



European Commission  
Information Society and Media  
Directorate –  
General



Delivery D 1.2		Document			
WP	Project name:	Project No.:	Editor:	Date:	Page
1	SmartCoDe	ICT-2009-247473	C. Wysoudil	30 June 2011	1/46
Filename: D1.2 Advanced Power Management			Lead Beneficiary No. 4 ennovatis		

# Methods for advanced power management in local grids

## SmartCoDe

---

Project No.:	ICT-2009-247473
Deliverable No.:	D-1.2
Deliverable Title:	Methods for advanced power management in local grids
Due Date:	June 30 <sup>th</sup> , 2011

---

Nature:	Report
Dissemination Level:	Public
Author:	Markus Damm Hartmut Freihofer, Milan Lukic, Fritz Schmidt, Nelly Vanvor, Christian Wysoudil,
Lead Beneficiary No.:	4



European Commission  
Information Society and Media  
Directorate –  
General



Delivery D 1.2		Document			
WP	Project name:	Project No.:	Editor:	Date:	Page
1	SmartCoDe	ICT-2009-247473	C. Wysoudil	30 June 2011	2/46
Filename: D1.2 Advanced Power Management			Lead Beneficiary No. 4 ennovatis		

## Contend

Summery .....	5
Abbreviations .....	5
1 Energy management – Introduction goals and basic approaches .....	6
1.1 Process model of the Energy Star project .....	6
1.2 Process model of the BuildingEQ project .....	7
1.3 Used EMS strategies and SmartCoDe .....	9
1.4 Energy management in local grids with distributed energy generation .....	9
1.5 SmartCoDe concept.....	11
2 SmartCoDe operation & energy management approach.....	11
2.1 Requirements.....	11
2.2 Cost-function based Energy Management .....	12
2.3 The global picture.....	14
3 SmartCoDe EMU implementation .....	16
3.1 Introduction ennovatis Smartbox .....	16
3.2 EMU Implementation General Description of the SmartCoDe EMU prototype .....	16
3.3 Application of the SmartCoDe EMU to controlling tasks.....	18
3.4 To reach the basic goals.....	19
3.4.1 Sensing of conditions.....	19
3.4.2 Control of appliances and comfort.....	20
3.4.3 Scheduling of appliances.....	20
3.4.4 Demand response based on present availability of energy.....	20
3.4.5 Demand response based on market signals .....	20
3.4.6 Demand response based on future energy availability .....	20
3.4.7 Strategy module.....	21
4 Power management with the SmartCoDe EMU .....	21
4.1.1 Introduction .....	21
4.1.2 Preliminary considerations.....	21
4.1.3 Highlight Concept .....	21
4.1.4 Description Highlights .....	22
4.1.5 Implementations.....	25
4.2 Detailed specification .....	25
4.2.1 Behaviour and application .....	25
4.2.2 The solar forecast Calculator.....	26
4.2.3 Q.met .....	29



European Commission  
Information Society and Media  
Directorate –  
General



Delivery D 1.2		Document			
WP	Project name:	Project No.:	Editor:	Date:	Page
1	SmartCoDe	ICT-2009-247473	C. Wysoudil	30 June 2011	3/46
Filename: D1.2 Advanced Power Management			Lead Beneficiary No. 4 ennovatis		

4.2.4	The wind forecast Calculator .....	29
4.2.5	The least cost function (LCF).....	29
4.2.6	User surface.....	30
4.2.7	Interfaces .....	31
4.2.8	Persistent Data .....	31
4.2.9	Performance, Capacity and Scalability (PCS) .....	32
4.2.10	Compatibility .....	32
4.3	Restrictions of functionality .....	32
4.3.1	Constraint #1.....	32
4.3.2	Constraint #2.....	32
4.4	Next steps in the development of the SmartCoDe EMU.....	32
4.4.1	Improvement of Module Power management.....	32
4.4.2	Update of EMU CPU.....	33
4.5	Description of interfaces .....	33
4.5.1	Description of the interface for weather forecasting .....	33
4.5.2	Description of the interface between EMU and the wind turbine.....	36
5	SmartNodes & Cost-function based EUP control .....	39
5.1	VSTSVC - Virtually Storable Service .....	40
5.2	SKDSVC - Schedulable Service .....	43
5.3	CHACON (Charge Control) and COMCON (complete control) .....	43
5.4	VAR SVC (Variable Services) and ETOSVC (EventTimeOut Service) .....	44
5.5	CUSCON – Custom Controlled.....	44
6	Conclusion and future work.....	44
7	References .....	45



Delivery D 1.2		Document			
WP	Project name:	Project No.:	Editor:	Date:	Page
1	SmartCoDe	ICT-2009-247473	C. Wysoudil	30 June 2011	4/46
Filename: D1.2 Advanced Power Management			Lead Beneficiary No. 4 ennovatis		

## List of Figures

Figure 1: ENERGY STAR process model for energy management .....	7
Figure 2: Phases of an ongoing commissioning process.....	9
Figure 3: Decision making in a local grid through an EMU (modified from /12/).....	10
Figure 4: SmartCoDe cluster overview.....	11
Figure 5 Bang-Bang control taking cost function into account.....	12
Figure 6 cost function as a step function.....	13
Figure 7 Format of the ZigBee Smart Energy Load Control Event Payload, taken from /25/ .....	13
Figure 8 SmartCoDe Energy Management: Overview.....	14
Figure 9 The SmartCoDe cluster as a control loop .....	15
Figure 10: Smartbox basic technical data .....	16
Figure 11: Smartbox functional overview .....	17
Figure 12: Smartbox wireless communication using wireless MBus or ZigBee.....	17
Figure 13: Smartbox peak load control .....	18
Figure 14: Smartbox integration of solar and conventional heating problem analysis.....	18
Figure 15: Smartbox integration of solar and conventional heating problem solution .....	19
Figure 16: Sensing climate data with the Smartbox.....	19
Figure 17: Leakage control using a MBus counter.....	20
Figure 18: Wireless connections to sensors and counters via MBus.....	20
Figure 19 Request handling - simple digital input output modules .....	23
Figure 20 SmartCoDe smart grid switch .....	24
Figure 21 event driven cost function calculation .....	25
Figure 22 status for error management.....	26
Figure 23 Example solar forecast configuration data.....	27
Figure 24 extended smartbox manager GUI.....	27
Figure 25 solar forecast sample calculation .....	28
Figure 26 weather forecast parameter definition for RSS feeds .....	35
Figure 27 EuP classification from D-1.1.1 /24/ .....	40
Figure 28 Modelling heater temperature with a low pass filter.....	41
Figure 29 Load balancing of 4 fridges with 4 cost functions .....	43
Figure 30 Typical Charge Profile for Lithium-ion Batteries, taken from /20/ .....	44

## List of Tables

Table 1 Criticality Levels from ZigBee SE (ZigBee Alliance, 2008) .....	14
Table 2 example for least cost function.....	30
Table 3 prototype user interface EUP I/O instruction.....	30

## Summery

Based on different Art of State Energy Management Ideas and Standards the reader finds in this Document a description of a combination of a centralized and decentralized EMS having the ability to control power consumption in manifold ways. It is a “cost function” based optimization process. The cost are to be seen in the meaning of eg Co2 ecological Footprints through improved usage of local Energy Production, Load balancing etc.

In Chapter 3 & 4 a introduction in the used Energy Management Unit, collecting Sensor data, Meter data and transmitting calculator of behaviour suggestions via cost functions and the develop principal algorithms introduction in developed interfaces and basic Codes are given.

Finally some impacts of the decentralized management approach will be shown in some samples (simulations) of different EUP classes.

Pls enjoy reading

## Abbreviations

short	Full Text
CC	Continuous Commissioning
CHACON	Charge Control
COMCON	Complete Control
CPU	Central Processing Unit
CUSCON	Custom Controlled
CxP	Commissioning Process
EM	Energiemanagement
EMS	Energiemanagementsystem
EMU	Energy Management Unit
ETOSVC	EventTimeOut Service
EuP	Energy using Produkt
GUI	Graphical User Interface
HVAC	Heating, Ventilating and Air Conditioning
IEA	International Energy Agency
LEP	Local Energy Production
SKDSVC	Schedulable Service
UTC	Universal Time Coordinated
VAR SVC	Variable Services
VST SVC	Virtually Storable Service

# 1 Energy management – Introduction goals and basic approaches

Energy management as defined in wisegeek /14/ is a form of energy conservation that seeks to identify and make use of various types of energy sources with the greatest degree of efficiency. In many cases, a particular approach will focus on not only using the least amount of energy for the most return, but also on utilizing energy sources that are considered beneficial for the environment. Individuals, businesses, and municipalities can apply this general concept in a number of ways. Building energy management is the process of monitoring and controlling the operating systems within a building. Though specific components may differ, these operating systems may include heating and air conditioning, ventilation, lighting and power. While building energy management techniques can be applied to a variety of building types, they are generally the most cost-effective when used in large commercial and industrial buildings. For such buildings the International Energy Conservation Code /15/ provides codes and standards that are often used in developing building energy monitoring standards at the local level.

Most current Energy Management Systems (EMS systems) were originally designed to control Heating, Ventilating and Air Conditioning (HVAC) processes and are being expanded to include the energy needs of the manufacturing process and the control needs of distributed generators. They are equipped with expert systems to analyze large numbers of variables with extensive algorithms. The use of these systems to control distributed generation is relatively new, and most manufacturers are just getting started in addressing the needs of distributed generation..

EMS systems of the future will manage information to improve overall profitability, not just simple process operations. Manufacturing and service companies will be able to maximize their returns while at the same time, maintaining high quality and minimizing downtime. These EMS systems will monitor for equipment degradation, assist in diagnosing process problems, and analyze overall process efficiencies. They provide advice to the system operator as to desirable set point and other process changes, and can make the changes themselves when authorized, or can simply provide guidance to the operator to maximize efficiency and availability.

To perform energy management most effectively it is necessary to measure energy consumption and energy production with high local and time resolutions and to analyse immediately the measured data in the in the actual context of the cost and comfort situation in the building. This requires a deep understanding of the building, its usage and its technical equipment and intelligent controllers to collect and analyse the data and to recommend appropriate actions.

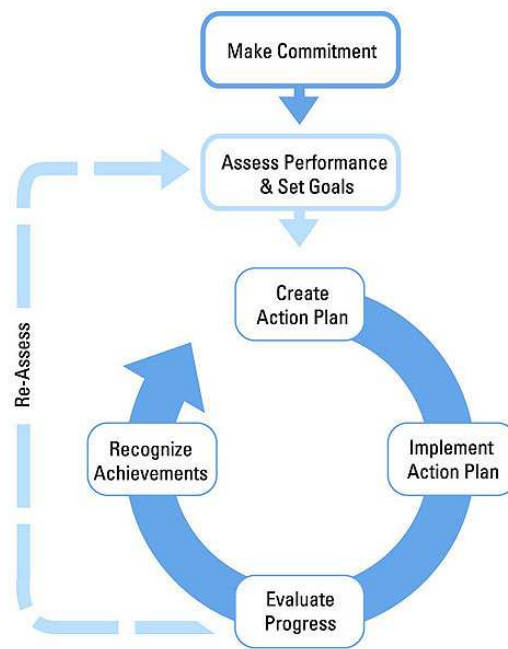
To support this by the energy management system requires considering energy management as a process, There are several process models available in the literature We concentrate in the introduction on two of them in this report. The first one is the process model provided in the Energy Star program of the US /16/. The second one was developed in the EU project BuildingEQ /2/, /4/

## 1.1 Process model of the Energy Star project

Based on the successful practices of ENERGY STAR /16/ guidelines for energy management were developed . These guidelines can assist in improving the energy and financial performance of a building. The process is shown in Figure 1. It consists of 2 starting and 4 continuous repeatable steps

The Steps are described in detail in the internet: Just follow the links given below to see details

- STEP 1: [Make Commitment](#)
- STEP 2: [Assess Performance](#)
- STEP 3: [Set Goals](#)
- STEP 4: [Create Action Plan](#)
- STEP 5: [Implement Action Plan](#)
- STEP 6: [Evaluate Progress](#)
- STEP 7: [Recognize Achievements](#)



**Figure 1: ENERGY STAR process model for energy management**

“

## 1.2 Process model of the BuildingEQ project

The process model of BuildingEQ is based on the idea of ongoing or continuous commissioning (“Continuous Commissioning” and “CC” are registered trademarks of the Texas Engineering Experiment Station) as developed in Annex 40 /17/ and in the frame of the Federal Energy management program /18/.

Ongoing-” or “Continuous Commissioning” is a term that comes from the USA. Originally, “Commissioning” denoted a task that is performed after a building is constructed and before it is handed over to the building owner to check operational performance. Today (at least in USA, but also in some other countries) “Commissioning” stands for a general quality assurance procedure.

The final report of the IEA ECBCS Annex 40 gives the following definition for commissioning /7/:

“Commissioning is a quality-oriented process for achieving, verifying, and documenting whether the performance of a building’s systems and assemblies meet defined objectives and criteria.”

To overcome the common understanding of commissioning as a task performed during handover, the Annex 40 report distinguishes between different kinds of commissioning:

- **Initial Commissioning (I-Cx)** Systematic process applied to the production of a new building and/or an installation of new systems.
- **Retro-Commissioning (Retro-Cx)** First time commissioning implemented in an existing building in which a documented commissioning process was not previously implemented.
- **Re-Commissioning (Re-Cx)** Commissioning process implemented after I-Cx or Retro-Cx when the owner hopes to verify, improve and document the performance of building systems.
- **On-Going/Continuous Commissioning (On-Going Cx / CC)** Commissioning process conducted continually for the purposes of maintaining, improving and optimizing the performance of building systems after I-Cx or Retro-Cx.

Most experts in building performance analysis probably would agree that a continuous commissioning process is a prerequisite for the energy-efficient long term operation of a building.

The “Continuous Commissioning Guidebook” of the FEMP /18/ gives the following definition for continuous commissioning:

*improve comfort, optimize energy use and identify retrofits for existing commercial and institutional buildings and central plant facilities.”*



A more detailed definition of continuous commissioning from Annex 40 stresses the aspect of quality assurance which strives to meet the requirements of the owner:

*“CC = Clarifying Owner’s Project Requirements (OPR) from viewpoints of environment, energy and facility usage, and auditing and verifying different judgments, actions and documentations in the Commissioning Process (CxP) in order to realize a performance of building system requested in the OPR through the life of the building.”*

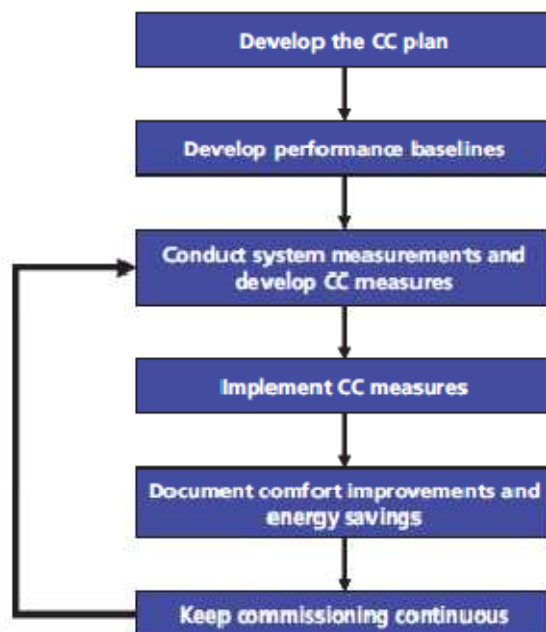
Typically, the commissioning process is described as a multi-level process. The “Continuous Commissioning Guidebook” of the FEMP describes the general phases for CC as follows:

- Develop the CC plan
  - Develop a detailed work plan (description of task, definition of targets, specification of monitoring, etc.)
  - Identify the entire project team
  - Clarify the duties of each team member
- Develop performance baselines
  - Document existing comfort conditions / problems
  - Document existing system conditions / problems
  - Document existing energy performance
- Conduct system measurements and develop CC measures
  - Identify current operating schedules, set points and problems
  - Develop solutions to existing problems
  - Develop improved operation and control schedules and set points
  - Identify potential cost effective energy retrofit measures
- Implement CC measures
  - Obtain approval for measures from building owner’s representative before implementation
  - Implement solutions to existing operational and comfort problems
  - Implement and refine improved operation and control schedules
- Document comfort improvements and energy savings
  - Document improved system conditions
  - Document improved comfort conditions
  - Document improved energy performance
- Keep commissioning continuous
  - Maintain improved comfort and energy performance
  - Provide measured annual energy savings

Although there are rules or even standards for energy auditing in many Member States, the complete list of tasks described above is seldom performed /9/. Especially the documentation of cost and energy savings is often neglected.

