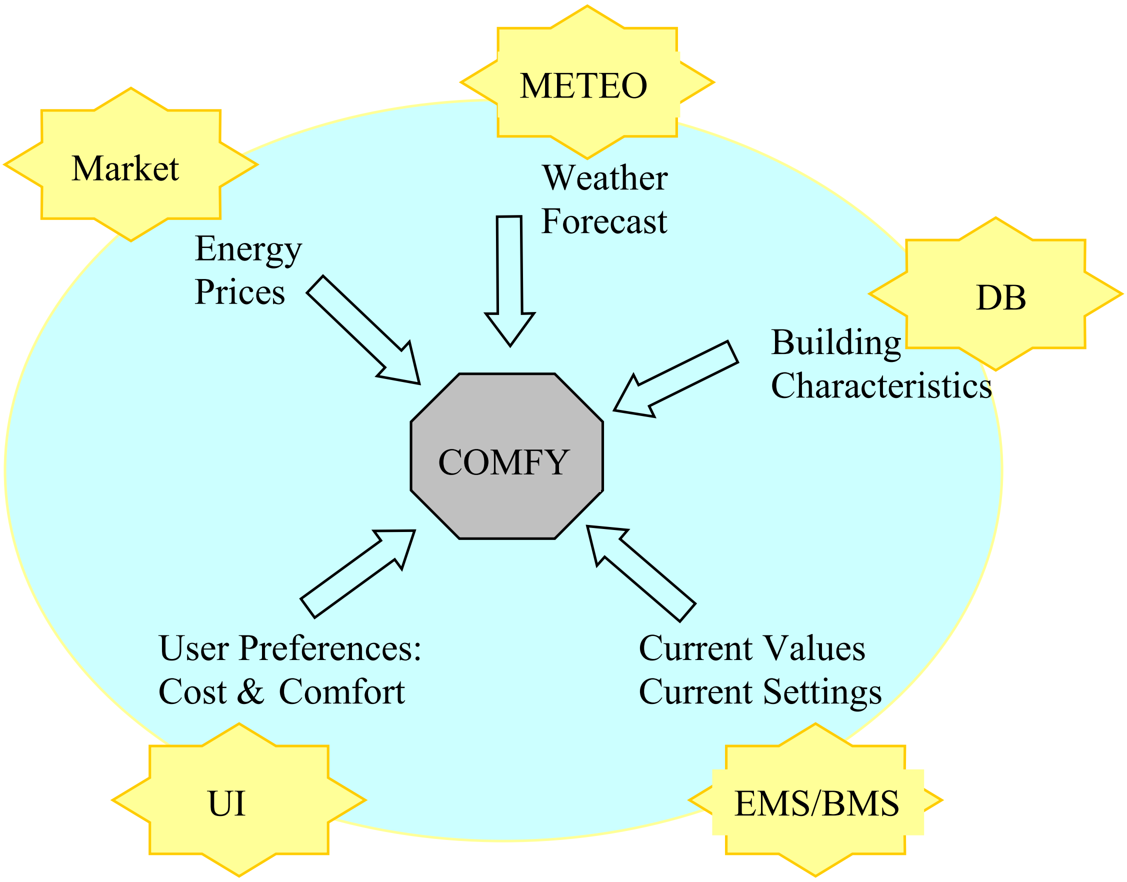


Figure 4.1. Comfy Context

Figure 4.1 shows the COMFY environment. The Internet network, with XML-structured portals, makes it possible to retrieve any information from any source and to use this information for better processing. In COMFY comfort is managed while making use of electricity prices from a liberalised energy market and weather forecasts from a meteo institute. With known building characteristics COMFY then can search for an optimised strategy to maintain a given comfort level. Knowledge of energy prices allows COMFY to adapt to price fluctuations by pre-heating a room or to postpone the heating. Weather forecasts can be used to calculate/predict the comfort several hours ahead. Knowledge of sunshine can be used to open the sunblinds for passive solar heating in the winter or to close the sunblinds in order to keep the room cool in the summer.
The user can define his or her setpoints for comfort and can set a cost factor which indicates the cost that a user wants to attach to this comfort. It will be clear that a poor student will accept larger deviations from his comfort setpoints if it saves him money. A wealthy person will put more stress on keeping a preferred comfort level than on saving money.

COMFY optimizes the comfort in a room taking into account both the comfort setpoints and the energy cost that is attended with it. As we will see, this also will lead to improved energy efficiency.

4.2 Comfort Management in COMFY-2

4.2.1 Thermal Comfort

Comfort, as we have used the term above, has a rather qualitative meaning which is difficult to quantify. In COMFY we have concentrated on the thermal sensation of comfort. Other aspects of comfort such as air quality, light and sound are not considered in COMFY. Thermal comfort is defined in the international ISO 7730 standard as being "That condition of mind which expresses satisfaction with the thermal environment". Hence thermal comfort is characterised as the personel experience of well-being based on environmental elements such as temperature, humidity and air velocity.
The ISO 7730 standard has adopted the Fanger thermal comfort model [4], [5], [6], which takes into account all aspects which affect thermal comfort:

- air temperature in the room,
- radiation temperature,
- humidity,
- air velocity,
- metabolism, i.e. the activity level and well-being of the user,
- level of clothing of the user.

The Fanger model leads to a comfort index on a scale from -3 (cold) via 0 (neutral) to +3 (hot). This comfort index is called the 'Predicted Mean Vote' (PMV). In addition a 'Predicted Percentage Dissatisfied' (PPD) can be derived from the PMV. The comfort index has been established experimentally such that PMV = 0 corresponds with PPD = 5% and PMV = -3 / +3 corresponds with PPD =95%.

4.2.2 COMFY Control

As can be seen in figure 4.1, COMFY uses a number of external resources to obtain information, which can be used to make decisions

- Meteorological forecasts are used to anticipate on expected weather conditions.
- Energy prices from a central energy market detemine the cost of comfort.
- Building and installation characteristics are needed to calculate the effects of comfort control.

- Momentary measurements are used to verify the calculated effects and to adjust for differences in calculated and actual comfort.

The following user preferences can be defined:
- Comfort setpoints are expressed as a series of PMV values, for each period one value. In the following paragraphs and chapters we will use the term 'preferred comfort' in stead of comfort setpoints.
- Allowed deviations on the PMV values can be set, ranging between 0.5 (strict settings) and 2 (large deviations allowed).
- A cost factor can be set which determines the weighing of the deviations. The cost factor ranges between +1 (luxurious scenario) and -1 (economical scenario).

The following user profiles can be set:
- Clothing level for each period.
- Metabolism m-rate, i.e. the rate of transformation of chemical energy into heat and mechanical work by aerobic and anaerobic activities within the body.
- Metabolism w-rate, i.e. the mechanical work produced by the human body (usually W = 0).

In the COMFY demonstration program the user, by means of a user interface, can define all this external information. The user interface is described in the next paragraph. Note that especially the building model and the thermo-physical calculations in COMFY are based on simple premises, which will be improved in later versions of COMFY. See also paragraph 6.3.

COMFY uses the information to calculate an optimal user comfort against minimal cost. In a utility function a trade-off is being made between comfort and cost, based on the user preferences. The COMFY output consists of four series of optimal control settings for each period: heating (based on electrical heating), sunblinds (on behalf of solar radiation), ventilation ant the ventilation grid.

4.2.3 User Interface

The COMFY demonstration program enables the user to define a large number of parameters and time vectors which influence the comfort sensation in a room. A complete description of the user interface is given in the Appendix. In this paragraph the parameters and time vectors are defined.

COMFY is based on calculations on a consecutive number of periods. The default timescale (plan horizon) is 24 hours (from 00:00 h. to 24:00 h), using periods of 1 hour, which can be changed into 15 minutes (plan horizon 96 quarters). Thus, starting at 0:00 hours, for 24 resp. 96 periods ahead the settings of the control parameters are calculated, until 24:00 hours. After that COMFY moves one period (1 hour or 15 minutes) ahead and restarts calculation on a timescale from period 2 to period 'plan_horizon' + 1. The total number of recalculations (default 1) is given by the parameter 'n_of_plan_periods'.
All time vectors need to be given for 'plan_horizon' + 'n_of_plan_periods' periods.

Program Control Parameters	
plan_horizon	total number of periods COMFY is looking ahead in one calculation. Default plan_horizon = 24
n_of_plan_periods	Number of plan periods for which recalculation is made. Default n_of_plan_periods = 1
PeriodLength	the number of seconds in a period. Default PeriodLength = 3600 (i.e. one hour).

The next step is to define the characteristics of the room and of the installation details (heating, ventilation and sunblinds).

Room Characteristics	
RoomLabel	The name by which the room is identified
DampingFactor	Damping factor of the building, as defined in [1], G1/43.
RoomVolume	Total volume of the room, in m^3
HeatingCapacityAir	Heating capacity of the room, in kJ/°C (follows from room volume)
HeatingCapacityWalls	Heating capacity of the walls of the room, in kJ/°C
WindowArea	Total area of the windows, in m^2 - it is assumed that the room has only windows on one side.
WindowOrientation	Orientation of the room window, in degrees (0-360), based on the convention North = 0, South = 180

Heater Characteristics	
HeaterCapacity	Total maximum capacity of the (electrical) heating, in W
SwitchingEnergyHeating	This is the total energy needed for switching the heater on or off, from 0 to 1 or backwards, in W

Ventilation Characteristics	
VentilatorCapacity	Capacity of the ventilator in the room, i.e. the maximum air velocity that the ventilator is able to realize, in m/s
VentilationPower	Power of the ventilator in the room, in W
VentilationGridOpening	Maximum ventilation grid opening, in m^2
SwitchingEnergyVentgrid	This is the total energy needed for switching the ventilation grid open or closed, from 0 to 1 or backward, in W
SwitchingEnergyVentilator	This is the total energy needed for switching the ventilator on or off, from 0 to 1 or backward, in W

Sunblind Characteristics	
SunblindShieldingFactor	Maximum shadow factor of the sunblinds, depending on the sunblind's material (see [1], table G1/3.2); normally between 0.5 and 0.7 (50% - 70%)
SwitchingEnergySunblind	This is the total energy needed for switching the sunblinds open or closed, from 0 to 1 or backward, in W

COMFY also needs initial values for four comfort variables: room temperature, radiation temperature, humidity and air velocity, all at the start of the first period.

Inside Climate Parameters	
InitialAirTemp	Initial air temperature in the room, in °C
InitialRadiantTemp	Initial radiation temperature in the room, in °C
InitialHumidity	Initial humidity in the room, in %
InitialAirVelocity	Initial air velocity in the room, in m/s

The comfort control takes advantage of the sun for solar radiation. To make use of it COMFY needs sun characteristics: sunrise, sunset and the solar power. These data can be derived from the day of the year and the geographical location (and the orientation of the building; see room characteristics). However, COMFY uses straightforward information. For simplicity reasons the solar power is taken as a constant over the day.

Environment Parameters	
SunUp	Time of sunrise, given as a decimal number between 0 and 24 hours
SunDown	Time of sunset, given as a decimal number between 0 and 24 hours
QZenith	Solar power, on a plane perpendicular to the sun beams, W/m^2

Besides solar characteristics other meteorological data are needed. COMFY only makes use of air temperature and humidity. Both are given as a timevector for each period. Other information, such as wind direction and windspeed and cloudiness are not yet considered.

Climate Forecast Timevectors	
TemperatureProfile	array with predicted outside temperature for each period, in °C
SolarRadiationProfile	array with predicted solar radiation reaching the window, for each period, in W/m^2; if this profile is not given, the Qzenith will be used, corrected for SunUp and SunDown and the orientation of the sun with respect to the window
HumidityProfile	array with predicted outside humidity for each period, in %

Cost is important control information. Therefore the electricity price for each period must be known. COMFY considers fixed electricity prices for each period. Note that COMFY only models electricity.

Market Timevectors	
ElectricityPricesProfile	array with valid electricity prices for each hour, in ct/kWh

Also the user preferences and user profiles are important control information. This information also is given for each period.

User Preference and Profile Timevectors - to be set by the user	
MetabolismMProfile	array with metabolism M-rate (activity level) for each hour, as given by the user, according to Fanger scale
MetabolismWProfile	array with metabolism W-rate (mechanical work of the human body) for each hour, as given by the user, according to Fanger scale
ClothingProfile	array with cloting indication for each hour, as given by the user, according to Fanger scale
ComfortProfile	array with preferred comfort level as given by the user for each hour, as PMV-values
CostProfile	array with weighing factors indicating the importance the user attaches to the given comfort level
CCDeviationProfile	array with allowed deviations from the preferred comfort; if the deviation exceeds the deviation profile, the utility function becomes 'infinite'

In order to be able to test the following parameters are built in the program:

Program Control Parameters	
EnergyUseAwareness	Energy_use_awareness
IOLevel	iolevel, used to control the level of program output

COMFY uses a reference scenario for comfort management, based on a simple thermostate control for heating and fixed settings per period for the sunblinds and the ventilation. This reference scenario can be used to get an indication of the possible savings using optimal control. The following timevectors are defined for the reference scenario:

Reference Setting Timevectors	
InitialTemperatureSettingProfile	array with thermostat settings for the room air temperature in the initial reference situation, in °C
InitialHeaterProfile	array with heater switch settings in the initial reference situation, between 0 and 1
InitialSunBlindProfile	array with heater switch settings in the initial reference situation, between 0 and 1
InitialVentilatorProfile	array with heater switch settings in the initial reference situation, between 0 and 1
InitialVentilationGridProfile	array with heater switch settings in the initial reference situation, between 0 and 1

4.2.4 Used Formulas for Thermophysics

This section explains how to calculate or estimate each of the comfort aspects. It is taken from [2]. The formulated thermophysics calculations are very basic. In a later chapter we give some directions for improvement.

4.2.4.1 Calculation of Air Temperature

We assume that the air temperature *air_temp*$_i$ in a room in a period *i* depends on:

- air temperature in previous interval *i-1*
- outside temperature in period *i* (measured or estimated)
- the opening of the ventilation grids
- the heating power
- the solar heating
- the thermal capacity of the room

$$air\_temp_i = air\_temp_{i-1} + T_{vent\_grid} + \frac{Q_{heating} + Q_{sun}}{C} \cdot \Delta t \qquad (4.1)$$

T_{vent_grid} is the contribution to the inside temperature caused by the difference in in- and outside temperature.

$$T_{vent\_grid} = z \cdot (T_{outside,i} - T_{i-1}) \qquad (4.2)$$

Where z is a correction factor depending on the opening of the ventilation grids and the volume of the room. By opening and closing of the ventilation grids, the contribution of the outside temperature can be adjusted. For simplicity reasons we take z = 1 if the ventilation grids are completely opened and z=0 if they are completely closed:

$$z = s_{vent\_grids} \qquad (4.3)$$

Note that *z* should be dependent on the windspeed and wind direction. This is not modeled in COMFY.

We assume the heating power to be linear dependent of the heating setting.

$$Q_{heating} = s_{heating} \cdot cap_{\max,heating} \quad (4.4)$$

The calculation of the solar power Q_{sun} is explained in the next section.

The heating capacity C of a room is the amount of energy needed to increase the air temperature with 1 degree (Kelvin or Celsius). C accounts for room volume, windows, etc.

The variable Δt is the length of the interval between the comfort management cycles (e.g. 1 hour).

4.2.4.2 Calculation of Radiant Temperature

From [1] we know that the solar energy entering a room can be calculated by:

$$Q_{sun} = A \cdot b \cdot I \cdot d \quad (4.5)$$

with
- A the window area [m^2]
- b the shadowfactor
- I entering solar power through normal glass
- d the building's damping factor ([1] table 3.4)

Factor I depends on the orientation of the sun in relation to the window's orientation:

$$I = b_{angle} \cdot q_{ze} \quad (4.6)$$

with

q_{ze} the solar power on a plane perpendicular to the sun beams (we take q_{ze} = 1000 W.m^2 [Cok&Wisselink]

$$b_{angle} = \cos(\alpha) \quad (4.7)$$

where α is the angle between sun and window

$$\alpha = \alpha_{window} - \alpha_{sun}(time) \quad (4.8)$$

with the convention that α = 0 corresponds with the north, α = 180 corresponds with the south. In calculating the angle of the sun, we assume that α = 90 at six in the morning, α = 180 at noon, and α = 270 at six in the evening.

The shadow factor depends on the shadow factor of the sunblinds and the setting of the sunblinds. We take:

$$b = s_{sunblind} \cdot b_{sunblind} \quad (4.9)$$

with $b_{sunblind}$ between 0.5 and 0.7, depending on the sunblind's material (see [1], tabel G1/3.2)

Note that these calculations are very basic. Especially the calculation on solar power should be based on a better model for expected solar radiation through the window..

From the solar power we calculate the radiant temperature. We assume that the radiant temperature is caused by the temperature of the wall and the sun light reflected by the wall. It takes some time for the wall to adjust to the air temperature. To account for this delay, we take the wall temperature equal to the air temperature in the previous period. We then get for the radiant temperature:

$$rad\_temp_i = air\_temp_{i-1} + \frac{Q_{sun} \cdot \Delta t}{C_{wall}} \quad (4.10)$$

Where C_{wall} is the heating capacity of the wall. The air temperature is calculated by equation (4.1). Variable Δt is the length of the period (e.g. 1 hour, Δt=3600).

4.2.4.3 Calculation of Humidity

We assume that the humidity in a room in period *i* depends on:
- the humidity in period *i-1*
- the outside humidity in period *i*
- the opening of the ventilation grids

We take the following very simple approach:

$$hum_i = hum_{i-1} + H \qquad (4.11)$$

with $$H = s_{vent\_grids} \cdot (hum_{i,outside} - hum_{i-1}) \qquad (4.12)$$

A more advanced calculation could take into account the <verhouding> between the opening of the ventilation grids and the volume of the room:

$$H = z \cdot \frac{s_{vent\_grids} \cdot A_{\max}}{V_{room}} (hum_{i,outside} - hum_{i-1}) \qquad (4.12b)$$

where A_{max} is the maximal possible opening of the ventilation grids in a room and V_{room} is the room's volume.

4.2.4.4 Calculation of Air Velocity

We assume that the air velocity is completely dependent of the current setting of the ventilator(s):

$$air\_vel_i = s_{ventilator} \cdot cap_{\max} \qquad (4.13)$$

where

$s_{ventilator}$ is the setting of the ventilator(s) in a room
cap_{max} is the mean of the values of the maximum air velocity that each ventilator is able to realize.

The calculations for both the humidity and the air velocity are based on simple models. These models need improvement in a next generation of COMFY .

4.2.4.5 Estimation of Person's Metabolism

When looking at the example activities, the metabolism is very likely to follow a rather fixed pattern, as these activities are often performed at a fixed period of the day. A good basis for estimation is therefore the period of the day. (Morning, afternoon, etc.).

