Chair IT in AEC
School of Engineering
University College Cork
Ireland
Supervisor: Professor Karsten Menzel

Analysis of Building Performance Data

Author:

Stephan Carlo Hoerster

M.Sc., Dipl.-Ing. (FH)

A thesis submitted to the National University of Ireland

In Candidature for the Degree of Doctor of Philosophy

February 2018

This page is intentionally left blank

Abstract

In recent years, the global trend for digitalisation has also reached buildings and facility management. Due to the roll out of smart meters and the retrofitting of buildings with meters and sensors, the amount of data available for a single building has increased significantly. In addition to data sets collected by measurement devices, Building Information Modelling has recently seen a strong incline. By maintaining a building model through the whole building life-cycle, the model becomes rich of information describing all major aspects of a building.

This work aims to combine these data sources to gain further valuable information from data analysis. Better knowledge of the building's behaviour due to high quality data available leads to more efficient building operations. Eventually, this may result in a reduction of energy use and therefore less operational costs.

In this thesis a concept for holistic data acquisition from smart meters and a methodology for the integration of further meters in the measurement concept are introduced and validated. Secondly, this thesis presents a novel algorithm designed for cleansing and interpolation of faulty data. Descriptive data is extracted from an open meta data model for buildings which is utilised to further enrich the metered data.

Additionally, this thesis presents a methodology for how to design and manage all information in a unified Data Warehouse schema. This Data Warehouse, which has been developed, maintains compatibility with an open meta data model by adopting the model's specification into its data schema. It features the application of building specific Key Performance Indicators (KPI) to measure building performance. In addition a clustering algorithm, based on machine learning technology, is developed to identify behavioural patterns of buildings and their frequency of occurrence.

All methodologies introduced in this work are evaluated through installations and data from three pilot buildings. The pilot buildings were selected to be of diverse types to prove the generic applicability of the above concepts.

The outcome of this work successfully demonstrates that the combination of data sources available for buildings enable advanced data analysis. This largely increases the understanding of buildings and their behavioural patterns. A more efficient building operation and a reduction of energy usage can be achieved with this knowledge.

Acknowledgements

- This work was funded by the Higher Education Authority of Ireland through their PRTLI-5 program. Additional funding was provided by Bilfinger HSG FM.

- I would like to thank my supervisor Karsten Menzel for his guidance throughout this PhD and also his family for their hospitality in the last stages.

- Thanks to all current and former members of the IRUSE team as well as my colleagues from Insight for countless constructive discussions.

Declaration

Herewith I declare that this thesis is my own work and has not been submitted for another degree, either at University College Cork or elsewhere.

Stephan Carlo Hoerster

Table of Contents

List of Figures

List of Tables

List of Listings

1 Introduction

The global drive for digitalisation affects buildings and their operation. Smart meters which monitor building energy consumption are becoming increasingly popular. Older buildings see retrofits while new buildings often get smart meters installed during their construction. In addition, Building Information Modelling (BIM) is eventually becoming a global requirement for the full building life cycle with individual adoption levels per country (McGraw Hill Construction, 2014).

Through digitalisation, building data suitable for analysis becomes available in large scales. The proper combination and examination of these data sets reveals new aspects on building operations and the disclosure of hidden energy saving potentials.

The motivation for this work is to develop novel methodologies supporting data acquisition, retrieval, storage and analysis. It focuses on the combination of data from different sources to enhance its significance. It aims to provide efficient analysis methods and added value to the Facility Management (FM) domain to optimise building operations while minimising efforts required for data preparation and processing.

1.1 Background and Motivation

1.1.1 Legal context

The common target for all member states of the European Union is to reduce the global energy demand. The Kyoto Protocol has been adapted by the EU with the so called 20/20/20 targets for the year 2020 (European Commission, 2013):

- Reductions of 20 % green-house gas emissions relative to emissions in 1990
- 20 % of EU’s energy consumption to be provided by renewable energy sources
- Reductions of 20 % of the total EUs energy consumption relative to emissions in 1990

In addition, energy conscious member states might have defined further goals. For example, Germany published in 2010 their energy transition concept which aims to fulfil additional requirements (Bundesregierung, 2010). Their aims are compared to the statistics in year 1990.

- Reduction of greenhouse gas by at least 80% by 2050
- Renewable energy should account for at least 60% of the total energy consumption
- Increasing the electricity efficiency by at least 50%

Ireland on the other hand has recently published their efforts to increase the share of renewable energy (Irish Government, 2012). Their strategy enumerates 36 actions which include e.g. the installation of wind farms and the roll out of smart energy networks. These strategies aim to maximise the potential gained from renewables.

The International Energy Association (IEA) has identified buildings and people's activities within buildings as being responsible for about 38% of the total energy consumption (International Energy Association, 2008). Therefore it seems appropriate to increase efforts for energy reduction specifically in this area. To achieve this, the European Parliament adopted directive 2002/91/EC (European Union, 2002). This directive is called Energy Performance of Buildings Directive (EPBD) with the main focus on the buildings total energy efficiency. This directive was renewed in 2010 under 2010/31/EC (European Union, 2010). One outcome of this directive is the Energy Performance Certificate which classifies a building from A (very good energy efficiency) to G (very bad energy efficiency).

1.1.2 Buildings

The EPBD enforces awareness to understand energy usage within a building. However, in reality it is often difficult to allocate consumption to systems, areas or users. Wang et al. discussed the available options to quantify energy consumption. Because there is often no sub-metering available, costs are allocated through a calculated model (Wang et al., 2014). However, this does not necessary reflect the actual distribution of consumption.

A building can be described as a complex, integrated system. A full understanding of the building and its operation contains saving potentials. Unfortunately, buildings are often considered as "black boxes" without any sub monitoring functionality (Trianni et al., 2013). In the worst case, energy consumption is only determined through the monthly utilities bill (Wang et al., 2014). Through frequent visual reading of meters, this condition can be marginally counteracted. However, due to low data occurrences (e.g. from the monthly bill) no detailed analysis is possible. Buildings which do offer some monitoring functionality are usually equipped with a Building Management System (BMS). However, their main focus is vested in managing and controlling the major building services and systems. Smart meters are rolled out to change this deficit (European Commission, 2012). These meters allow automated reading of meters at a high interval. This creates high potentials for improved behaviour analysis (Alahakoon, Yu, 2016)

In recent years, buildings were often retrofitted with renewable energy systems (Ma et al., 2012). There is a huge commercial market promoting various systems. These are e.g. district heating systems, solar roof panels or the installation of combined heat and power plants (CHP). The installation of renewable energy systems is popular (Hagemann, 1996) (Zhang et al., 2014), as it is also often supported through subsidies from public bodies (Maroušek, 2015). Their commitment is justified through the EU targets that each country should fulfil. Customers' interest in renewable systems is often attributed to long-term economic gains.

The desire to have transparency in building energy consumption has dramatically increased. Besides the legislative requirements, reasons for transparency vary. Examples are:

- Increased energy cost
- Desire to decrease energy consumption
- Achieve certifications, e.g. ISO 50001
- Better understanding of complex building processes
- Better control of interconnected building systems

This PhD thesis aims to provide methodologies which work towards these requirements.

1.1.3 Energy Monitoring Systems

The monitoring of energy-related data can be achieved through sensed or metered data. After collecting from these data sources, an analysis can be conducted. Sensed data is retrieved from (wireless) sensors, while metered data is acquired from energy meters. Energy Monitoring System (EnMS) can be specialised to work with sensed data, with metered data, or with a combination of both (Zach, Mahdavi, 2010).

An EnMS is a fundamental component required for successful energy management. Through its application, evaluation of data becomes meaningful and credible. Weaknesses and inefficiencies can be revealed by critically analysing load curves and the performance of technical systems. The corrections of these weaknesses typically result in decreased energy consumption. Ultimately, the outcome is a reduction of operational cost (AMEV, 2001). Besides monetary reasons, other benefits should not be neglected, e.g.

- Ensuring quality and availability of technical systems
- Sustainable saving of energy
- More adequate maintenance of systems
- Saving time through automated energy monitoring
- Increased energy awareness of tenants

These benefits are not immediately obvious to customers. This is because the acquisition of such a system does not automatically result in energy savings. Instead, a customer faces installation costs for the monitoring equipment required by the EnMS. The utilisation of EnMS alone does not save any energy. Instead, the data gathered by an EnMS must be analysed. An EnMS may save energy once a data analysis has been conducted and measures to optimise building operation have been identified and realised.

A driver for the implementation of an EnMS is the international standard ISO 50001 Energy Management. One of its aims is to provide support in establishing, implementing and maintaining an EnMS in order to sustainably achieve an improvement in energy performance. The standard is outlined in a generic way, enabling all industries and organizations to benefit from it. Within the standard, specifications and requirements to an EnMS are defined. The following list is a summary of the key demands to an EnMS from the ISO's perspective (ISO 50001, 2011):

- Evaluate past and present energy consumption.
- Identify significant energy users.
- Determine energy performance of systems.
- Estimate future energy consumption.
- Definition of performance indicators tailored to the organization.

These requirements make the ISO 50001 standard particularly interesting as they set the framework for this thesis.

Fulfilment of the ISO 50001 can be certified by verified auditors. In Germany, the ISO standard is also associated to the peak compensation scheme, enabling companies which fulfil its requirements to request an energy and electricity tax relief (Currie, 2012).

A recent study (Erek et al., 2013) did a comparison of 13 commercially available EnMS. The authors came to the conclusion that none of the discussed systems fulfils their requirements to act as a KPI-based management interface for energy efficiency in technical systems. While their analysis did not focus on an EnMS system to be used by the FM industry, it concludes that other industry sectors are also not satisfied with current EnMS functionalities.

Nowadays, some utility providers offer their customers the consumption data they record for their billing purposes. This data is usually recorded from the main meter as this is the handover point for the delivered energy. Likewise, some manufacturers of renewable energy sources (e.g. solar panels) offer their customers interfaces to determine how much energy was produced (SolarEgde, EnPhase). Both concepts do not provide a holistic building view and are therefore not suitable as EnMS for comprehensive analysis (Wei, Li, 2011).

For FM operations, availability of an EnMS is a huge asset as it enables the analysis of consumption data and technical systems (FM Interview, appendix 1).

1.1.4 Facility Management perspective

For Facility Management (FM) providers, sustainable operation of buildings is an essential task to fulfil contractual requirements. The total life cycle of a building plays a central role. In this case, the operational cost represents one major component. Hence, the reduction of energy cost is an ongoing target for any FM provider.

By having an EnMS in their portfolio, FM companies see a chance to either expand business relationships with existing customers, or to acquire further customers (see FM interview, appendix 1). An EnMS contains multiple business opportunities, e.g.:

- The FM provider could be contracted as liaison for an ISO 50001 certification
- Through identification of saving potentials, additional revenue can be generated (e.g. in the form of new building systems being installed)
- Detailed system monitoring allows the identification of load peaks, unusual consumption pattern or the misconfiguration of building systems
- Higher customer satisfaction through increased service quality
- To provide qualified statements and recommendations to the customer.

In addition to these business opportunities, it is desirable to operate buildings and systems in the most economical way without affecting daily operations. This can be achieved e.g. through optimised system operations or by replacing energy wasting systems with energy efficient systems.

Besides energy aspects, Facility Managers are often approached with a requirement for comfort monitoring. Comfort is mostly used in relation to temperature, humidity and luminance. Some FM contracts require that rooms (e.g. office spaces) are kept within a certain comfort temperature threshold (GEFMA, 2014). Deviation from this defined temperature range often results in penalties to be paid by the FM provider. When proactively monitoring and controlling thermal comfort, FM providers act in their own interest. Thus, an EnMS may avoid sanctions and provide a more efficient building operation. Ultimately, FM companies may utilise an EnMS as a key component towards sustainable energy reduction.

1.2 Hypothesis

The digitalisation of the building domain significantly increases the amount of available data. However, from a holistic point of view this data is unstructured as interfaces rarely exist. Combining the independent data sources enables profound data analysis. This leads to a better understanding of the building and its technical systems. The result is an increase in energy transparency and improved sustainable building operations.

However, standardised processes are required for the installation of monitoring devices for data integration.

1.3 Aim and Objective

The aim of this research is to increase efficiency and accuracy in data acquisition, retrieval and analysis.