However, results from USA show that by introducing CC an average of 20% of energy savings could be achieved while the implemented measures had a pay back time of a few months up to 2 years. If these results could be transferred to Europe was one of the issues to be investigated in Building EQ.





**Figure 2: Phases of an ongoing commissioning process**

(from BuildingEQ final report) /4/

### 1.3 Used EMS strategies and SmartCoDe

Out of the basic research done in Project eDiana /27/ state of the art management strategies are described in an excellent overview. Once the side parameters described in /27/ /28/ are located, strategies can be applied successfully. SmartCoDe follows let's say a combined idea of direct and indirect control described in /29/ with the major weight on the indirect control influence through the local EMS described in Chapter 1.4. (also Figure 3) The public (Energy- and or Network-) supplier hands forward signals of general balancing features over the whole network via tariffs, bonus malus systems and or supply failures. SmartCoDe includes additionally local energy production forecasting signals. Consumption peaks as mentioned in /30/ /31/ /32/ are topics to be balanced in this project. The direct signals from the utilities (public grid) will be handled as a suggestion for optimization in the local grid (behind the public main meter). With respect to the ideas of load shedding as eg described in /35//33/ the SmartNode in this SmartCoDe Project will take over parts of this strategies. Aspects as load forecasting will be handled in similarities with the idea presented in /36/.

Principals given by Stephan Mahlnecht in /37/ will be discussed as an enhancement for a future improvement – the optimization within the wireless Network is part of the present development in Work package 2.

Most of the described EMS consider that the user comfort is at least not less than before. We like to give a chance also to the possibilities to reduce the comfort in such cases when energy bottlenecks as seen in many countries outside of Europe are ruling the scenario. SmartCoDe will try to cover this with the possibility to influence upper and lower thresholds, more transparency for the use to different consumption profiles via new user interfaces and the use of alternative calculated timeslots for different EUPs like Boiler, Battery charger etc. Through the possibilities of the wireless communication between SmartNodes and the basic guiding Costfunction directives (see in the following Chapter beginning with 2.2) given by the SmartCoDe EMU SmartCoDe like to hit also in further steps principals described in /38/.

### 1.4 Energy management in local grids with distributed energy generation

When building a new or renovating an existing home, the selection of energy sources is a key element of effective energy management. Along with incorporating access to traditional sources such as the

local power grid and gas lines, the homeowner may also opt for the inclusion of a solar energy system or even a wind energy generator. The use of these alternative energy sources helps to minimize the use of traditional sources by using these easily renewable sources when and as possible. As a result, the homeowner saves money and also decreases the demand on traditional utilities.

Locally distributed controllers will support the management of such grids. They are particularly effective /11/ when some energy is generated at the building or in a nearby district, such as from photovoltaic cells or wind generators. When installed in homes and businesses the controllers have a couple of tasks to full fill:

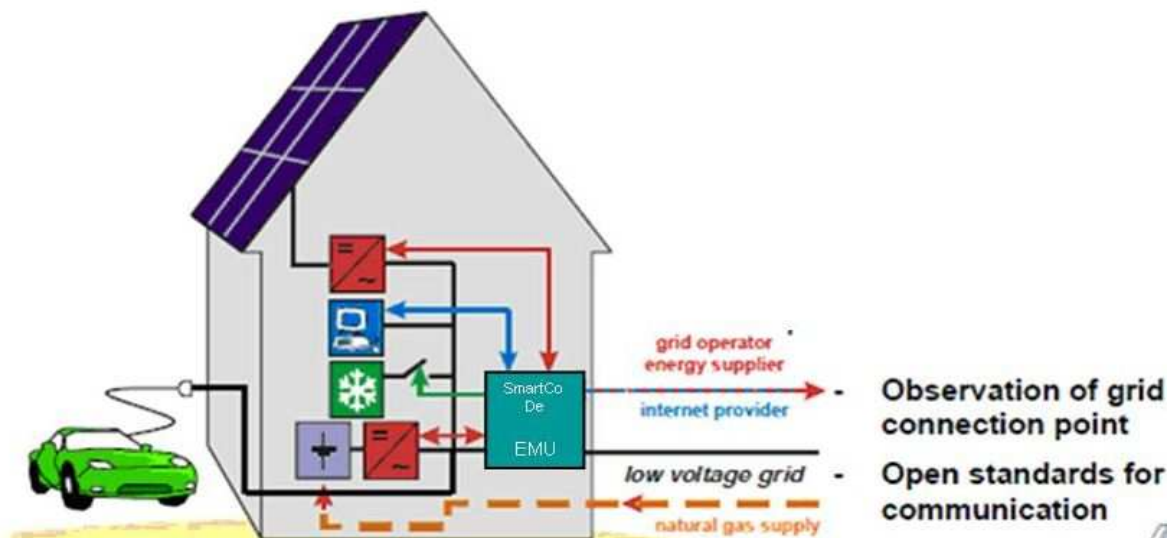
- sense conditions such as temperature
- control appliances and small local electricity generators, based on customer preferences
- receive information such as customer preferences, price signals and weather conditions, and communicate with all major appliances (EuP) and with the locally available energy sources (LEP) as well as with neighbouring sites and remote electricity utilities
- process the information to determine immediate and future optimum actions.
- communicate also with the electricity company to find out when peaks in demand are expected, and with the meteorological office for weather forecasts to predict future energy production

Customers can reduce demand by more careful scheduling of appliance use, and through real-time control they participate in and get paid for demand responses based on network and market signals.

With this in mind the smart control of demand for consumption and supply to enable balanced, energy-positive buildings and neighbourhoods as anticipated in the SmartCoDe project requires a quite ambitious future EMS system with new approaches for data collection, data analysis and energy management actions.

For domestic energy management intelligent controllers will help to manage appliances efficiently, integrate renewable generation, and inform customers about options and consequences of different energy choices. Home energy management can be coordinated across multiple households, small neighbourhoods and offices for aggregated benefits.

To implement these ideas it was proposed in the DERlab project /12/ the solution shown in Figure 3. It is based on an intelligent bidirectional energy management interface as described in /12/or /19/.



**Figure 3: Decision making in a local grid through an EMU (modified from /12/)**

A further control feature of the future, larger scale EMS will be managing non-critical loads so that they can serve to provide reserve power within the local grid. It is anticipated that not all "reserves" will come from energy storage. Energy storage devices, such as batteries and ultra capacitors, will be ideal for short term power requirements (seconds to minutes) for such needs as fault current, motor starting, etc., but for longer term needs, such as 10 minutes to one hour, it will be useful to have "reserves" by shutting down noncritical loads that can be stopped for this period of time with no deleterious effect.

## 1.5 SmartCoDe concept

The general concept of SmartCoDe considers a “local energy resource cluster” with:

- Local renewable energy resources like small-scale wind turbines and building-integrated PV. Energy generation forecasts support the integration of the renewable’s energy output into the energy management process.
- Local energy storages such as car batteries (plug-in hybrids, electric vehicles) which might also provide energy to the cluster if needed.
- Energy using Products (EuPs) such as HVAC, electric lighting, consumer electronics and white goods, whose power consumption is monitored periodically.

The EMU gathers the available data and controls the components of the cluster via the wireless nodes. Since the cluster can also provide (sell) energy to the grid, it is necessary to have predictable consumption / production behaviour. A common business model today for large consumers is that they pass their planned or expected load profiles to the grid operator, and are charged according to how far the profiles could be met. Therefore SmartCoDe assumes that the local cluster has a certain target load profile to the grid. This load profile might be issued by the customer to the grid operator like described above, but could also be issued by the grid operator implicitly via time-dependent tariffs. It might even change over time due to dynamic tariffs and/or automated negotiations by both parties. Figure 4 shows an overview of the overall scenario.

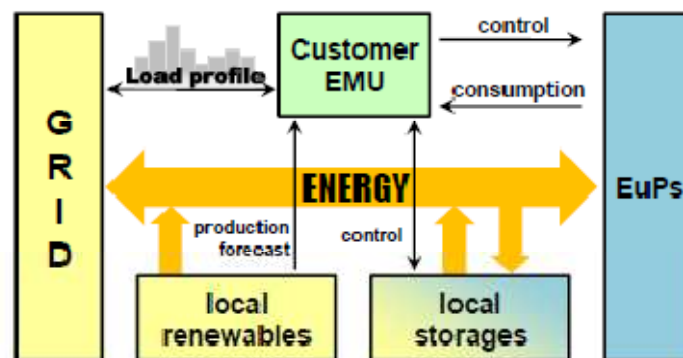


Figure 4: SmartCoDe cluster overview

As a consequence energy management in SmartCoDe means primarily power management. Other functionalities of ennovatis energy management will be used but will not determine the critical path.

## 2 SmartCoDe operation & energy management approach

### 2.1 Requirements

Here a list of the requirements identified for the SmartCoDe Energy Management (EM) approach.

1. The EM-algorithm should be able to be **adaptable to different optimisation goals**. The main goal of SmartCoDe is maximizing the usage of locally produced renewable energy. But other goals like monetary saving for user acceptance and load balancing for grid stability are worthwhile as well, especially when if locally renewable energy resources are *not* available.
2. The **communication means between EMU and SmartCoDe node should be generic** in the sense that it should not depend on specific circumstances like tariff model, optimisation goal or the different power states of an EuP. This ensures that the EM approach is robust regarding future developments.
3. **The knowledge required within the EMU about the EuPs should be minimal**. This ensures that cumbersome system updates due to changes in the network like adding, removing or replacing EuPs are kept to a minimum.
4. The EM-algorithm **should not “patronize” the user** – ideally the user should not be aware of EM activity and definitely should not experience discomfort. This is at the moment a key requirement for user acceptance in an average European Union household or office environment, which is connected to the public grid. In an *islanded scenario* (not connected to

the grid), users might be more willing to accept discomfort in order to ensure vital services, but this is not considered here.

5. The EM-Algorithm **is not required to be perfect**: There is no need for perfect load balancing or hard deadlines to obey to regarding load shifting/shedding since we assume connection to the public grid which can even out all imperfections. Again, in an islanded scenario this might look different, e.g. loads might have to be switched within eg seconds.

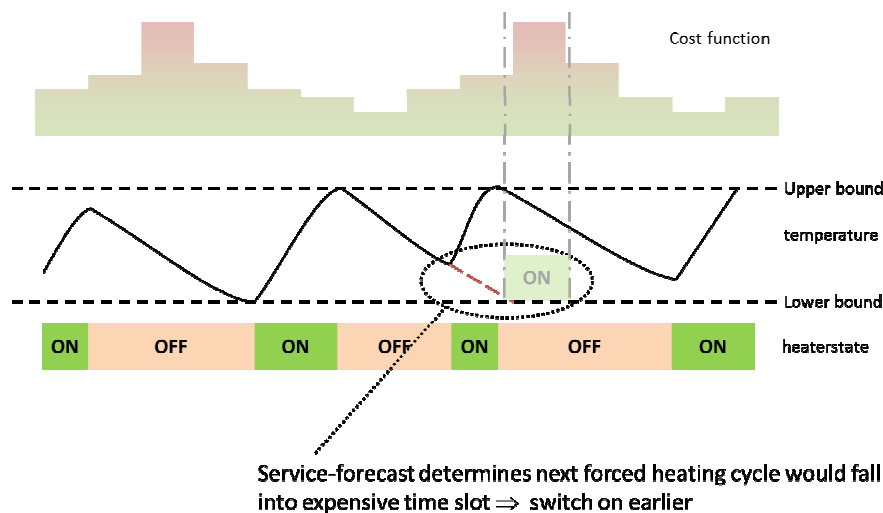
## 2.2 Cost-function based Energy Management

In Deliverable D-1.1.1, we outlined an EM approach where the EMU issues a cost function on which the SmartCoDe nodes act autonomously /24/. This was called a decentralized approach; however, it is in fact a *partially decentralised* approach since the cost function is still computed by a central instance, namely the EMU.

The idea of the cost function is that it provides **abstract energy costs** for a certain future time period, taking into account tariff, forecasts of local energy production and even power consumption forecasts based on usage statistics. For example, the abstract energy cost for a certain time span might be minimal even if the grid tariff during that time is at a maximum because the local power production is so high (e.g. a sunny & windy day) that it covers the lot of the possible local power consumption.

The SmartCoDe nodes now take this abstract energy cost figures into account regarding their local control decisions. Figure 5 shows how this would work with a heater as an example of the VSTSVC (virtual storages) class.

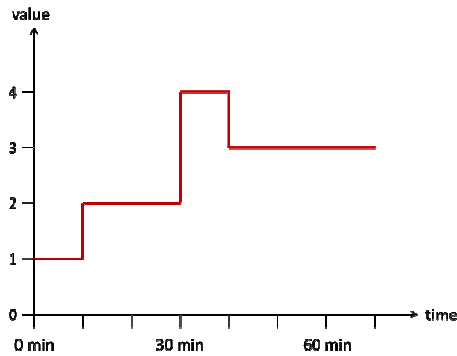
A convenient by-product of this approach is that it provides more privacy. Since the values of a cost function are basically aggregated, it is virtually impossible to figure out the concrete reasons for specific values. A potential eavesdropper can't know if the abstract energy cost for the next two hours is low because the EMU projects that the user remains absent for that time span.



**Figure 5 Bang-Bang control taking cost function into account**

The SmartCoDe node algorithm sketched in Figure 5 basically performs a simple bang-bang control, but switches on or off earlier if the cost function suggests this. It tries to avoid heater activation in costly times, and tries to turn it on in cheap times. To this end, it makes use of an internal forecast for the temperature it controls.

The format of the cost function is straightforward: It has a starting time, followed by several pairs of a time period and the value the function has during that period. That is, we describe the cost function as a step function. Figure 6 shows a simple example which would be described by the pairs (10 min , 1) , (20 min , 2) , (10 min , 4) , (30 min , 3).



**Figure 6 cost function as a step function**

The implementation of this cost function in the SmartCoDe ZigBee profile is called load control vector; see Deliverable D-2.3 /21/. The minimal time granularity can be set from 1 second to 1 hour, allowing for maximal flexibility.

The values of the cost function are abstract in the sense that they don't represent monetary costs or energy thresholds, although the computation of these values might be directly linked to these parameters. The basic approach is motivated by the ZigBee Smart Energy Load Control Event Payload, shown in Figure 7. The idea of the cost function in SmartCoDe is that it bundles several consecutive load control events, but concentrates on the criticality level.

Octets	4	2	1	4	2	1	1
Data Type	Unsigned 32-bit integer	16-bit BitMap	Unsigned 8-bit integer	UTC Time	Unsigned 16-bit integer	Unsigned 8-bit integer	Unsigned 8-bit integer
Field Name	Issuer Event ID (M)	Device Class (M)	Utility Enrollment Group (M)	Start Time (M)	Duration In Minutes (M)	Criticality Level (M)	Cooling Temperature Offset (O)

Octets	1	2	2	1	1	1
Data Type	Unsigned 8-bit integer	Signed 16-bit integer	Signed 16-bit integer	Signed 8-bit integer	Unsigned 8-bit integer	8-bit BitMap
Field Name	Heating Temperature Offset (O)	Cooling Temperature Set Point (O)	Heating Temperature Set Point (O)	Average Load Adjustment Percentage (O)	Duty Cycle (O)	Event Control (M)

**Figure 7 Format of the ZigBee Smart Energy Load Control Event Payload, taken from /25/**

The elements in the red frame are basically reused in the load control vector (the cost function message). The Criticality Level in ZigBee SE is explained in Table 1. The EMU can ask a device to save energy with varying intensity. From Criticality Level 1 - 6 the participation of the device is optional, while the levels 7 - 8 require participation, and have also concrete semantic assignments (e.g. Emergency).

We could choose to adopt this criticality level semantic; the advantage would be that we also could be backwards-compatible with ZigBee SE in the sense that we could use such load control events as "1-step step-functions". However, it does not seem as if the criticality level is adopted also in the upcoming ZigBee SE 2.0 standard. Therefore we can also decide to define our own criticality level, which could make use of the full 8-bit range. The current simulations work with a criticality level from 0-7. In any case, the requirement 2 from Section 2.1 is satisfied.