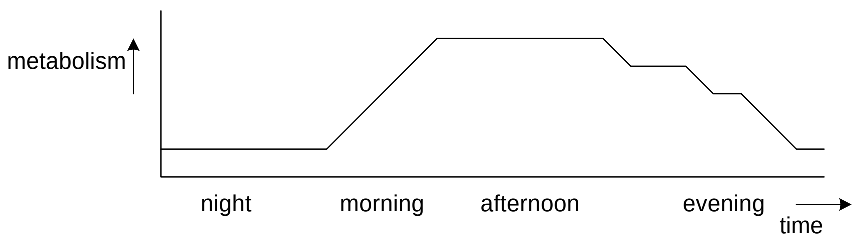


Figure 4.2. Example of typical pattern of metabolism

Activity	Metabolism	Examples of in-house activities
In rest, laying	0.8	(sleeping)
In rest, sitting	1.0	(watching television, sleeping)
Light activity, sitting	1.2	(embroidery, drinking coffee, eating)
Light activity, standing	1.6	(cooking)
Medium activity, standing	2.0	(cleaning)

The time at which each period starts and ends differs from person to person. One person gets up at 8, another at 6. The system might cope with this by measurement: it detects (e.g. by getting a signal from the alarm clock, or by movement detection) when a person gets up. For reasons of simplicity we chose to have the user specify the various start and end times.

4.2.4.6 Estimation of person's clothing

The thermal resistance value for clothing that people normally wear inside the house varies from 0.3 (underwear, shorts, socks, and T-shirt) to 0.95 (underwear, pants, socks, shoes, shirt, sweater) [5]. Values higher than these represent clothing that normally would be worn outside only (e.g. a jacket).

A proper basis for estimation might be the outside temperature. When outside temperatures are low, people tend to wear more clothing, regardless of the inside temperature. To keep things simple, we take the following value for the thermal resistance:

$$clo = 0.95 - 0.65 \cdot \frac{T_{outside} - T_{Cold}}{T_{Hot} - T_{Cold}} \qquad (4.14)$$

where T_{cold} is e.g. –5 degrees and T_{hot} is e.g. 25 degrees.

4.2.5 Remarks on the Comfy Model

COMFY suffers from a number of incompletenesses, due to the fact that it is still a prototype. Some of the problems are mentioned here. They should be worked upon when COMFY evaluates into a program for real comfort management.

- At the end of the day COMFY tends to let the temperature drop to a very low level. This is due to the fact that at night the preferred comfort (PMV) is set to a low value and larger deviations are allowed. The utility does not punish enough for cold temperatures. At the same time the COMFY system sees no reason to keep the temperature at a reasonable level, since it does not anticipate on a situation more than 24 hours ahead of the starting period (0:00 h.). We have temporarily overcome this problem by defining smaller allowed deviations at the end of the day. Better alternatives are: setting a minimum allowed temperature or defining a short-cut constraint to the beginning of the day.

- Allowed deviations are modeled symmetrically, which in many cases is not realistic. At night or when not at home the comfort level can be set at a low value, e.g. PMV = -1. Higher comfort, e.g. PMV = +1, is allowed as long as it leads to optimal cost. A PMV-value of -3 however might be too low. Asymmetric deviations can be achieved by reshaping the net utility function.

- The utility function is sensitive to preferred comfort, allowed deviations, cost, user profiles. Logical dependencies exist between these parameters and the thermostat settings in the reference situation, such as: preferred comfort with low allowed deviations requires a corresponding temperature setting of the thermostat. If not then the utility function can get unrealistic values and comparisons with the COMFY optimal situation cannot be made on a realistic base.

- The COMFY model is built around a special configuration (electrical heating, mechanical ventilation) which can be controlled separately for each room. Also in the building model a number of simplifications are used. This can be improved by coupling of COMFY to a verified building model, such as TRNSYS.

4.3 Modeling of Energy Storage in Buffers

Implementation of energy storage in a buffer can be realized with two additional actuators with settings between 0 and 1:

s^i_{buf-in} denotes the setting of the buffer filling as a 'percentage' of its filling capacity.

$s^i_{buf-out}$ denotes the setting of the buffer output as a 'percentage' of its emptying capacity.

In an optimal situation one wil expect s^i_{buf-in} to be zero if $s^i_{buf-out} > 0$ and vice versa.

The heating system now has two possibility to heat the room: by using the heating installation itself and by using the heat buffer. The temperature equation now becomes:

$$air\_temp_i = air\_temp_{i-1} + T^i_{vent\_grid} + \frac{Q^i_{heating} + Q^i_{buf-out} + Q^i_{sun}}{C} \cdot \Delta t$$

The $Q^i_{buf-out}$ is dependent on the capacity of the buffer, its filling percentage and of course whether or not the buffer is used. As a first approximation we consider a linear dependency on the current buffer capacity:

$$Q^i_{buf-out} = s^i_{buf-out} \cdot cap^{i-1}_{buf} \cdot speed_{buf-out} \qquad \text{with } cap^i_{buf} = cap_{\max-buf} \cdot fill^i$$

where cap^i_{buf} is the current buffer capacity in period i, $fill^i$ is the filling percentage of the buffer and $speed_{buf\text{-}out}$ is the emptying speed of the buffer.

We also have to account for heat losses from the buffer. We also model the energy loss in the buffer as a linear dependence on the current buffer capacity:

$$Q^i_{buf-loss} = cap^{i-1}_{buf} \cdot speed_{buf-loss} \qquad \text{with } cap^i_{buf} = cap_{\max-buf} \cdot fill^i$$

where $speed_{buf\text{-}loss}$ is the energy loss speed of the buffer.

In order to make use of a heat buffer we also have to fill the buffer. Similar as above the filling energy can be stated as:

$$Q^i_{buf-in} = s^i_{buf-in} \cdot cap^{i-1}_{buf} \cdot speed_{buf-in} \qquad \text{with } cap^i_{buf} = cap_{\max-buf} \cdot fill^i$$

where $speed_{buf\text{-}in}$ is the filling speed of the buffer.

The heating energy to fill the buffer has to adapt for the buffer efficiency:

$$Q^i_{heating-buf} = Q^i_{buf-in} / conv_{eff}$$

Using the buffer for heating ($Q^i_{buf-out}$) does not contribute to the cost. The contribution of the filling of the buffer to the cost in period i equals $k^i_{buf-in} = Q^i_{heating-buf}$

In the above equations:

$cap_{\max-buf}$	the maximum capacity of the buffer
$conv_{eff}$	the conversion efficiency to the energy storage

At each point in time we are interested in

The current capacity (or the filling percentage) of the heat buffer, i.e.

$$cap_{buf}^{i} = cap_{buf}^{i-1} + Q_{buf-in}^{i} - Q_{buf-loss}^{i} - Q_{buf-out}^{i}$$

$$cap_{buf}^{0} = cap_{\max-buf} \cdot fill^{0}$$ $fill^0$ is the initial filling percentage

The cost of filling the buffer, i.e.

$$c_{buf-in}^{i} = Q_{heating-buf}^{i} \cdot energyprice^{i} = k_{buf-in}^{i} \cdot energyprice^{i}$$

And of course the total heating energy $Q_{heating-buf}^{i}$ itself.

The new input-parameters generated are:

	`Buffer`	`BufferId % the type of buffer`
$cap_{\max-buf}$	`Capacity`	`(kWth)`
$speed_{buf\text{-}loss}$	`EnergyLoss`	`(percentage/hr)`
$fill^0$	`InitialFilling`	`(percentage)`
$speed_{buf\text{-}in}$	`FillingSpeed`	`(percentage/hr)`
$speed_{buf\text{-}out}$	`EmptyingSpeed`	`(percentage/hr)`
$conv_{eff}$	`ConversionEfficiency`	`(percentage)`

New output settings are:

s_{buf-in}^{i} (BufferInput, between 0 and 1), $s_{buf-out}^{i}$ (BufferOutput, between 0 and 1), and cap_{buf}^{i} (BufferFilling in kW or in percentage) per timestep.

4.4 Comfy Architecture improvements

The version of COMFY, developed in this project, is implemented as a front-end user interface program, written in Java, and a back-end calculation program, written in Fortran, called *Comfortran*. The user interface allows parameter input and viewing of a number of standard result charts. The current version COMFY-2 contains a number of improvements on the original version of COMFY:

4.4.1 Optimisation Model

An advanced optimisation algorithm using the NAG module E04JYF replaces the calculation algorithm -simple Newton-Raphson -. This leads to a spectacular improvement in the performance (from 10 - 20 minutes to 5 - 10 seconds in the default 24 hours schedule). This makes COMFY-2 suitable for real-time demonstration.

4.4.2 Reference Situation

In the original version of COMFY arbitrary initial settings were used for the control variables (heating, sunblinds, ventilator and ventilation grid). Therefore it was difficult to compare optimization results with a reasonable reference situation.
COMFY-2 makes it much easier to study the optimisation effects since it starts computation with a reference situation based on standard thermostat settings for room heating. The other control variables are set on a fixed value for each period. Thus COMFY-2 does not only give an optimized comfort strategy, but also gives a representative basic strategy. Improvement and worsening of comfort, cost and energy efficiency can be considered useful indications.

4.4.3 New Parameters

Several new parameters have been added to the model:

- SolarRadiationProfile
  Replaces the – constant – Qzenith with reasonable values for solar radiation in each period. If no solar radiation profile is given, Qzenith will still be used.
- InitialTemperatureSettingProfile
  For the reference situation a thermostat setting profile is given for each period. If the room temperature falls below the thermostat setting then the heater is switched on with a setting given by InitialHeaterProfile (between 0 = 'off' and 1 = 'full capacity').
- DayNumber
  The daynumber of the starting day is used in order to be able to calculate the position of the sun and the times of sunrise and sunset. The reference location is the Netherlands.
  The daynumber is not yet used for calculations. In stead the SunUp and SunDown parameters are used together with either the SolarRadiationProfile or the Qzenith.

4.4.4 Standard Graphics

In order to ease interpretation of the COMFY-2 results, the user frontend provides the user after optimization with some useful graphic charts. Standard charts which are given are:

- Settings of control variables: referential situation versus optimized strategy.
- Temperature graphs of outside temperature, room temperature, radiation temperature, experienced temperature after correction for clothing: referential situation versus optimized strategy.
- Comfort indicators, such as preferred comfort, predicted mean vote (PMV), difference between preferred comfort and PMV, and percentage dissatisfied (PPD): referential situation versus optimized strategy.
- Utility and cost functions, both direct and cumulative graphs; also a cumulative energy usage fucntion (kW): referential situation versus optimized strategy.

For a complete overview see the Appendix.

4.4.5 Program Control

Optimisation based on a period length of one hour (i.e. changing the settings of each control variable every hour) tends to be rather coarse. COMFY-2 is adapted in order to handle other period lengths. Experience shows that a period length of 15 minutes gives reasonably balanced results, also in the reference situation. Calculation time (Pentium 266) over 24 hours however increases from 10 - 15 seconds to several minutes. The scenario study results in the next chapter are based on a 15 minute period length.

COMFY-2 is not only fit for optimisation of one standard interval (default 24 hours). It also can perform optimisation using an adaptive strategy, i.e. starting with a standard interval (e.g. 8 hours) adapt the calculation after each period to the actual comfort and thus optimise 24 hours in 1 + 24 - 8 = 17 steps. This allows COMFY-2 to anticipate on deviations between expected or calculated values and actual values for comfort control variables (temperatures, climate, electricity prices, etc.).

In the user interface this option cannot be selected. It is built in for later purposes. Nor any effort is put into improving the output style of COMFY since adaptive scenarios only are useful when applied in a real life situation. Then COMFY will need to react to discrepancies between calculated comfort and actual comfort.
Note, that this option also allows users, in a real life scenario, to reschedule their preferred comfort, e.g. when changing plans on short notice.

4.5 References

[1] *Poly-technisch Zakboekje*. Koninklijke PBNA, Arnhem, 1997.
[2] Boertjes, E. *COMFY, Comfort Management System, design document*. Free University Amsterdam, December 1999.
[3] Boertjes, E., J.M. Akkermans, R. Gustavsson, I.G. Kamphuis. *Agents to Achieve Customer Satisfaction: The COMFY Comfort Management System.* Proceedings of the Practical Application of Intelligent Agents and Multi-Agents Conferences (PAAM), April 10-12, 2000.
[4] Fanger, P.O. *Thermal Comfort.* McGraw-Hill, 1972.
[5] ISO Standard 7730:1994. *Moderate Thermal Environments - Determination of the PMV and PPD Indices and Specification of the Conditions for Indoor Thermal Comfort.* ISO 1994, European Standard Report EN-ISO-7730:1995, CEN, Brussels, 1995.
[6] Innova. *Thermal Comfort.* http://www.innova.dk/books/thermal/thermal.htm. Innova AirTech Instruments, 1997.

5. COMFY BASIC SCENARIO STUDIES RESULTS

Using COMFY a number of basic scenarios have been calculated in order to determine the energy and cost saving potentials by means of comfort management. We modelled a living room faced south with the following characteristics:

Four scenarios have been modeled, varying two parameters: day of the year (early summer day and winter day) and cost level (luxureous and economic).
The day is divided in four main periods: the morning from 7:00 to 10:00, on which the customer is at home; the daytime from 10:00 to 16:00, on which the customer is out; the evening, from 16:00 to 22:00, when the customer is at home again; and the night from 22:00 to 7:00, when the customer is at sleep. We assume that the customer is not well dressed in the first hours of the day. Comfort is required when the customer is at home and not asleep.

<u>Summer versus Winter</u>

Summer season: The summer season is characterised by a day on which a heat request exists during the night and a cooling request during the day. The outside temperature varies from 15 °C at night up to 23 °C in the afternoon. The sun is shining and up from 5:00 to 22:00 hours.
The clothing level is set to Icl = 1, i.e. normal summer clothing: pants, shirt, overhemd. Early in the morning and late at night the Icl is somewhart lower. The activity level (M-rate) is kept constant at +1.

Winterseason: The winter season is characterised by a day on which a heat request exists both at daytime and at night. The outside temperature varies from −5 °C at night up to +5 °C in the afternoon. The sun is shining and up from 6:00 to 18:00 hours.
The clothing level is set to Icl = 1.3 / 1.5 in the morning and the afternoon, i.e. winter clothing (sweater). At night we presume Icl = 2 (sleeping under a warm blanket). The activity level (M-rate) is kept constant at +1.

For a complete overview see figure 5.1.

<u>Luxurious versus Economic</u>

Comfort luxureous: We set PMV = 0 (optimal comfort) when the customer is at home. At daytime, when the customer is out, we set PMV = −0.5. At night we set PMV = −0.25.

Comfort economic: We set PMV = −0.2 (sub-optimal comfort) when the customer is at home. At daytime and at night we set PMV = −0.5.[1]

Weighing luxureous: We allow a maximum deviation of +1 when the customer is at home. The customer wants to pay for better comfort at a high cost level (0.5 to 0.88, (scale −1 to +1). At daytime and at night the allowed deviation is set to 2, with a cost level from −0.5 (daytime) to −0.2 (night).

Weighing economic: We allow a maximum deviation of +1 when the customer is at home. The customer wants to pay for better comfort at a low cost level (near 0, scale −1 to +1). At daytime and at night the allowed deviation is set to 2, with a cost level of −0.6 (daytime and night).

For a complete overview see figure 5.2.

[1] In the summer we should set a PMV > 0 as a sub-optimal comfort., since cooling (not considered in COMFY) will be the main issue. We can also set PMV = 0 and allow higher deviations on comfort.

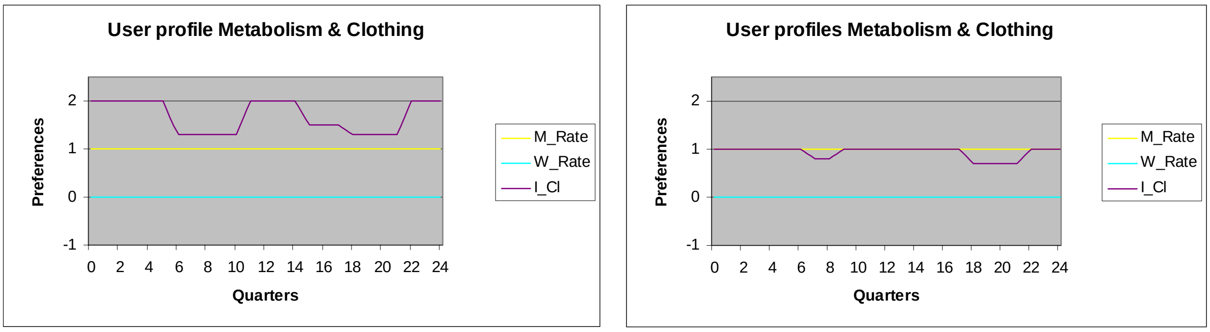


Figure 5.1. Winter (left) and Summer User Profiles for Metabolism and Clothing

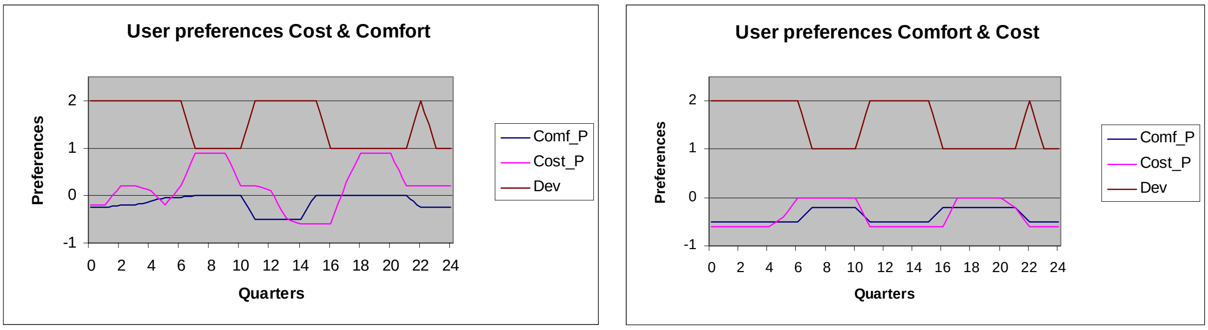


Figure 5.2. Luxurious (left) and Economical Comfort & Cost Scenario

5.1 Scenario 1: Winter Luxurious

Specific settings: optimal PMV, high cost level, low deviation.
Grafical results: comfort, utility/cost functions, energy use, appliance settings.
User profiles ans settings: as described above.

The reference thermostat control (figure 5.3a) shows regular switching on and off. at night with full power, at daytime with partial power (in accordance with the thermostat settings). This guarantees a fairly reasonable level of comfort with temperatures around ± 22°C on times of presence. The utility function (which is a measure between preferred comfort and actual comfort) shows a value of −542, the energy cost is fl.2.48 and the total energy usage is 13.9 kW_{el}.

Comfort management with COMFY shows a different picture. By using a more gradual scheme for operating the heater, a completely different picture is generated. A more even temperature control maintains a 22 °C temperature during presence hours. The difference between desired and delivered comfort is minimal in hours of presence. During absence and sleeping time comfort never exceeds the the limit of PMV = -0.5. (the end evening dip is an artefact from the optimiser). PMV can be seen to never exceed PMV = 0.25. The complete picture is far more constant compared to using the thermostat.
In figures: The utility spectacularly declines from -542 to -8.6. Costs diminish by an amount of 24% (from DFl.2.48 to fl.1.88), and the energy usage with 25% to 10.4 kW_{el}.