The objectives of this work are:

- Develop a monitoring concept for a unified approach to gather performance data in a building. This concept should be a standardised guideline to conduct installations of monitoring equipment in buildings.
- Provide a data cleansing and interpolation algorithm to process faulty and missing meter readings.

- Develop methods for extraction of selected information from BIM models to enrich meter data with descriptive information.
- Development of a Data Warehouse architecture compatible with the open BIM format IFC as single data repository for efficient data analysis.
- Generate added value to analysis through the combination of metered and modelled data.
- Perform building data analysis through Key Performance Indicators.
- Load curve analysis of metered consumption data through mathematical clustering.

These objectives are derived from the ISO 50001 requirements as outlined in section 1.1.3. The first four objectives aim to structure data acquisition and retrieval while the last three objectives focus on the analysis of data.

1.4 Scientific Contribution

The main scientific contributions made by this work are outlined below.

- The development of a metering concept allows for the categorisation of meter types and installation scenarios. Solutions are provided to retrofit highly diverse buildings with standardised approaches.
- The automated cleansing of faulty consumption readings can greatly improve the data quality while no additional manual efforts need to be undertaken. Potentially all EnMS could benefit from this method as it increases data readability and usability.
- Currently, descriptive data needs to be acquired and put in place manually. Instead, the extraction process can be highly automated which results in better quality while very little time and effort needs to be put into data extraction. Through extraction of descriptive data from open Building Information Modelling (BIM), monitored data can be enriched significantly.
- The introduction of a Data Warehouse (DWH) that is compatible with Industry Foundation Classes (IFC) brings the IFC definition into the DWH domain. Processing of IFC objects through SQL queries enables the analysis of BIM data with database functionality.

- A set of aggregation methods was developed to provide KPI for the holistic evaluation of building performance.
- The application of machine learning algorithms to cluster metered data creates a different perception of monitored load curve data. This potentially highlights buildings' different types of usage patterns which cannot be easily seen when analysing raw data sets.

1.5 Thesis Outline

The structure of the further chapters in this thesis is as follows:

Chapter 2 presents the results of a state of the art analysis. It discusses various ways for the acquisition of time series data. Furthermore, it discusses BIM and IFC as source for descriptive data. Data Warehousing and database principles are elucidated. The chapter discusses data interpolation and cleansing techniques before it concludes with a debate of mathematical clustering algorithms.

Chapter 3 discusses the methodologies presented in this work. Namely, these are (i) defining a monitoring concept for the data acquisition process, (ii) the cleansing of faulty or missing data, (iii) the acquisition of descriptive data from BIM models, (iv) definition of an IFC-compatible database schema and development of a DWH, (v) data analysis through application of holistic KPI and (vi) data analysis of load curve data through mathematical clustering and machine learning algorithms.

Chapter 4 outlines the installation of monitoring equipment in three pilot buildings. These installations should verify the metering concept introduced in chapter 3. The pilot buildings were carefully selected to be of different kinds. Their operational usage behaviour aims to underpin that this concept may be further applied to different types of buildings. Finally, it outlines how faulty collected data gets cleansed with the algorithm introduced in chapter 3.

Chapter 5 presents analysis obtained by applying the methods introduced in chapter 3. It starts by providing an overview of the load curve data analysis. It identifies that the methodologies introduced in this thesis reveal building behaviour patterns not obtainable by standard load curve analysis. This is done by clustering the load curve data which provides a new perspective on the data sets. Additionally, the application of holistic building KPI is executed for various building domains.

Chapter 6 concludes this thesis by critically discussing the conducted work. It claims that the functionality of EnMS can be positively enriched by incorporating some/all of the proposed methodologies. It highlights limitations which may affect the quality of the methodologies presented in this work. It will also point out that the combination of methodologies contains additional caveats which need to be considered. An outlook for potential future work is presented before a final summary is given.

2 State Of The Art Analysis

This chapter discusses the current state of the art with regards to the aims and objectives of this thesis. It is divided into 5 parts:

- Benchmarking buildings and processes inside buildings is usually done with Key Performance Indicators. This section discusses KPI often used in Facility Management and Energy Performance Indicators (EnPI) required by EnMS
- It is discussed which time series data is available from buildings and how this data may be acquired. Current possibilities are introduced and their advantages and disadvantages are discussed.
- Building Information Modelling is discussed as potential source for the extraction of selected descriptive data. This type of data enriches the context of the acquired time series data.
- It is discussed how building data may be managed holistically using Data Warehouse technology. Here, database principles and limitations are elaborated.
- Existing algorithms to cleanse and interpolate faulty data are discussed. The application of mathematical clustering algorithms for the analysis of load curve data is evaluated.

2.1 Benchmarking

Performance Indicators are information derived from collected data. Key Performance Indicators (KPI) are measures for key business objectives and success. These KPI’s usually reflect the managerial performance (Marr, 2012). Their focus lies on the aspects which are most critical for current and future success (Parmenter, 2007). KPI’s are also key components in Facility Management operations. They are used as an instrument to evaluate the performance of processes and components in buildings against systems, departments, employees. In the case of a Service Level Agreement (SLA) between the Facility Manager and the landlord is in place, these KPI’s are defined in a contract. KPI’s are used to measure success or failure in building operations. Failure to achieve set targets is often related to penalty payments. KPI’s could either be part of strategic goals which need to be met, or could be defined on their own, or could also be part of a recurring goal. In Industry, KPI’s are currently measured on a monthly interval (Morré, 2014). One main requirement for the definition of a KPI is the standardisation of data to ensure that comparison of KPI’s is possible e.g. at different time intervals. KPI’s can be classified in various types, for example absolute and relative KPI. Absolute KPI’s are figures like e.g. total number of employees or total energy consumption. A relative KPI is derived by referencing an absolute KPI with reference values, e.g. employees per building or energy consumption per m².

Common KPI’s are listed in Table 1

Table 1 List of common KPI in FM

KPI	*Example/explanation*
Client satisfaction	e.g. derived from thermal comfort
Number of tickets	e.g. tickets opened via help desk
Number of incidents	e.g. technical failures
Response & resolution time	e.g. after issuing an incident
Loss of floor area due to failure	e.g. after power outage
EnPI	e.g. consumed kWh per square meter, energy consumption per hospital bed, water consumption per visitor, prisoners per jail block

An EnMS which is used to achieve energy targets should provide an Energy Performance Indicator (EnPI, see Table 1). Ireland’s Sustainable Energy Authority states that the approach of using EnPI compared to traditional methods has multiple advantages (SEAI, 2015), e.g. only limited data needed for comparison, focus on organization level performance, easy to understand.

2.2 Compiling Time-Series Data

Buildings, especially larger ones, often are managed by a Building Management System (BMS). A BMS is a computer-based building control system (Knibbe, 1996). Connected through a bus system, the BMS can operate building services and systems. The actual functionality of a BMS greatly depends on its capabilities to actuate and the age of the building equipment. The BMS often does not only control the building, they often also keep a short history about systems' status. Additionally, they often have many sensors, e.g. temperature or CO_2 connected to it. In some instances, the BMS is also in control of meter data. This makes the BMS a desired system for data harvesting.

Figure 2-1 outlines a schematic for a standard BMS. Its functionality is versatile and powerful. The block on the left lists modules usually found in BMS. The automation in control module changes e.g. system behaviours may change once set points of thresholds have been met. It could also operate based on an integrated scheduler, e.g. a special routine for bank holidays where systems are run at a lower capacity. The monitoring module tracks system states and logs any operational changes. The fault detection gets activated whenever an undesired system state has been identified. Building access controls may also be maintained through a BMS. Alarms get generated when pre-defined conditions or thresholds have been breached. Modern BMS usually control the building's technical systems to coordinate building operations. Lastly, BMS often also control the lighting in buildings. All the information handled by the BMS can be visualised and/or controlled by a BMS operator (Knibbe, 1996) (Desjardins, 1983).

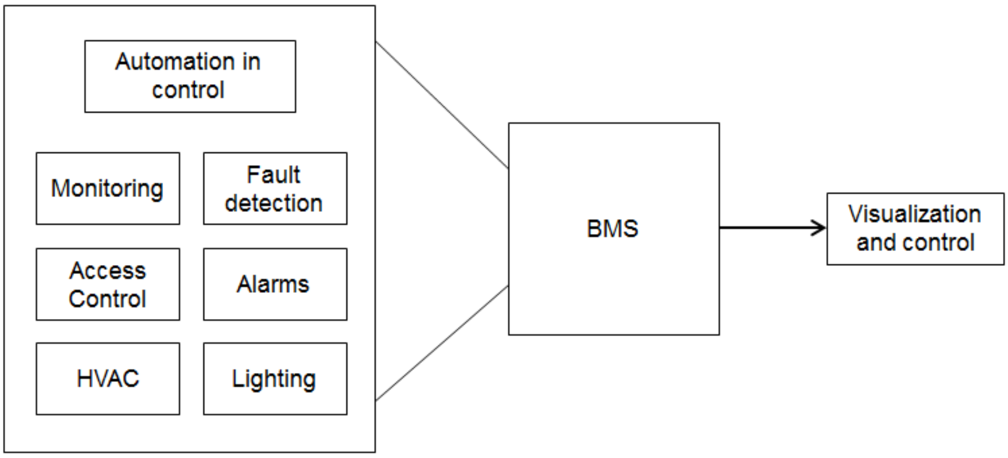


Figure 2-1 Functional schematic of a BMS

Third-party systems can be connected to the BMS and exchange data with it. BMS often communicate through vendor specific, proprietary protocols. More recent BMS often either support open protocols natively, or they offer an open gateway interface to their proprietary bus (Kastner et al., 2005).

One open standard which has emerged in recent years is BACnet. BACnet is a protocol which is utilised in buildings for communication and control between BMS, technical systems and meters (Bushby, 1997). Since 2004, it has become a world-wide standard for buildings (ISO 16484-5). Due to its open documentation, more vendors are implementing the BACnet protocol in their products. BACnet can be described by its three major layers: the application layer, the network layer and the physical layer. The application layer is usually a front-end or graphical interface where information is interpreted by users and from where certain building controls can be actuated. The network layer enables communication between the application and physical layer. The physical layer is compatible to a variety of existing standards and protocols. The layers of BACnet are shown in Figure 2-2. For example, the permission to communicate to a device via RS 485 is triggered in the BACnet application layer.

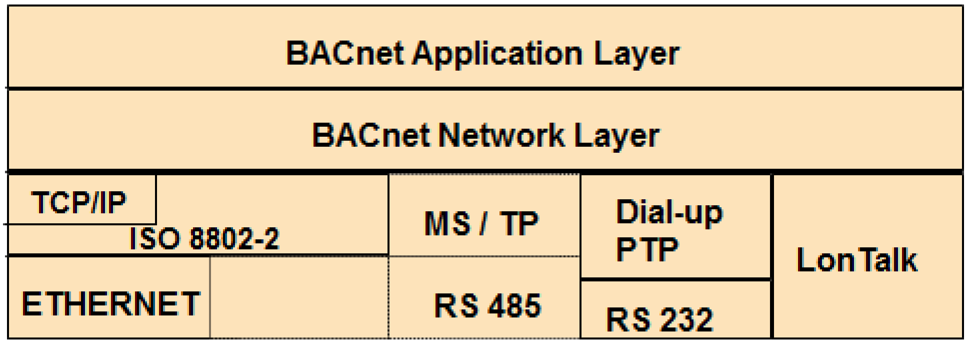


Figure 2-2 Layers in BACnet (BACnet Interest Group, 2005)

BACnet, however, is only available in newer installations whereas many older BMS installations either do not support BACnet directly or do not offer a gateway to BACnet. The same applies for other open standards like LonWorks or OPC. This deficit should not be neglected when searching for a generic solution for an EnMS.

2.2.1 Meters and field layer

Traditionally, main meters in buildings are read manually in defined intervals. These readings are followed up by an invoice of the utility provider. While this procedure is sufficient to justify claims, the granularity of the data does not allow a closer inspection of the building's behaviour.

By utilising smart meters, readings are taken automatically. This results in a much higher reading interval. The objective of automated systems is usually to be in accordance with the 15 minute interval used e.g.by German distribution network operators (Bundesministerium, 2005). Capturing data in a 15 minute interval is a reasonable trade-off between accuracy and high volume of data (Balachandran 2014).