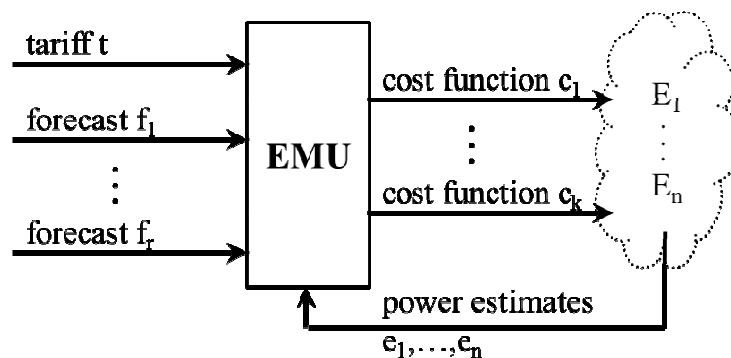
Criticality Level	Level Description	Participation
0	Reserved	
1	Green	Voluntary
2	1	Voluntary
3	2	Voluntary
4	3	Voluntary
5	4	Voluntary
6	5	Voluntary
7	Emergency	Mandatory
8	Planned Outage	Mandatory
9	Service Disconnect	Mandatory
0x0A to 0x0F	Utility Defined	Utility Defined
0x10 to 0xFF	Reserved	

**Table 1 Criticality Levels from ZigBee SE (ZigBee Alliance, 2008)**

## 2.3 The global picture

Figure 8 shows a coarse overview of the SmartCoDe energy management approach. There are two aspects not discussed yet:

- To manage load balancing, several cost functions are used. That is, each cost function “controls” a certain subgroup of the EuPs.  $E_1, \dots, E_n$ , where a group might only consist of one EuP. This approach avoids the severe imbalances that might be caused if a large number of EuPs obey to a single cost function. The EuP grouping (with possible dynamic re-grouping) is done by the EMU.
- Since we want the EMU to need as little knowledge as possible about the specific EuPs, it also does not know how an EuP reacts on a cost function. In the case of VSTSVC, for example, this would involve knowing the current temperature managed by the EuP. Therefore, we foresee the EuPs to report an estimate of the future power consumption. The idea is as follows: when the wireless nodes get their cost function, they make a control plan for their EuP for a certain time interval. Out of this control plan, they can then compute the expected power consumption, and pass this information on to the EMU, possibly together with a record of the past power consumption. This could be realised as a simple load-profile, again expressed as a step-function. The EMU can now take these power estimates into account when computing the next round of cost functions. This covers requirement 3 from Section 2.1.



**Figure 8 SmartCoDe Energy Management: Overview**

The exact format of tariff and the forecasts (for the renewables power output) are of no concern here. Indeed, an advantage of the cost function approach is that it is independent of tariff models, forecast formats and even optimisation targets. Any changes there only have to be implemented in the EMU, the cost function format and the node algorithms can stay the same. Therefore, the requirement 1 from Section 2.1 is covered by our approach.

It is obvious from Figure 9 that the overall system forms a control loop, possibly with several sub-control loops included in the EuPs  $E_1, \dots, E_n$ . Figure 9 shows a more detailed view of Figure 8; the items in red are interpretations of this situation in terms of classical control theory.

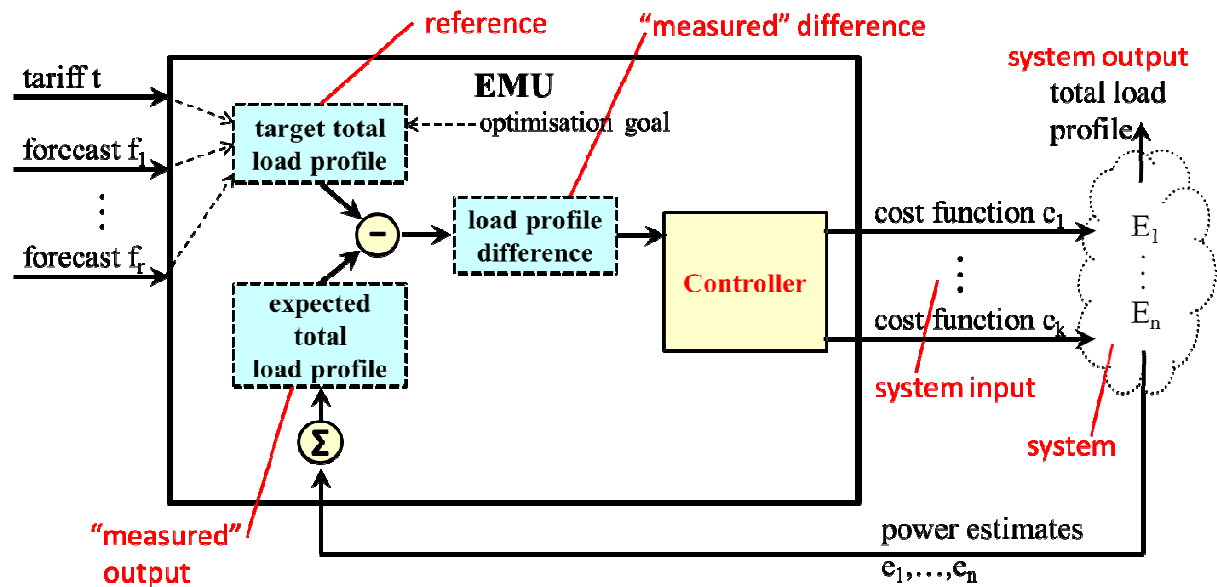


Figure 9 The SmartCoDe cluster as a control loop



### 3 SmartCoDe EMU implementation

#### 3.1 Introduction ennovatis Smartbox

The ennovatis Smartbox collects all energy data of a building and transmits it for further processing. It enables a completely new, high-efficient energy management of buildings. All data concerning energy consumption of buildings are collected fully automatic and round-the clock and transmitted for further detailed processing.


ennovatis Smartbox: technical data (samples)	
	<b>Inputs</b> 24 8 x analog Inputs (0-10V, 0-20mA) 8 x temperature Inputs (Pt 1000) 8 x digital Inputs (e.g. pulse/status/tariff) M-Bus via external level converter (RS 232) RS485 Onboard
	<b>Interfaces</b> modem / ISDN / GSM Ethernet / 2 x RS 232 / RS 485
	<b>Protocols</b> TCP/IP, M-Bus, Email...
	<b>Switch procedures:</b> connection up to 2 relays or with RS485 almost boundless possibilities
	<b>Memory</b> 2 Mbytes Flash memory

Figure 10: Smartbox basic technical data

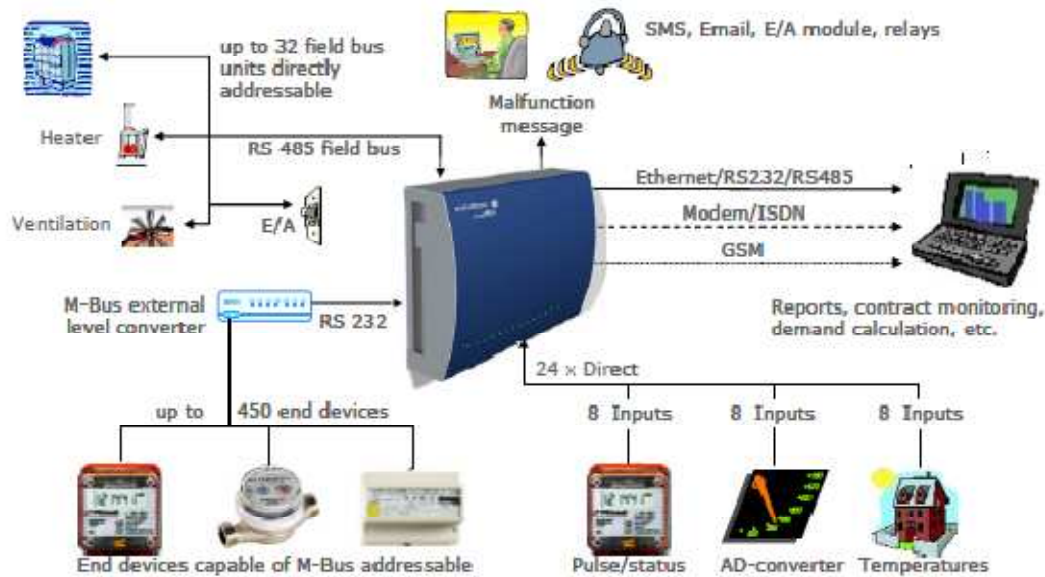
#### 3.2 EMU Implementation General Description of the SmartCoDe EMU prototype

In connection with ennovatis controlling there is a variety of utilizations. Peak demand management is for recognizing and avoiding extremely high energy consumptions to achieve considerable savings.

**Detect faults and defects, irregularities and unnecessary energy wasters** through permanent observation of energy processes. Unintentional or superfluous energy consumption becomes obvious quickly and can be remedied- often with easy and cheap methods.

**Fault indication management**, e.g. permanent comparison of parameters; automatic fault signalling by SMS, E-mail or relay. Switching of relays is based on predefined incidents.

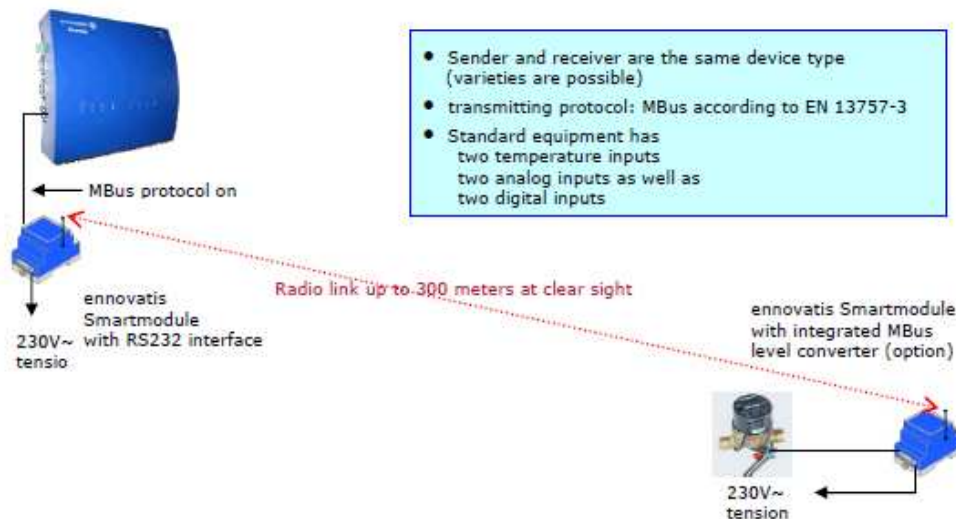
As a consequence the ennovatis Smartbox is more than a data logger for energy consumption data. We support and complete your building control technology.



**Figure 11: Smartbox functional overview**

All relevant parameters of energy consumption will be gathered:

- Counter readings (electricity, heat, water, gas etc.)
- Temperatures (outside-/ indoors-/ flow-/ return temperature, etc.)
- States (burner/ pump running time, etc.)
- Analogue signals of external signal and probing converters
- M-Bus protocol via external level converter, ModBus
- Create your own controlling with the aid of field bus Modules based on RS485 interfaces



**Figure 12: Smartbox wireless communication using wireless MBus or ZigBee**

### 3.3 Application of the SmartCoDe EMU to controlling tasks

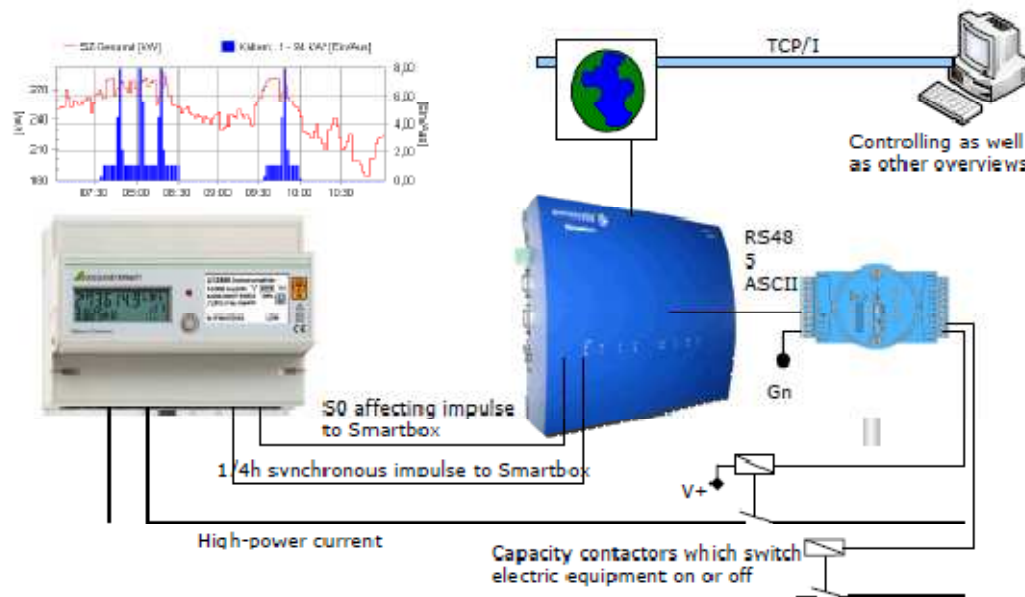


Figure 13: Smartbox peak load control

characteristic solar system schemata:

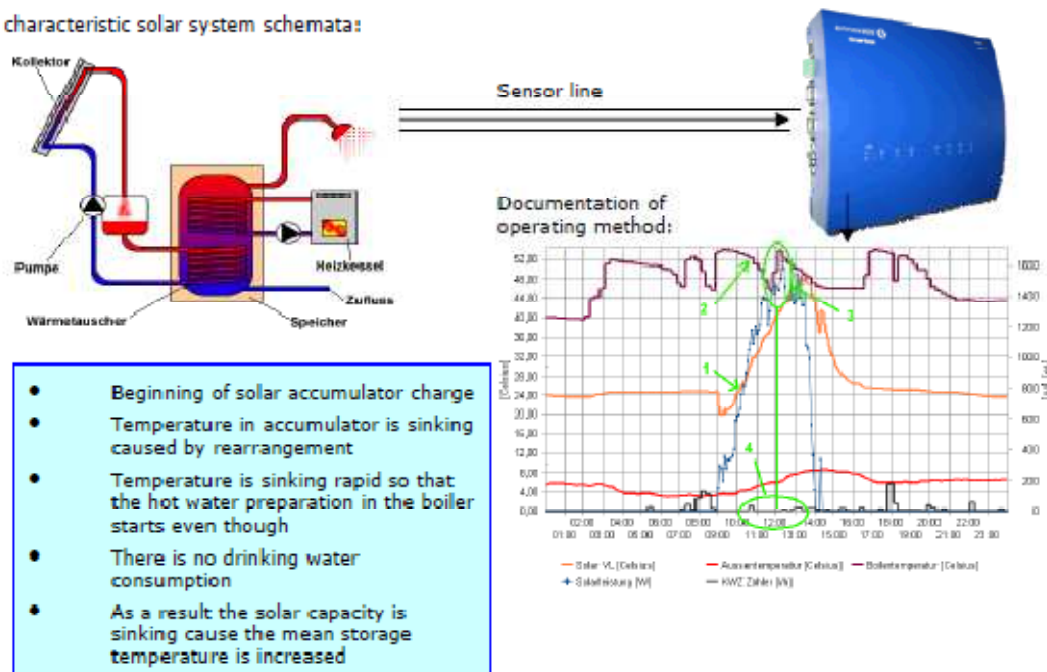


Figure 14: Smartbox integration of solar and conventional heating problem analysis

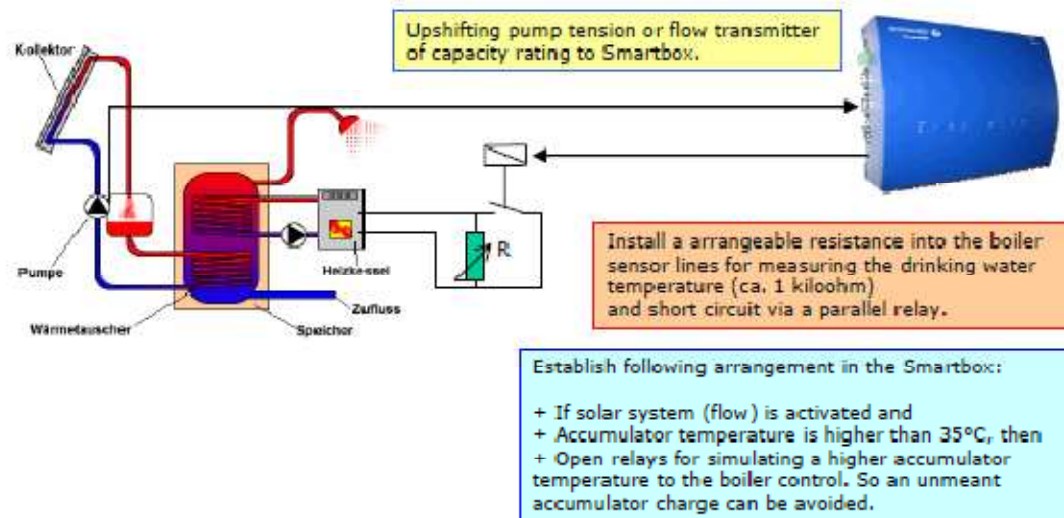


Figure 15: Smartbox integration of solar and conventional heating problem solution