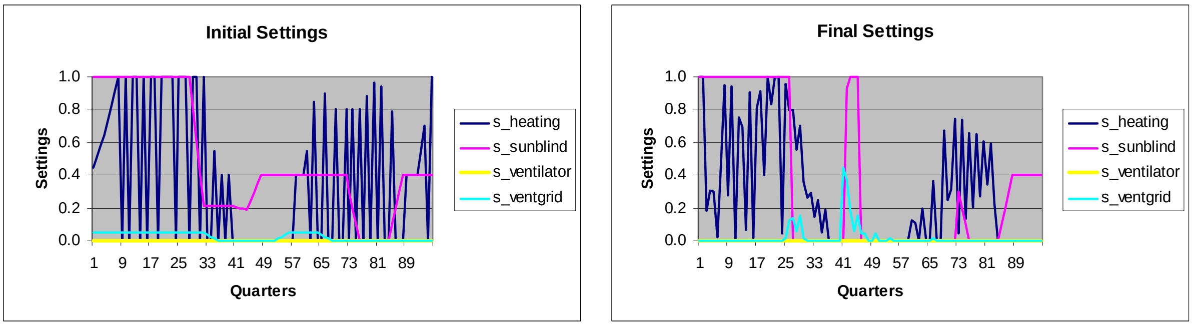


Figure 5.3. Scenario Luxurious in Winter:

5.2 Scenario 2: Winter Economical

Specific settings:	basic PMV, low cost level, high deviation.
Grafical results:	comfort, utility/cost functions, energy use, appliance settings.
User profiles ans settings:	as described above.

Using the settings described above, the simulation was done for the economic scenario. In the reference situation the thermostat control was set at 15 °C at night. The rather Spartan life style is expressed in the reference situation. The comfort level in the presence periods is satisfactory. The utility amounts to a value of only -2.85 (Note that the weight parameters do allow this compared to the previous scenario). 11.8 kW_{el} is used with a price of Dfl. 2.14. Note that the luxurious scenario is cheaper in this respect.

The comfort management system hardly uses the heater at night. The temperature in the living room is lowered considerably. By switching on the heater at full pace during the morning at low prices, a reasonable morning temperature is achieved. Total realised comfort, however, can be seen to lag behind the desired comfort. In the simulations, probably, too wide margins in the weighting factors have been granted. Prime advantage is the dramatic decrease in the costs. The energy usage decreases by 50 % to 6.07 kW_{el} and the cost to DFl.0.99.

Possibly, the comfort can be further optimised by setting more narrow bandwidth to the optimiser with respect to the desired comfort level (PMV = 0 in stead of. -0.2) or by allowing the cost level to increase somewhat.

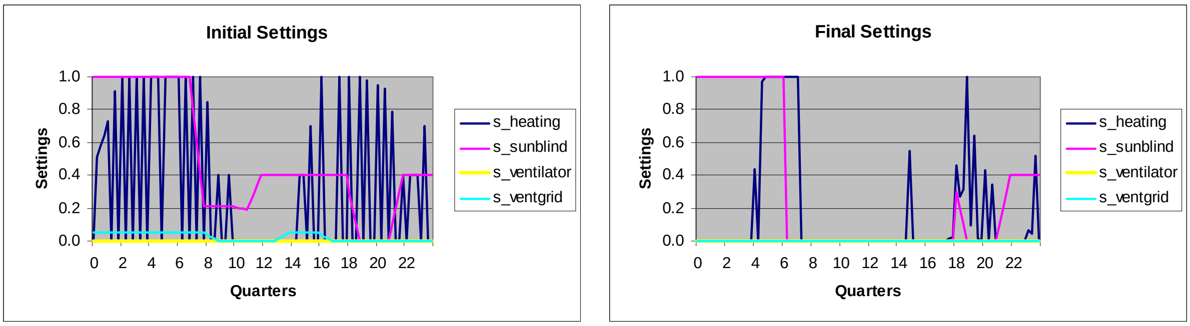


Figure 5.4. Scenario Economical in Winter:

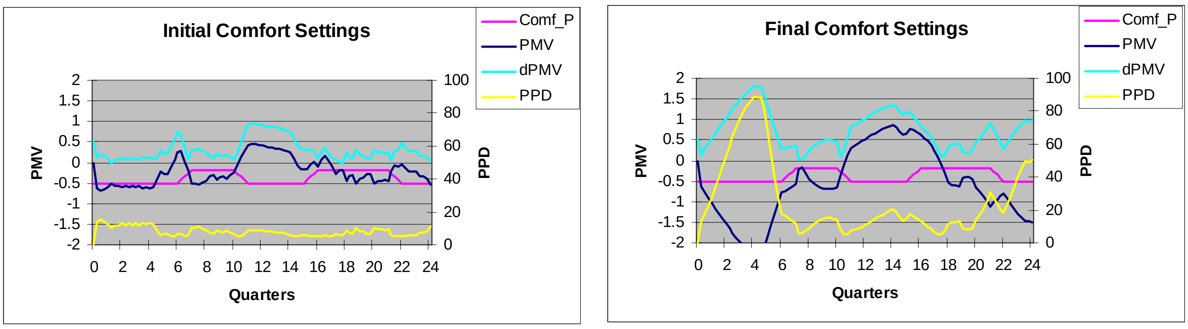
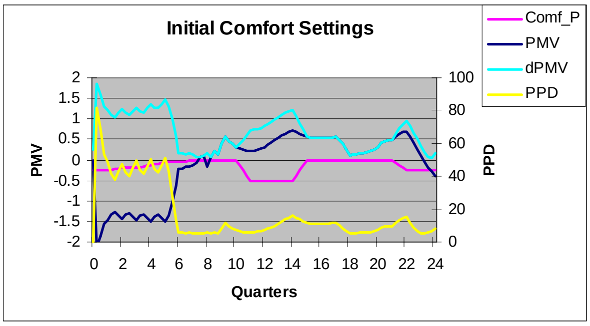


Figure 5.5. Scenario Economical in Winter:

5.3 Scenario 3: Summer Luxurious

Specific settings: optimal PMV, high cost level, low deviation.
Grafical results: comfort, utility/cost functions, energy use, appliance settings.
User profiles ans settings: as described above.

In the reference situation it can be seen, that the heater only switches at night; during the rest of the day no additional heating is necessary. Now the overheating aspect (26 °C) appears during the daytime. The comfort level (PMV) attains a value of 0.5, which is considered slightly too warm. The utility function has a value of -427; cost is DFl. 0.56 and a total of 4.1 kW_{el} is used.
Comfort optimisation gives a completely different behaviour. Realised and desired comforts in periods of presence closely follow each other. The complete picture is more flat than the strategy without comfort management. The utility attains a level of –4.25. The cost are reduced by 29% (to DFl.0.40), and the energy usage by 28%, from 4.5 kW_{el} to 2.9 kW_{el}. Especially noteworthy are the concerted control of the sun blinds to optimally use the radiation temperature for maintaining comfort.



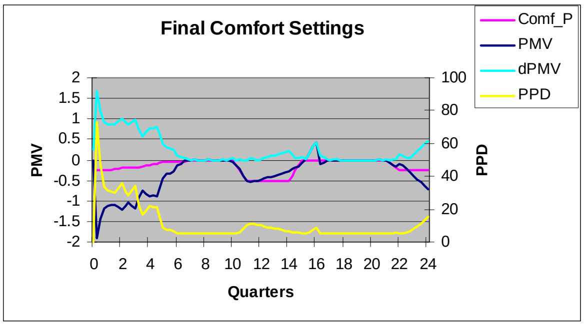


Figure 5.6. Scenario Luxurious in Summer:

5.4 Scenario 4: Summer Economical

Specific settings: basic PMV, low cost level, high deviation.
Grafical results: comfort, utility/cost functions, energy use, appliance settings.
User profiles ans settings: as described above.

In the reference scenario the heater operates early in the morning as in the winter case. The afternoon tends to be too warm, since there is no cooling apart from a fixed setting for the sunblinds. The total utility function has a value of -247. The energy usage to achieve this is 2.1 kW_{el} at a price of DFl. 0.28.

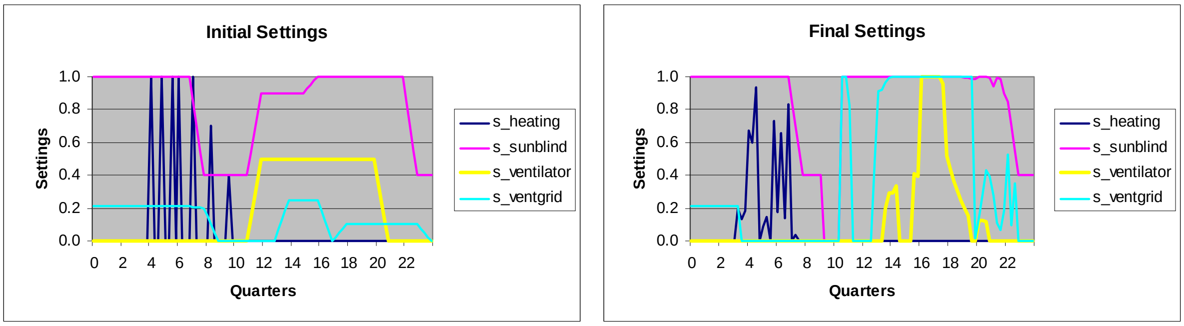


Figure 5.7. Scenario Economical in Summer:

In contrast to the winter situation, comfort management leads to a strongly improved comfort; especially in the morning hours. This can be attributed to the fine-meshed control actions to the solar screen, that leads direct solar radiation in the room. In the afternoon the sunblinds are closed to prevent overheating. There is over 50% decrease of the energy usage to 0.9 kW_{el} and in cost, DFl 0.28 to 0.12.

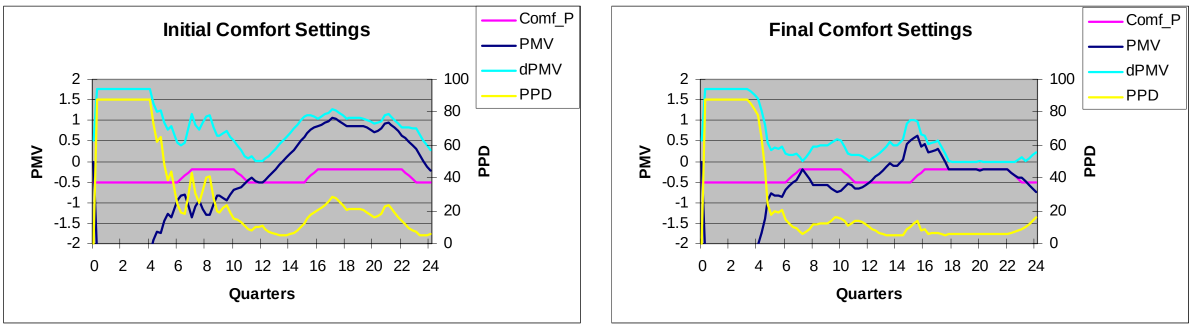


Figure 5.8. Scenario Economical in Summer:

5.5 Buffer Strategy

5.5.1 Results of Energy Storage in Buffers

Applying the buffer model as described in chapter 4.3 to the Comfy luxurious winter scenario (chapter 6) shows that the buffer will not be used in the optimal strategy . The buffer settings are:

Capacity	2800.0	(Wth)
EnergyLoss	9.0	(percentage/hr)
InitialFilling	1.0	(percentage)
FillingSpeed	21.0	(percentage/hr)
EmptyingSpeed	10.0	(percentage/hr)
ConversionEfficiency	70.0	(percentage)

Note that the initial buffer filling is set to 0, in order to eliminate the side-effect that the optimal strategy just empties the buffer at the end of the day. The results in terms of utility and energy are:

Total Energy	from	14.78	to	14.00
Total Utility	from	-671.92	to	-15.47
Total Costs	from	313.20	to	289.96
Total NetUtility	from	-985.11	to	-305.43

In an optimal strategy buffering will be used only if the dimensions of the whole infrastructure are right. Three parameters clearly influence the usefulness of buffering: the ratio between emptying speed and energy losses from the buffer, the conversion efficiency, and the differences in price levels between hours for energy. In the above scenario we changed the following settings:

EmptyingSpeed	20.0	(percentage/hr)
ConversionEfficiency	95.0	(percentage)
EnergyLoss	1.0	(percentage/hr)
InitialFilling	1.0	(percentage)
EnergyPrices	Doubling of energyprices from 1700 to 2000 hours	

As result we see that the optimal strategy starts filling the buffer from 1400 to 1600 hours (when energy prices are at the lowest). The results for the utility and energy functions now become:

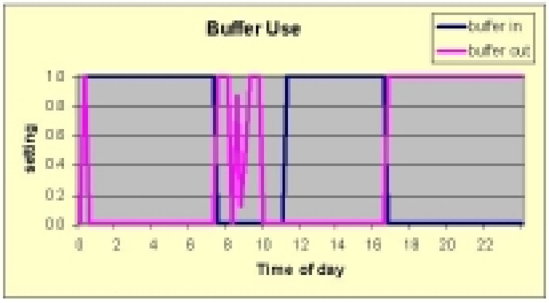
Total Energy	from	14.78	to	15.03
Total Utility	from	-671.92	to	-15.88
Total Costs	from	-413.25	to	329.52
Total NetUtility	from	-1085.17	to	345.40

Note that the cost (and hence the total net utility) cannot be compared to the earlier buffer scenario, since we drastically raised the energy prices.[1] We see a large improvement in the comfort utility function and the cost, whereas the total energy needed slightly increases. This is not surprising as buffer utilisation always will be less efficient due to (1) loss of energy efficiency and (2) buffer energy losses.

A picture of the buffer use is given in Figure 5.9. As can be seen the buffer starts loading in the early hours of the day, when prices are low. After 7:00 h, when prices rise, the buffer is partly

[1] The optimal solution from the base buffer scenario would generate a cost of 385.39 if prices were set at the level of the second buffer scenario.

used until ± 11:00 h. Then the buffer is filled again until the next price raise. Note that at the end of the day the buffer fill is at a higher level than at the beginning of the day. The value of this energy is not included in the total utility.



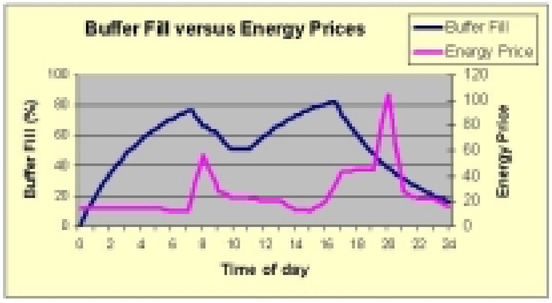


Figure 5.9. Buffer use for optimal buffer and large energy price deviations

Remark: the optimisation algorithm forces us to start with an initial buffer fill > 0. If not, the iteration process probably suffers too much from boundary effects.

5.5.2 Energy Storage in Buffers - First Conclusion

We have seen from the results that with respect to energy efficiency the implemented buffer strategy does not lead to improvements. Cost efficiency is possible under special circumstances.

Energy buffering becomes a valuable element in the energy infrastructure when local energy is supplied with no (or low) operational cost (PV cells, WKK, waste heat) and it is not needed immediately. When the redelivery is not possible or compensations are low compared to the delivery price both energy and cost advantages may be achieved using buffering.

6. COMFY LIFESTYLE SCENARIO STUDIES

6.1 Goal and scope

In this chapter we further research the potential energy savings using optimised comfort management, by defining different scenarios reflecting different life-styles. Our basic reference is a standard residential building with a energy performance index of 1.0.

6.2 Archetypes

6.2.1 Residential building usage and smart control

6.2.1.1 Introduction

Smart control in a residential environment requires a good impression of the usage of rooms in the house. Presence detection and activity knowledge are two main deciding factors. Simulations, such as with the COMFY model, can be used to make comparisons between optimally equipped computerised networks having access to all necessary information through the WEB and adapting to the lifestyle of the inhabitants and "dumb" equipment only programmed with few day types.
In this chapter we use the COMFY comfort management model to compare presence-dependent HVAC control on a per compartment base, controlled also accounting for predicted short term weather with HVAC-schemes. The reference situation is a standard NOVEM "doorzonwoning". The EPC-value of this home is currently legally fixed at 1.0. Existing houses normally have a higher EPC, the future norm EPC will be 0.5. In the simulations a complete climatic year is calculated.

Complete comfort control also should take into account air quality as the measured CO_2 percentage. In simulation conditions also the predictability of the usage and volume of the heating water flow as a function of time might be taken into account. Filling the buffers takes place in favour of renewable generation of hot water. Lighting can be controlled by presence and taking into account outside lighting conditions. All these extensions lie outside the scope of COMFY.

6.2.1.2 The usage of residential buildings

The usage of buildings can be derived from time-occupation enquiries from SCP (Social and Cultural Planning Bureau). Most recently this was done in 1995 [1]. The results of the enquiries are approximate and derived values. Aggregation is done over family types. Looking at all family types we get the results as listed in Table 6.1. If we divide further, for two-job households the presence at home is 15 % lower under working time condition and 10 % lower in the evening. For families with children and elderly people one can subtract 10 % during working hours. Elderly people have a 10 % higher evening residential presence.

Time	Presence (%)
7.00	88
8.00	70
9.00	54
10.00	45
11.00	40
12.00	45
13.00	45
14.00	38
15.00	38
16.00	44
17.00	57
18.00	73
19.00	72
20.00	68
21.00	70
22.00	75
23.00	83
24.00	90

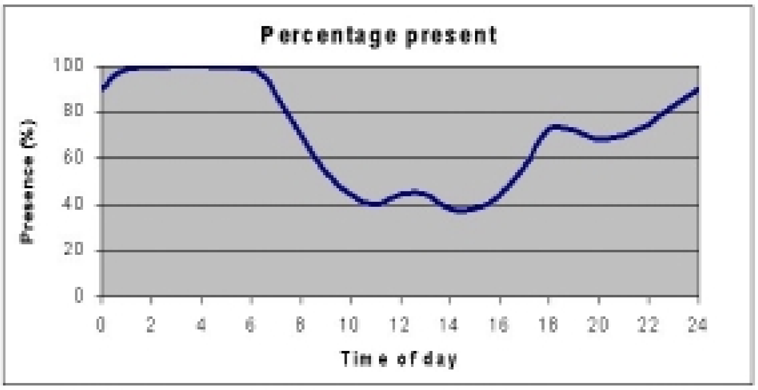


Table 6.1. Presence in residential buildings

6.2.1.3 Room characteristics

A living room in a standard Dutch house with an energy performance (EP) of 1.0 is modelled.[1] The orientation of the room is such that it contains a large window on the south side. The room has the following characteristics:

Room label	EP10	
Room volume	100 m^3	
Area window	3 m^2	*
Orientation window	180 ° N	
Heating capacity walls	1500 kJ/°C	**
Heating capacity air	600 kJ/°C	***
Damping factor	0.5	
Sunblind shielding	15 %	
Heater capacity	1500 Wt	
Ventilator capacity	0.8 m/s	
Ventilation power	300 Wt	
Ventilation grid opening	0.1 m^2	
Switching energy heater	0 Wt	
Switching energy sunblind	0 Wt	
Switching energy ventilator	0 Wt	
Switching energy ventilation grid	0 Wt	

Table 6.2. Room characteristics for EP = 1.0.
** A window area of 3 m^2 is rather small*
*** The heating capacity of the walls is used with respect to the reflection of infrared radiation.*
**** In the heating capacity of the air the air ventilation is included*

[1] The Dutch EPC 1.0 standard is based on gas heating. The COMFY-2 model however is based on electrical heating. By imposing the COMFY-2 model on a Dutch home we make a concession to the basis of the EPC 1.0 standard.