Due to the high number of meter vendors and communication protocols on the market a generic way to acquire meter readings is not existent. Various communication protocols exist to transfer meter data. Meters can be classified into three categories:

- Digital meters
- Analogue meters
- Basic meters (without any communication interface)

Digital meters utilise communication protocols. Analogue meters provide an analogue pulse output which sends a pulse whenever a certain consumption threshold has been metered. Lastly, basic meters do not provide any interface at all. These meters cannot be used for the outlined approach unless they get upgraded. An exemption for this might be the release of meter scanners. These scanners are small objects which get mounted on top of the analogue meter. The scanners contain an optical camera. This camera takes frequent pictures of the meter. Through OCR scanning technology, the picture is then converted to a meter reading. While already suggested by current research as an accurate replacement for manual meter readings (Dayama et al., 2014), its reliability and accuracy of OCR meter reading still needs to be evaluated and is not part of this research.

In terms of meters and their data transmission one can distinguish between push and pull modes. A meter pushes information when the acquired meter reading is sent at specific requests. A meter follows the pull integration when it is queried for meter readings from an external source. A combination of both push and pull is possible, e.g. where information is pulled in a set interval but also pushed by the meter when an alarm is raised or set threshold has been reached.

The main advantage of digital meters is that other devices can communicate to the meter through a common protocol, e.g. M-Bus (see section 2.2.2) or Modbus (Swales, 1999). Meters could be queried e.g. for their current meter readings or timestamps of previous readings. If the data stream between external source and meter would be interrupted, the next received value would still be in accordance with the physical meter. While missing readings can only be interpolated, the consumption can still be calculated by subtracting the old reading from the new one. This is a huge benefit compared to analogue meters which only output pulses. As there is no relation between a pulse and a meter reading, one cannot determine how many pulses were sent in an outage period. A manual reading of the meter would be required to determine how many pulses were missed. If multiple outages happen, it becomes impossible to distribute the additional consumption shown on the meter to their corresponding periods.

The main disadvantage of digital meters is the complexity that comes with their communication protocol. The protocol stack creates information overhead on the bus. While the bus can surely handle the overhead, it makes debugging more difficult as one need to understand the protocol in order to identify problems (Ericsson, 2017). Additionally, any device added to the bus might interrupt the existing communications due to misconfiguration or incompatibilities. This could raise alarms or result in packet loss during communication.

It is a requirement from the bus system that each connected device has a unique address (M-Bus User Group, 1998) (Swales, 1999). Otherwise, transmitted information would be useless as it cannot be related to a source. The address also allows the devices to communicate amongst each other and with any further device connected to the bus. Any device added to the bus has not to be assigned duplicated addresses. If senders and receivers would not be unique, malfunctions and misbehaviour can be expected.

On the contrary, analogue meters are not able to digitally exchange information. They only send pulses whenever the meter reached a predefined threshold. For electricity meters e.g., this could be defined as one pulse gets generated for every used kWh. The actual threshold can be determined either by (a) labelling on the meter, (b) specification in the data sheet, or (c) by monitoring the pulse output with a multimeter. Since an analogue meter cannot be queried for readings, it is essential to receive every pulse that is being sent. They do not send any additional information. If no consumption takes place, no pulse is sent by the meter. The sending interval of the meter can therefore be defined as completely random. Every analogue meter has a transformer ratio which states the magnitude of the pulse. An incorrect setup of the transformer ratio in the monitoring equipment will still produce results but they will be incorrect. Table 2 summarises the meter attributes.

Table 2 Different meter categories

Property	**Basic Meter**	**Analogue Meter**	**Digital Meter**
Able to communicate	No	Yes*	Yes, communication via bus
Meter reading can be queried	No	No	Yes, they can be queried via bus any time
Protocol based	No	No	Yes, byte-encoded data can be sent & received
Easy to configure	No**	Yes, it is always a pulse that needs to be interpreted	No, parameterisation requires knowledge about the underlying bus system
Data loss can be compensated	No	No	Yes, it can be queried again at a later time
Flexible pulling intervals	No	No	Yes, they can be pulled for information any time

* Can only send information

** No configuration possible

Digital signals should be always preferred over analogue signals. This is because analogue meters only send pulses; i.e. there is no way to recover the information in case of lost signals. A connected system counting the pulses would reflect a different value than the meter, resulting in inconsistencies. On the contrary, digital meters transfer the actual meter reading and can, in case of issues, be queried later again to retrieve the actual meter reading. Digital meters are characterized by an improved interference resistance. Additionally, they allow the communication between devices and they can communicate meaningful information (e.g. meter reading, consumption). In comparison, a pulse cannot be interpreted without further information.

2.2.2 M-Bus

In buildings, gas or water meters are often found in areas where no electricity socket is nearby. The installation of a regular digital meter would require an electricity socket placed at the disposal. This increases installation cost and time.

The M-Bus is a digital field bus which is of particular interest for energy monitoring. This is because M-Bus meters are powered by the bus itself, rendering an external power supply obsolete (M-Bus User Group, 1998). Furthermore, the application of the M-Bus protocol can be freely used without royalty fees.

2.2.3 Data sources

This section discusses the acquisition of fact data within a building. The different ways to acquire sensor and meter data can be grouped into three major categories (AMEV, 2010) such as manual acquisition, compiled from BMS and compiled through data loggers.

2.2.4 Manual Acquisition

To satisfy the requirements outlined by ISO 50001, period manual reading of meters is sufficient. For this, an on-site technician audits every meter reading. Another very basic possibility is manual processing of utility bills.

Microsoft Excel seems to be the tool most widely used to accommodate manual meter readings (see FM interview, appendix 1). These manually read values could potentially be used to populate a monitoring system. In this case, the information needs to be fed manually into the database. Various reasons suggest avoiding manual data acquisition. These are given in Table 3. The sole advantage of having manual readings is that it does not generate any installation, labour and maintenance cost.

Table 3 Reasons to avoid manual data acquisition

Reasoning	***Explanation***
Human error	This acquisition method involves the danger that meter readings are incorrectly read or taken.
Low reading interval	Due to the low interval of meter readings, data has a very limited benefit for analysis purposes
Read at different times	The irregular meter reading likely happens at different times, therefore making comparison difficult. Even defined intervals, e.g. every 30 days might not be met if the day occurs on a weekend or bank holiday.
Personnel cost	The manual acquisition method requires operational staff to carry out scheduled inspections.

2.2.5 BMS

Open protocols like BACnet and customised interfaces for proprietary protocols allow third party devices to interact with BMS. Depending on the BMS it is possible to acquire all current set points, historical readings and, in some cases, even write values back to the BMS which enables third parties to take control. In order for these third party devices to interact with the BMS, they either need to be connected to the field bus network, or directly to the BMS via Internet. The engineering effort to incorporate BMS data into an EnMS should not be neglected. Due to BMS complexity it is often necessary to involve its vendor for the acquisition of building data. Since BMS software is generally closed-source with no common Application Programming Interface (API) available, a generic solution which is compatible to any available BMS is unfeasible. Finally, one has to make sure that existing bus systems and the BMS cannot be affected by introducing an additional connection for data harvesting.

2.2.6 Data logger

For the collection of meter readings from analogue and digital meters, a monitoring device called data logger can be used. This device will cyclically query connected meters and save their readings to its internal storage.

A data logger is a programmable device capable of retrieving, storing and forwarding meter readings. It usually has interfaces to allow the communication with multiple field bus systems. Generally, it offers various ways to retrieve the stored meter data over Internet, e.g. using FTP/HTTP connections or Email transfers.

The sequence diagram in Figure 2-3 illustrates the operating principle of a data logger. One can see that the data logger either just retrieves meter readings or queries meters for their readings. In the meter case 1, the sending meters are analogue meters which cannot be queried. They only send their pulses whenever a threshold was reached. The meter case 2 depicts digital meters which are queried by the data logger periodically. Lastly, the acquired data from all meters is unified and stored in the database.

The choice of the right data logger is essential for a standardised procedure. While there are numerous commercially available models, there are also custom built data loggers, which are especially used in research projects (Campus21 D4.1, 2011) (Baker, 2013).

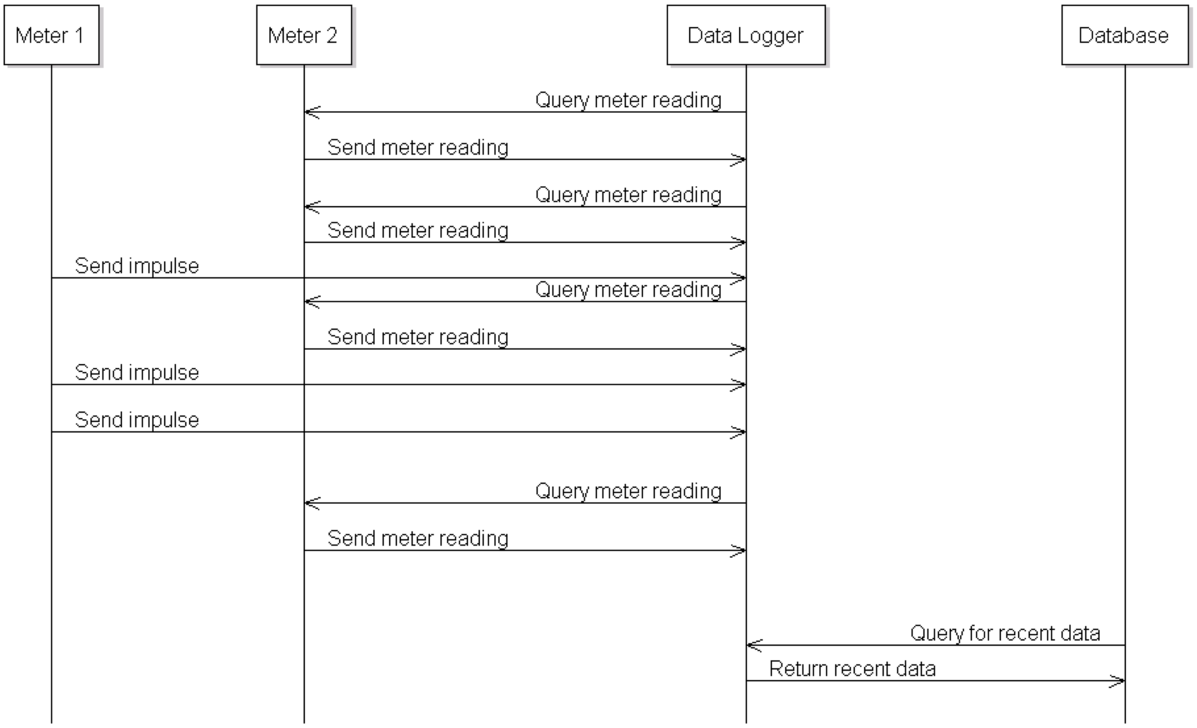


Figure 2-3 Data logger sequence diagram

For an EnMS being deployed in a FM environment, requirements for a data logger can be defined as follows:

- to support digital meters and their communication protocols
- to support analogue meters
- to support open protocols (BACnet)
- to require minimal parameterisation and programming
- to communicate via Internet to transfer meter readings
- to support adjustable data retrieval intervals

A data logger cannot interfere with already existing systems unless installed in the same physical network. This becomes necessary e.g. when information from a BMS should be retrieved. With individual solutions, metered data can be gathered without interaction and interference with existing systems. A data logger is mainly used to collect data from connected sensors and meters. These devices are either already installed or will be installed along with the data logger. This allows the automated reading of connected devices at any desired interval. Additionally, data loggers could be equipped with basic fault detection logic that e.g. sends an email when a device cannot be read any longer. The acquired information is stored in the data logger. From there, it can be automatically transferred into the monitoring systems database through any Internet connection.

2.2.7 Comparison of the three methods

Table 4 provides a comparison between the three acquisition methods. Automated data acquisition should be always preferred as manual readings happen less frequently. Manual readings should be only used to verify that the digital acquisition is working properly.

Table 4 Different data acquisition methods

	Manual	BMS	Data Logger
Reading of meters	Manually	Automatically	Automatically
Reading interval	Flexible / rare	Set by BMS / often	Set by data logger / often
Transfer to EnMS	Manually	Automatically	Automatically

Figure 2-4 summarises the means of data acquisition in a graphical way. Acquisition of similar data sets can be achieved with method 2 and 3 as only the hardware to acquire the readings is changed (BMS vs data logger). Method 1 provides no automatism and will not reach the potentially high sending intervals and accuracy of the other methods.