### 3.4 To reach the basic goals

#### 3.4.1 Sensing of conditions

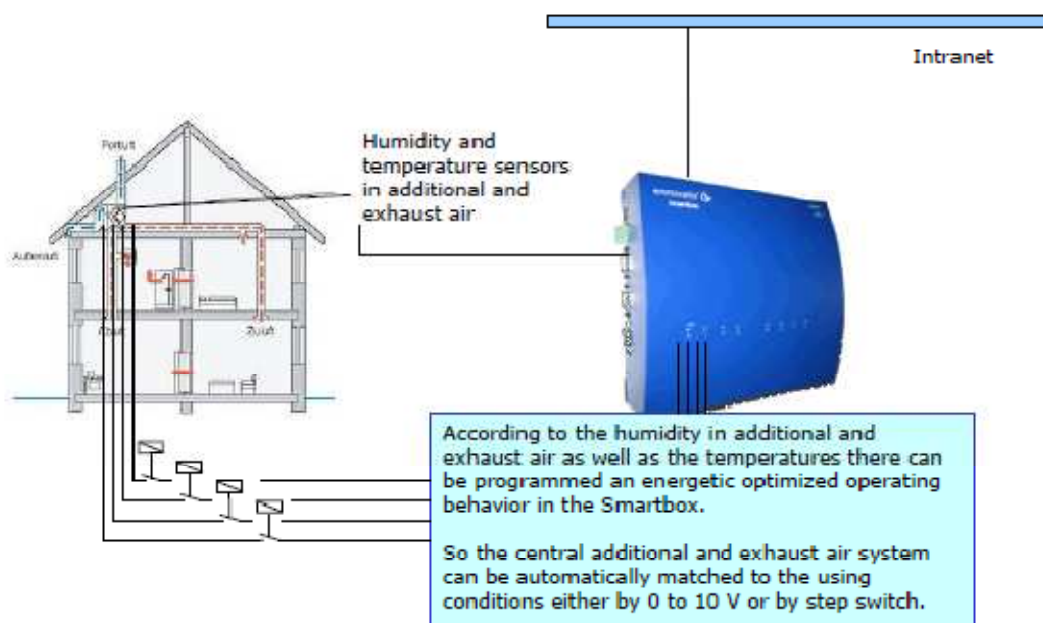


Figure 16: Sensing climate data with the Smartbox

### 3.4.2 Control of appliances and comfort

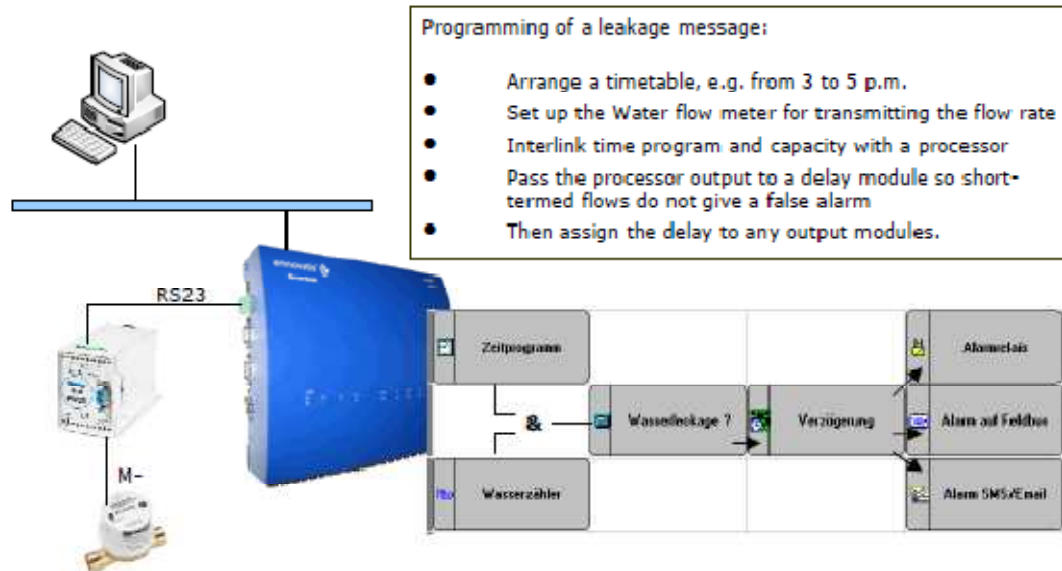


Figure 17: Leakage control using a MBus counter

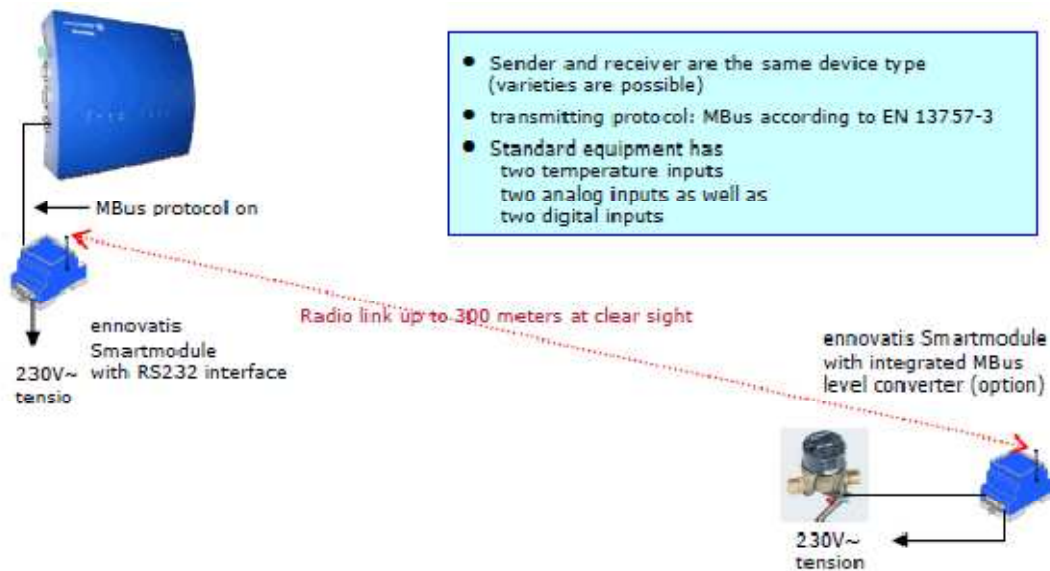


Figure 18: Wireless connections to sensors and counters via MBus

### 3.4.3 Scheduling of appliances

The Smartbox calculator will be used to determine priorities for the use of the different appliances according to the need of these appliances and the status of the network

### 3.4.4 Demand response based on present availability of energy

Available power will be distributed to the available EuPs according to the priorities given by the power scheduler. Presently a peak load manager was used for each LEP. However this will be changed using the modified PES function described in a first step in this document.

### 3.4.5 Demand response based on market signals

Market signals will be used to modify available power. Low prices will result in bigger loads. Price predictions will be taken into account if available

### 3.4.6 Demand response based on future energy availability

Power predictions will be treated in a similar way as price predictions.



### 3.4.7 Strategy module

If this approach proves to be practical it is anticipated to develop a strategy module to further reduce consequences of uncertainties in price, demand and availability predictions

## 4 Power management with the SmartCoDe EMU

### 4.1.1 Introduction

The project SmartCoDe requires a function to calculate the cost of electrical energy within the conglomerate of energy users, energy producers and energy suppliers. Each of which consists of different profiles for power and the according cost of using or selling this power.

The function of calculating this is to be implemented within the Smartbox System (= EMU).

The resulting cost profile is used internally to start wired machines directly and it is propagated wireless to different other machines which decide about their consumption by their own.

To achieve this, it is necessary to have all the information in the Smartbox System.

The information consists mainly on different profiles, which have to be balanced against each other.

The main task is then to find the cost-effective time to start a dedicated machine with a dedicated machine profile. For this each request to start a machine is calculated for the next hours as long as it is allowed to delay the start.

With this in mind (memory) it is possible to calculate a Cost-profile. This Cost-profile is to be sent to the wireless connected machines. Consequentially this Cost-profile changes with every calculation step (typically 15 minutes), with every new forecast, with every actual not controlled power consumption, every not foreseen change of power production and last but not least every new request running a dedicated machine.

Therefore the Cost-Profile should be sent out to the wireless connected consumers every time it is changed.

### 4.1.2 Preliminary considerations

There are a few conditions for the functionality like it is researched in SmartCoDe in the real world:

- There is only one power meter which is capable of counting forwards and backwards as power is drawn (buy) or delivered (sell) to the grid.
- Furthermore this power meter must be capable of counting actual and quarterly different costs of electric energy.
- To do this the power meter have to store and forward each counted quarter hour with current buy and sell charge to the local energy supplier and to the customer. This is necessary to keep the electric bill comprehensible for all parties.

If this is the case then it is only possible to calculate a Cost-Profile for the future if the load of the local network (behind the power meter) doesn't change. See also constraint #1.

Alternatively it is NOT possible to have two "intelligent" power meters for buy and sell as it is currently in Germany the case. Even if this power meters have the same quarterly counting capabilities.

In this scenario there is no calculation necessary because it is simply the actual buy profile.

### 4.1.3 Highlight Concept

#### Daily Profile:

List of 16 bit numbers. Within this project we are using profiles with a resolution of 15 minutes. So each daily profile consists of 96 values plus a starting time.

All profiles have to be updated at least twice a day, so that it is possible to calculate the Cost-Profile for at least the next 12 hours.

#### Machine Profile:

List of 16 bit signed integer numbers. Machine profiles do not change with time. They have no starting time. Each machine profile is different in length because different machines have different runtimes. Its unit is 10 x Watts. So the value range is -327,6kW

to +327,6kW. Negative Values represent power producing machines. With this the optimum time of running a heat controlled block heating machine could be found.

**Charge Profile:**

The unit of the profile is 1/10 cents/kWh. Typically there are three Charge-profiles: The charge for buying, the charge for selling and the internally calculated charge. The internally calculated charge is to be propagated to the wireless connected devices.

**Loadprofile:**

The unit of the profile is average 10 x Watts.

Typically it is calculated by summing all the planned machines running and the basic load.

**Basic Load:**

The Basic Load is a fixed daily consumption profile. It consists mainly of standby devices and a bunch of not planned running machines (e.g. the wireless connected devices).

**Forecast:**

The unit of the profile is average Watts. Forecasts are used for the photovoltaic production, the wind turbine production and many more (e.g. block heat and power plant).

**Request:**

A request is a user initiated event to start a pre defined machine profile within a requested delay time.

**Delay time:**

Due to the daily profiles update cycles (24 hours long and every 12 hours) one could only calculate a maximum delay time of 12 hours. This is the maximum delay able time in a request.

**Least-Cost-Function:**

The Least-Cost function takes a request and searches for the cost-effective time to start the pre defined machine profile in the given delay time. In the first approach there are no further constraints to consider. For example an electrical heater request to start heating because it's temperature falls below a threshold. To circumvent a cold shower, the requested delay time could be reduced as the difference to the threshold temperature increases.

**Request table:**

All requests are stored in a table in the order they are received. If a new request arrives the least-cost function starts and the calculated cost effective time is stored together with the request itself.

If any part of the information to be considered changes (e.g. forecast, actual power production, buy or sell profile, ...) the request table is used to recalculate all start times again.

#### 4.1.4 Description Highlights

The main task is now to get all the information into the system. If some data is not available – e.g. a missing update connection – the data has to be replaced by default values or alternative data calculations.

The tables (profiles) used to run the least-cost functions are:

1. Cost profile buy: The charge in cents/kWh to buy electrical power.
2. Cost profile sell: The charge in cents/kWh to sell electrical power.
3. Forecast-sun: for the next 12 hours
4. Forecast-wind: for the next 12 hours
5. Load profile: for the next 12 hours
6. Requested machine profile(s)



### Prototype Version:

To get the prototype version quickly, and due to the lack of getting real-time buy and sell charge profiles, these tables are permanently imported into the system and do not change.

Also the different machine profiles are permanent and not changeable. The machine profiles are limited to 10 tables each 3 hours in length.

The forecast-sun table is build up of a sun power calculator which calculates the maximum possible power out of the power plant data (gradient, latitude, square meters, efficiency, time of day). This maximum power is then reduced by the value yesterday achieved and – if available – the weather forecast. If the sun power production starts, the actual values are used to correct the daily forecast-sun table.

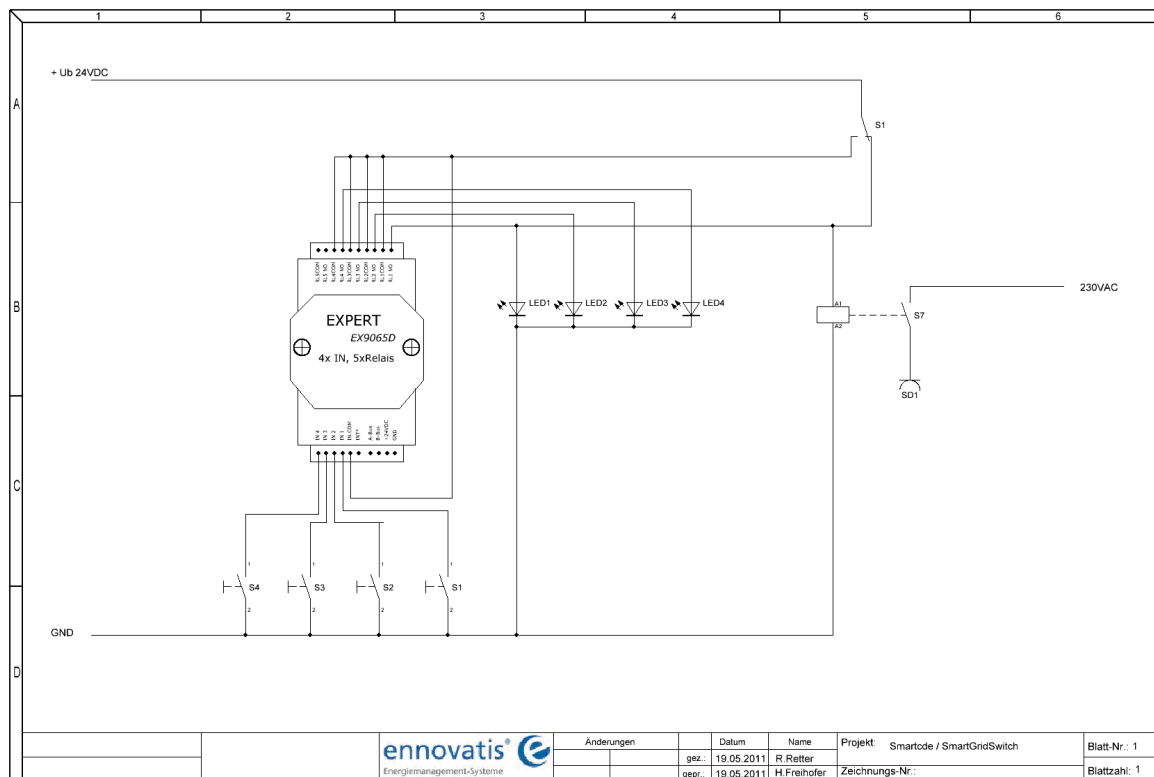
The forecast-wind table is build up of the current wind power. This current power is prolonged for the next 12 hours as long as it isn't changed. If this current power is not available, it is set to zero. If this data could be retrieved from the wind turbine it is read from there. In an alternative scenario (e.g. second prototype Version) the wind-forecast table could be corrected by the weather forecast.

### Processing:

As mentioned above the main processing takes place within the Least-Cost function. This function calculates the cost effective start time by summing all already requested runs and start times in the request table. It then compares them against the power balance (buy or sell) and summarizes the charge for each quarter hour to a total charge of request. This sum is moved by quarter hours for as long as the maximum allowable delay time is valid. The earliest time step with the least cost of request is then the time to start the machine.

### Request inputs and outputs:

To simplify request inputs, they are done by simple (wired) digital input modules. Such a wired module is for example the EX9060D from Expert. This module has 4 digital inputs and 4 relays outputs. The inputs are used in the following way (see [Figure 19](#)):



**Figure 19 Request handling - simple digital input output modules**

Input one starts the machine immediately.

Input two: The machine should start within 4 hours.

Inputs three: The machine should be ready within in the next 12 hours and

Input four deletes the request.

The first development of Hardware plug can be seen in [Figure 20](#)

The desired input can be given by short cutting the digital in by a switch key as long as there is no signalling on the corresponding output relay.