6.2.2 Seasons

6.2.2.1 Day characteristics

In COMFY version 1 the solar power is calculated from the times of sun up and sunset. The Qzenith is taken as a constant value, which is adjusted to angle of the sun. In COMFY 3 we introduced a solar radiation profile. Therefore we no longer need day characteristics.

6.2.2.2 Outdoor climate

The outdoor climate is given using profiles: for each period (hour or quarter) a forecast is known for the outdoor air temperature, the humidity and the solar radiation per m^2.

The data for the outdoor climate are from the DEGO reference year for the Netherlands. We used the following typical days::

- a sunny winter day, i.e. 6 February, constant heating demand,
- a sunny summer day, i.e. 14 August, cooling demand at daytime

6.2.2.3 Indoor climate

Initial values for the indoor climate are given: initial air temperature and radiation temperature, initial humidity and initial air velocity. The indoor climate for each concurrent period is taken as calculated by the COMFY model.

In winter, the initial air and radiant temperature are set equal to the thermostat value at period 0 (and thus differs in each scenario). The initial air and radiant temperature in the summer are set to 21 °C, being close to the temperature at the end of the day at 24:00 hours..

Initial humidity is set to 60 %, initial air velocity to 0.2 m/s.

6.2.2.4 Energy prices

See Figure 6.1.

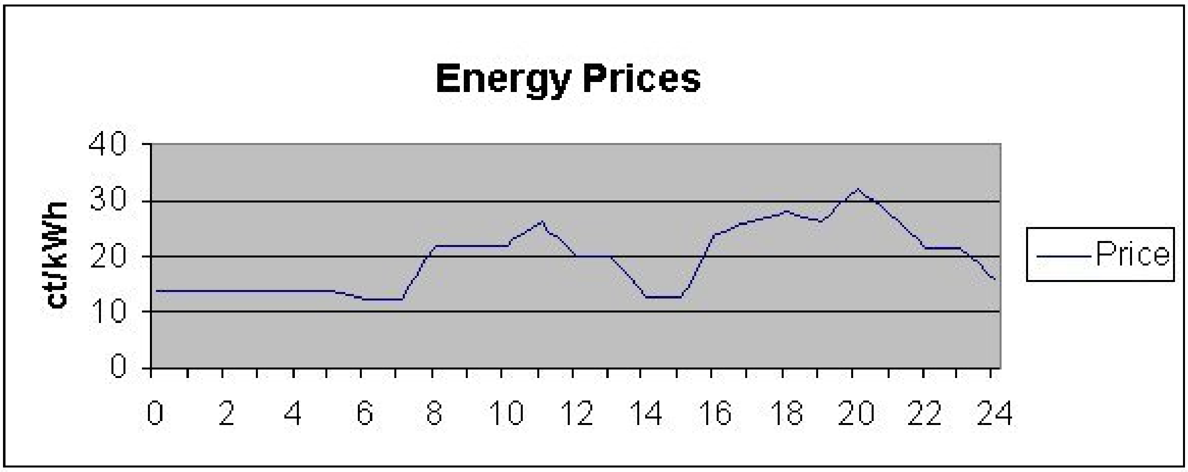


Figure 6.1. Energy market prices for one day

6.2.3 Life-style

- We set up the following life-styles
  - yup (young urban professional) or dink (double income, no kids): absent during daytime; luxury comfort at night – Erica serves as the role model,
  - budget dweller with no kids (student): absent during daytime; economic comfort at night – Eric serves as the role model,

- senior: during daytime and at night passive and present; high comfort at higher PMV level, low cost.
- family with children: during daytime active and present, at night present but less active; pleasant comfort, cost level between Erica and the senior.

6.2.3.1 User profiles

The metabolism W-rate is set to 0, the M-rate is expressed in MET and is dependent on the activity level in the different life-styles. The clothing level (Icl) is expressed in 'clo' and is dependent on the time of the day.

<table>
<tr><th></th><th colspan="4">Presence</th><th colspan="4">M-rate</th><th colspan="4">Icl</th></tr>
<tr><th></th><th>Eric</th><th>Erica</th><th>Senior</th><th>Family</th><th>Eric</th><th>Erica</th><th>Senior</th><th>Family</th><th>Eric</th><th>Erica</th><th>Senior</th><th>Family</th></tr>
<tr><td>00:00 – 07:00</td><td colspan="4">1</td><td colspan="4">0.8</td><td colspan="4">1.4
1.2</td></tr>
<tr><td>07:00 – 08:00</td><td colspan="4">1</td><td colspan="4">1.2
1.0</td><td colspan="4">0.8</td></tr>
<tr><td>08:00 – 09:00</td><td colspan="4">1</td><td colspan="4">1.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr>
<tr><td>09:00 – 10:00</td><td colspan="4">1</td><td>1.6
1.2</td><td>1.4
1.2</td><td>1.2
1.0</td><td>2.0
1.6</td><td>1.0</td><td>1.0</td><td>1.2
1.0</td><td>1.0</td></tr>
<tr><td>10:00 – 13:00</td><td>0</td><td>0</td><td>1</td><td>1</td><td>-</td><td>-</td><td>1.0</td><td>1.6
1.4</td><td>-</td><td>-</td><td>1.2
1.0</td><td>1.0</td></tr>
<tr><td>13:00 – 14:00</td><td>0</td><td>0</td><td>1</td><td>1</td><td>-</td><td>-</td><td>1.0</td><td>1.0</td><td>-</td><td>-</td><td>1.2
1.0</td><td>1.0</td></tr>
<tr><td>14:00 – 16:00</td><td>0</td><td>0</td><td>1</td><td>1</td><td>-</td><td>-</td><td>1.0</td><td>1.6
1.4</td><td>-</td><td>-</td><td>1.2
1.0</td><td>1.0</td></tr>
<tr><td>16:00 – 18:00</td><td colspan="4">1</td><td>1.4
1.2</td><td>1.2</td><td>1.2
1.0</td><td>1.2</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr>
<tr><td>18:00 – 19:00</td><td colspan="4">1</td><td colspan="4">1.0</td><td>1.2
1.0</td><td>1.0</td><td>1.2
1.0</td><td>1.2
1.0</td></tr>
<tr><td>19:00 – 22:00</td><td colspan="4">1</td><td>1.2
1.0</td><td>1.0</td><td>1.0</td><td>1.2
1.0</td><td>1.2
1.0</td><td>1.0</td><td>1.2
1.0</td><td>1.0</td></tr>
<tr><td>22:00 – 24:00</td><td colspan="4">1</td><td colspan="4">0.8</td><td colspan="4">1.4
1.2</td></tr>
</table>

Table 6.3. User profiles for the COMFY scenarios.
If a cell contains two numbers, the first number is valid
for the winter and the second for the summer

<u>Metabolism M-rate profiles:</u>

The metabolism M-rate profile is built up following these rules of thumb:

- M-rate = 0.8 during sleep (22:00 – 07:00)
- M-rate = 1.2 after waking up (07:00 – 08:00)
- M-rate = 1.0 during breakfast and dinner (08:00 – 09:00 resp. 18:00 – 19:00)
- M-rate between 1.0 (senior) and 1.6 (family work) during daytime, when present (09:00 – 18:00)
- M-rate = 1.0 at night (19:00 – 22:00).

Clothing profiles:

The clothing profile is built up following these rules of thumb:

- Icl = 0.8 / 1.2 during sleep, in summer resp. winter
- Icl = 0.4 after waking up
- Icl = 1.0 during daytime, in summer resp. winter

Exceptions:

- Eric and the senior are prepared to put on a sweater at all times when it is too cold outside, having an Icl = 1.2 when the outdoor air temperature < 10 °C.
- When metabolism (M-rate) > 1.2, a higher clothing level is set back to 1.0.

6.2.3.2 Comfort profiles

Comfort indicator

The comfort indicator PMV (Predicted Mean Vote) is a number between –3 and +3 which represents the sensation of thermal comfort. PMV = -3 corresponds with too cold, PMV = +3 with too hot. A PMV of 0 is experienced by 95 % of all users as an agreeable comfort.
Both Erica and the family prefer optimal comfort (PMV = 0) when present and not asleep. At night less comfort (PMV = -0.2) is accepted. The senior requires a higher preferred comfort (PMV = 0.2 to 0.5) during daytime. At night PMV = -0.2 is accepted. Eric rather saves some money and is satisfied with low comfort (PMV = -0.2) when present. At night the PMV value can drop to –0.5.

	PMV				**Allowed deviation**				**Cost indicator**			
	Eric	**Erica**	**Senior**	**Family**	**Eric**	**Erica**	**Senior**	**Family**	**Eric**	**Erica**	**Senior**	**Family**
00:00 – 07:00	-0.5 -0.6	-0.2	-0.2	-0.2	2.5	2.0	2.0	2.2 2.5	-0.6	-0.2	-0.6	-0.4
07:00 – 08:00	-0.2	0.0	0.5	0.0	1.0	1.0	1.0	1.0	0.0	0.8	0.0	0.4
08:00 – 09:00	-0.2	0.0	0.5	0.0	1.0	1.0	1.5	1.5 1.0	0.0	0.8	0.0	0.4
09:00 – 10:00	-0.2 0.2	0.0 0.2	0.2	0.0 0.2	1.0	1.0	1.5	1.5	0.0	0.8	0.0	0.4
10:00 – 16:00	-0.8 0.2	-0.5 0.2	0.2	0.0 0.2	2.0	2.0	1.5	1.5	-0.6	0.0	0.0	0.0
16:00 – 18:00	-0.2 0.4	0.0 0.2	0.2	0.0 0.2	1.5 1.0	1.0 1.5	1.5	1.5	-0.3	0.3	0.0	0.0
18:00 – 19:00	-0.2 0.4	0.0 0.2	0.5	0.0 0.2	1.0	1.0 1.5	1.0	1.0	0.0	0.8 0.5	0.0	0.4
19:00 – 22:00	-0.2 0.4	0.0 0.2	0.5	0.0 0.2	1.0	1.0 1.5	1.0	1.0 1.5	0.0	0.8 0.5	0.0	0.4
22:00 – 24:00	-0.5 -0.6	-0.2	-0.2	-0.2	2.0	2.0	1.5	2.0 1.5	-0.6	-0.2	-0.6	-0.4

Table 6.4. Comfort profiles for the COMFY scenarios. If a cell contains two numbers, the first number is valid for the winter and the second for the summer

Allowed deviations

Allowed deviations vary from 1.0 / 1.5 when present to 2.0 / 2.5 when absent or at sleep, according to the values in Table 6.3.

Note that one should use different allowed deviations for lower and higher PMV values. This is not implemented in COMFY as yet.

Cost indicator

Eric wants to save money as much as he can. Especially at night and when he is not present the cost indicator can be set very low. Also the senior is cost-aware. Erica doesn't grudge the cost to receive optimal comfort, especially when present. The family cost level falls between Erica and the senior, except when at work.

6.2.3.3 Reference settings

In the reference situation the heater is used at full power when needed. During daytime the heater is only used at partial power (50%).
The thermostat in the reference situation is used according to the preferred comfort settings: Eric puts on the heating at an economical level of 20 °C and turns the thermostat down at night to 15 °C. For Erica these values are 22 °C and 18 °C resp. The senior sets the thermostat level even higher (23 °C) during daytime, the family lower, to 21 °C, due to a higher activity level.
The heater switches on or off for one period (15 minutes), depending on whether the temperature at the beginning of the period lies below or above the thermostat setting.

	PMV (see Table 6.4)				**Heater settings**				**Thermostat**			
	Eric	**Erica**	**Senior**	**Family**	**Eric**	**Erica**	**Senior**	**Family**	**Eric**	**Erica**	**Senior**	**Family**
00:00 – 06:00	-0.5 -0.6	-0.2	-0.2	-0.2	1.0	1.0	1.0	1.0	15°C	18°C	17°C	16°C
06:00 – 07:00	-0.5 -0.6	-0.2	-0.2	-0.2	1.0	1.0	1.0	1.0	18°C	20°C	20°C	18°C
07:00 – 08:00	-0.2	0.0	0.5	0.0	1.0	1.0	1.0	1.0	20°C	22°C	23°C	21°C
08:00 – 09:00	-0.2	0.0	0.5	0.0	0.5	0.5	0.5	0.5	20°C	22°C	23°C	22°C
09:00 – 10:00	-0.2 0.2	0.0 0.2	0.2	0.0 0.2	0.5	0.5	0.5	0.5	20°C	20°C	22°C	20°C
10:00 – 16:00	-0.8 0.2	-0.5 0.2	0.2	0.0 0.2	0.5	0.5	0.5	0.5	16°C	18°C	22°C	20°C
16:00 – 18:00	-0.2 0.4	0.0 0.2	0.2	0.0 0.2	0.5	0.5	0.5	0.5	20°C	22°C	22°C	21°C
18:00 – 19:00	-0.2 0.4	0.0 0.2	0.5	0.0 0.2	0.8	0.8	0.8	0.8	20°C	22°C	23°C	22°C
19:00 – 22:00	-0.2 0.4	0.0 0.2	0.5	0.0 0.2	0.9	0.9	0.9	0.9	20°C	22°C	23°C	21°C
22:00 – 23:00	-0.5 -0.6	-0.2	-0.2	-0.2	1.0	1.0	1.0	1.0	18°C	20°C	20°C	18°C
23:00 – 24:00	-0.5 -0.6	-0.2	-0.2	-0.2	1.0	1.0	1.0	1.0	15°C	18°C	17°C	16°C

Table 6.5. Reference settings for the heater in the COMFY scenarios. If a cell contains two numbers, the first number is valid for the winter and the second for the summer

The settings of the sunblind and ventilation depend on the season. In the winter season the sunblind will be opened during the daytime in order to make use of solar power. In the summer the sunblind usage is reversed.
In the winter ventilation will be used very carefully. In the summer season ventilation is used especially in during the afternoon to lower the temperature.

	Sunblind settings		Ventilator settings		Ventilator grid settings	
	Summer	**Winter**	**Summer**	**Winter**	**Summer**	**Winter**
00:00 – 07:00	0.0	1.0	0.0	0.0	0.05	0.05
07:00 – 08:00	0.4	0.2	0.0	0.0	0.05	0.0
08:00 – 09:00	0.4	0.2	0.0	0.0	0.05	0.0
09:00 – 10:00	0.4	0.0	0.2	0.0	0.1	0.0
10:00 – 13:00	0.7	0.0	0.2	0.0	0.1	0.0
13:00 – 16:00	1.0	0.0	0.2	0.0	0.1	0.0
16:00 – 18:00	1.0	0.0	0.4	0.0	0.2	0.05
18:00 – 19:00	1.0	0.0	0.4	0.0	0.2	0.0
19:00 – 22:00	0.4	0.5	0.2	0.0	0.1	0.0
22:00 – 24:00	0.0	1.0	0.0	0.0	0.05	0.05

Table 6.6. Reference settings for sunblind and ventilator in COMFY scenarios

6.3 Results

6.3.1 Winter

6.3.1.1 Settings

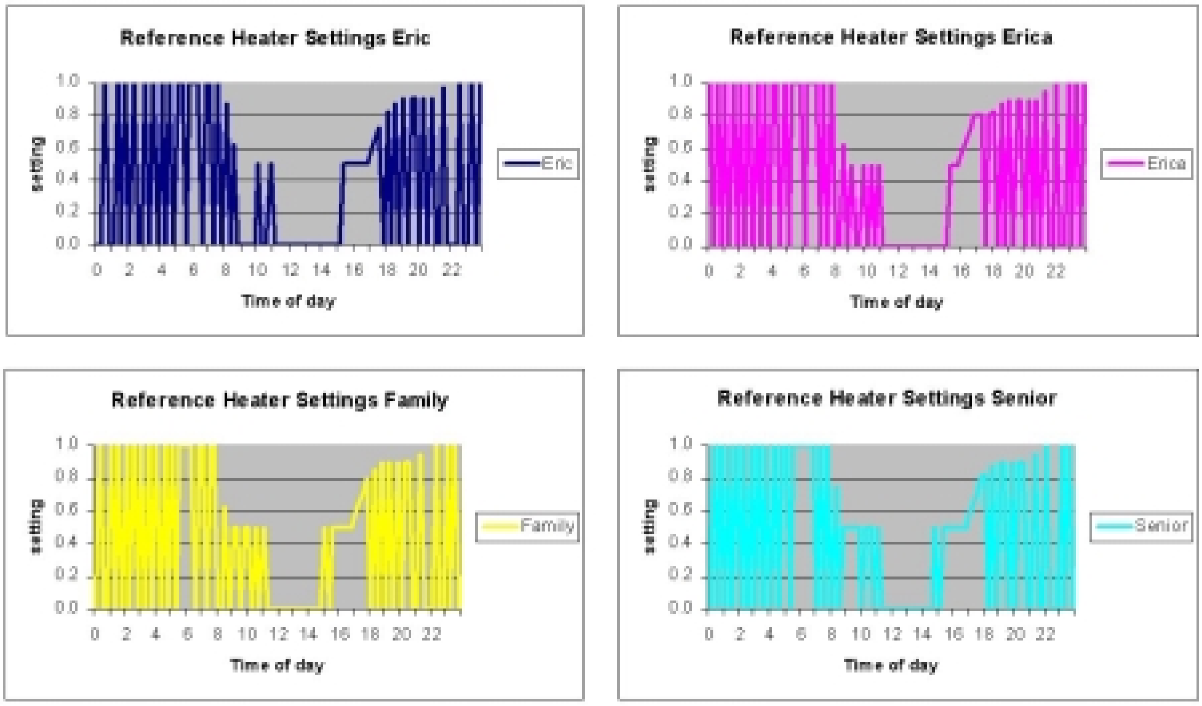


Figure 6.2. Heater settings for the different winter scenarios in the reference situation

In the winter the heater settings are the most relevant. However, we expect the COMFY system to control the sunblind as well, in order to make optimal use of passive solar heating.
The reference situation is characterised by a pattern of on-off switching of the heater, as we have seen before. As expected, the family and the senior make use of the heater during daytime. Eric and Erica have a ‘cooling off’ period after 9:00 hours.
In the optimal situation the control is more modulated. The heater is seldom switched off completely. Very clearly the preheating period before 8:00 hours (when the electricity price starts rising) can be seen in the scenarios for Eric, the family and the senior. The same effect can be seen before 10:00 hours (the senior), 16:00 hours and 19:00 hours. Erica is less cost aware and does not care about price changes.

In Figure 6.4 the sunblind settings are shown. All scenarios start with the same settings: closed at night and open during daytime. This is te be expected near optimal. The COMFY optimisation makes optimal use of the sunshine by opening the sunblinds at sunrise, thus improving energy efficiency up to the last sunbeam. After sunrise the sunvlind settings are no longer relevant, so they are unchanged.

Note that we have not modelled the isolation effect of the sunblinds. Undoubtedly COMFY would have closed them completely after 18:00 h. Even at daytime, when the solar radiation is not very strong, it could become more efficient to close the sunblinds if the effect of isolation is larger.