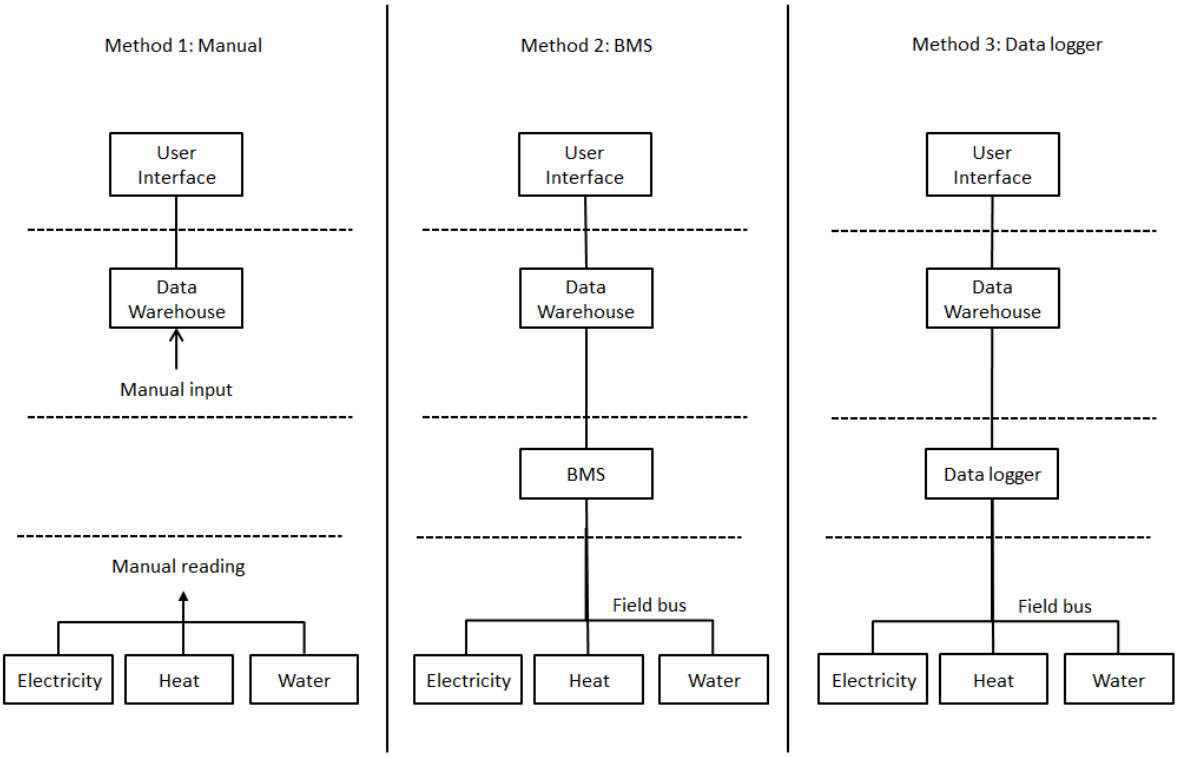


Figure 2-4 Comparison between the data acquisition methods

Table 5 gives a summary about the quality of the meter reading and the main advantages and disadvantages between the three acquisition methods. Manual meter reading appears to be very cost efficient as no hardware installation is required. However, the process of taking the meter readings on recurring basis will always pull resources from the site personnel. Automated solutions come with investment cost but pay off over time as no recurring reading of meters is required.

Table 5 Advantages and disadvantages of acquisition methods

Method	Advantage	Disadvantage
Manual	Existing meters can be used. No hardware installation cost (unless sub-metering of new systems is required).	Very low data granularity. Likely to oversee peaks in consumption or incorrect system behaviour. High chance of human error. Time consuming and thus expensive.
BMS	No manual reading required. Existing infrastructure can be used. A single, consistent solution for a building. Reading of meters at high interval possible. Open standards like BACnet may be used to reuse existing ways of communication	New meter installations need to be incorporated into the BMS network. Interference with BMS network, possibly due to bad set up. Open standards may only be available for recent BMS installations.
Data logger	No manual reading required. Independent from any systems already in place. Reading of meters at high interval possible.	Requires meters and installation of additional hardware. Hardware may need maintenance.

2.2.8 Data transmission

Various ways exist to transfer data acquired from a building to a database system. Table 6 compiles the common possibilities. From a FM perspective, an independent connection (either through a 3G modem or a dedicated internet line) is the preferred option as there cannot be any interference with the existing customer's network.

Table 6 Data transfer alternatives

	Advantage	**Disadvantage**
Existing Internet connection	This is usually considered as the cheapest solution.	Due to IT regulations it is often not possible to jointly use the existing IT infrastructure.
3G Modem	With a mobile connection, data acquisition is done through an independent network which does not interfere with existing IT equipment.	Utilising the mobile broadband involves monthly costs. Moreover, signal coverage is not always given. This is due to the nature of meter installations, which can usually be found in remote areas of buildings
Dedicated Internet line	A dedicated line is independent from existing IT infrastructure.	Similar to 3G Modem, a dedicated Internet line generates monthly cost. The hooking up and enabling of the line requires the assistance of third party telecommunication companies.
Manual transmission	*none*	This option required personnel on site and continuous human interaction. Meter readings need to be collected in tables or lists. It should be avoided whenever possible.

2.3 Compiling Descriptive Data

This section discusses Building Information Modelling (BIM) as it will be utilised as a source of dimensional data for the work in this thesis.

The BIM model of a building aims to integrate product and process data and to increase productivity in building design and construction (Holness, 2008). It is envisaged that the BIM model acts as holistic meta model to manage a building through its entire life cycle (Lee et al., 2005). BIM features the many individual aspects, such as handling of building geometry in 3D models, relationships between objects, component properties and material composition and cost information. The BIM model stores and keeps track of all changes throughout the building's life cycle. It covers design, construction, managing of performance property sets and maintenance processes of a building. Each model object must have a Global Unique Identifier (GUID). This GUID never changes during lifetime. This allows the tracking and versioning of individual objects within the model.

Information stored in the model will be available on demand to all building stakeholders. It is fundamental in the BIM concept that all information can be shared among contributing parties (Menzel, 2014).

It can be safely anticipated that demand and utilisation of BIM will greatly increase in the near future. Public sector projects in the UK for example require that BIM will be utilised from April 2016 onwards (BIM Task Group, 2015).

In the context of this thesis, dimensional data may cover e.g. building storeys and their rooms, meters and sensors, their spatial relationship, building systems and components.

2.3.1 Industry Foundation Classes

In recent years, Industry Foundation Classes (IFC) has been established as one potential neutral and open meta data model for BIM (Steel et al., 2012). Eastman claims that the IFC data model is becoming the industry standard for data exchange and integration in the building domain (Eastman, 2011).

Proprietary file formats of software vendors are forcing many companies and individuals to invest into software for the single reason to work with the building model. By using open standards like IFC, information held in BIM models can be exchanged between applications developed by different software vendors.

IFC has been developed since 1994. With each new version it supports more building elements, products, processes and resources. Since then, more than 150 software applications worldwide support the standard. The standard has been extended multiple times with more capabilities. The last release of IFC version 2x4 was released in March 2013. IFC is maintained by buildingSMART and standardised in ISO 16739 (buildingSMART, 2014).

As of 2016, IFC consists out of 776 entities (IFC4 Addendum 2, 2016). Each type of object may contain information about entities found in the building domain. These could be for example:

- Building elements, e.g. a pipe or a wall
- Spaces
- Persons or organizations
- Tasks or procedures
- Properties for the above, e.g. performance data or materials

The objects are structured in a hierarchical tree where child objects inherit attributes from all parent objects. Figure 2-5 gives an inheritance example for the IFC object IfcSpace.

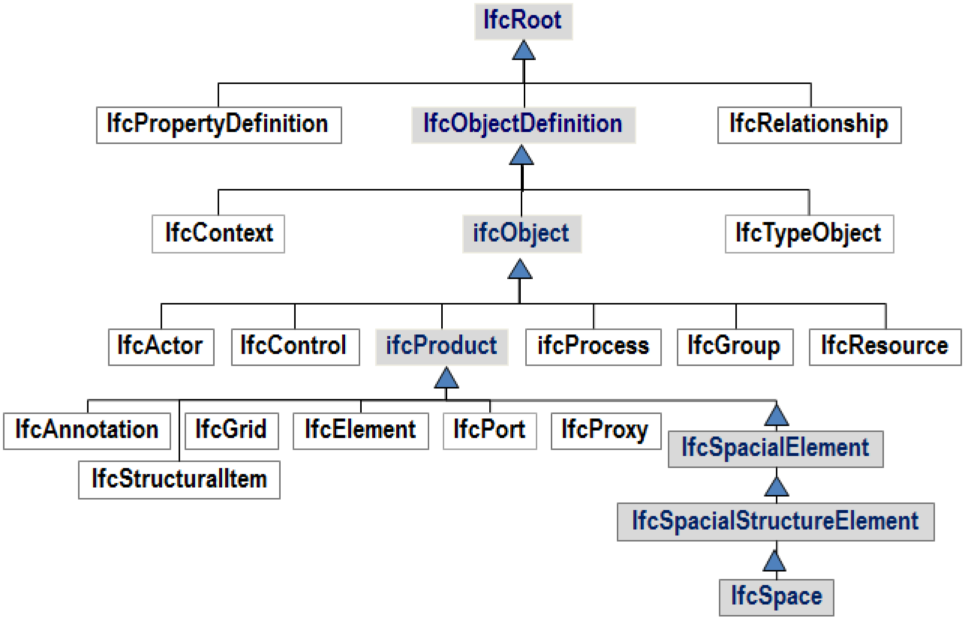


Figure 2-5 IFC inheritances of objects (Menzel, 2014)

IfcRoot is the parent element of all model elements participating in the inheritance structure. The main responsibility of IfcRoot is the allocation of GUID’s. In addition to the GUID, it also provides a name and a description for any subtype object. Every entity that inherits from IfcRoot is classified as independent entity, whereas entities which are not sub-types of

IfcRoot cannot be independent. These dependent objects (also called resource objects) can only exist when they are referenced by an object which has a GUID (and therefore are a subtype from IfcRoot).

2.3.2 Modelling approach

References between entities are realised in IFC through relationship objects. Through IfcRelationship (and its specialised subclasses), it is possible to link independent objects to each other. For example, a boiler could be linked to a heat pipe element. IFC can depict either 1-to-1 or 1-to-many relationships. IfcRelationship also inherits from IfcRoot; therefore any relationship also possesses a GUID. Figure 2-6 illustrates how IFC realises the relationship be-tween buildings, sites and the project.

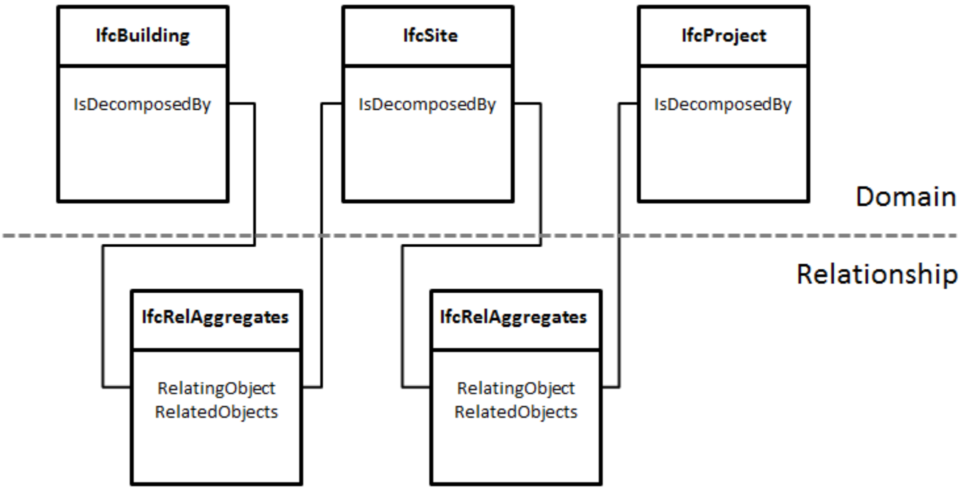


Figure 2-6 Relationship between three IFC classes

IFC property sets are properties which can be attached to any IFC object by a relationship object. They provide a meta-model for custom information. This information can be linked into the IFC without changing the model. These can e.g. be utilised to store acquired fact data. In Figure 2-7, it is illustrated how monitored sensor data is linked in IFC to a sensor. In this model, all sensor history can be stored in the PHistory property of the IfcPropertySet definition.