The outputs are used in this way:

If relay four is active, the (currently) calculated start time is four hours or more away.

If relay three is active, the (currently) calculated start time is less than four hours away.

If relay two is active, the (currently) calculated start time is less than one hour away.

If relay one is active the machine is running. It is connected to the machine.

If the machine profile is done (the duration time has elapsed) the request is being deleted from the request table and the relay output is released.

If an input request is pressed again the entry in the request table is being updated.

Respectively if the delay time wants to be changed and if anything else changes, all entries in the request table are updated.



**Figure 20 SmartCoDe smart grid switch**

Within WP2 further future development of a wireless Hardware plug is planned in combination with the SmartNode

### **Wireless Outputs:**

For the wireless connected machines the cost-profile for the next 12 hours is propagated every time it is changed. The cost-profiles changes every time anything is changed. This could be an incoming request, a change of the weather forecast or a cloud covering the photoelectric power plant and thus changes the currently available power.

Because of the unpredictable changing, this cost-profile change is send out without request to the serial interface two (Field 2 of the Smartbox System) were a wireless transmitter receives this information.

### 4.1.5 Implementations

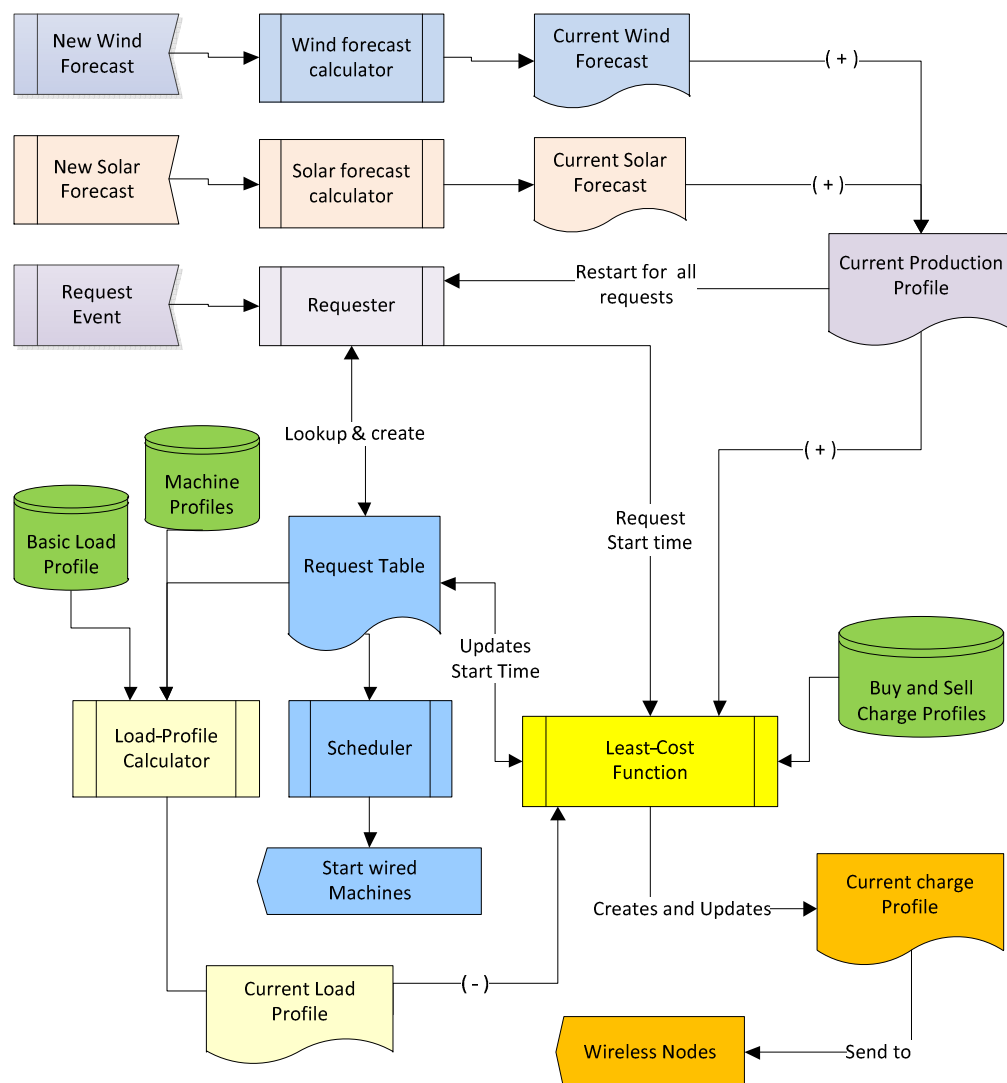
The new function is to be implemented for the first prototype only within the Smartbox Firmware. The profile tables are hard coded.

The made decisions (request table delays) are stored on the internal flash and could be read out for later researches. For this the Smartbox interface as well as the Excel tool must be adapted.

## 4.2 Detailed specification

### 4.2.1 Behaviour and application

With the start off the system the profile tables are read in and the least-cost function starts to calculate. The following diagram in [Figure 21](#) shows the different objects and there interconnections.



**Figure 21 event driven cost function calculation**

The total system is build up and works with events. Events could be "System start", "new Wind forecast", "new sun forecast", "new machine request" or "new charge profile". Each event starts a process chain which calculates all or a dedicated start time. Additionally the current charge profile is updated and sent to the wireless nodes.

#### System start event (not shown on the diagram above):

Retrieves Wind forecast, retrieves solar forecast, creates default load profile, starts least cost function, creates current charge profile.

#### New solar forecast event:

Starts least cost function, creates current charge profile.

#### New wind forecast event:

Starts least cost function, creates current charge profile.

#### New request event:

Lookup request table, request least cost start time, creates current charge profile

If an error occurs the system sends automated notifications to predefined recipients. For this the new status error management was built in (see [Figure 22](#))

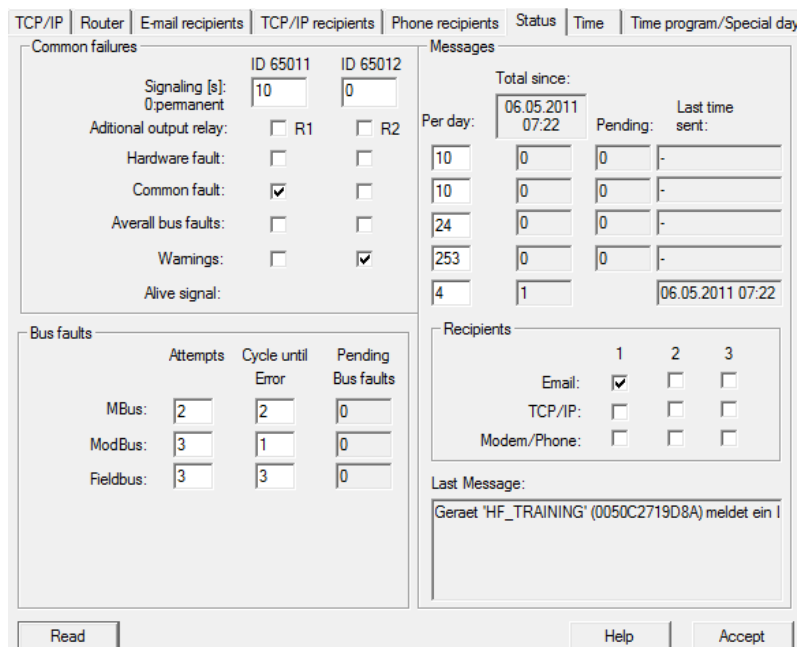


Figure 22 status for error management

### 4.2.2 The solar forecast Calculator

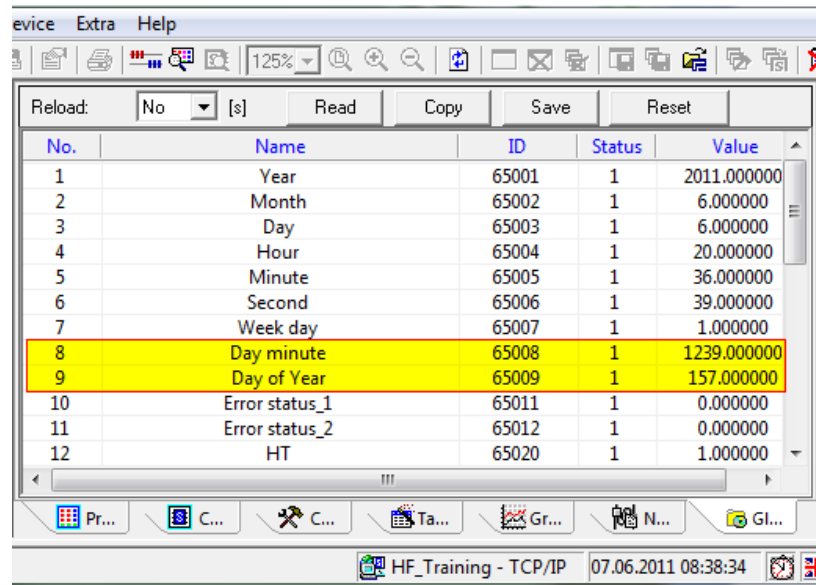
The solar forecast Calculator takes the following parameters (written into the Smartbox via the Smartbox interface)

ID of current Sunpower	10604	ID that delivers W W Kornwestheim MEZ Kornwestheim of PV Survice (Neigungswinkel: 0 = horizontal)
Maximum	3800	
longidute local	9,2	
longidute of timezone	15	
latitude local	48,9	
Inclination	46	

Orientation	-12	of PV Surface (0 = South, -90 = West, +90 East )
-------------	-----	--

**Figure 23 Example solar forecast configuration data**

With this data it is possible to calculate the maximum sun beam power to an arbitrary oriented surface. There for one have to calculate the real sun time in minutes of the day. An extension was added to the Smartbox to relieve this calculation:



**Figure 24 extended smartbox manager GUI.**

This minute have to be corrected for the local longitude and the invariance of the 23.45 inclination angle of the earth.

The longitude correction is as simple as:

$$\text{Minute of Day Correction} = 4 \times (\text{longitude local} - \text{longitude Time zone}) + E$$

Where E is the invariance of the inclination of the earth angle. It could vary from -13 to +14 Minutes during the year.

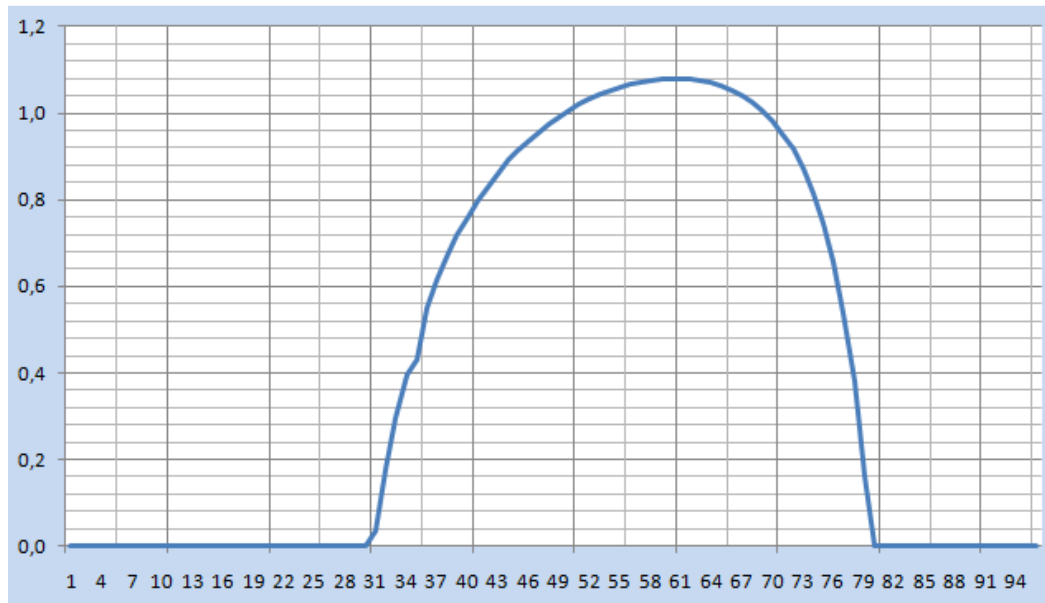
Then the altitude and the Azimuth of the sun could be calculated. With these angles the incident angle to the PV receiver could be determined.

The incident angle together with the current sun height angle builds up the sun Ratio.

The detailed formulas are released for the public in the German by the "VDI-RICHTLINIE 3789, Blatt 2". /39/ The formulas for this implementation were taken from the script written by Prof. Dr. M. Norbert Fisch /40/.

The total ratio calculation within the Smartbox is done in approximate 200 ms (0.2 seconds) for all 96 values a day. The reason for the fast calculation is because most of it is done by optimized machine near assembler code. So this CPU runtime could be neglected and thus could be done every 15 minutes.

The following chart (Figure 25) shows the result with the example data above for the 6<sup>th</sup> of July 2011 with a resolution of 15 minutes.



Ratio for the 6<sup>th</sup> of July 2011 in Kornwestheim. The surface has an azimuth of 12 degrees to the west.  
It is displayed with daylight saving change on.

**Figure 25 solar forecast sample calculation**

One has to notice, that the ratio could exceed 1.0. This is due to the definition of  $E_{dir/h}$ .  $E_{dir/h}$  is defined to be the horizontal sun energy. And this energy could be less than the energy received by a surface inclined to the direction of the sun.

(The little buckle at step 33 is due to a known calculation bug. It could arise when the sun crosses direct east or direct west. It is neglected for the moment.)

This ratio could then be used to calculate the maximum sun power with the following formula:

$$E_{PV} = E_{dir/h} * \text{Ratio}$$

$E_{dir/h}$  is assumed to be a fixed value which is for clear sky approximately 800 W/m<sup>2</sup>. The dependency from the moisture and clouds in the sky is considered in the following calculation steps:

With this ratio for the next 24 hours and the real power meter data, one could estimate the PV-production for next 24 hours.

First the ratio is multiplied with the maximum device output power. This power is then compared against the real power meter at the very moment now (if power is available).

Together with the weather forecast from Q.met /41/ where the cloud coverage is delivered, an average production forecast could be estimated.

If Q.met is not available or if the forecast did not fit the real power measured by the sun power meter, the power of the last 24 hours is used for the estimation. This fits at least as good as a 60% weather forecast.

Since this calculation changes with every 15 minute, the total Least Cost Function have to be recalculated every 15 minutes. So the time calculated to start a dedicated machine as well as the cost profile could change every 15 minutes.

### 4.2.3 Q.met

For the duration of this project <http://www.qmet.de/> is delivering a detailed weather forecast as an RSS feed. The cloud coverage is shown with a resolution of  $1/8^{\text{th}}$ .

The ID's where the cloud cover data ("bedeckung") are stored were written together with the sun forecast configuration data.

```

- <stadt id="2968">
  <name>Maria Anzbach</name>
  <plz>3034</plz>
- <prognose datum="06.06.2011">
  <tmax>26</tmax>
  <tmin>15</tmin>
  <wetter_id_tagsueber>2</wetter_id_tagsueber>
  <nw>5</nw>
  <sd>90</sd>
  <windrichtung>4</windrichtung>
  <windgeschw_kmh>31</windgeschw_kmh>
  <temp_0>17</temp_0>
  <temp_3>15</temp_3>
  <temp_6>15</temp_6>
  <temp_9>19</temp_9>
  <temp_12>24</temp_12>
  <temp_15>25</temp_15>
  <temp_18>24</temp_18>
  <temp_21>20</temp_21>
  <bedeckung_0>4</bedeckung_0>
  <bedeckung_3>3</bedeckung_3>
  <bedeckung_6>1</bedeckung_6>
  <bedeckung_9>2</bedeckung_9>
  <bedeckung_12>6</bedeckung_12>
  <bedeckung_15>3</bedeckung_15>
  <bedeckung_18>2</bedeckung_18>
  <bedeckung_21>1</bedeckung_21>

```

### 4.2.4 The wind forecast Calculator

The wind forecast calculator is much simpler than the sun forecast, since all forecast data is delivered as an RSS Feed from the wind turbine producer "Quiet Revolution". Nevertheless the received data have to be checked and put into the wind forecast array to its appropriate forecast step. Additionally the timeliness of the forecast array has to be supervised very closely to be sure not using forecast data for the wrong time range.

The RSS feed is read every 5 minutes to get as close as possible the 10 minutes forecasts.

Prior every RSS feed read, the total wind forecast array is invalidated. So it is assured, that if there is no data available for all or only some time range, the wind turbine forecast is not accounted by the Least Cost Function for the missing forecast time.