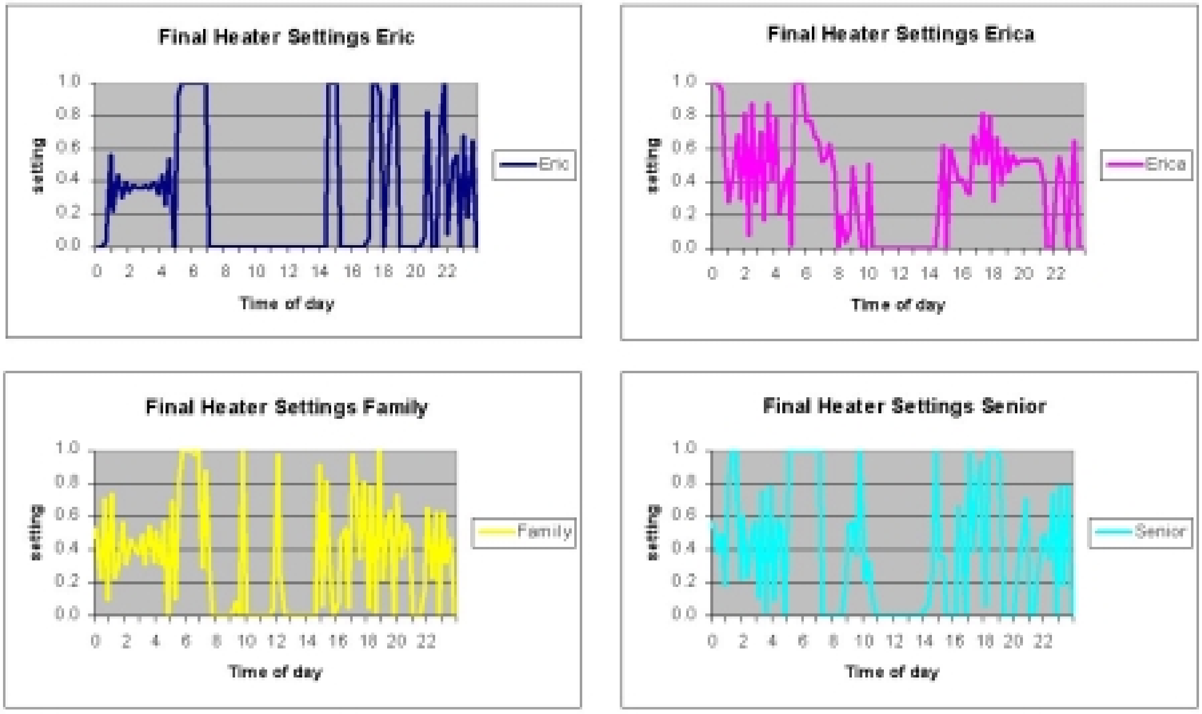


Figure 6.3. Heater settings for the different winter scenarios in the COMFY optimisation

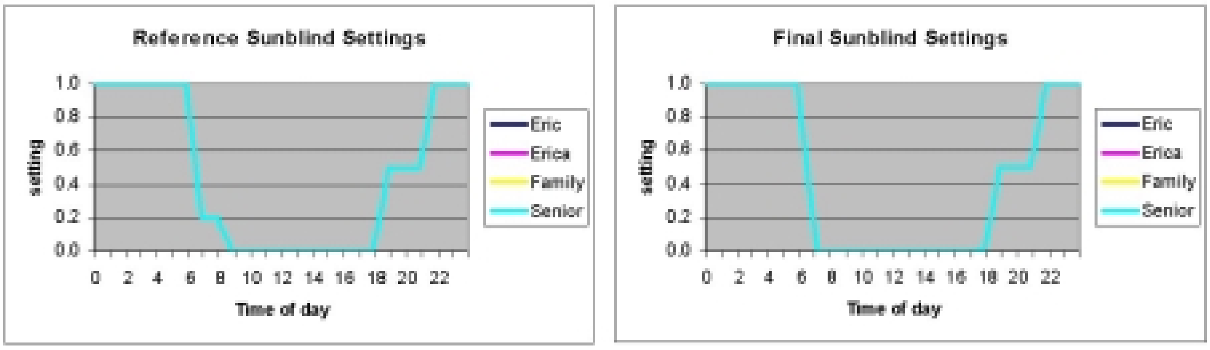


Figure 6.4. Sunblind settings (equal for all winter scenarios)
the reference situation and the COMFY optimisation

6.3.1.2 Comfort

The main component for comfort is the air temperature. The Fanger comfort index PMV also considers other factors, such as radiation temperature, humidity, air velocity, metabolism and clothing level.

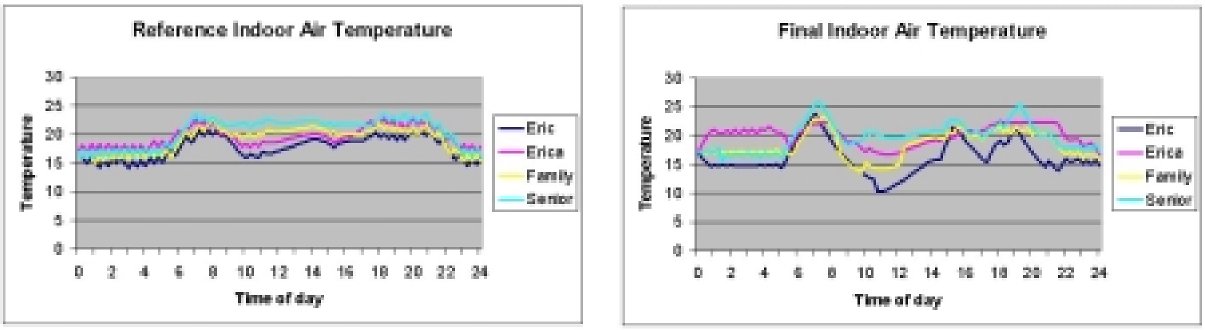


Figure 6.5. Temperature progress for the different winter scenarios
in the reference situation and the COMFY optimisation

In Figure 6.5 the room temperature is given for each scenario, both in the reference situation and with optimised control. The reference situation follows the thermostat control as close as possible. However due to the on/off switching of the heater the temperature fluctuates around these settings. The same is true for the Fanger PMV index (Figure 6.6). Note that changes in comfort are considered as less pleasant than a constant level.

The COMFY optimisation shows more differentiation. Eric definitely falls short on air temperature and comfort. Early in the morning at 7:00 hours the temperature reaches 24 °C and then slowly falls back due to high energy cost. The senior suffers from the same effect, yet at a higher temperature level. The peaks for Erica and the family are less prominent, since they are less cost aware and (especially Erica) prefer high comfort.
Note that in the COMFY situation less fluctuations on the short term occur, thus leading to a better comfort sensation. As might be expected the temperature and the PMV index rise just before energy cost rising. The base level is just below preferred level in the scenarios which allow larger deviations. It is clear that COMFY has a preference for lower values since this saves energy and cost. This is a disadvantage of the model as we have defined and may be under study in a next version of COMFY.

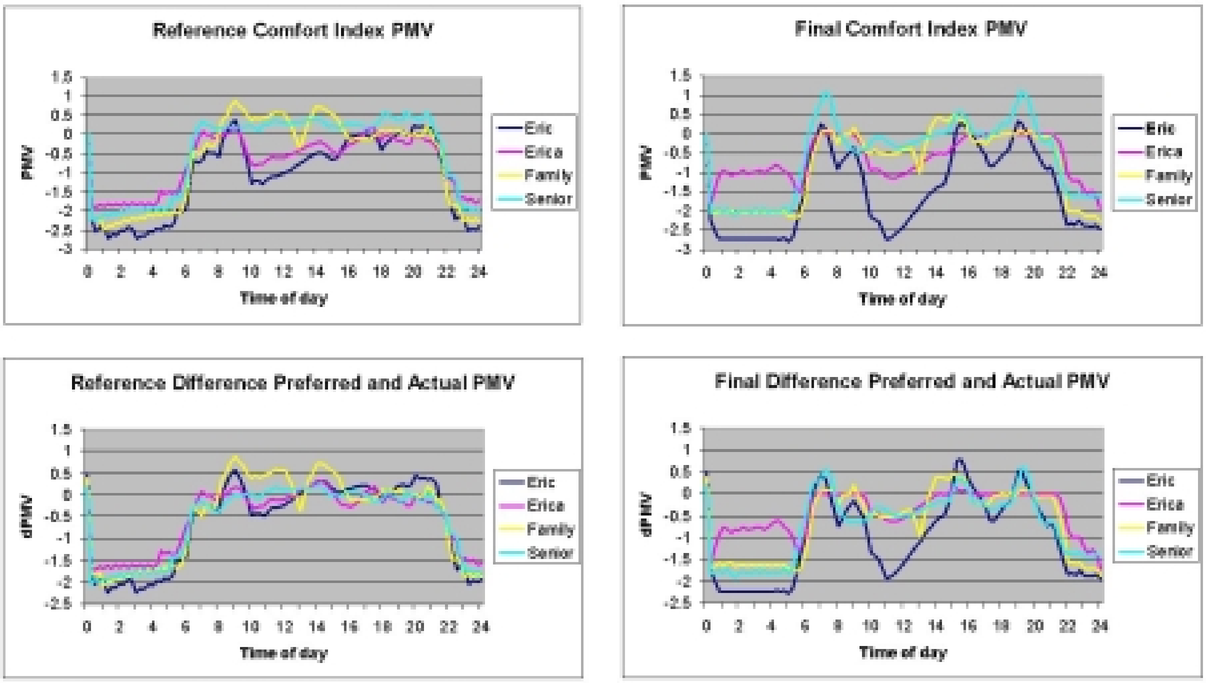


Figure 6.6. Comfort index for the different winter scenarios in the reference situation and the COMFY optimisation

COMFY: Senior:Preheating effect at 7:00 h. After this time the cost rises. Abstinence of heating until almost 9:00 h. leads to less comfort. The difference between actual and preferred confort however never becomes very large. Preheating also occurs before 10:00 h., 16:00 h. and 20:00 h.
Eric has the same effects. Although he is not at home we see preheating at 15:00 h.
The profile for Erica is completely different. Because she is not concerned about cost she gets the comfort she prefers when she is present. Fluctuations in comfort level are minimal.
The family profile shows the same effacts as for Erica, however less pronounced. The optimised comfort index shows a more regular pattern than in the reference situation.

6.3.1.3 Energy and cost

Figure 6.7 and Figure 6.8 show the potential cost and energy efficiencies of the COMFY optimisation. As can be seen in the graphics all scenarios show improvement in effinciency for both cost and energy.
Eric definitely gains the most, which is not surprising, since he is cut down on comfort. Further tuning might give him slightly better comfort with less efficiency. The family and the senior also save a large percentage of cost and energy. Even Erica, getting improved comfort, pays less.
Note that the senior uses slightly more energy than Erica, but pays considerably less. The senior seems to make the best use of periods with low energy prices.

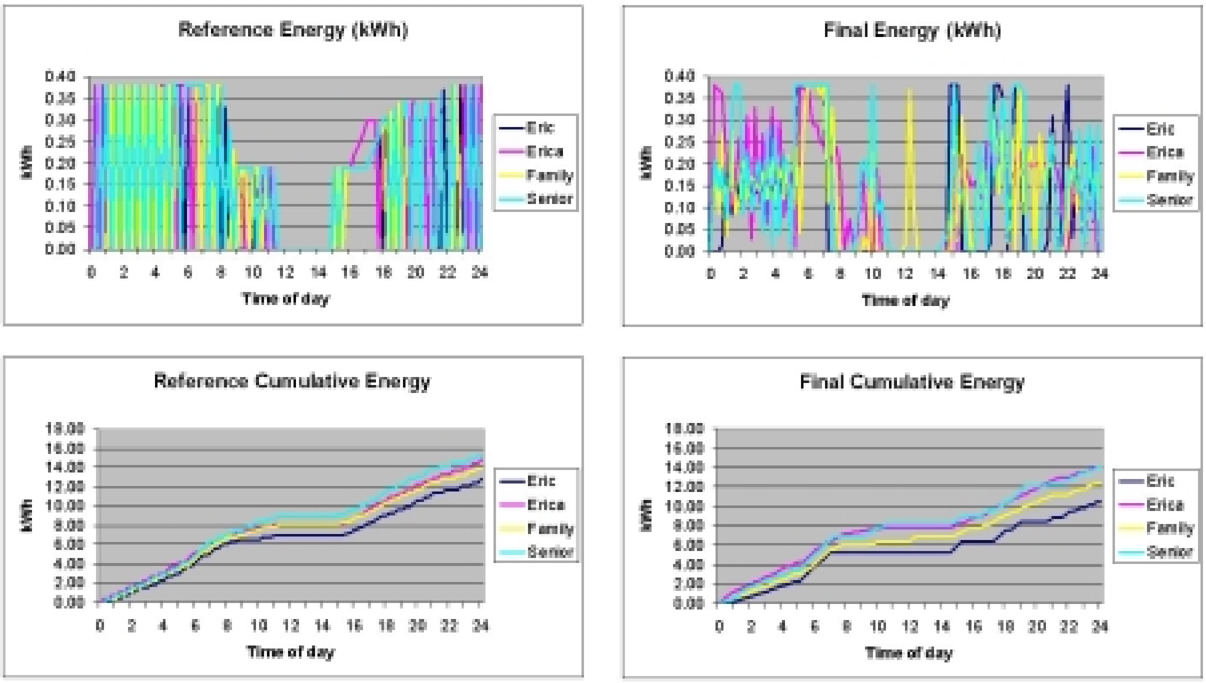


Figure 6.7. Energy usage for the different winter scenarios in the reference situation and the COMFY optimisation

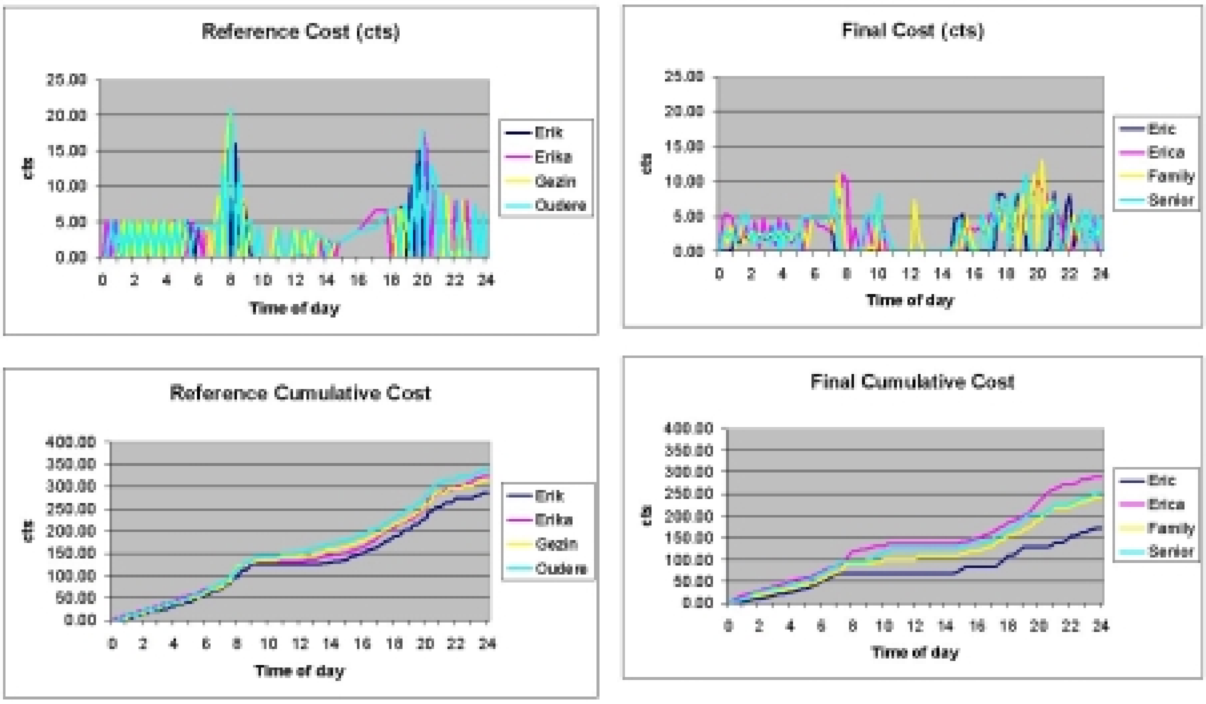


Figure 6.8. Cost for the different winter scenarios in the reference situation and the COMFY optimisation

6.3.2 Summer

6.3.2.1 Settings

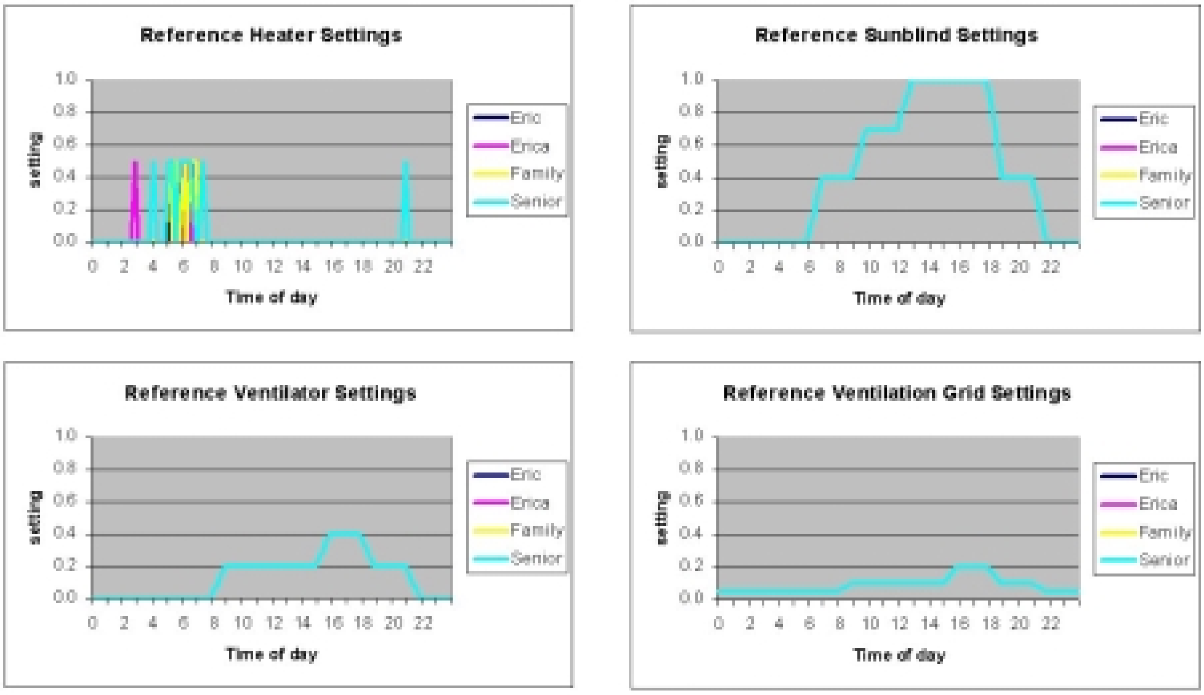


Figure 6.9. Heater, sunblind, ventilator and ventilation grid settings for the different summer scenarios in the reference situation

In the summer the heater is only used early in the morning. COMFY explicitly makes use of low prices before 7:00 h in order to preheat the room.
Note that it may not be realistic that the heater is used at a summer night. This might be due to flaws in the thermodynamic model.