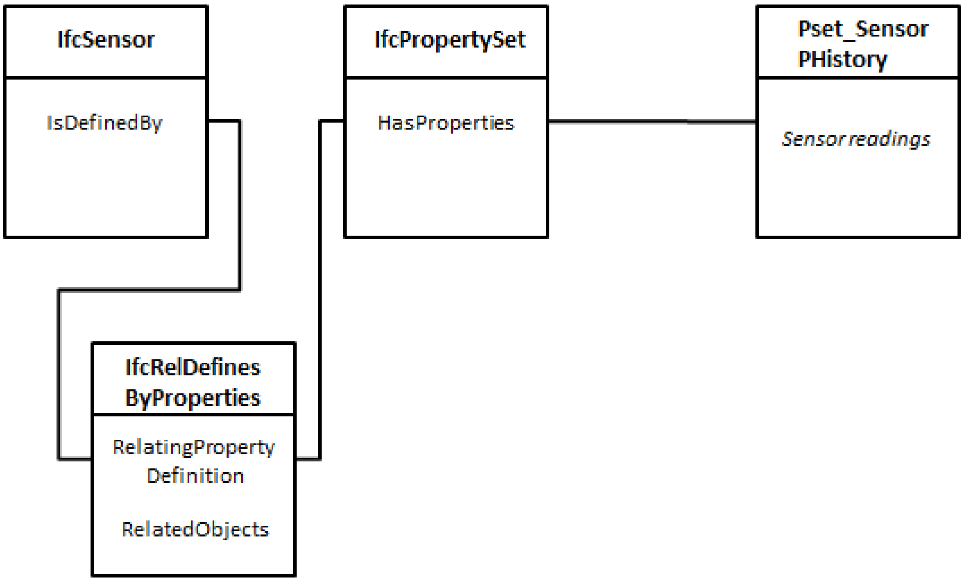


Figure 2-7 Linking sensor data to spatial information

2.3.3 IFC file formats

The IFC specification uses amongst others the EXPRESS modelling language of ISO 10303-11 to describe its metadata model (ISO 10301-11, 2004).

In software engineering, an Entity-relationship model (ER model) is an abstract way to describe data or information. Diagrams created to visualise these entities and relationships are called Entity-relationship diagrams. There are three main elements in ER models:

- Entity objects: Where information is stored
- Attributes: Where information from entities is stored
- Relationships: The means to access information from entities

An example for an EXPRESS entity model (in this case: IfcSpace) is given in the following code:

Listing 1 IfcSpace EXPRESS definition

```
ENTITY IfcSpace
 SUBTYPE OF (IfcSpatialStructureElement);
      InteriorOrExteriorSpace :IfcInternalOrExternalEnum;
      ElevationWithFlooring : OPTIONAL IfcLengthMeasure;
 INVERSE
      BoundedBy : SET [0:?] OF IfcRelSpaceBoundary FOR RelatingSpace;
END_ENTITY;
```

The code defines the IFC object IfcSpace. It is revealed that IfcSpace is a subtype of an entity called IfcSpatialStructureElement. Furthermore, IfcSpace has two attributes called

"InteriorOrExteriorSpace" and "ElevationWithFlooring". Lastly, it consists of an inverse element called "BoundedBy". This element allows creating a relationship with the IfcSpace entity through the relationship object "IfcRelSpaceBoundary".

As presented in the hierarchical tree in Figure 2-5 one can see that IfcSpace is hierarchically linked to IfcRoot. Its superior object is IfcSpatialStructureElement which again is a subtype of IfcProduct. Continuing this lookup process eventually leads to IfcRoot.

IFC allows the export of metadata models into files. These file types and layout have been standardised. Firstly, there is IFC-XML, a definition based on the widely used XML file type as defined by W3C (W3C, 2008). The application of XML is the IFC domain is also standardised in (ISO 10303-28, 2007). The format is suitable for interoperability with XML enabled tools.

Due to the nature of the XML standard, the XML file comes with an overhead that increases the total file size compared to the other standardised format, IFC-STEP.

IFC-STEP has been standardised in (ISO 10303-21, 2002). In recent years, STEP has received a big boost due to the wider acceptance and application of IFC. This format stores every instance of an IFC object in a single line. One example is given in the following snippet. Here, it is shown how a building storey is linked to a building and to a site through the IfcRelAggregates relationship.

Listing 2 IfcRelAggregates Example in EXPRESS

```
#42287=IFCSITE('1e$PJwy2v56gfAEvUiYYwZ',#33,'Default',$,'',#42286,$,$,
      .ELEMENT.,(51,53,48,813171),(-8,-29,-10,-736618),-0.,$,$);
#42289=IFCRELAGGREGATES('0hAxe3PXj8FvFlzFxJeZBa',#33,$,$,#42287,(#36));
#36=IFCBUILDING('1e$PJwy2v56gfAEvUiYYwW',#33,$,$,$,#25,$,$,.ELEMENT.,$,$
      ,#35);
#42352=IFCRELAGGREGATES('3m1gcSsJT0zOe8S0AjPt5$',#33,$,$,#36,(#40,#44,#4
      8,#56));
#40=IFCBUILDINGSTOREY('1e$PJwy2v56gfAEvTJTT1c',#33,'Level -01 FFL',$,
      $,#39,$,$,.ELEMENT.,5500.);
```

Each IFC object has a defined number of attributes. The STEP format enumerates these attributes chronologically, where unset attributes are represented with a dollar sign ($). IFC STEP and IFC-XML formats can be converted into each other without losing information.

2.4 Managing Building Data Holistically

All data received via communication channels needs to be stored for analysis purposes. A storage system which holds readings up to 5,000 data points and retrieves readings every 15 minutes acquires after 5 years of operation about 864 million data sets. For FM companies or utility providers which operate many buildings, 5,000 data points can be easily reached. Each building is usually equipped with a gas, electricity and water meter while bigger buildings often come with many additional meters for sub-areas (Action Energy, 2003).

Data Warehouse (DWH) systems were originally developed to increase productivity and to consolidate data from different sources. Companies with large sets of data found it increasingly difficult to efficiently analyse their own collection of data. The spread of data across multiple systems, database products and files further increased this obstacle (Davenport, Dyche, 2013).

2.4.1 Data Warehouse

At a first glance, a DWH is very similar to a conventional, transaction-based database. Both systems offer tables for data storage, as well as indexes and keys. But a DWH has additional layers of complexity where data is categorised into fact data and dimensional data for analytic processing. The separation of the analysis workload from the ordinary transactional database enables fast and comprehensive accesses for analytical purposes. This aggregated information is often used for reporting purposes. Data Warehousing is usually required when a huge number of data sets need to be efficiently processed. They are specialised to consolidate data from multiple sources and store it in a single repository (Inmon, 2002) (Widom, 1995).

The different aggregation layers involved in a DWH are usually populated through additional ETL processes (Extract, Transform and Load). In order to consolidate data from different sources, it gets pre-processed for unification. At this stage, errors and inconsistencies need to be eliminated. Extraction deals with the selection of data from various sources. Transformation converts the data into the desired storage format and loading stores the data into the specific data target (Kimball and Ross, 2002).

A DWH structures data in pre-defined materialized views and data cubes. By using categorised information stored in DWH dimensions, information can be aggregated in logical groups. One related example would be the room consumption in a building, in a specific room or from a specific tenant.

Profound tools and methodologies for the analysis of building performance data are essential for high quality results. Huge amounts of gathered fact data result in demand for DWH storage technology. Dimensional data is used to slice and dice the fact data needs to be of high granularity and great accuracy in order to benefit the analysis.

Especially for historical data analysis, the size of the storage and its query speed needs to be efficient. Conventional database management systems (DBMS) emphasise on transactions and provide only limited mechanisms to aggregate and generate reports (Lane, 2013).

Advantages and disadvantages of DBMS and Data Warehouse (DWH) systems have been worked out in a recent PhD thesis (Ahmed, 2011). Its findings were processed and summarised in Table 7.

Table 7 Differences between DBMS and DWH, derived from Ahmed, 2011

Database Management System	Data Warehouse
Easy to understand for business users.	DWH design increases complexity
Design is clear and logical.	
Support only for predefined operations. Usually designed to operate in compliance with business requirements.	Optimised for ad-hoc queries to enable wide variety of operations
Full support for transactional activities.	No support for transactional processes as a DWH is updated on a regular interval.
No history of changed records is kept.	Historical records are kept consistent to increase understanding of business processes
The DBMS is always up to date with business transactions.	
Low query efficiency when joining multiple tables or handling big data	Many years of historical data may be processed
Very limited mechanisms for decision applications such as trends and reports	Wide support for decision support applications such as trends and reports
It reflects business entities and their relationship.	Changes in business requirements are more complex to adapt, e.g. KPI’s need to be modified
Changes in business requirements can be adapted easily	

The key benefits of a DWH are the consistent storage of historical records for business processes, and the compatibility with decision support applications for trends and reports.

2.4.2 Data Warehouse for building performance analysis

For the analysis of building performance data, query speeds and support for data aggregation and reports is a fundamental requirement (Gupta and Mumick, 1995). For these reasons, DWH technology is commonly being used as intelligent storage for metered and sensed building data. By utilising DWH technology, Online Analytical Processing (OLAP) can be deployed to analyse building performance data (Pedersen and Jensen, 2001) (Chaudhuri and Dayal, 1997). Performing an energy performance analysis requires capable tools and reporting (Augenbroe and Park, 2005), which is functionality provided by a DWH.

2.4.3 Dimensional and fact data

In DWH terms, fact data is the bulk historical data usually stored for long term. It stores numerical values, e.g. meter measurements. Additionally, it stores foreign keys which reference to dimensional objects. Each row in a fact table relates to one measurement. Fact data may be spread across multiple tables e.g. fact data of different kinds. A single table can hold many millions of data sets. Fact data is accessed most efficiently by selection of certain criteria from the dimensions. Each dimension is linked to the fact data by a column with a dedicated foreign key (Kimball et al., 2008).

Dimensional data is descriptive data which aids users in structuring and analysing of fact data. Dimensions are defined in hierarchical levels. These levels allow the processing of fact data at different degrees of granularity. This enables analysis at any desired level of abstraction. As an example, energy consumption could be analysed at a yearly, weekly or hourly time interval. Each dimensional element may have thousands of corresponding records in the fact table due to its aggregation capabilities (Rob and Coronel, 2009). A DWH enables the modelling and representation of information in multiple dimensions.

Dimensions are required to meet certain conditions to function:

- A dimension is required to maintain a 1:n relationship between parent and children. One parent may have many children, but each child must have only one parent. Example: One building storey has many rooms, but one room relates only to one building storey.

- For each attribute in a dimension, there must be a 1:1 relation with an attribute in the corresponding hierarchy. For example, a hierarchy attribute “month” could be assigned to the dimensional attribute “month name”.
- Hierarchy elements can only be assigned to a single dimension and may not be re-used in a different dimension.