As shown in the RSS example, all time ranges are given as UTC time stamps. This UTC time has to be converted into the local Smartbox MEZ time.

```

- <energyForecast>
  <dateTime>2011-10-30T06:44:29Z</dateTime>
  <baseplateID>ael-base-d</baseplateID>
  <baseplateName>AEL Base D</baseplateName>
  <powerUnits>Watts</powerUnits>
- <periods>
  - <period>
    <periodStart>2011-10-30T06:44:29Z</periodStart>
    <periodEnd>2011-10-30T06:54:29Z</periodEnd>
    <averagePower>102.9</averagePower>
    <sigma>30</sigma>
  </period>
  - <period>
    <periodStart>2011-10-30T06:54:29Z</periodStart>
    <periodEnd>2011-10-30T07:04:29Z</periodEnd>
    <averagePower>502.2</averagePower>
    <sigma>30</sigma>
  </period>
- <period>

```

### 4.2.5 The least cost function (LCF)

With all the preliminary work done, it is now simple to estimate a cost function. Supposing that all the tables and forecasts are up to date, starting at the very moment (quarter hour).



First the LCF takes all the wired (or direct controlled) requests from the request table and calculates them one by one. This is done by summing each request for each period for as long as the maximum allowable time frame:

$$E_{LCF} = \min t \left\{ \sum_t^{MP_{t+n}} MP_x \right\}, t = \text{time}, n = \text{period of machine profile}$$

Where

$MP_x$  = cost for single quarter hour running the request.

Example:

Delay time (minutes):	t = 0	15	30	45	60	75	90	105	120	135	150	165
Buy profile (cents/kWh):	25	25	25	25	25	25	25	25	25	25	25	25
Sell profile (cents/kWh):	50	50	50	50	50	40	40	30	15	15	15	20
Total (planned) Load (Watts)	800	850	900	900	500	600	700	750	800	100	150	200
Total forecast (Watts)	200	300	400	500	700	800	1200	1400	1600	200	100	0
Machine profile to calculate (W):		700	700	500	100	→						
Cost runnint at t (cents):		4,4	4,4	3,1	1,25							
Total cost for starting at t (cents):		13,1										

**Table 2 example for least cost function**

This calculation is done for all time slots (15 minutes) as long as the maximum delay time is allowed to do. The wired machines are then planned to start at the least cost time in the array. The machine profile is added to the total load profile at the least cost time.

For the wireless machines the array is simply outputted to the wireless transceiver. And since the time could not be determined, when the machine really starts, the profile could not be added to the total load and hence not be considered for further calculations.

## 4.2.6 User surface

The first prototype is made up of fixed tables. Nevertheless the connection between the input and output of the requests must be parameterized. Therefore for each machine profile (still fixed for the first prototype) the input instruction ID and the output instruction ID must be put in.

The input data (Start immediately, in 4 hours, ... ) is then read from the permanently updated network table.

The output data is directly written into the current value of the output instruction which in turn drives directly the connected relays.

Read Connections		Write Connections	
Profile Nr.	Machine Name	Input Instruction	Output Instruction
1	dishwasher	54	55
2	washing m.	122	235
3	dryer	487	324
4	block heating	47	35
5	-	0	0
6	-	0	0
7	-	0	0
8	-	0	0
9	-	0	0
10	-	0	0

**Table 3 prototype user interface EUP I/O instruction**

## 4.2.7 Interfaces

As already said: the system lives and dies with data and the up-to-dateness of the data. There for the system needs interfaces to transfer the data to and from the system.

These Interfaces are used for:

- **Wind forecast [input]:** Not yet finalized. Still to be defined.
- **Solar forecast [inputs]:** The solar forecast calculator relies on a bunch of Interfaces: Weather forecast (RSS Feed), last day sun and current sun power (electrical power meter / MBus).
- **Request event [input]:** Wired Expert Field bus module (RS485)
- **Switching machine [output]:** Wired Expert Field bus module (RS485)
- **Current charge profile [output]:** Wired RS232 (Field 2) to wireless transceiver.

This last interface is new and has to be defined for both sides. Proposal could be:  
The current charge profile sends on every change its content to the RS232 without any further actions (forward push). This content looks like:

STX	CM D	L	TIME	Cents 0	Cents 1	Cents 2	Cents 3	Cents n	CRC16	EOT
-----	---------	---	------	---------	---------	---------	---------	---------	-------	-----

Where:

STX	=	0x02
CMD	=	0x00 // left for future use
L	=	Length of Data in bytes (incl. TIME ... CRC16 ). Max 255 bytes.
TIME	=	32 bit unsigned integer: Seconds since 1.1.1980
Cents 0...n	=	16 bit signed integer. The number has to be divided throw 10 to get cents/kWh.
CRC16	=	CRC (from L ... Cents n )
EOT	=	0x04

The transfer takes place at: 115200, 8, N

Typically there will be a data count (L) of  $12 \text{ hours} * 4 \frac{1}{4} * 2 \text{ bytes} + \text{CRC16} + \text{TIME} = 102$ .

But – if there is no forecast available – it could be less!

The wireless transceiver has to acknowledge with a single ACK (0x06) within 2 seconds. Otherwise the system repeats the messages three times. If all attempts failed a warning is created and will be send to all recipients defined on the state page.

## 4.2.8 Persistent Data

To supervise the new behaviour, all decisions made by the least cost function, the requester as well as the forecast calculators is stored into the local flash memory.

This data could be read out via the COM-interface – which in turn – has to be expanded to read the extra marked data.

The new data will become new data type numbers:

Request Table:	9011 (request with LCF start times)
Forecast Wind:	9012 (current forecast for the next 12 hours)
Forecast Sun:	9013 (current forecast for the next 12 hours)
Charge Profile	9014 (current forecast for the next 12 hours)

Each Table and profile is stored on every change. At least every 15 minutes since the total calculation must recalculate every quarter hour to maintain the 12 hour forecast.

The data stored looks like the current charged profile which is already defined above. However without the STX, CMD, L, EOT and CRC since these entries are already available.

The expanded COM-interface will then be able to extract the stored data for further examinations. An Excel example shows how to play with the new data. If one wants read the data into a professional data base (e.g. ennovatis controlling) 48 data points has to be defined for each profile.

#### **4.2.9 Performance, Capacity and Scalability (PCS)**

There is no issue about performance. The time used to calculate the new start times doesn't concern the overall system performance.

#### **4.2.10 Compatibility**

The new functionality does not affect the Smartbox manager tool nor the ennovatis controlling. However there is a lot off forecast data stored and read out, it is not intended to import this data directly into these systems.

The rest of the Smartbox system (Firmware functionalities, SBM, ...) is to be used as if the new functions where not there.

### **4.3 Restrictions of functionality**

#### **4.3.1 Constraint #1**

As already said in the preliminary consideration it is only possible to calculate a cost profile, if the local load doesn't change.

Reason:

Supposed the charges for buy and sell changes and differ (as it is necessary in a smart grid world). Then consider the following example:

Actual charge to buy: 95cents (high noon)

Actual charge to sell: 25cents (same time but the local supplier wants to earn too)

Actual balance: Selling 200 Watts (for 25 cents).

Well the cost function calculates 25 cents which is send to the local (wireless connected) devices for the current charge.

The (wireless connected) dryer then decides to start drying with the calculated 25 cents. And now the balance alters from 200 Watts sell to 1800 Watts buy. The dryer draws now 1800 Watts from the main grid with 95 cents.

This problem increases the more wireless devices are connected with the same algorithm.

#### **4.3.2 Constraint #2**

The total forecast time range is limited to 12 hours with a resolution of 15 minutes.

Elongated forecasts need more memory, detailed and more accurate wind and sun forecasts. All of which isn't available yet.

### **4.4 Next steps in the development of the SmartCoDe EMU**

#### **4.4.1 Improvement of Module Power management**

Energy management in SmartCoDe will be based on the power management module described in chapter 4. The basic idea is to determine the optimal power to be used in the actual situation and to distribute it to the available EuPs according to the decided rules of network operation. To make this module flexible enough to handle all the energy management tasks relevant to SmartCoDe several improvements are necessary. The goal is to allow more general input and to make the formulation of the optimum power more flexible. To reach this goal it is necessary to allow both impulse and

analogue input. From these it will become possible to calculate the optimum power as a function of time and network conditions like price or availability. In addition output will be allowed to influence network variables to enable the inclusion of more general operation rules.

After such modifications we will be able

- To include MBus meters into the power optimisation
- To allow for variable optimal power to reflect price or availability information
- To integrate several LEPs
- To use the power calculator of the Smartbox manager not only for heat and cold but also for electricity

It is anticipated to install these changes in the second quarter in 2011.

#### 4.4.2 Update of EMU CPU

The present version of the Smartbox has limited memory and processing capabilities. Therefore the Smartbox in its present stage is not capable to handle the wireless protocol completely. Therefore for a first approach a concept was agreed upon, which is that we implement a simplified and more general protocol for communication between Smartbox and SmartNode (the format of message would be as we discussed earlier), and to leave the SmartNode to handle the conversion to and from the ZigBee. According to the experiences made during this first phase a more general solution will be sought in the third year of the project. To allow such a solution we plan to upgrade the Smartbox by implementing a new processor and a more moderate operation system. Details will be described as soon as they are available.

### 4.5 Description of interfaces

#### 4.5.1 Description of the interface for weather forecasting

In order to acquire weather data needed for estimating future energy consumption and production, the Smartbox uses web-based weather services. There are two kinds of such services: free and commercial. First, Smartbox is establishing a TCP/IP connection with weather forecast server, by sending a request which contains information about geographical location of a place of interest. A server provides meteorological data through RSS feeds. Unfortunately, format of feeds is not standardized, which means that different weather services may provide the same data that is represented in different ways. As an example, we observe RSS feeds from two free weather forecast services for Ludwigsburg, Germany:

##### Example 1

A request sent to "world weather online":

```
"GET
http://free.worldweatheronline.com/feed/weather.ashx?q=Ludwigsburg,Germany&format=xml&num_of_days=2&key=xxxxxxxxxxxxxxxxxxxxxxxxxxxxx"1
```

An RSS reply from "world weather online":

```
"<?xml version="1.0" encoding="UTF-8"?><data><request><type>City</type><query>Ludwigsburg,
Germany</query></request><current_condition><observation_time>12:47
PM</observation_time><temp_C>7</temp_C><temp_F>45</temp_F><weatherCode>113</weatherCode><weatherIconUrl><![CDATA[http://www.worldweatheronline.com/images/wsymbols01_png_64/wsymbol_0001_sunny.png]]></weatherIconUrl><weatherDesc><![CDATA[Sunny]]></weatherDesc><windspeedMiles>1</windspeedMiles> . . ."
```

##### Example 2

<sup>1</sup> "API key" is generated after registration process on <http://www.worldweatheronline.com/register.aspx>

A request sent to "yahooapis":

```
"GET http://weather.yahooapis.com/forecastrss?w=20066863&u=c"2
```

An RSS reply from "yahooapis":

```
"<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
..<rss version="2.0" xmlns:yweather="http://xml.weather.yahoo.com/ns/rss/1.0"
xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
...<channel>
..
<title>Yahoo! Weather - Ludwigsburg, DE</title>
<link>http://us.rd.yahoo.com/dailynews/rss/weather/Ludwigsburg__DE/*http://weather.yahoo.com/f
orecast/GMXX5286_c.html</link>
<description>Yahoo! Weather for Ludwigsburg, DE</description>
<language>en-us</language>
<lastBuildDate>Mon, 18 Oct 2010 3:00 pm CEST</lastBuildDate>
<ttl>60</ttl> . . ."
```

In summary we can say:

- The configuration of the RSS feeds must be definable, thus allowing the possibility of using many different sources.
- First, the configuration will be set up in very simple way by using Excel macro. This document describes how it is set up.
- There can be up to 3 RSS feeds simultaneously defined.
- The feeds contain up to 50 readable and addressable variables.
- The feeds can be saved in the moment of reading for monitoring purposes. The current values are again read by Excel macro.
- At the start and 6 hours after an RSS server becomes unavailable, the feed data in memory become invalid. As a result, referenced instructions suspend work.
- If format or content of the feed changes, the defined values are no more valid as well, thus the values become invalid after 6 hours.
- Refresh rate of the feeds is adjustable in minute steps from 0 (off) to 254 (4h 14min).
- The feeds are independent of the program list created in Smartbox. The normal program execution is delayed only slightly.

When a feed has been defined by Excel macro, a minute counter starts counting. After the first minute and then every n minutes starts reading of an RSS feed. The data will be parsed - as defined - in a list of values stored in Smartbox. From there, all the data can be accessed by ID.

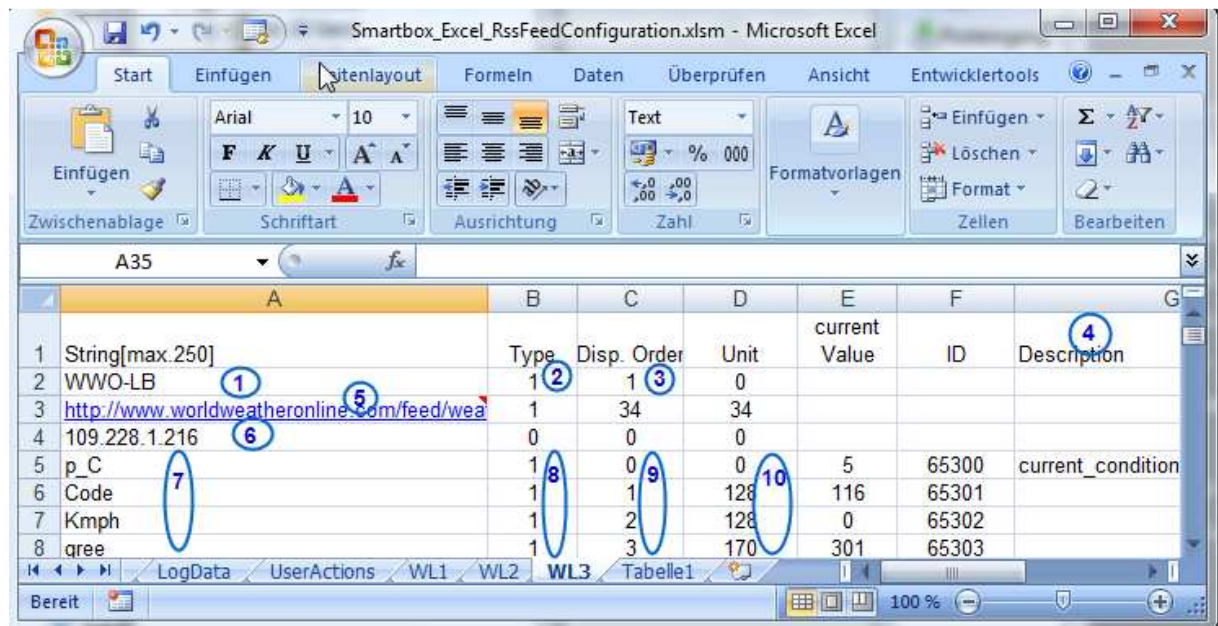
The IDs go from 65200 to 65249 for the first feed, from 65250 to 65299 for the second feed, and from 65300 to 65349 for the third feed.

### The Excel macro

The Excel macro is used to define parameters needed to parse data from RSS feeds:

---

<sup>2</sup> Yahooapis do not require personal user ID. Instead, a Location ID (red) and the unit in "C" (blue underlined) are used.



	A	B	C	D	E	F	G
1	String[max.250]	Type	Disp. Order	Unit	current Value	ID	Description
2	WWO-LB	1	1	0			
3	<a href="http://www.worldweatheronline.com/feed/wea">http://www.worldweatheronline.com/feed/wea</a>	1	34	34			
4	109.228.1.216	0	0	0			
5	p_C	1	0	0	5	65300	current_condition
6	Code	1	1	128	116	65301	
7	Kmph	1	2	128	0	65302	
8	qree	1	3	170	301	65303	

**Figure 26 weather forecast parameter definition for RSS feeds**

- Any name, up to 12 characters long. Longer names are truncated automatically. The name is used when importing into the parent folder.
- By "Type", the cycle in minutes is entered in line 2. The value of 1 means: update every minute. A cycle of 0 or 255 switches the feed off permanently.
- By „Display Order" in row 2, a "1" or "0" will be saved in the database.
- In column "G" can be any textual comment.
- In A3 cell, the "Request URL" is defined (without "GET" command).
- In the fourth line is the IP address of the feed server because Smartbox doesn't support the DNS.
- Then follows the information used to parse the feed:

The first entry "p\_C" is a string preceding the first value to be parsed. Here it is the current temperature. It should be noted that the recognition string always ends with the same terminator character (in this case ">"), after which the actual value follows. Note: parsing string is case-sensitive!

```

- <data>
- <request>
  <type>City</type>
  <query>Ludwigsburg, Germany</query>
</request>
- <current_condition>
  <observation_time>12:01 PM</observation_time>
  <temp_C>5</temp_C>
  <temp_F>41</temp_F>
  <weatherCode>116</weatherCode>

```

- Type determines whether an integer value (1), a floating point value (2) or a string (3) is read. The value 3 (not processed) is used, if there is a need to jump on a point further in the feed, and various ambiguities lay in between.
- By display order, we set in which place (ID) the value is written. In the simplest case, it starts with 0 and increments with each additional value. In the case of change, or when a different server is chosen, the "old" ID can remain in the program which is to be executed.
- Under "Unit", the encoded unit is registered, as used in the Smartbox. In the worksheet "units" a list of valid values is stored.