More interesting are the controls of the sunblinds and the ventilation. As can be seen from Figure 6.10 to Figure 6.12 it is most essential to close the sunblinds at daytime to prevent the solar radiation leading to too warm comfort. Also, the ventilator and the ventilation grid are used more extensively with COMFY control.

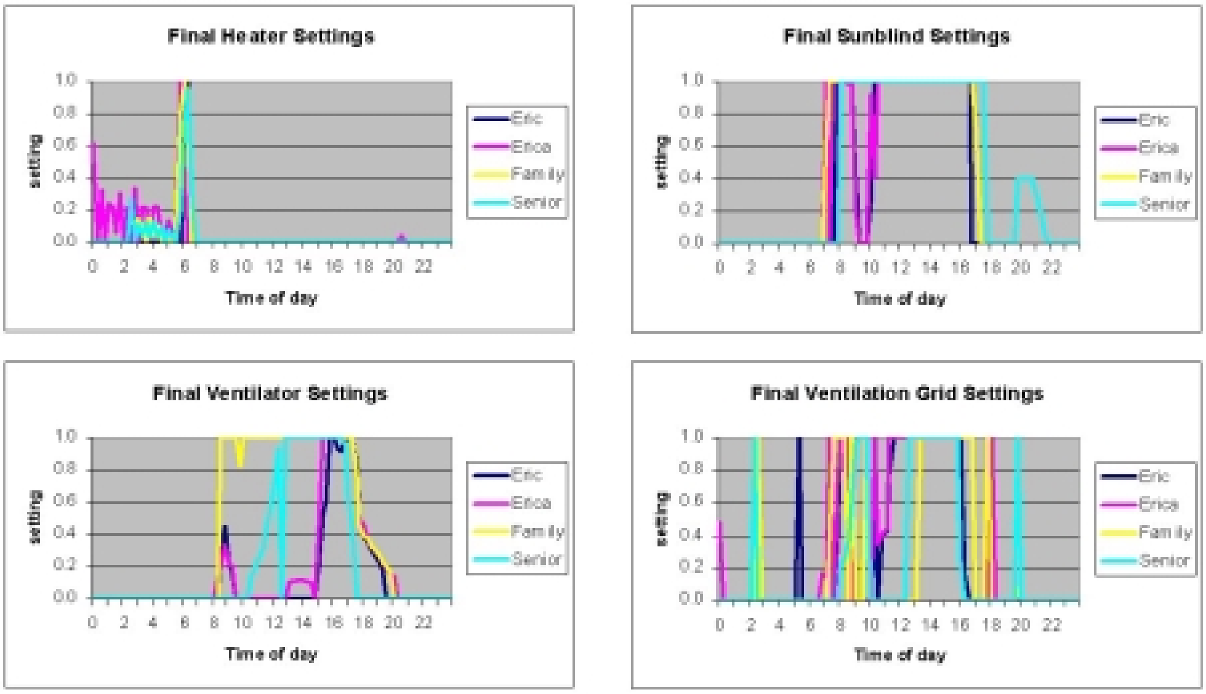


Figure 6.10. Heater, sunblind, ventilator and ventilation grid settings for the different summer scenarios in the COMFY optimisation

6.3.2.2 Comfort

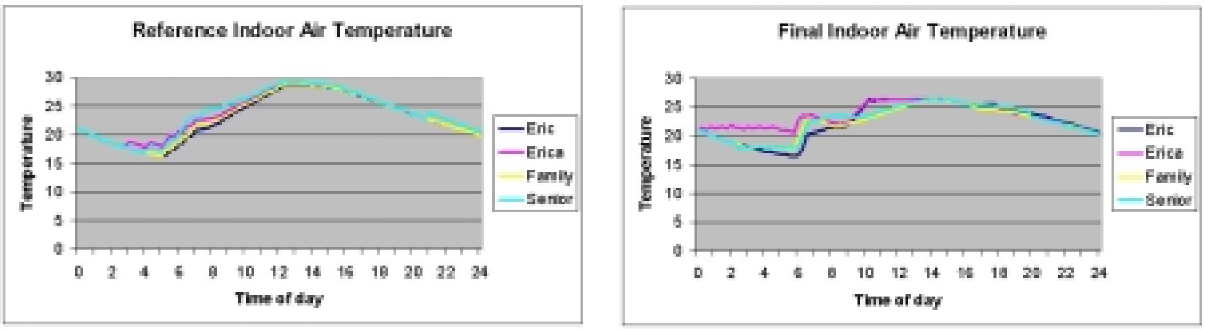


Figure 6.11. Temperature progress for the different summer scenarios in the reference situation and the COMFY optimisation

From Figure 6.10 and Figure 6.11 it can be seen that the COMFY control leads to improved air temperature and more optimal comfort as compared with the reference situation. Even Eric gets near-optimal comfort, since control of sunblinds and ventilation can be delivered at minimum cost. The family comfort control is the weakest. The reason clearly is the high activity level compared with the others, which leads to a warmer comfort sensation although air temperature is at the same level.

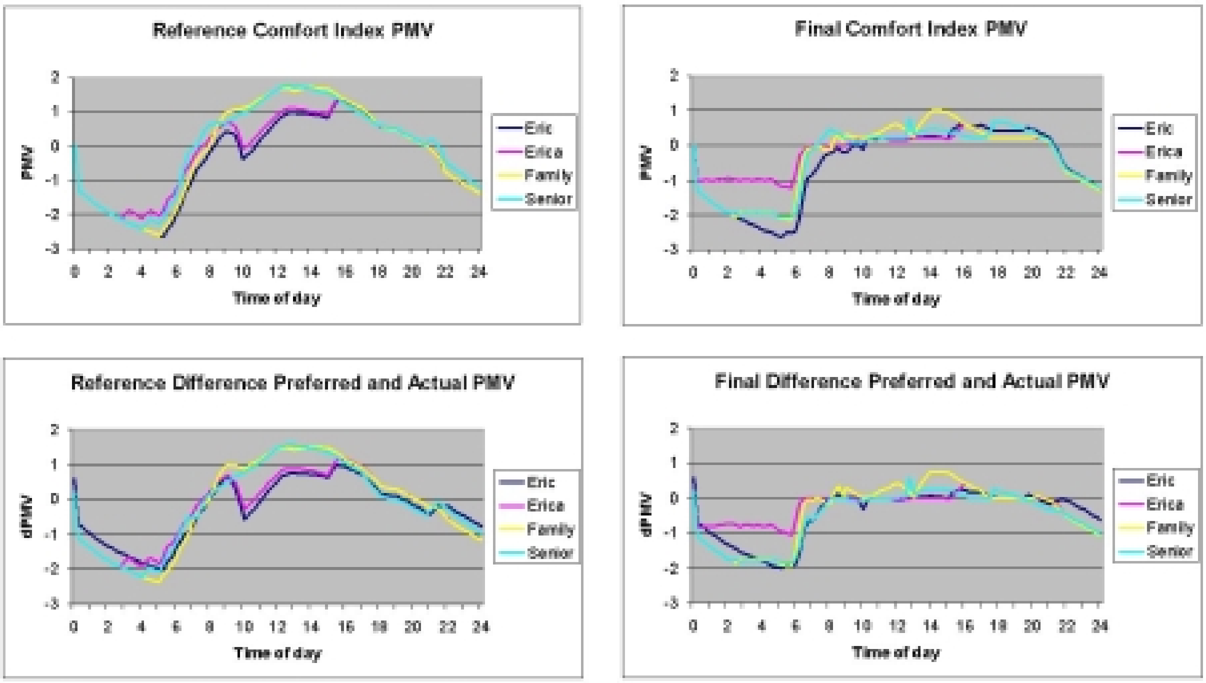


Figure 6.12. Comfort index for the different summer scenarios in the reference situation and the COMFY optimisation

6.3.2.3 Energy and cost

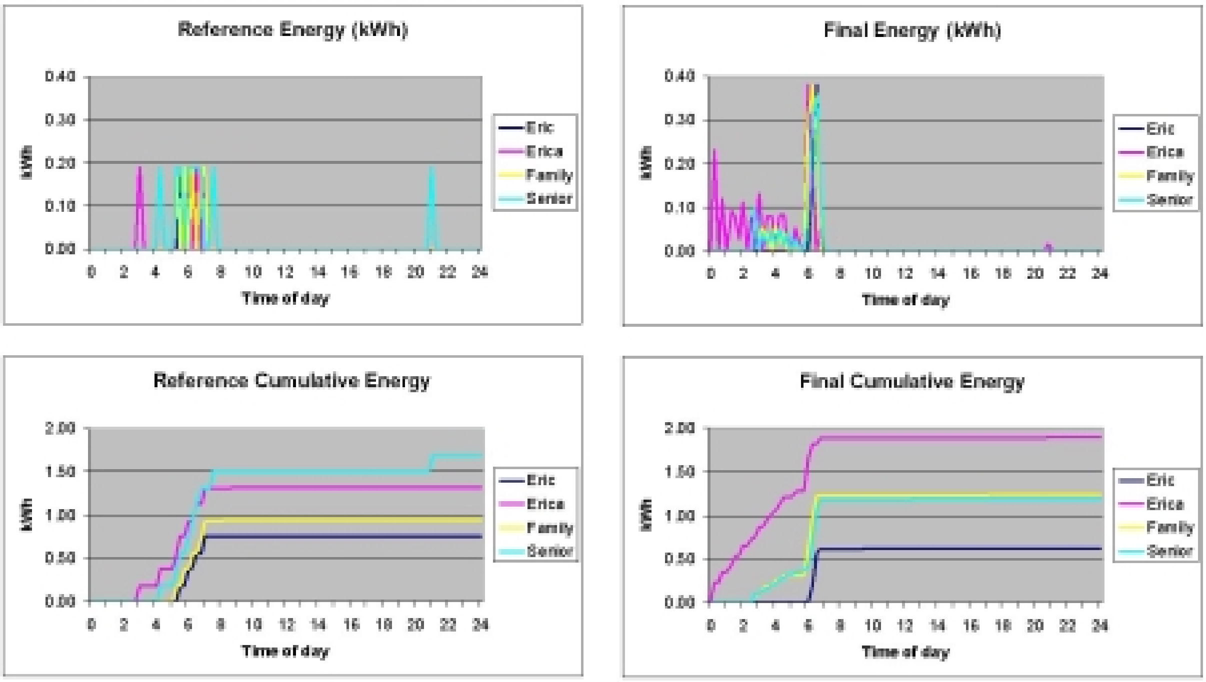


Figure 6.13. Energy usage for the different summer scenarios in the reference situation and the COMFY optimisation

Although energy usage and hence cost are expected to be minimal in the summer season, we see a different effect from the winter scenarios. The cost for Erica and the family rises with COMFY control, which is due to the fact that COMFY wants to deliver optimal comfort at the start of the day (7:00 h), which requires usage of the heater.

Eric and the senior both improve their cost and energy efficiency in the summer. Both also get improved comfort.

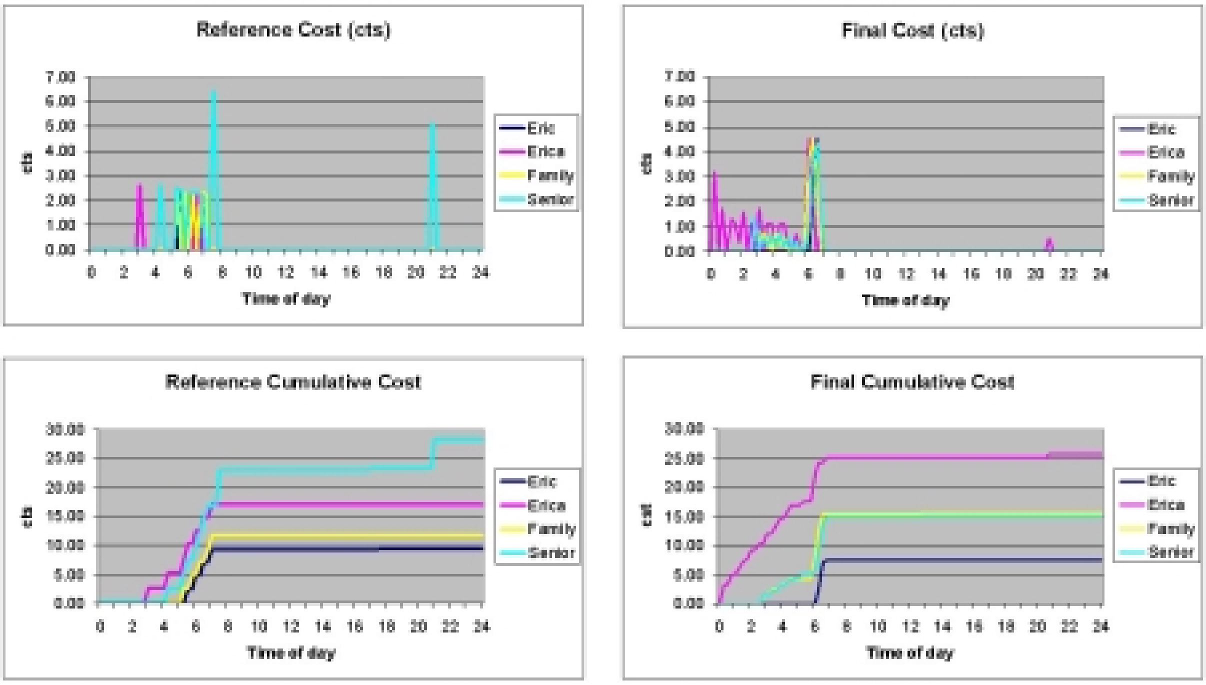


Figure 6.14. Cost for the different summer scenarios in the reference situation and the COMFY optimisation

6.3.3 The reference year

In order to get a picture of the COMFY effects over a complete year we also simulated COMFY control over the DEGO reference year. In order to get a representative result we used the Family comfort settings from the previous paragraphs..

6.3.3.1 Settings

It is not very useful to give a picture of the settings of the controls, since the detail will not show in a year graphic.

6.3.3.2 Comfort, cost and energy

Figure 6.15 to Figure 6.18 give an impression of the comfort control over the refence year for the family situation. We see an overall efficiency improvement of around 20%, both in energy and cost efficiency. The main conclusions are:

- COMFY shows less bandwidth in comfort than the reference scenario.
- COMFY shows less peaks during the summer period, although no direct cooling is used.
- In the cold periods there seems to be no significant difference in behaviour, athough COMFY uses significantly less energy.
- The difference between preferred and actual comfort for COMFY is concentrated around zero and keeps on thelow side. This is easyly explained by the fact that COMFY optimizes comfort on minimal energy and cost and hence will not provide more comfort than needed. The difference between preferred and actual comfort in the reference situation significantly lies above the zero level. This might be improved by setting a lower reference temperature. In summer it is always too warm, since no optimal use is made of the sunblinds.

We also ran the luxury scenario of Erica for the reference year. Although Erica is using slightly more energy (< 2%), her comfort improves significantly, both in the smmer and in the winter.

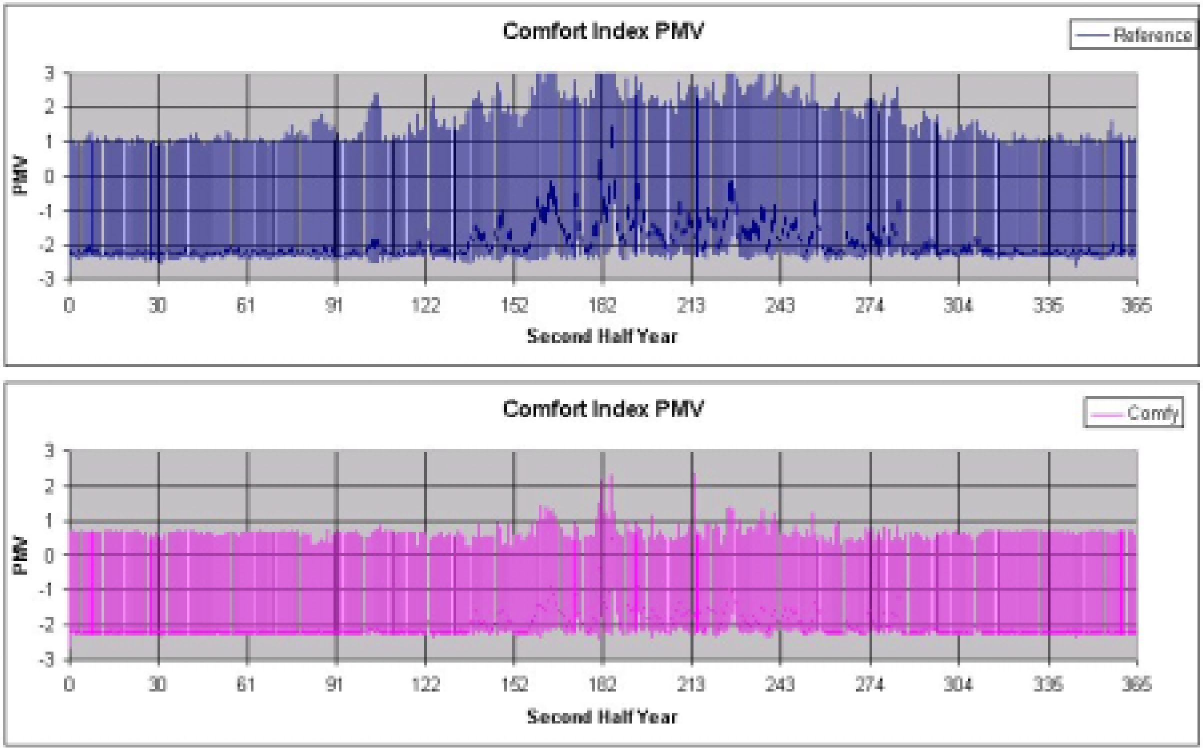


Figure 6.15. Comfort index for the reference year in the reference situation and the COMFY optimisation