In set theory, dimensions can be defined as follows (Lehner, 2003):

$$(\{ D_1, \dots , D_N, Top_D\} ; \rightarrow) \tag{1}$$

Here, a dimension D consists out of ordered elements $D_1, \dots , D_{n,}$, whereas Top_D is the maximum element which can be selected by all elements:

$$\forall_i \; (1 \; \leq \; i \; \leq \; n) \colon D_i \; \rightarrow \; Top_D \tag{2}$$

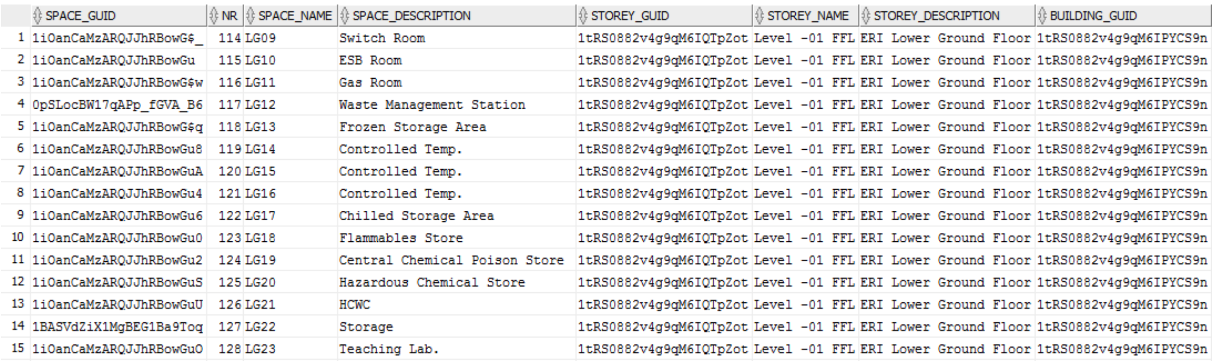
Furthermore, there exists exactly one element Di which defines all other elements. Thus, Di has the highest granularity.

$$(\exists_i \; (1 \; \leq \; i \; \leq \; n) \; \forall_j \; (1 \; \leq \; j \; \leq \; n, i \; \neq \; j) \colon D_i \; \rightarrow D_j) \tag{3}$$

2.4.4 Star and snowflake schema

When designing a DWH, literature often refers to either star or snowflake schemas. These are widely used schemas in data warehousing (Sandhu et al., 2015).

A star schema has a central fact table which is surrounded by dimensional tables. Each dimension is linked to the fact table through a foreign key relationship. The name star schema is derived from its star appearance in the dimensional model. Star schemas are commonly considered to be easier understandable due to their dimensional structure (Krippendorf and Song, 1997). Due to the simplistic dimensional model, the data retrieval process operates quickly. Only schemas with extremely large dimensional tables might suffer from performance issues in a star schema (Martyn, 2004). Due to its nature, hierarchy levels depicted in a single dimension may increase the redundancy of information within the dimension (Kimball and Ross, 2002). Figure 2-8 illustrates an example for each room in the spatial dimension the full storey description and also where the building information is stored. One can see a high redundancy of certain information which is the result of dissolving relationships. Large dimensions therefore contain huge information overheads.



	SPACE_GUID	NR	SPACE_NAME	SPACE_DESCRIPTION	STOREY_GUID	STOREY_NAME	STOREY_DESCRIPTION	BUILDING_GUID
1	1iOanCaMzARQJJhRBowG$_	114	LG09	Switch Room	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
2	1iOanCaMzARQJJhRBowGu	115	LG10	ESB Room	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
3	1iOanCaMzARQJJhRBowG$w	116	LG11	Gas Room	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
4	0pSLocBW17qAPp_fGVA_B6	117	LG12	Waste Management Station	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
5	1iOanCaMzARQJJhRBowG$q	118	LG13	Frozen Storage Area	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
6	1iOanCaMzARQJJhRBowGu8	119	LG14	Controlled Temp.	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
7	1iOanCaMzARQJJhRBowGuA	120	LG15	Controlled Temp.	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
8	1iOanCaMzARQJJhRBowGu4	121	LG16	Controlled Temp.	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
9	1iOanCaMzARQJJhRBowGu6	122	LG17	Chilled Storage Area	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
10	1iOanCaMzARQJJhRBowGu0	123	LG18	Flammables Store	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
11	1iOanCaMzARQJJhRBowGu2	124	LG19	Central Chemical Poison Store	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
12	1iOanCaMzARQJJhRBowGuS	125	LG20	Hazardous Chemical Store	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
13	1iOanCaMzARQJJhRBowGuU	126	LG21	HCWC	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
14	1BASVdZiX1MgBEG1Ba9Toq	127	LG22	Storage	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n
15	1iOanCaMzARQJJhRBowGuO	128	LG23	Teaching Lab.	1tRS0882v4g9qM6IQTpZot	Level -01 FFL	ERI Lower Ground Floor	1tRS0882v4g9qM6IPYCS9n

Figure 2-8 Redundant information in a dimensional table

In contrast, the snowflake schema has little redundancy and provides clear relationships between all hierarchy levels of a dimension. Its downside is that queries to join the dimension have a greater complexity (Petrenko et al., 2012). Furthermore, Kimball and Ross claim that snow flaked schemas generally compromise understandability and the browsing performance (Kimball and Ross, 2002). The same author emphasised this again, praising star schemas to cause less confusion. Additionally, the join conditions in a snowflake schema can be rather complicated. A minor change of the model may already require the recoding of query commands (Kimball and Caserta, 2004). Figure 2-9 provides an illustration of the two schema types.

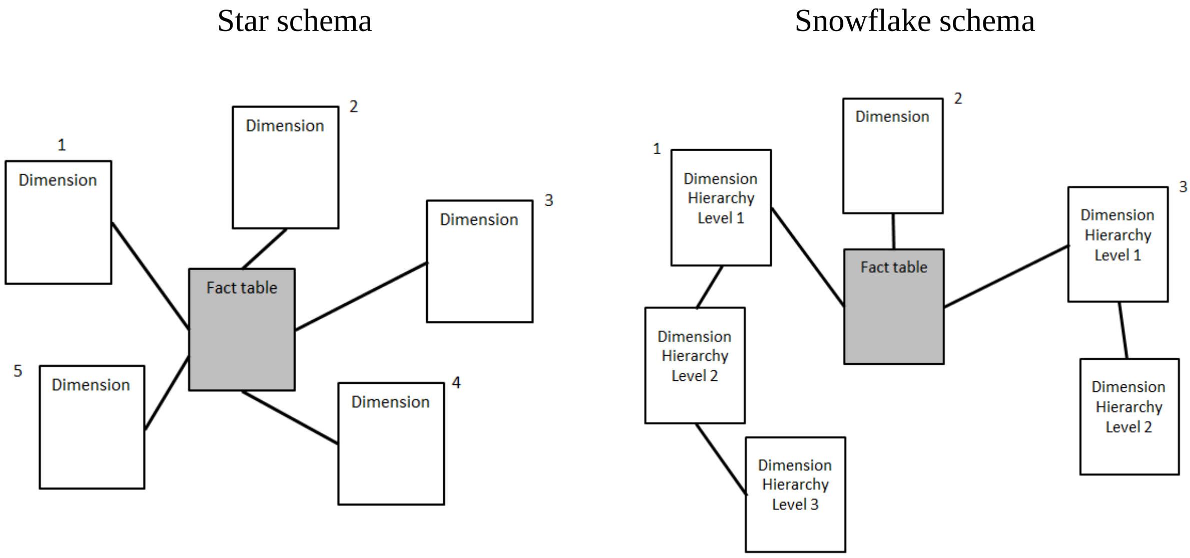

Star schema
Snowflake schema
1
Dimension
2
Dimension
3
Dimension
Fact table
5
Dimension
4
Dimension
1
Dimension Hierarchy Level 1
2
Dimension
Fact table
3
Dimension Hierarchy Level 1
Dimension Hierarchy Level 2
Dimension Hierarchy Level 3
Dimension Hierarchy Level 2

Figure 2-9 Comparison between star and snowflake schema

2.4.5 Managing derived data

Materialized Views (MV) are DWH objects which contain results from complex retrieval patterns. They can be implemented once and then refresh themselves automatically at given events. MV's can be efficiently used as storage for pre-calculated information. This allows users to access the essence of large bulk data sets. Since the information in MV’s is already aggregated, response times are reduced to a minimum. MV’s are widely accepted as one of the main DWH tools to optimise execution time (Kimball and Ross, 2002).

The implementation of a useful MV relies on accurate knowledge about the business requirements and frequency of queried information. Instead of a full refresh, MV’s also provide functionality to update incrementally. Incremental updates are more economic as compared to a full rebuild from scratch (Gupta, 1995).

2.4.6 Example

To give an example, a dataset, of 59 million sensor readings, was processed (Hoerster 2013). The fact table, holding all data sets, was called FACT_2013_RAW. The table includes information such as the measured value from each sensor, the ID of each measuring point, the time stamp of each measurement and unique reading identification for each value. A MV that contains monthly aggregated information can be implemented with the following SQL code:

Listing 3 Materialized View CREATE statement

```
create materialized view mv_fact_year as
select id,
TO_CHAR(timestamp, 'MM-YYYY') as time,
AVG(value) as avg_value,
min(value) as minvalue,
MAX(value) as max(value)
from FACT_2013_RAW
group by TO_CHAR(timestamp, 'MM-YYYY'), id
order by id;
```

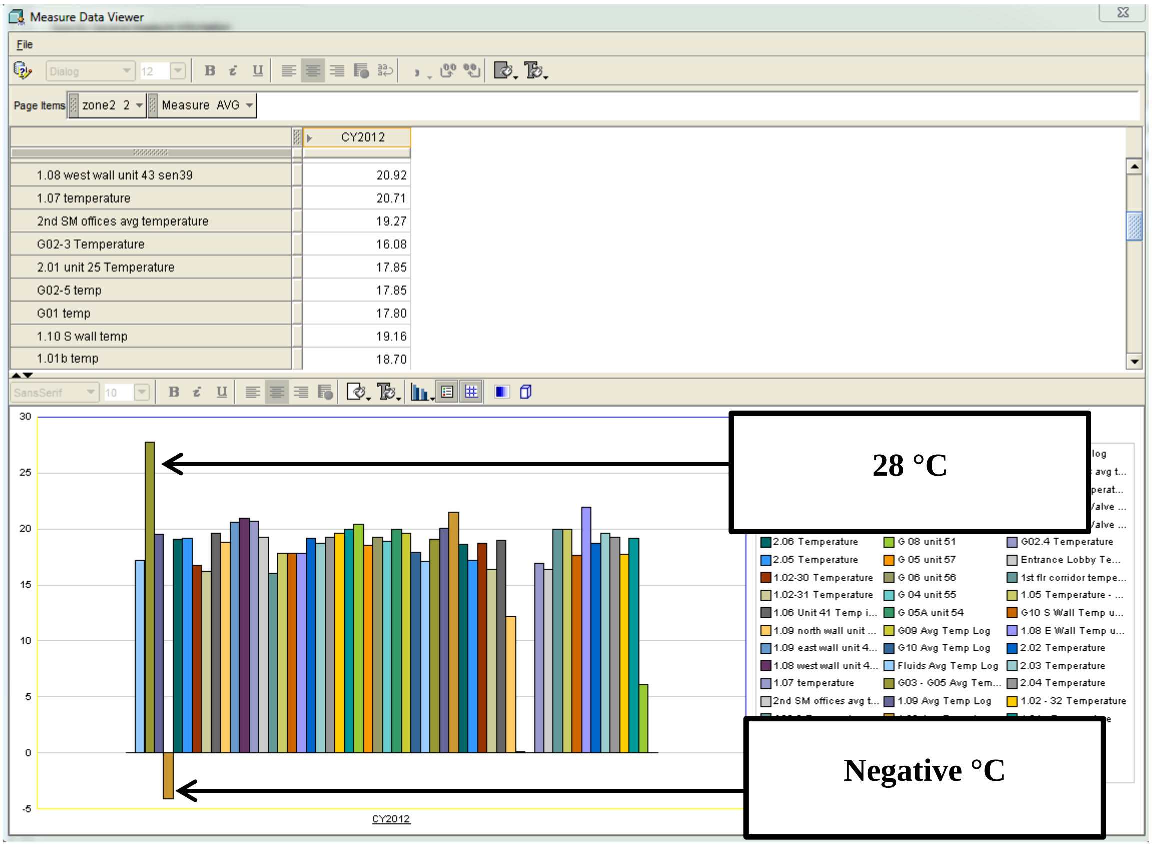
This SQL code creates a MV that calculates for each device the recorded minimum, maximum and average value on a monthly basis. Table 8 displays an excerpt of the SQL snippets output.

Table 8 Materialized View output

ID	TIME	AVG_VALUE	MAX_VALUE	MIN_VALUE
1	01-2013	11.03	12.81	0
1	02-2013	7.13	9.2	0
1	03-2013	8.16	10.65	0
1	04-2013	9.03	12.88	0
1	05-2013	8.45	11.82	0

Figure 2-10 Sample data representation provides an example where average room temperatures per year are listed and plotted. The figure reveals that e.g. one room reports negative temperatures while another room has an average temperature of roughly 28 °C.