When all the detection strings, types, disp. orders and units are defined, the configuration is transferred to the Smartbox with the "Write WL" command. After a short wait, the result can then be read by "Read Values".

It should be noted, that a TCP/IP socket is used for reading a feed, which is normally used for communication with or controlling the SmartboxManager. If all of the 3 sockets are in use simultaneously, the reading of the feed fails. Conversely, if a feed is active, a maximum of 2 connections remain.

## **4.5.2 Description of the interface between EMU and the wind turbine**

### **4.5.2.1 SmartCoDeNodes & Cost Summary of the Energy Forecast Interface Specification /26/**

The quietrevolution Energy Forecast Interface consists of an ethernet connection giving access to a web server. Standard HTTP GET or POST requests are sent to this web server by the client's software. The server responds with XML documents containing predictions of the amount of energy that a quietrevolution turbine will generate over the coming hours and days.

#### **4.5.2.2 Interface Specification**

##### **4.5.2.2.1 Ethernet Hardware**

quietrevolution will provide a standard ethernet interface via an 8P8C ("RJ45") socket. The interface will switch automatically between 10 or 100MB/s depending on the capabilities of other devices on the network.

The client should connect to this socket using a standard ethernet ("Cat 5e") cable and our system will switch automatically between straight and cross-over configurations if needed - a cross-over cable is not required.

##### **4.5.2.2.2 Internet Protocol (IP)**

The Energy Forecast interface will use a static IPv4 address, subnet mask and router IP address. If these are not specified by the client prior to installation then default values will be used. The client may specify any legal values for these settings.

Automatic configuration of the Energy Forecast interface's IP settings (for example via DHCP) is not possible.

##### **4.5.2.2.3 HTTP**

The Energy Forecast interface runs a standard web server on TCP port 80. Requests are served at a maximum rate of approximately one per second, and the server will only serve around five concurrent clients. Requests which exceed these limits may be ignored or may receive an HTTP error response.

The web server does not authenticate users or encrypt any data. It is the client's responsibility to protect the turbine's web server from access by unauthorised users. Because of the rate limits set in the server, data supply to client's software may well be disrupted if any unauthorised users are also accessing the server.

Both GET and POST types of HTTP request are acceptable. The interface serves a single page called "energyForecast".

For example, if the IP address you have specified for the Energy Forecast interface is "192.168.2.1", then you can request the following URI using an HTTP GET request:

<http://192.168.2.1/energyForecast>

This can be done from a web browser (for testing purposes) or by some custom software (for integration into a client system). If the energy forecasting system is not working then the web server will return a "503 Service Unavailable" HTTP status code.

##### **4.5.2.2.4 XML**

The data is returned in XML format. Information about XML is widely available on the internet and an example can be found later in this document.



Libraries to parse XML are available for many common programming languages. Alternatively, custom software can be written to extract the required data.

An XML schema in RELAX-NG format is included in Appendix One. Note that the XML documents themselves do not mention the RELAX-NG schema. Including a schema reference in the XML document is becoming less popular.

The XML namespace used throughout is <http://quietrevolution.com/smartcode>. Note that this is not an internet address and no file is available at this location, it is merely the name of the namespace.

The request described in the preceding section should result in an XML document being returned via HTTP. The format of this XML document is best explained using an example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE energyForecast SYSTEM "http://static.quietrevolution.co.uk/xml/schemata/energyForecast.dtd">
<energyForecast xmlns="http://quietrevolution.com/smartcode">

  <dateTime>2011-10-30T06:44:29Z</dateTime>
  <baseplateID>ael-base-d</baseplateID>
  <baseplateName>AEL Base D</baseplateName>
  <powerUnits>Watts</powerUnits>

  <periods>

    <period>
      <periodStart>2011-10-30T06:44:29Z</periodStart>
      <periodEnd>2011-10-30T06:54:29Z</periodEnd>
      <averagePower>102.9</averagePower>
      <sigma>30</sigma>
    </period>

    <period>
      <periodStart>2011-10-30T06:54:29Z</periodStart>
      <periodEnd>2011-10-30T07:04:29Z</periodEnd>
      <averagePower>502.2</averagePower>
      <sigma>30</sigma>
    </period>

    <period>
      <periodStart>2011-10-30T07:04:29Z</periodStart>
      <periodEnd>2011-10-30T07:14:29Z</periodEnd>
      <averagePower>102.9</averagePower>
      <sigma>50</sigma>
    </period>

    <period>
      <periodStart>2011-10-30T07:14:29Z</periodStart>
      <periodEnd>2011-10-30T07:24:29Z</periodEnd>
      <averagePower>102.9</averagePower>
      <sigma>60</sigma>
    </period>

    <period>
      <periodStart>2011-10-30T07:24:29Z</periodStart>
      <periodEnd>2011-10-30T07:34:29Z</periodEnd>
      <averagePower>602.7</averagePower>
      <sigma>30</sigma>
    </period>
  </periods>
</energyForecast>
```

#### 4.5.2.2.5 Example energy forecast XML file

The encoding and DOCTYPE will not change. The following table details the various data fields:

Tag Name	Description
baseplateID	A string set by quietrevolution during installation. It uniquely identifies each turbine to the quietrevolution operations system.
baseplateName	A string used to identify the turbine to human users (for display on websites etc).
dateTime	The time that the forecast was generated, in ISO 8601 format YYYY-MM-DDThh:mm:ssZ. The T separates the date from the time and the Z indicates that the time is Co-ordinated Universal Time (UTC) - i.e. it does not include the local time zone or any allowance for Daylight Saving Time ("Summer Time"). UTC is always used.
powerUnits	Units for the power forecasts. Within the SmartCode project this will always be "Watts" indicating that the power is measured in Watts.
period	Contains information about a specific time-period in the future. Does not contain any child text, just the tags listed below.
periodStart	Start of the time period, formatted as per dateTime field (see above).
periodEnd	End of the time period, formatted as per dateTime field (see above).
averagePower	Predicted average power output of the turbine during the period.
sigma	A measure of the uncertainty associated with the prediction. A larger value indicates greater uncertainty. Nominally the standard deviation of the posterior distribution over the possible power output levels.

The time periods will always be contiguous (one will start when the previous one ends) and in ascending time order (earliest time first).

#### 4.5.2.3 Appendix: Schema

This appendix contains the schema for the XML document in the RELAX-NG format.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
  quiet revolution energyForecast XML schema in RELAX-NG format
  The XML files that conform to this schema provide forecast of the turbine's energy output.

  www.quietrevolution.co.uk
  (c) quietrevolution 2011
-->
<grammar datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes"
  xmlns="http://relaxng.org/ns/structure/1.0"
  ns="http://quietrevolution.com/smartcode">

  <!-- Our top-level element is energyForecast -->
  <start>
    <ref name="energyForecast.element" />
  </start>

  <define name="energyForecast.element">
    <element name="energyForecast">
      <ref name="energyForecast.content" />
    </element>
  </define>

  <!-- Define the content of the top-level element separately so that we can extend it -->
  <define name="energyForecast.content">
    <interleave> <!-- note that elements may occur in any order -->
      <element name="dateTime"><data type="dateTime" /></element>
      <element name="baseplateID"><text /></element>
      <element name="baseplateName"><text /></element>
      <element name="powerUnits"><text /></element>
      <element name="periods"><ref name="periods.element" /></element>
    </interleave>
  </define>

  <!-- Define the types that we used above -->

  <define name="periods.element">
    <element name="periods">
      <zeroOrMore><ref name="period.element" /></zeroOrMore>
    </element>
  </define>

  <define name="period.element">
    <element name="period">
      <interleave> <!-- note that elements may occur in any order -->
        <element name="periodStart"><data type="dateTime" /></element>
        <element name="periodEnd"><data type="dateTime" /></element>
        <element name="averagePower"><data type="decimal" /></element>
        <element name="sigma"><data type="decimal" /></element>
      </interleave>
    </element>
  </define>
</grammar>

```

## 5 SmartNodes & Cost-function based EUP control

In D-1.1.1 /24/, we presented a classification of EuPs as clusters. To the EMU SmartNodes will be will be wireless connected. To the SmartNodes EUP of different classes and or sensor (meter) will be connected. The control of EUPs through the SmartNode is the topic of the next chapters.

In the next part the description is given how each class is handled regarding Energy Management - using the cost-function paradigm.



**Figure 27 EuP classification from D-1.1.1 /24/**

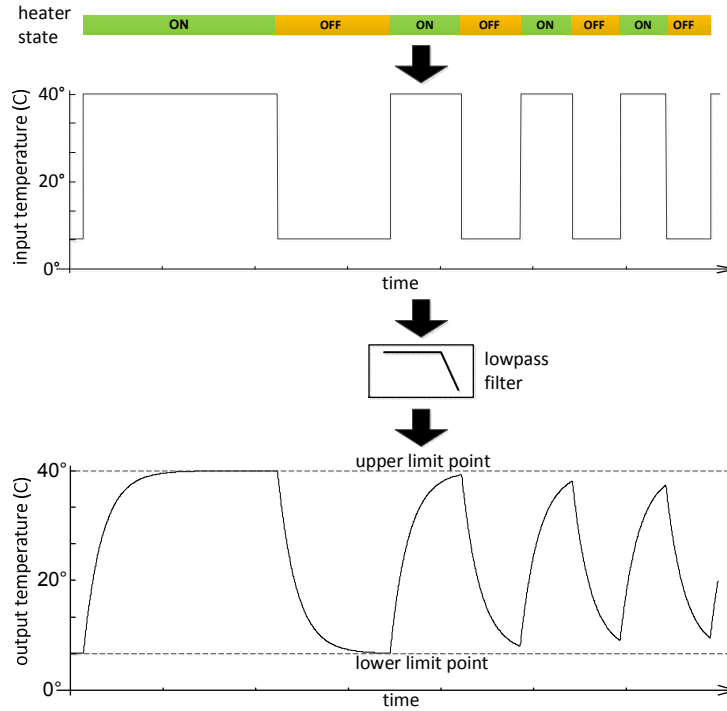
On the wireless to the EMU connected SmartNodes will be a EUP and or sensor (meter) connected. The control of EUPs through the SmartNode is the topic of the next chapters

## 5.1 VSTSVC - Virtually Storable Service

The advantages and properties of virtual storages like freezers or heaters in that regard are well known, see for example /22/ or /23/. The core idea is to cool down (resp. heat up) in favourable times (e.g. when wind energy is abundant or grid energy is cheap), such that the appliance can be turned off in unfavourable times. This is possible since the (virtually always thermal) services provided by VSTSVC EuPs arise inert enough. However, the time frames involved can reach from several minutes (fridge) to even a day for large building air-conditioning. Forecasts on the energy conditions are not essential for DSM with VSTSVC EuPs. They can react instantaneously to changing conditions, unless they are at the "opposite end" of their boundary conditions, e.g. fully cooled down to their lower threshold already when the solar panel output increases. If energy forecasts are taken into account, such circumstances can be avoided. However, a temperature forecasts must be used also in order to make control plans.

The temperature forecast used in the SmartCoDe node for a VSTSVC EuP works as follows: It is a well-known fact that the temperature profile of heaters, fridges etc. can be modelled using a lowpass filter in the time domain.

Figure 28 shows how this works for a heater:



**Figure 28 Modelling heater temperature with a low pass filter**

Depending On the current state of the heater, the input to the low pass is set to the temperature where the room temperature (of the room heated by the heater) would converge to if this state would be kept indefinitely. In Figure 28 we assume that the room would heat up to 40°C if the heater keeps turned on, and would cool down to 10°C if the heater keeps turned off. On the bottom of

Figure 28, the resulting room temperature is shown. However, a heater is usually controlled such that the room temperature stays in good distance from these upper and lower limit points.

In *discrete time*, a low pass can be modelled as follows: If  $x$  is the input of the low pass and  $y_{i-1}$  was the output of the low pass in the last time-step, then the output in the next time-step is

$$y_i = \alpha \cdot x + (1 - \alpha) \cdot y_{i-1} \quad (1)$$

where  $\alpha$  is the so called *smoothing factor*, which lies between 0 and 1, and characterises the low pass in this discrete time view. This can be used for temperature forecasting as follows: The SmartCoDe node takes temperature measurements in constant time steps, e.g. every minute, while storing the last measurement. The aforementioned limit temperatures are basically known: The temperature for switching off is always the temperature of the environment, e.g. the temperature of the kitchen the fridge is located. The temperature for switching on is device-dependent, possibly with some influence of the environment temperature. Therefore, if  $t_{current}$  is the temperature measured in the current time step,  $t_{old}$  is the temperature measured in the previous time step, and  $t_{limit}$  be the limit temperature for the current state (e.g. on or off) of the EuP, we get the following equation from (1):

$$t_{current} = \alpha \cdot t_{limit} + (1 - \alpha) \cdot t_{old} \quad (2)$$

Solving for  $\alpha$  yields:

$$\alpha = \frac{t_{current} - t_{old}}{t_{limit} - t_{old}} \quad (3)$$

provided  $t_{limit} \neq t_{old}$ , which will always be the case in practical applications. That is, we can characterise the thermal process by a single number (namely  $\alpha$ ), which can be learned successively by simple temperature measurement and the knowledge of the current state and the limit temperatures. This method has been implemented in the simulation environment with good results.

Another possibility is to learn the limit temperatures *together* with  $\alpha$  by successive measurement and curve-fitting, which would work as follows: When the SmartCoDe node switches its VSTSVC EuP it successively changes the temperature from the current temperature  $t_0$  to the unknown temperature  $t_{limit}$ . Let's assume we take  $k$  temperature measurements  $t_1, \dots, t_k$  before the node switches the EuP into a new state. If we solve the recursion in (1) applied to this situation we get

$$\tilde{t}_k = t_{limit} + (1 - \alpha)^k ((\alpha - 1)t_{limit} + t_0) \quad (4)$$

Now we can determine  $t_{limit}$  and  $\alpha$  by fitting this curve to the temperature measurements  $t_1, \dots, t_k$ . Experiments with the least square method applied to values determined in the simulation yielded good result. However, the least square method is most likely too costly to be implemented on the microprocessor.

If  $t_{limit}$  and  $\alpha$  are known, it is now straightforward to compute a forecast of the state changes for a certain future time interval. Let the function

```
decide_new_state(time, temp, state);
```

compute the new state of the EuP out of the current time (which is needed to access the cost function), the current temperature, and the current state. Then we just successively call this function with the respective temperature forecasts for the future time steps:

```
int state = current_state;           // start with EuPs current state
double temp = current_temp;         // ... and the current temperature
int control_plan[max_steps];
for(i=1; i<=max_steps; i++)
{
    temp = alpha*t_limit(state)+(1+alpha)*temp;
    state = decide_new_state(current_time+i*time_step, temp, state);
    control_plan[i-1] = state;
}
```

The variable `time_step` holds the time difference between each function calls. The array `control_plan` now holds the forecast for the subsequent states of the EuP for the next `max_steps` steps. Since the power consumption for each state is known, it is now straightforward to compute the power consumption forecast.

The `decide_new_state` algorithm is implemented as follows:

Let `cost` be the average weighted cost (normalized to the interval [0,1]) from *now* to time *now+p/2*, where *p* was the last switching period, i.e. the time-difference between the last two switch-on events. Let `lb` be the lower bound for the temperature, and `ub` the upper bound, and `temp` the current temperature. Then switch according to the following rules (in case of a cooling device like fridges or air-conditioning):

```
if(state==0off && temp > ub - (1-cost) * 0.5 * (ub - lb) )
    state = 1on;
if(state==1on && temp < lb + cost * 0.5 * (ub - lb) )
    state = 0off;
```

For heaters, it is

```
if(state==0off && temp < lb + (1-cost) * 0.5 * (ub - lb) )
    state = 1on;
if(state==1on && temp > ub - cost * 0.5 * (ub - lb) )
    state = 0off;
```

That is, we alter the upper and lower temperature thresholds bounds according to the cost function. Figure 29 Load balancing of 4 fridges with 4 cost functions

shows this algorithm in action. We have four fridges with similar parameters, and use 4 periodic (sine-wave) cost functions shifted by 90°. The cost function values range from 0 to 7. The left half shows the outcome before the cost functions are transmitted and usual bang-bang control is used. Obviously, the total power consumption ("power\_total") is evened-out significantly when the cost function-based control kicks in, as can be seen by the sample variance plotted at the bottom.