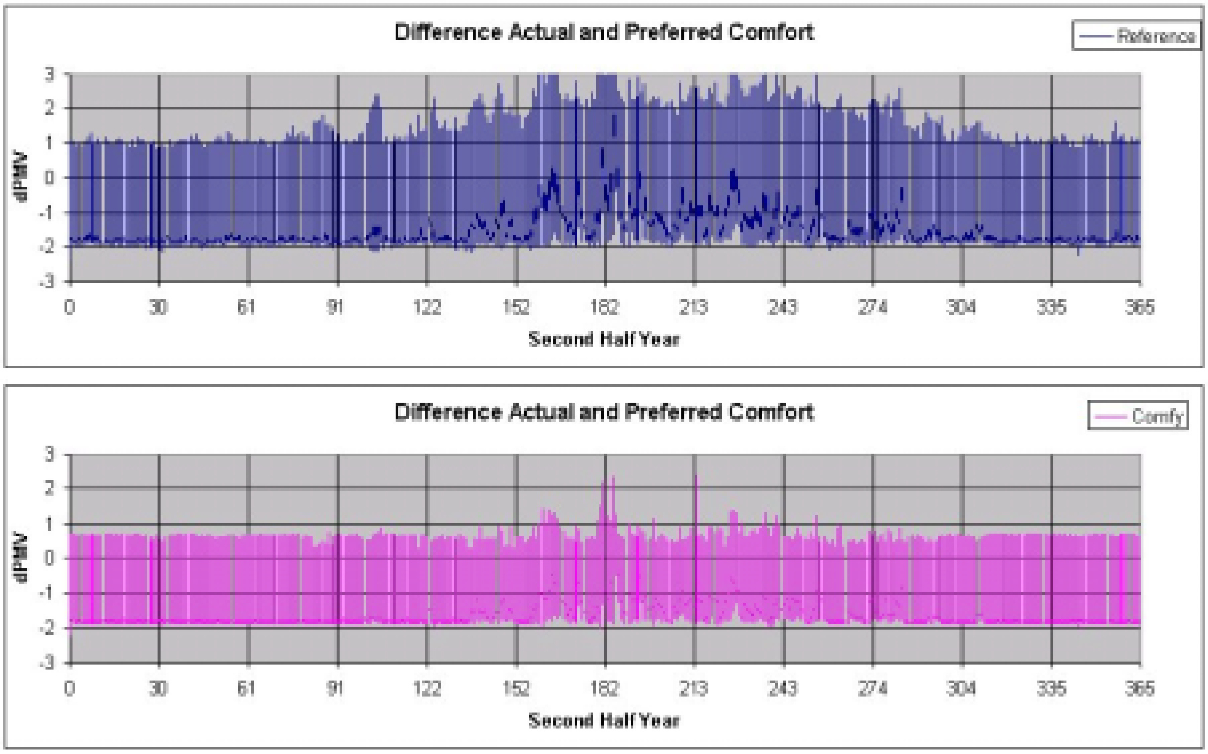


Figure 6.16. Difference between actual and preferred comfort for the reference year in the reference situation and the COMFY optimisation

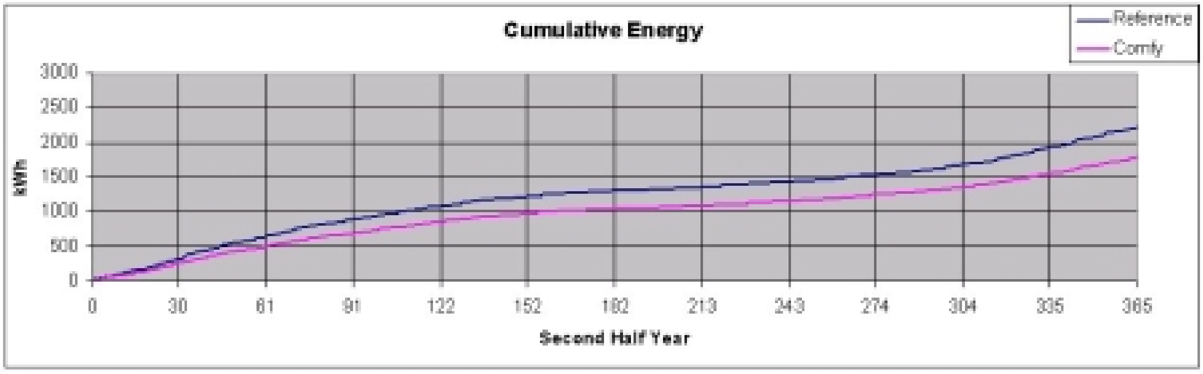


Figure 6.17. Energy usage for the reference year in the reference situation and the COMFY optimisation

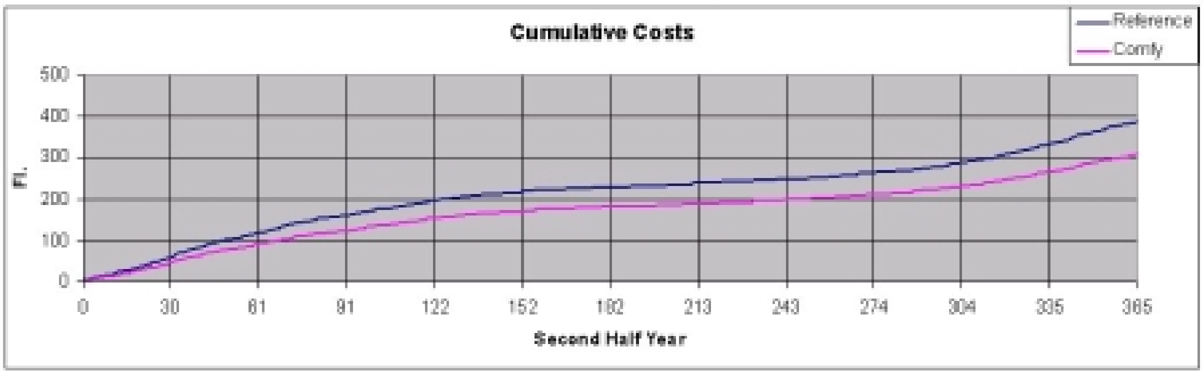


Figure 6.18. Cost for the reference year in the reference situation and the COMFY optimisation

6.3.4 Utility buildings

Above scenarios are based upon a standard living room in a residential home. The SMART project [2] will project the COMFY approach on the office environment. There are several differences between a residential environment and an office environment:

- An office environment has to provide comfort during working hours, i.e. 8:00 – 18:00 hours; the home has to provide comfort on presence, which in many cases will be early in the morning and in the evening.
- In an office environment user profiles are more or less predictable, e.g. metabolism and clothing; the home user will act less predictable.
- The office user has less need to control on cost; a large percentage of the home dwellers will be more cost aware.

In order to get a feeling what COMFY might achieve in an office scenario we modelled this user behaviour as well.

Note: energy prices profile is different from the above scenarios.

6.3.4.1 Profiles

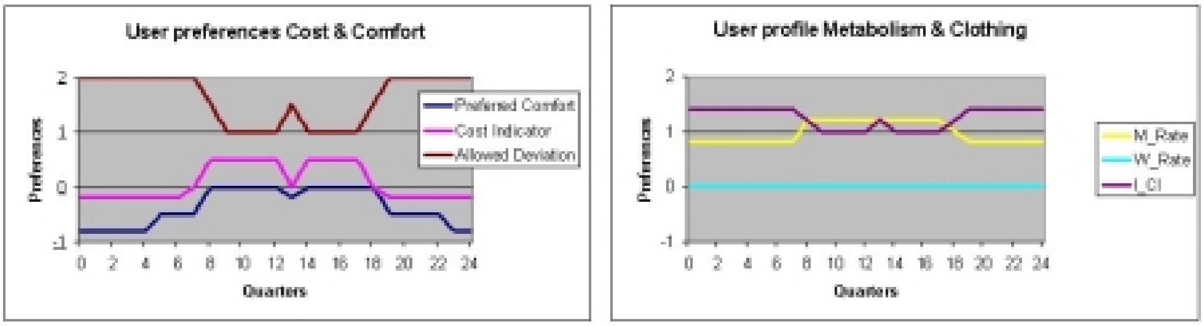


Figure 6.19. Left the user profiles for comfort, allowed deviations and cost indicator in an office environment. Right typical activity and clothing profiles in an office environment.

6.3.4.2 Settings

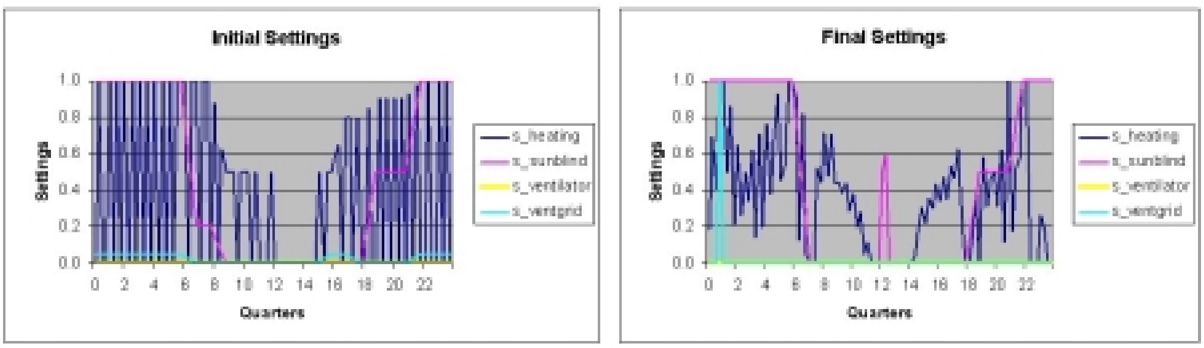


Figure 6.20. Left the initial and Comfy-controlled temperature progress for indoor air temperature; as a reference also the outdoor temperature and the thermostat settings are given. Right the initial and Comfy-controlled comfort; as a reference the preferred comfort is given.

6.3.4.3 Comfort

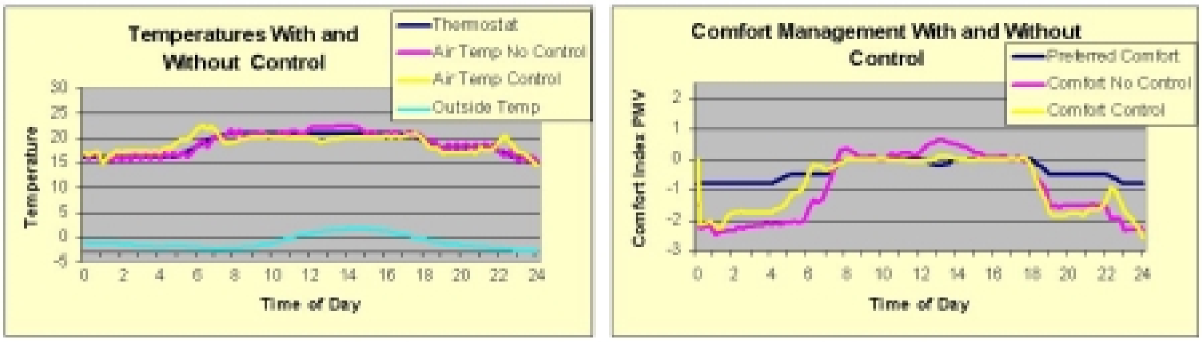


Figure 6.21. Left the initial and Comfy-controlled temperature progress for indoor air temperature; as a reference also the outdoor temperature and the thermostat settings are given. Right the initial and Comfy-controlled comfort; as a reference the preferred comfort is given.

6.3.4.4 Energy and cost

The following graphics are presented:

- Energy prices, energy usage with no control and with COMFY control.
- Cumulative energy with no control and with COMFY control.
- Cumulative cost with no control and with COMFY control.

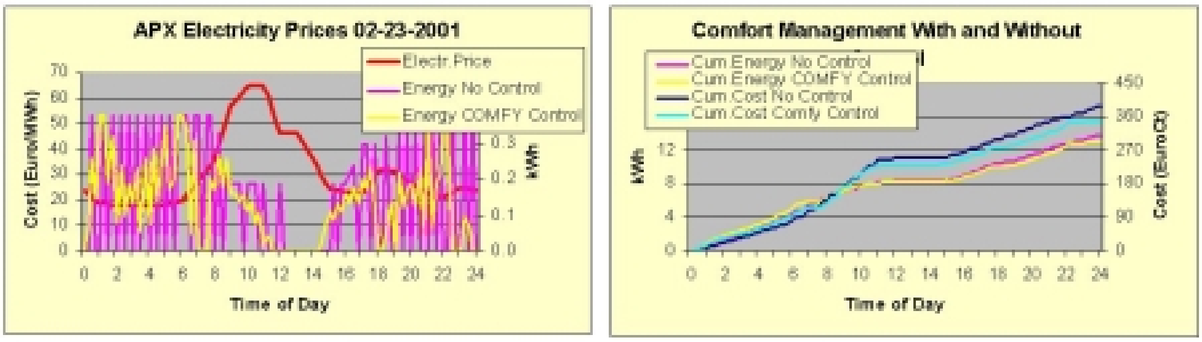


Figure 6.22. Left the electricity usage for each period; note that the COMFY-controlled situation avoids energy usage when prices are high.
Right the cumulative progress of the energy usage and the corresponding cost.

6.3.5 Potential efficiencies

We have tried to define the reference scenarios as faithfully as possible. However, they are based on electricity heating and a very basic thermodynamic model.

The COMFY optimisation shows that a definite improvement can be made with respect to the reference situation. Improvement is made both in energy and cost efficiency and in comfort: improved comfort against lower energy usage and lower cost.

	Eric		**Erica**		**Senior**		**Family**	
	Winter	**Summer**	**Winter**	**Summer**	**Winter**	**Summer**	**Winter**	**Summer**
comfort	–	++	++	++	++	++	+	+
cost	++	+	+	– –	++	++	+	–
energy usage	++	+	□	– –	□	++	+	–

Table 6.7. Potential efficiency for different scenarios

Several remarks:
Especially in the winter the comfort tends to be on the low side compared to the preferred comfort. This is due to the fact that the optimisation rewards delivery of less comfort since it leads to low energy / low cost. This effect may be neutralised by constraining the total difference between actual and preferred comfort over the periods of presence.
The weighing of cost against comfort is dominated by cost in the winter (they make out ± 90% of the total utility). In the summer the weighing is more balanced. Further study might lead to adjustment of the utility function.
The energy usage is not expressed directly in the utility function, but is reflected indirectly by its cost. Inclusion of energy usage also can lead to adjustment of the utility function.

References

[1] Van den Broek, A., W. Knulst, K. Breedveld. *Naar andere tijden; tijdsbesteding en tijdsordening in Nederland, 1975 – 1995*. Sociaal en Cultureel Planburo Studies – 29. See also www.scp.nl.

[2] Kamphuis, I.G.: *Smart projectplan*.

7. MULTI-PARAMETER OPTIMISING PACKAGE COMFY-3 DESCRIPTION.

7.1 Modules

The comfy front-end is implemented in a number of Java2-packages, based on GUI Swing components. These are:

- `batchrun`. Containing the modules to invoke a DOS-session with a batchrun.
- `day`. Containing the attributes of a day.
- `gui`. Containing the graphical user interface primitives.
- modelstorage. In COMFY there is the opportunity to store all relevant data in a serialized fashion on disk.
- `plots`. The plotting package. A number of plots is defined in a standard manner.
- `profiles`. Files in this package contain the timeseries settings as profile information.
- `room`. This package contains the room specific attributes.

In the following figure the directories are shown.

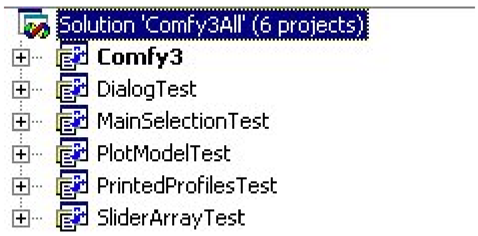


Figure 7.1 Directory tree of COMFY-3

Apart from the packages a number of module testing programs are part of the development environment. These files are in packages preceded by Test…. As a plotting package JCChart version 3.6 is used from KL Group. The plotting package is in `jcchart36-classes.zip`. For user interface primitives Swing classes are used. In the current development environment these contained in `swingall.jar`. Both files have to be in your CLASSPATH for things to work. The package structure of the Comfy3-directory is shown in figure 2.

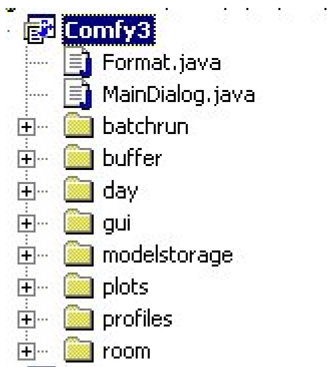


Figure 7.2 Package structure of COMFY-3

In the Format-class is a collection of 'C'-like textbook formatting-routines (`atof, atoi` and reverse). In MainDialog.java the main-menu and the main screen is composed. The larger packages are:

- `batchrun`. This packages contains the flat-file communication mechanism to the COMFY-backend, COMFORTRAN. The structure of batchrun is as depicted in figure 3.

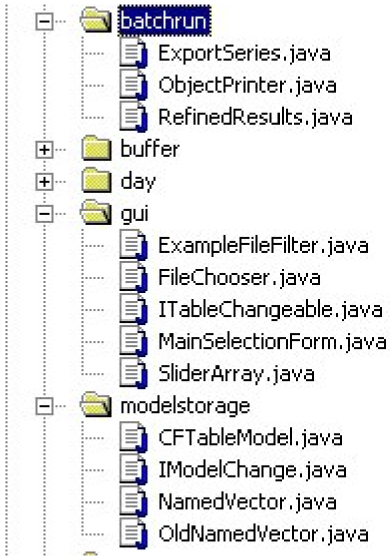


Figure 7.3 Package batchrun,gui and modelstorage

Functions exist for exporting the relevant user data as flat files (`ExportSeries` and ObjectPrinter) and for reading back the values as produced by Comfortran for plotting (RefinedResults).

- `gui.` In this package a standard swing file handling filter and dialog is implemented. The `ITableChangeable` -interface names a number of functions for handling time-series and individual field data in a uniform way.
- `Modelstorage.` In this package internal storage of the profiles and the transfer to disk is programmed. For compatibility reasons for previous versions a deprecated data-type is included. The main class is CFTableModel, which contains the root of the COMFY-3 data-structures and serialisation structures.
- `Plots`. In this package the link to the JClass36-package is laid. Main interface to the other classes is `PlotModel`. Once a time-series is defined as `IPlottable,` it can be addressed by any specific plotting module.

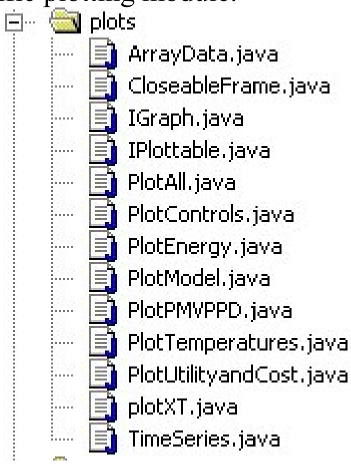


Figure 7.4. Plot related sources

- Profiles. In this package default initialisation takes place. The packages contained are listed in figure 5.

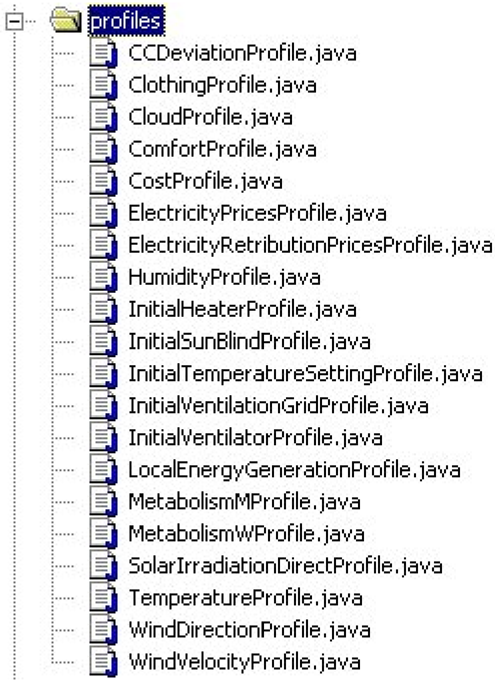


Figure 7.5 Profile containing packages

7.2 Installation

The program sources can be unpacked by unzipping the file delivered. The package should contain the above-mentioned files in the src-directory. Furthermore:

1. `Comfy3.exe`. The graphical front-end.
2. `Comfortran.exe`. The optimising engine.