Further examples for the processing of energy data in MV’s can be found in a recent doctoral thesis (Ahmed, 2011). The present work will also utilise MV’s but it will go a step further and populate OLAP cubes with information aggregated in MV’s.



Page Items: zone2 2, Measure AVG

	CY2012
1.08 west wall unit 43 sen39	20.92
1.07 temperature	20.71
2nd SM offices avg temperature	19.27
G02-3 Temperature	16.08
2.01 unit 25 Temperature	17.85
G02-5 temp	17.85
G01 temp	17.80
1.10 S wall temp	19.16
1.01b temp	18.70

Figure 2-10 Sample data representation

2.4.7 OLAP Cubes

A data cube is a DWH component that allows efficient acquisition of structured data. The design of individual cubes usually meets a specific business requirement. For example, a cube could be developed which contains average temperature readings across areas in a building. Cubes take advantage of the inheritance in dimensional data to efficiently retrieve only selected data sets (Elmasri and Navathe, 2004).

Cubes consist of a fact table and multiple dimensions. A Cube with only two dimensions is still called Cube, even though technically it is a slice. Cubes with four or more dimensions are sometimes also called Hypercubes but the principle remains the same (Rob et al. 2009).

Users can obtain information from Cubes by slicing entire sections, dicing individual segments or by specifically drilling down / rolling up the Cube (Adamson, 2009). An explanation for each term is given below:

- Roll-Up and Drill-Down changes the level of granularity for the inspected data, e.g. inspecting measurements on a monthly or a yearly basis.
- Slice and Dice allows selection of specific values for a dimension, e.g. slicing a time dimension for the year 2013 would retrieve all readings which occurred in that year only
- Pivoting (also called rotation) changes the cube's data representation by changing the orientation of the axes.

A comprehensive mathematical definition of the individual cube processes is given in (Aalst, 2013).

Existing fact and dimensional tables cannot be used as basis for Cubes if certain requirements aren't met:

- The fact table needs to have a primary key set and for each dimension a foreign key that is pointing to the dimensional table
- Each dimensional table needs a primary key
- A time dimension has to exist that is also linked through keys to the fact table

A cube C consists out of dimensions D and measures M (Lehner, 2003). A cube is defined in set theory as

$$C = (D, M) \ = \ (\{D_1, \dots, D_N\}, \qquad M_1, \dots, M_N\}) \tag{4}$$

Whereas $M_1, \dots, M_N$ are the measures which the cube provides. These could be e.g. aggregate or scalar functions.

2.4.8 Example

An example for data retrieval from a cube is given below. In this SQL snippet, a cube is queried that has three dimensions, namely ZONE, TIME and SENSOR. The query selects exemplary the sum of all measurements from all sensors in a zone called "G.09" aggregated to a yearly data representation. Its output can be seen in Table 9.

Listing 4 Data retrieval from OLAP cube

```
select  s.type_long_description as medium,
        c.sum as sum[kWh],
        z.long_description as zone,
        t.long_description as timeperiod
from    sensor2_view s,
        cube_view c,
        zone2_view z,
        time2_view t
where  (s.dim_key = c.sensor2
and     z.dim_key = c.zone2
and     t.dim_key = c.time2
and     s.level_name = 'TYPE'
and     t.level_name = 'YEAR'
and     z.level_name = 'ROOM'
and     z.room_long_description = 'G.09'
);
```

Table 9 Cube sample output

MEDIUM	SUM[kWh]	ZONE	TIMEPERIOD
Electricity	6548.45	G.09	CY2012
Electricity	6483.33	G.09	CY2013

OLAP cubes are an efficient and easy to operate tool to visualise and analyse building performance data. The flow of information inside the DWH is shown in Figure 2-11. The import of building information model data into the ETL tools will be discussed in the next chapter.

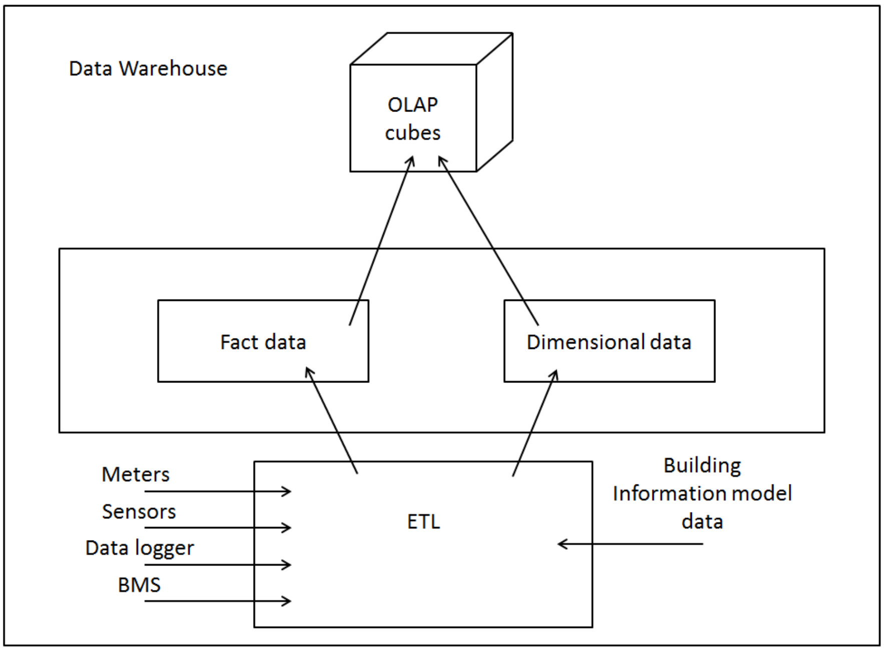


Figure 2-11 Data Warehouse model

2.5 Mathematical methods for energy data analysis

Load curves are data series captured by energy meters in defined time intervals. Some utility providers capture load curves with their own meter. This is usually conducted in large commercial sites. This data can be requested for internal analysis by the customer. For FM operators, a load curve is an essential element for analysing building performance.

A typical load curve can be seen in Figure 2-12. The ordinate shows the consumption in kWh while the abscissa depicts 35,040 samples taken in a 15 minute interval. The period shown in the figure is exactly one year. Various observations can be made from the data: the building's base load is around 200 kWh. It peaks at almost 900 kWh. There was one visible outlier where consumption was 0 kWh. The load curve shown is from a production with multiple shifts per day. Lower consumption is expected on weekends only.

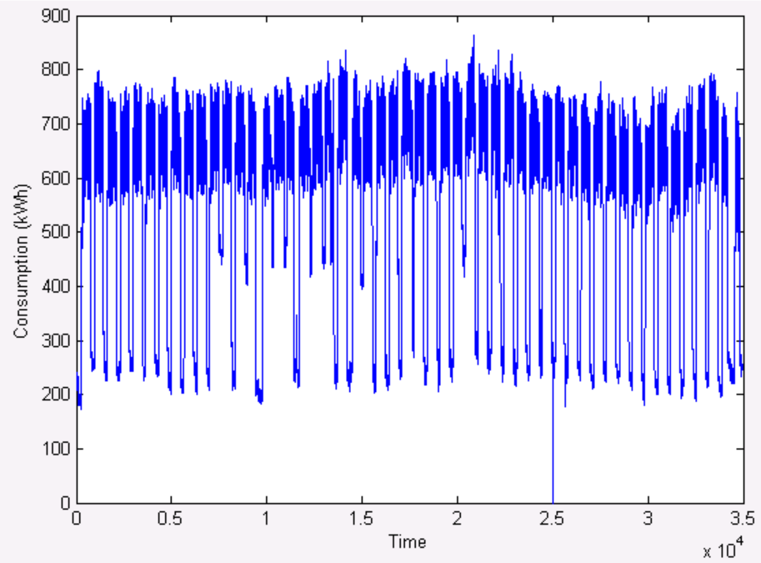


Figure 2-12 Typical load curve

Different building types have different types of load curves. For example, a school building would see increased load during school hours, while at night time or weekends the building is expected to run on a minimal basis (Base load). In a hospital, the difference between weekdays and weekends should be significantly lower as patients stay over weekends and surgeries also take place on weekends.

Li already stated in 2005 that the analysis of load curves has positive impact on daily operations, running systems and results in reduced energy costs (Li 2005). By analysing a load curve, expected behaviour can be verified and wrong behaviours can be identified. For example, a likely waste of energy is a technical system that is running on full capacity during a bank holiday in a school.

Unfortunately, metered load curve data is often inconsistent and not free of faults. This results in increased cleansing efforts in order to evaluate and analyse the data. Therefore, monitoring solutions often offer methods discussed in this chapter to counteract flaws. For example, the commercial EnMS i4Energy provides linear interpolation which is functionally described in their online documentation (Webfactory, 2015).

Two different methods are of interest when correcting the data: interpolation and cleansing. When a given data set has missing data points, the calculation (or approximation) of new data points is called interpolation. Data cleansing deals with detecting and removing errors and inconsistencies from data in order to improve its quality (Rahm, 2000). Errors are often outliers, which are values that are bigger by a magnitude compared to their adjacent values. Data cleansing can also be the correction of transformer ratio or unit.

2.5.1 Data cleansing

Reasons for bad or missing values in load curves are manifold. These could be e.g. malfunctioning meters, bad wiring, errors while transmitting the data, or even power outages. Nowadays, utilities providers and facility management companies manage many thousand buildings. For them, reliable data is essential for billing, analysis and optimisation. For efficiency reasons, in order to fulfil these purposes, manual correction of data has to be minimal.

Replacing bad or missing values through the construction of estimated values is called interpolation. If either future values or values from before the measurement period are desired to be estimated, then one speaks of extrapolation.

Mathematically, the problem of interpolation can be described in the following way: single values of a function f(x) are known, but the analytical expression for f(x) is not known. The knowledge of this function would enable the calculation of any specific point. If f(x) is not known, a curve needs to be fitted with all known values from the function. This curve then acts as a replacement function that may be used to estimate values of f(x). The two most common interpolation methods will be outlined in the next section.

For data cleansing and the detection of outliers, various research is available, e.g. statistical methods have been studied to identify outliers (Davies & Gather 1993) (Ferguson 1961). Here, it is mostly assumed that the underlying distribution is known. However, for load curves, this is not guaranteed. Data mining techniques have also been utilised to identify outliers (Knox & Ng 1998) (Ramaswamy et al 2000). The downside of their approach is that these techniques are usually designed for data sets of fixed length and might not work well on load curve data which continuously grows.

2.5.2 Example for the interpolation of a data set

MATLAB allows the application of various interpolation algorithms (MATLAB, 2015). It will be used to demonstrate the linear and spline interpolations. These two algorithms are widely used in many fields and are not specifically designed for energy data. Research on data interpolation and cleansing which is focused on energy data will be discussed in the following section.

Assuming we have a small data set which consists out of 10 values as seen in Table 10. Upon inspection, a few values are bad and discarded. Our data therefore is:

Table 10 Sample data for interpolation

Y Values	3	5	-	1	3	-	-	8	6	7
X Values	1	2	3	4	5	6	7	8	9	10

A graphical representation of these values is shown in Figure 2-13. One can see vast gaps in the graph which make it difficult to estimate the curve shape.

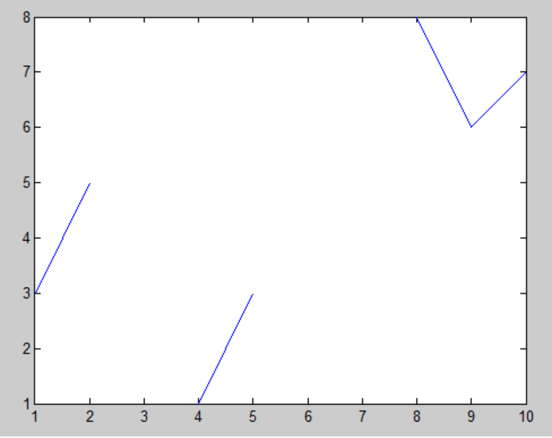


Figure 2-13 Incomplete data set before interpolation

Figure 2-14 and Figure 2-15 show the data after applying linear and spline interpolation. While the data set is very limited, one can see that the linear interpolation always creates straight connecting lines. On the other hand, the spline interpolation is taking a curved shape into consideration and, especially with the missing points at x=6 and x=7, introduces a curve which is not visible in Figure 2-13.