Figure 29 Load balancing of 4 fridges with 4 cost functions

also shows the results of the  $\alpha$ -estimation and the forecasts for the state changes (the latter starts also only when the cost functions are available).

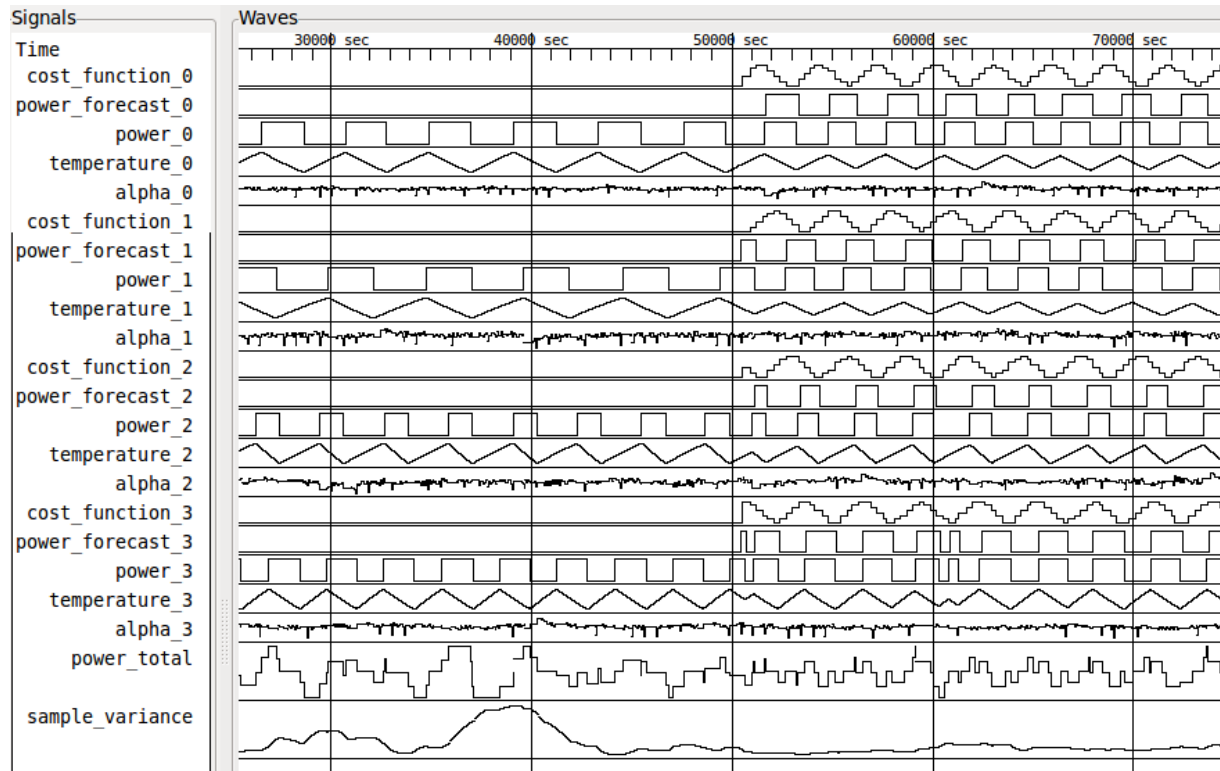


Figure 29 Load balancing of 4 fridges with 4 cost functions

## 5.2 SKDSVC - Schedulable Service

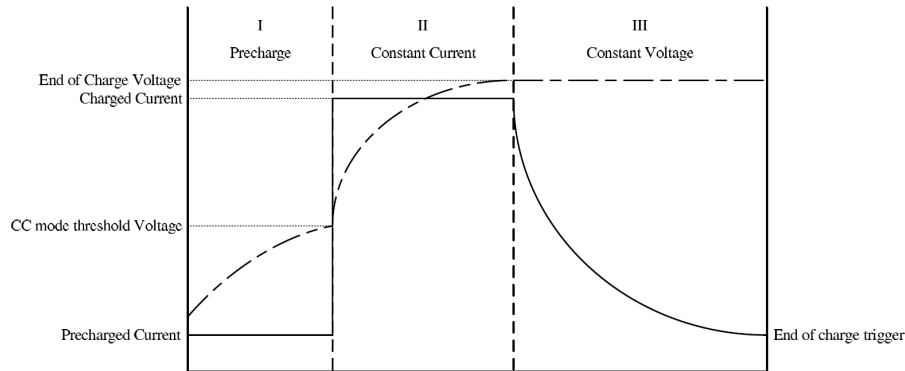
The idea of using EuPs with a schedulable service (like washing machines or dishwashers) for DSM is simple: In addition to a program, the user inputs a latest stop time until the service has to be finished. Then the program is started within this time frame such that the runtime lies within favourable energy conditions. This can be done effectively only if a forecast on the energy conditions is available. Also, having a typical load profile available for the specific program is helpful for planning, since SKDSVC EuPS will seldom draw a constant load when they run. A washing machine, for example, has typically a load peak at the program start, when water is heated up, and at the end of the program during centrifuging.

In principle, it would even be possible to interrupt a specific program. However, this might result in a quality loss in the respective process (e.g. washing clothes), as was suggested when consulting an associated partner in the white goods sector. Therefore, scheduling strategies including interruption have to be developed closely with the process experts, i.e. the white goods manufacturers. At the moment, we only consider to run a specific program completely once it's started without any interruption.

## 5.3 CHACON (Charge Control) and COMCON (complete control)

Regarding DSM, battery chargers are somewhat in between VSTSVC and SKDSVC. On one hand, they can be regarded as providing a service (the charge status), which only deteriorates marginally when the charging is stopped; this resembles VSTSVC. On the other hand, the ultimate goal is a full charging, possibly until a certain deadline. This aspect is more like SKDSVC, especially when keeping in mind that the charging process might even exhibit a specific load profile, see Figure 30.





**Figure 30 Typical Charge Profile for Lithium-ion Batteries, taken from [20]**

Depending on the rechargeable battery technology it might not be advisable to interrupt the charging process once it has started (battery lifetime issues), i.e. the safest strategy is to plan charging in one go according to an energy forecast, taking charging profiles like in Figure 30 into account. Therefore, CHACON EuPs can indeed be treated like SKDSVC. However, with a more detailed knowledge on the battery's characteristics, more elaborate schemes can be developed.

An interesting variation of CHACON is the COMCON class, which encompasses robotic services where the EMU can have complete control on when to charge and also when to use up the stored energy. A good example here is a robot-lawnmower which is controlled such that it mows the lawn of an estate completely on average every two weeks. We won't work out details on DSM of such robotic services in the project, since they are still very exotic, and there are also many ways to trigger the operation (e.g. sensors, minimal duty cycles or user demand) depending on the precise application. But it is obvious that this class provides excellent leverage regarding DSM.

## 5.4 VARSVC (Variable Services) and ETOSVC (EventTimeOut Service)

We collect these two classes together here since they mostly cover lighting applications, although fans fall also into the VARSVC class (the cooling effect is not inert), and e.g. a public message screen operated by a presence detector would be part of the ETOSVC class. For our typical application scenario, a household or office in the European Union which is connected to the public grid, VARSVC provides virtually no leverage regarding DSM. While it would be possible in general to dim lights according to the current energy conditions, it is unlikely that the typical user in this scenario would tolerate that. However, this may be different in islanded scenarios (i.e. with no grid connection), where such unpleasantness might be tolerable to ensure more vital operations like a running fridge.

The DSM possibilities of the ETOSVC class are minimal regardless of the scenario. The only possibility would be to manipulate the timeout-thresholds according to the energy conditions.

Therefore, we foresee no cost-function dependent EM for VARSVC and ETOSVC in SmartCoDe. However, the basic approach in these classes, e.g. diming lights in order to exploit natural light and using presence detection is energy-saving per se. Also, the two approaches could be combined.

## 5.5 CUSCON – Custom Controlled

This collects all these EuPs which won't allow for automated DSM activities, mostly because of high user interaction. While we don't foresee cost-function based EM, we still allow for the EMU to switch CUSCON EuPs directly. That way, the user can still define custom EM schemes, or even use the SmartCoDe infrastructure for safety purposes, e.g. switching off an oven when no one is present in an apartment.

# 6 Conclusion and future work

- The main direction of the developed EMS is fixed. The progress with first SW implementations eg forecasting was successful, the simulation work shows in this first step good results.
- The next step is to develop, granulate and implement the new software for first tests into the Hardware.

## 7 References

- /1/ Alexander M. Zhivov, et al. IEA ECBCS Annex 46, Subtask A, Energy and Process Assessment Protocol for Buildings and Building Sites <http://www.annex46.org/>,
- /2/ Christian Neumann Operating Agent Building EQ homepage [www.buildingeq.eu](http://www.buildingeq.eu)
- /3/ Christian Neumann et al., Requirements for data and measurement equipment -Guidelines for the evaluation of building performance, Building EQ WP 3 report, [www.buildingeq.eu](http://www.buildingeq.eu)
- /4/ Fritz Schmidt et al., Building EQ WP 6 report
- /5/ „A Specifications Guide for Performance Monitoring Systems“, Haves et. al., Lawrence Berkley National Laboratories, USA, 2006  
<http://cbs.lbl.gov/performance-monitoring/specifications>
- /6/ „Continuous Commissioning Guidebook“, Claridge et.al., Energy Systems Laboratory, Texas A&M University, Federal Energy Management Program, USA, 2002  
<http://eber.ed.ornl.gov/commercialproducts/contcx.htm>
- /7/ „Advanced Utility Metering“, National Renewable Energy Laboratory, USA, 2002  
[http://www1.eere.energy.gov/femp/operations\\_maintenance/om\\_advmetring.html](http://www1.eere.energy.gov/femp/operations_maintenance/om_advmetring.html)
- /8/ „International Performance Measurement and Verification Protocol“, USA, 2001  
[http://www.evo-world.org/index.php?option=com\\_content&task=view&id=40&Itemid=63](http://www.evo-world.org/index.php?option=com_content&task=view&id=40&Itemid=63)
- /9/ “Operation & Maintenance Best Practice – A guide to achieve Operational Efficiency”, PNNL for FEMP, USA 2004  
[http://www1.eere.energy.gov/femp/operations\\_maintenance/om\\_bpguide.html](http://www1.eere.energy.gov/femp/operations_maintenance/om_bpguide.html)
- /10/ “Using intelligent data analysis to detect abnormal energy consumption in buildings”, John E. Seem, Energy and Buildings, Elsevier, 2006
- /11/ Energy management by linked local controllers, found at the CSIRO homepage  
<http://www.csiro.au/science/Intelligent-Energy-Controllers.html#domestic> Feb. 2011
- /12/ D. Nestle et al. Chapter 3 in INTERNATIONAL WHITE BOOK ON THE GRID INTEGRATION OF STATIC CONVERTERS, NoE DERlab Deliverable 2.8 Sept. 2009  
[http://www.derlab.eu/media/pdf/docs/DERlab\\_D2.8\\_whitebook\\_static\\_converters\\_rev0.pdf](http://www.derlab.eu/media/pdf/docs/DERlab_D2.8_whitebook_static_converters_rev0.pdf)
- /13/ J. D. Kueck, R.H. Staunton, S. D. Labinov, B.J. Kirby, Microgrid Energy Management System, ORNL/TM-2002/242, Jan. 2003
- /14/ Energy management as defined in wisegeek, <http://www.wisegeek.com/topics/energy-management-systems.htm>, found February 2011
- /15/ International Energy Conservation Code <http://www.internationalcodes.net/2009-international-energy-conservation-codes.shtml>, found February 2011
- /16/ Process model of ENERGY STAR  
[http://www.energystar.gov/index.cfm?c=guidelines.guidelines\\_index](http://www.energystar.gov/index.cfm?c=guidelines.guidelines_index) partners , found February 2011
- /17/ “Commissioning tools For improved energy Performance”, Results of of IEA ECBCS ANNEX 40 (International Energy Agency, Energy Conservation in Buildings and Community Systems), 2004
- /18/ „Continuous Commissioning Guidebook“, Claridge et.al., Energy Systems Laboratory, Texas A&M University, Fed-eral Energy Management Program, USA, 2002
- /19/ Bidirektional Energiemanagement Interface, [http://www.iset.uni-kassel.de/pls/w3isetdad/www\\_iset\\_new.main\\_page?p\\_name=7012091&p\\_lang=eng](http://www.iset.uni-kassel.de/pls/w3isetdad/www_iset_new.main_page?p_name=7012091&p_lang=eng), found February 2011
- /20/ Dung, L.-R., & Yen, J.-H. (2010). ILP-based algorithm for Lithium-ion battery charging profile. Proceedings of the 2010 IEEE International Symposium on Industrial Electronics (ISIE), (S. 2286 - 2291). Bari, Italy.
- /21/ Hájek, J., Špánik, J., & Hvizdák, F. (2011). Architecture and Protocol software of a SmartCoDe node, initial version.

- /22/ Kupzog, F., & Roesener, C. (2007). A closer look on load management. Proceedings of the 5th IEEE International (S. 1151–1156). Vienna, Austria: IEEE Computer Society.
- /23/ Malik, A. S., & Cory, B. J. (1997). Impact of DSM on energy production cost and start-up and shut-down costs of thermal units. Proceedings of the Fourth International Conference on Advances in Power System Control, Operation and Management, (S. 650-655). Hong Kong.
- /24/ Wysoudil, C., Damm, M., Bertényi, T., & Young, T. (2010). Model of local energy resource cluster.
- /25/ ZigBee Alliance. (2008). ZigBee Smart Energy Profile Specification, Revision 15, ZigBee Document 075356r15.
- /26/ Tom Young, Principal Engineer for Control and Monitoring; Prepared for Internal Quiet Revolution Ltd and SmartCoDe project partners.
- /27/ eDiana Embedded Systems for Energy Efficient Buildings / D3.2-B In-home energy management strategies development (Artemis GA# 100012)
- /28/ eDiana Embedded Systems for Energy Efficient Buildings / D4.3-A Control Strategies Analyses (Artemis GA# 100012)
- /29/ Y.Penya, P.Palensky, M.Lobashov; "Requirements and Prospects for Consumers of Electrical Energy regarding Demand Side Management"; 3. Internationale Energiewirtschaftstagung an der TUWien, IEWT'03, Feb. 2003, Vienna, Austria
- /30/ M.Newborough, P.Augood; "Demand-side management opportunities or the UK domestic sector"; IEE Proceedings Generation, Transmission and Distribution, May 1999.
- /31/ B.Rautenbach, I.E.Lane; "The Multi Objective Controller: A novel Approach to domestic Hot Water Load Control"; IEEE Transactions on Power Systems, Nov. 1996.
- /32/ N.Beute; "Initiatives to Shift the Domestic Load to Improve the Load Factor"; Domestic Use of Electrical Energy Conference, 1999.
- /33/ J.Short, S.Leach; "Using Smart Appliances to Provide Peak-load Management and Increase the Efficiency of the Electricity Network"; International Energy Efficiency in Domestic Appliances & Lighting Conference, Jun. 2006.
- /34/ A.S. Wilken, G.J. Delport; "Using Smart Home Technology to Implement Residential Demand Side Management"; Domestic Use of Electrical Energy Conference, 1999.
- /35/ P.Govender, A.Ramballee; "A Time of Use Controller for Managing Domestic Loads During Peak Demand Periods"; Domestic Use of Electrical Energy Conference, 2005.
- /36/ A.Walker; "Exploring the Limits of Demand Side Management"; MSc Thesis, University of Strathclyde, Glasgow, 2006.
- /37/ C.Tamarit, S.Mahlknecht; "Situation-Dependent Behaviour for Energy Management"; Domestic Use of Energy Conference, 2003.
- /38/ H.Oh, A.Moholkar, S.Douglas, P.Klinkhachorn; "System Reliability and Price Responsiveness of Residential Loads"; West Virginia University, Advanced Power & Electricity Research Center (APERC), Jan 2006.
- /39/ VDI-RICHTLINIE 3789,: see also [http://www.vdi.de/uploads/tx\\_vdirili/pdf/9217456.pdf](http://www.vdi.de/uploads/tx_vdirili/pdf/9217456.pdf)
- /40/ Solartechnik by Prof. Dr. M. Norbert Fisch" 1991
- /41/ Q.met GmbH D-65189 Wiesbaden Internet: <http://www.qmet.de>
- /42/