3. `JClass36s.zip`. A collection of graphical classes, that are used for producing graphical output.
4. `Swingall.jar`. The swing-classes in JDK 1.2
5. This document.

For operation of COMFY-3 it is necessary to have the Fortran optimiser "`comfortran.exe`" in the `c:\`-directory. The front-end software requires a Java Virtual Machine to be installed in your computer.
Comfy-3 uses some temporary files, which are copied from the CDROM including their protection. Remove the read-only protection-flag of all files copied to the hard-disk, otherwise problems deleting files from previous COMFY-3 runs may occur.

7.3 Graphical Input Functions

7.3.1 The main menu.

The graphical front-end allows easy input of a large number of parameters. Once the comfy3.exe program has started the following screen appears:

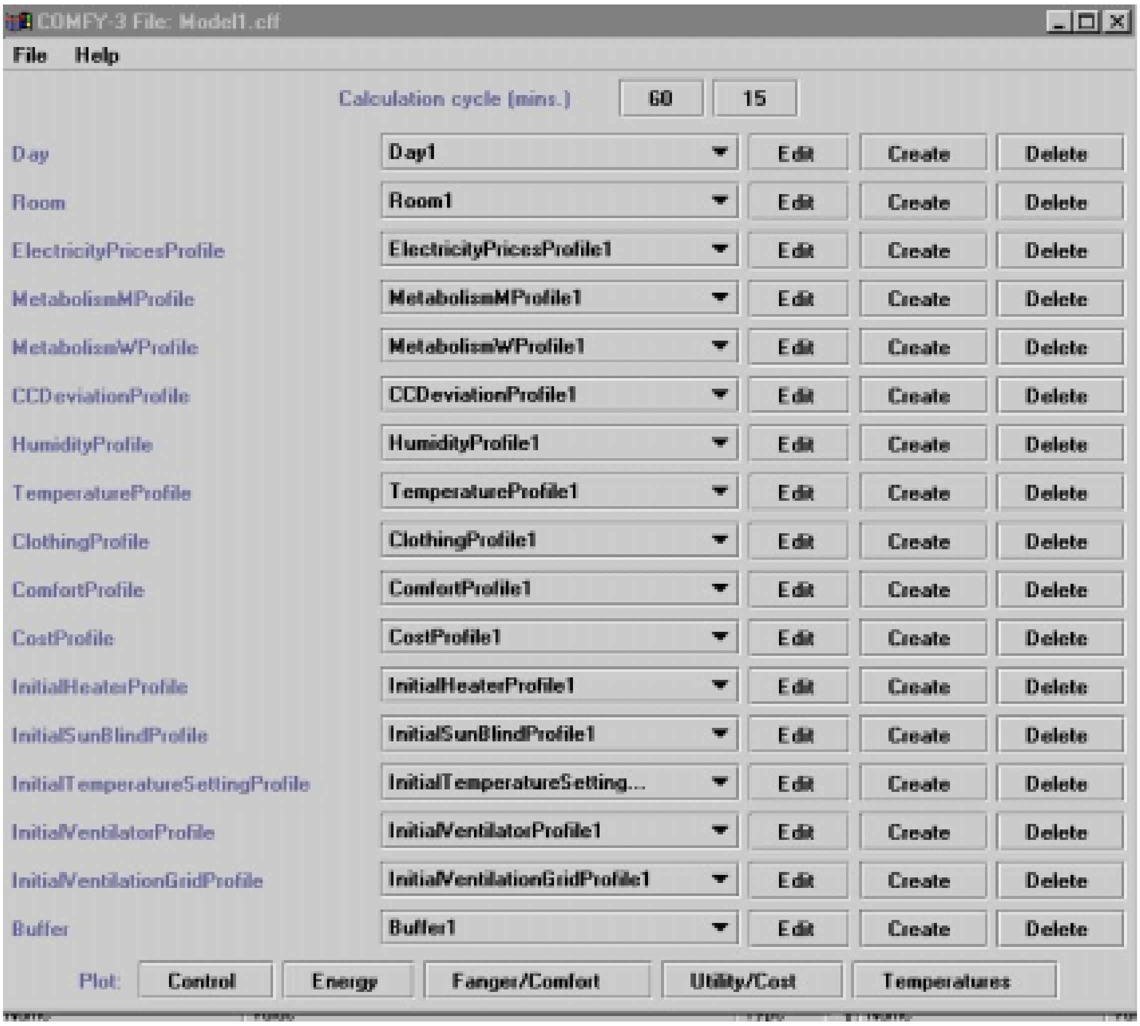


Figure 7.6 Main screen of COMFY-3

Edit and delete operations take the current record as target. On create a default tabel is made. Deleting should result in at least one record to be kept. Via the File-menu it is possible to store all model information in a .cff-file. The file menu also has a function to export the current model in a format to export to other packages. The TimeProfile-submenu gives the user the opportunity to give input parameters, that are a function of time, to the system. The Properties-submenu lets the user fill-out specific parameters, that are described later on. The pane on the screen describes the current setting. All time-profiles have an identifier, that is displayed with it's current value. Two buttons enable calculations with an hourly and a 15-minutes scale. At last, selection of a number of plots is possible. This selection is only possible if calculated parameters are in the model-file. The FileMenu is clarified in the following screen:

Figure 7.7 File menu for COMFY-3

A tabular output of results of the calculations can be made for further analysis with Excel. Furthermore a .2xl-file is generated by the calculation back-end, which can be imported by Excel. The Import and Export functions allow exporting and importing the current model information in an ASCII-formatted way. Imported information extends the current model information.

7.3.2 Time-profiles

A number of time-profiles can be set on an hour scale. By clicking the white area of the current setting the corresponding profile is shown. A time-profile-input screen is shown below:

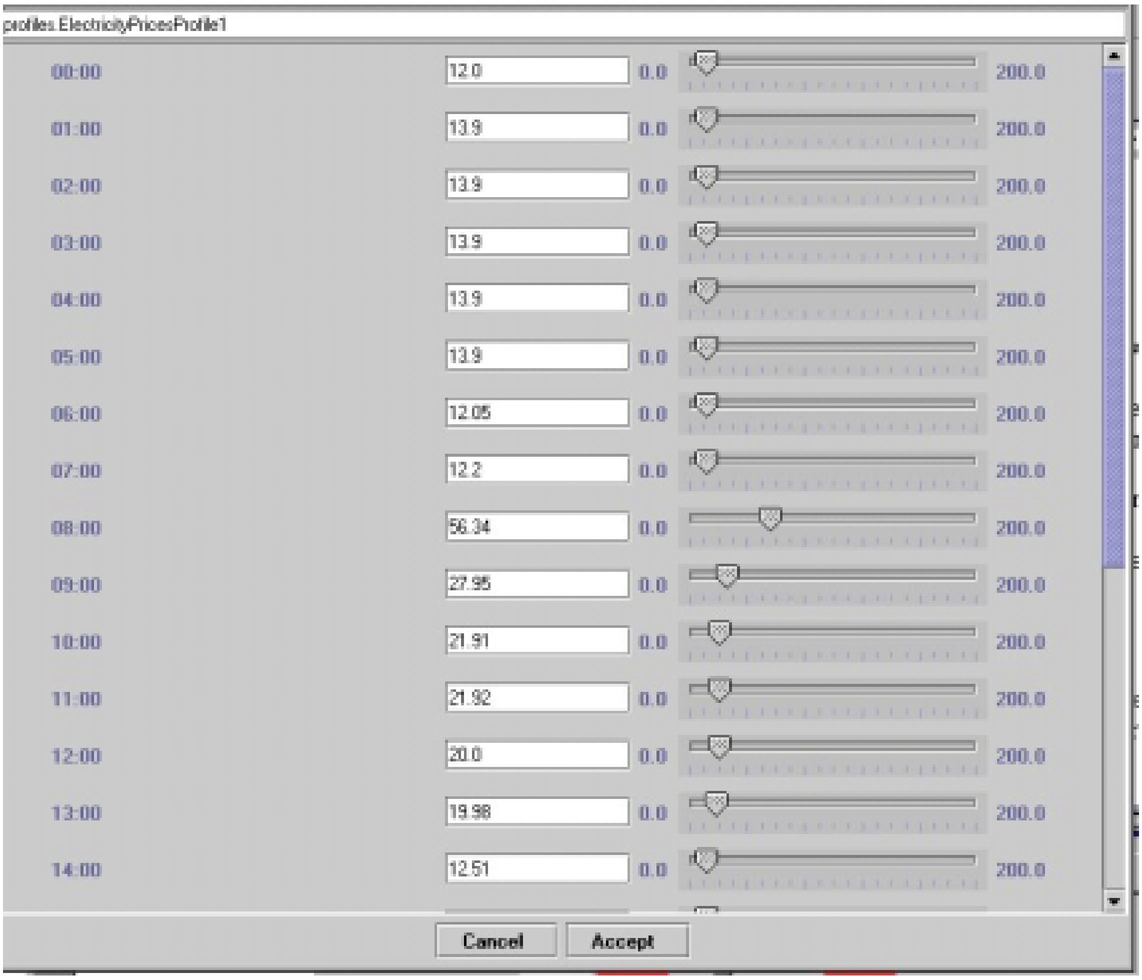


Figure 7.8 Time profile input

The profiles are shown as hourly changeable values. The track-bars can be used to alter the values. Direct input in the fields is also possible. The name of a profile can be altered using the textfield on top. The profile-input also serves input of room and day-attributes.

7.3.3 Saving the model information

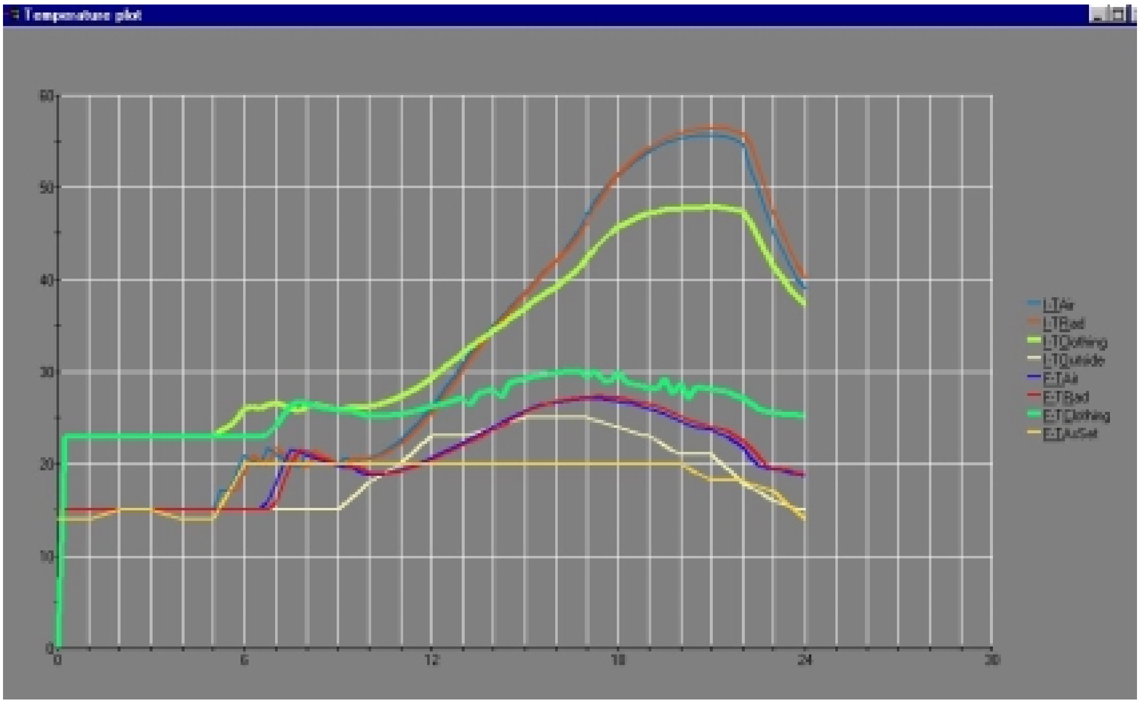
Model settings may be saved on COMFY-3 model files, which have the .cff extension. Apart from the currently selected values, all other variants are kept in this file. In the COMFY-3-release delivered, a number of reference situations can be found. COMFY-3 starts with a model file scenario1.cff. Apart from the input parameters, the output parameters are also contained in the model file for later inspection.

7.3.4 Inspecting the model information

Once a calculation has finished, plots can be made of a number of output parameters. There are four plot-types defined:

- A temperature versus time-plot. In this type of plot the inside radiation, air and clothing temperatures, outside and temperature settings are displayed.
- Control signal plot. In this plot the calculated initial and optimized control settings are displayed as 0 to 1 values.
- The PMV(Predicted Mean Value)/PPD(Predicted Percentage Dissatisfied) plot. This plot shows the 24 hours behaviour of Fanger's comfort indices.
- The utility, energy and cost plot. In this plot the cost, utility and energy usage as a function of time are displayed.

In the plots, in the legend characters are underlined. If these characters are entered, the corresponding curve is highlighted. E.g. pressing the 'I'-key highlights the initial values. The 'u'-key highlights the utility function. Pressing the right-mouse key in a plot gives access to standard charting functionality as adapting the settings to personal preferences. Holding the shift-key and the left-mouse button simultaneously allows zooming into the graph. The control-key/left-mouse button makes the chart area selected moveable. The 'r'-key resets the editing on the graph. Finally, pressing the 'p'-key make you enter a standard Windows print dialogue. A sample graph is shown below:



7.4 Model calculation part

Once the input-parameters are set, the model optimisation can start by pressing the calculate button.. The model optimisation is started in a separate DOS-box. After the optimisation the DOS-box pauses. Give a simple return and the graphical front-end takes over. Currently, quarterly and hourly intervals are calculated. Half-hourly and minute intervals are under implementation.

7.5 Export to other packages

On the file-menu the export-option can be chosen to export the parameters in a format suitable for EXCEL or other packages. The current model is written to *modelname*.dat in the current directory in a comma-separated format.

7.6 Fortran optimization code

7.6.1 Comfortan

The Comfortan program is used for the mathematical calculation of the optimal comfort. It receives its input from the COMFY2 user interface program through the files `FromComfyUI.upd` and `scenario.dat`. Comfortran uses the `E04JYF` subroutine from the NAG Library to calculate the optimum of the COMFY total utility function.
The main output files are `ToComfyUI.out`, `InitialSettings.plt` and `FinalSettings.plt`. These files are interpreted by the COMFY2 user interface in order to produce its graphic output.

`ComfySim.f90` is the main program for the COMFY optimisation:
Input file `FromComfyUI.upd` contains `plan_horizon` and `n_of_plan_periods`. The plan horizon typically is 24 or 96 periods (based on one day with period of 1 hour resp. 15 minutes). The number of plan periods normally is 1 (one day). Each next plan period adds another iteration step one period ahead of the previous step. The program loops over all plan periods by calling each time the `Comfortran.f90` subroutine.

The `Comfortran.f90` subroutine contains the basis program structure:

1. Defines the output files .
2. Input of the scenario (file `scenario.dat`), calling subroutine `ComfyInput.f90`.
3. Calculating the reference scenario, calling `ReferenceSettings.f90.`
4. Printing final results for reference scenario, calling `Total_net_utility_sub.f90.`
5. Calling `E04JYF`, the optimisation function from the NAG library.
6. Printing final results for optimal scenario, calling `Total_net_utility_sub.f90.`
7. `Buffer_function.f90` and `ThermoPhysical_Variables.f90` are used to reset the initial values at a next iteration (`n_of_plan_periods` > 1).

The `Total_net_utility_sub.f90` calculates the value of the total net utility function:
The subroutine makes use of `Buffer_function.f90`, (in order to calculate the buffer usage), `ThermoPhysical_Variables.f90` (in order to calculate the thermophysical effects in the room, `Fanger.f90` (in order to calculate the Fanger PMV-index), `Utility_function.f90` and `Cost_function.f90` (in order to calculate the comfort and the cost utility).
`Total_net_utility_sub` is also passed to `E04JYF`, since this function has to be optimised.

7.6.2 E04JYF Description

For a complete specification of the routine see the NAG Fortran Library Manual [1].

`E04JYF` is an easy-to-use quasi-Newton algorithm for finding a minimum of a *function F(x(1), x(2),..., x(n))*, subject to fixed upper and lower bounds of the independent variables *x(1), x(2),..., x(n)*, using function values only. The algorithm is based upon [2].

It is intended for functions which are continuous and which have continuous first and second derivatives (although it will usually work even if the derivatives have occasional discontinuities).

Mathematical Description
This routine is applicable to problems of the form:

$$\text{Minimize } F(x_1, x_2, ..., x_n) \text{ subject to } l_j \le x_j \le u_j \text{ , j = 1 ,2 ,...,n}$$

when derivatives of $F(x)$ are unavailable.

Special provision is made for problems which actually have no bounds on the x_j , problems which have only non-negativity bounds and problems in which $l_1 = l_2 =...= l_n$ and $u_1 = u_2 =...= u_n$. The user must supply a subroutine to calculate the value of $F(x)$ at any point x.
From a starting point supplied by the user there is generated, on the basis of estimates of the gradient and the curvature of $F(x)$, a sequence of feasible points which is intended to converge to a local minimum of the constrained function. An attempt is made to verify that the final point is a minimum.
A typical iteration starts at the current point x where n_z (say) variables are free from both their bounds. The projected gradient vector g_z ,whose elements are finite-difference approximations to the derivatives of $F(x)$ with respect to the free variables, is known. A unit lower triangular matrix L and a diagonal matrix D (both of dimension n_z), such that LDL^T is a positive-de finite approximation of the matrix of second derivatives with respect to the free variables (i.e., the projected Hessian) are also held. The equations

$$LDL^T \; p_z = -g_z$$

are solved to give a search direction p_z , which is expanded to an n-vector p by an insertion of appropriate zero elements. Then α is found such that $F(x + \alpha p)$ is approximately a minimum (subject to the fixed bounds) with respect to α; x is replaced by $x + \alpha p$,and the matrices L and D are updated so as to be consistent with the change produced in the estimated gradient by the step αp . If any variable actually reaches a bound during the search along p, it is fixed and n_z is reduced for the next iteration. Most iterations calculate g_z using forward differences, but central differences are used when they seem necessary.
There are two sets of convergence criteria –a weaker and a stronger. Whenever the weaker criteria are satisfied, the Lagrange-multipliers are estimated for all the active constraints. If any Lagrange-multiplier estimate is significantly negative, then one of the variables associated with a negative Lagrange-multiplier estimate is released from its bound and the next search direction is computed in the extended subspace (i.e., n_z is increased). Otherwise minimization continues in the current subspace provided that this is practicable. When it is not, or when the stronger convergence criteria are already satisfied, then, if one or more Lagrange-multiplier estimates are close to zero, a slight perturbation is made in the values of the corresponding variables in turn until a lower function value is obtained. The normal algorithm is then resumed from the perturbed point.
If a saddle point is suspected, a local search is carried out with a view to moving away from the saddle point. A local search is also performed when a point is found which is thought to be a constrained minimum.

7.7 References

[1] A complete description of E04JYF can be found at http://www.nag.com/numeric/FL/manual/pdf/E04/e04jyf_fl19.pdf .

[2] Gill P.E. and Murray W. (1976) - *Minimization subject to bounds on the variables*. NPL Report NAC 72, National Physical Laboratory.