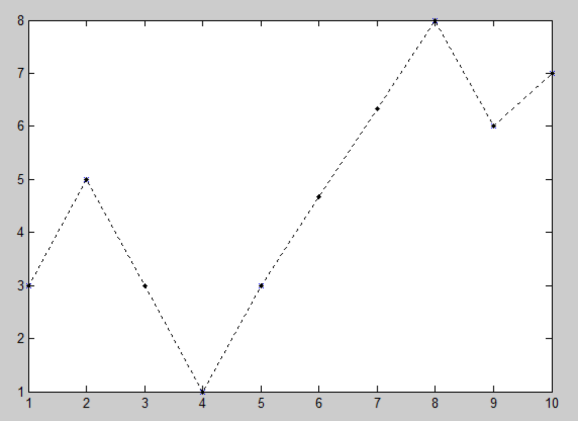


Figure 2-14 Linear Interpolation of sample data

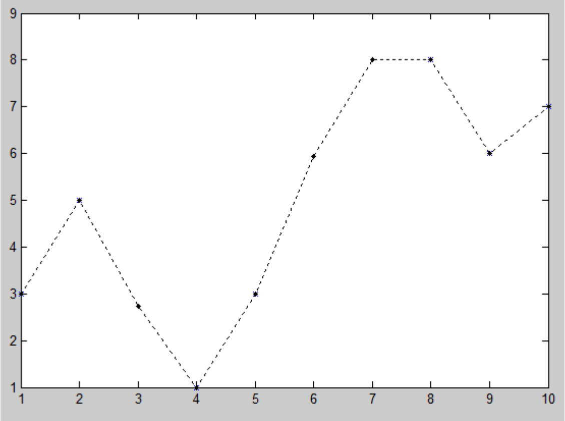


Figure 2-15 Spline interpolation of sample data

2.5.3 Interpolation of energy data in related research

The methods discussed in the previous chapter are generic and may be used on any kind of data. This section will provide an overview for data cleansing and interpolation in research related to energy data. The identification of outliers and cleansing of data has been discussed widely in previous research (Abraham & Chuang 1989) (Ljung 1993). Here, regression analysis has been utilised to determine outliers in time series curves. However, their work considers time series as non-changing, static values which renders their algorithms inappropriate for frequently changing load curves.

More recently, Chen et al. have introduced a smoothing method that corrects gaps and outliers in a load curve (Chen et al 2010). They claim that their algorithm is applicable to all kind of energy load curves. However, it requires a little amount of user interaction as their system needs the user to specify the ideal smoothing parameter. This is not desired by an EnMS which performs data cleansing without user interaction.

This limitation has also been criticized in (Høverstad et al 2013), where a system is envisaged which would automatically configures itself. For their work, data cleansing is performed in order to increase the accuracy of energy consumption forecasts for the next 24 hours. Their work involves several fairly complex load prediction models.

2.5.4 Clustering

With the availability of Big Data the desire to draw different kind of analysis emerges. For FM operators analysing load curve data, it is of interest to identify different behavioural patterns of buildings. These could be e.g. building in operation and its base load. Gaining knowledge by analysing data from different perspectives is called data mining (Nisha, 2015).

One notorious example is everyone's usage of web search engines. Their providers can store search queries to optimise their results but also e.g. to create user profiles and custom ads by tracking and analysing user inputs.

The clustering of data is one popular data mining technology. Clustering takes a data set as input and outputs N clusters. These clusters represent the input data assigned into common groups (Nizar, 2005). One example is given in Figure 2-16, where clustering of random sample data was conducted in MATLAB. After plotting the sample data (seen in the upper plot), one can see two distinct areas where values are concentrated. After applying the algorithm, MATLAB has identified the individual clusters (highlighted in blue and red).

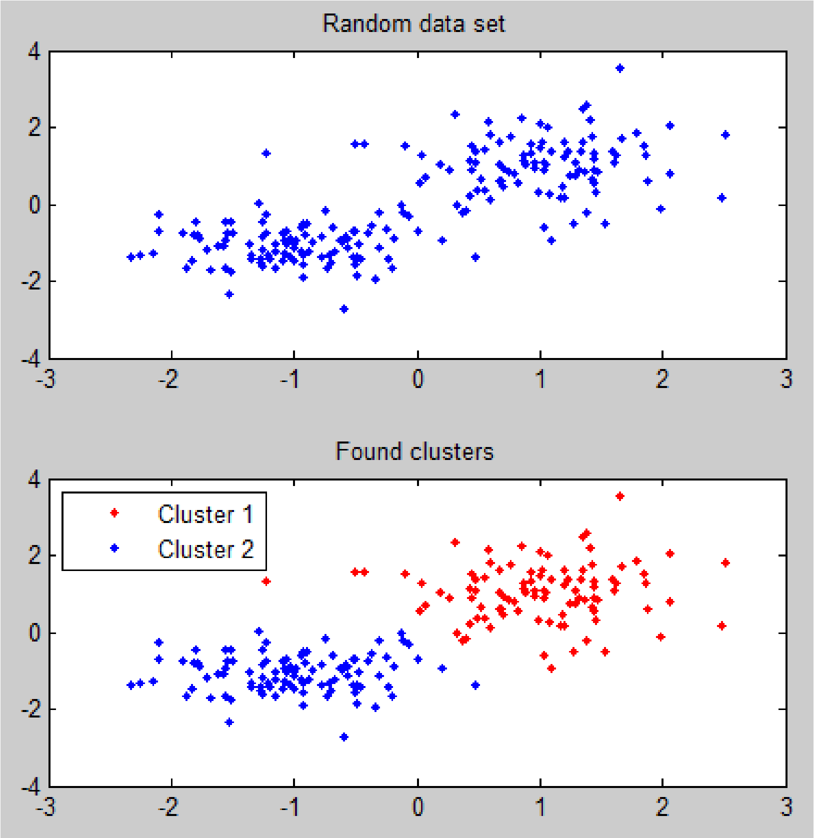


Figure 2-16 Clustering of random sample data

A recent Master thesis research conducted under the author's supervision concluded that more than 100 clustering algorithms exist (McSwiney, 2015). However, they can mostly be grouped into three different categories. These are summarised in Table 11.

Table 11 Clustering methods, derived from McSwiney, 2015

Algorithm	*Description*
Centroid based clustering (often called k-means)	Each identified cluster has a centre. This centre is the mean (also called: centroid) of all points in a cluster. The algorithm tries to minimise the Euclidean distance (distance between point and centre). Ideally, clusters are formed as a sphere and are not overlapping. (Manning, 2008)
Gaussian mixture models	Clusters are identified by probabilities. For each data point, the probability is determined how likely it belongs to a cluster. Thus, the relation of all data points and centroids are estimated. Based on these probabilities, the clusters are identified. (Nikolaou et al. 2012)
Hierarchical clustering	This method is also called connectivity clustering. The aim of this method is to build a hierarchy of clusters. There is two possible ways to perform hierarchical clustering: (a) the bottom up approach (also called: Agglomerative) where small clusters are continuously merged into larger clusters until only one cluster is left. (b) the top-down approach (also called: Divisive) where all data is represented as one cluster and then split into smaller clusters. (Nisha, 2015)

Both the centroid based clustering and the Gaussian mixture models require that the number of clusters k is known before clustering. This contradicts a fully automated clustering process. The hierarchical clustering has the disadvantage that, depending on top-down or bottom-up approach, the splitting or merging is never reversed. This limitation prevents any potential clusters found at a later stage of the algorithms runtime to become a cluster.

To identify the clustering algorithm(s) most suitable for energy and comfort data, McSwiney performed an exhaustive research which is summarised below in Table 12 (McSwiney, 2015). This research suggests that the centroid based clustering k-means is the most appropriate clustering algorithm for the data discussed in this thesis.

Table 12 Related research and their choice of algorithm, derived from McSwiney, 2015

Author(s)	*Verdict*
Nikolaou et al., 2012	Their work concludes that k-means clustering is the most suitable clustering algorithm for heating demand.
Abreu et al, 2012	k-means was utilised with satisfying results to cluster electricity consumption to detect habitual behaviours.
Bogen et al, 2013	Applied both k-means and hierarchical clustering to multiple building performance data sets. They conclude that the hierarchical clustering had poor performance when big data sets were applied. On the other hand, k-means produced satisfying results after comparing expected and actual resource patterns.
Yu et al, 2011	k-means was utilised to identify occupants influence on building energy consumption. They concluded that k-means allows insight into energy usage patterns.
Heidarinejada et al, 2014	k-means was used to classify simulated energy consumption for buildings. The authors found clustering as a useful statistical tool to analyse energy consumption data.

3 Methodology

The methodologies developed within the context of this thesis are outlined in this chapter. Its work focuses on improving, standardising and enriching the state of the art of Energy Monitoring Systems (EnMS).

This chapter is structured as follows:

- The acquisition of fact data through a metering concept is outlined. Its aim is to outline an approach which is generally applicable to a wide range of buildings
- The acquisition, of dimensional data by extracting selected information from Building Information Models (BIM), is explained
- The identification and cleansing of faulty fact data is discussed. Specifically, gaps in data and outliers can be cleansed through the proposed algorithm.
- A database schema compatible to Industry Foundation Classes (IFC) is defined and a Data Warehouse (DWH) infrastructure to support efficient analysis of Key Performance Indicators (KPI) is developed.
- A clustering algorithm to analyse metered performance data is applied.

The above topics contribute to the individual layers of an EnMS shown in Figure 3-1.

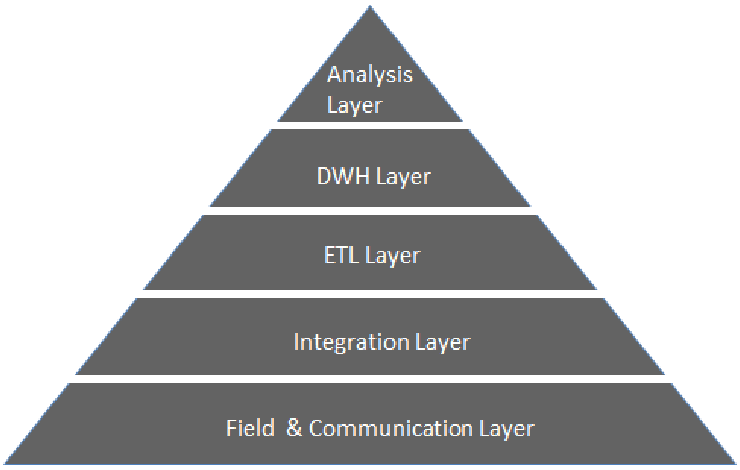


Figure 3-1 Layers in an EnMS

The field & communication layer contains a metering concept to acquire readings from field devices. The integration layer enriches acquired data with selected information from BIM models. The ETL layer interpolates and cleanses selected readings of bad quality. The DWH layer stores, combines and aggregates the data from the field layer and from the BIM models. Lastly, the analysis layer enables users to benchmark their data through KPI’s and to observe their load curves through application of a mathematical clustering method.

3.1 Metering Concept

In order to populate an EnMS with raw data to calculate KPI’s and generate charts, data needs to be acquired. Measured (often also called recorded) values, e.g. energy consumption or temperature readings are data retrieved on the field & communication layer.

As discussed in the state of the art analysis, the meters which provide this data can be either analogue or digital. Due to the variety of meter vendors on the market, it is likely to find heterogeneous meter installations in buildings. The approach of this work is a hybrid monitoring solution that aims to acquire readings from both types of meters.

In order to consolidate meter readings from various meter types and vendors, a unification of meter signals is necessary. Methodologies which aim to achieve this requirement are discussed in the following sections. Some installations may not require any of these methods to be applied.

3.1.1 Splitting of pulses

Sometimes, pulse signals need to be acquired twice. This could be for example when the local BMS already established a connection with the meter. In this case, a pulse can be split. Splitting a pulse creates two pulse signals of the same properties (cloning). By doing so, it is guaranteed that there will be a second pulse available. The splitting of pulses guarantees that there won’t be any interference with the two receiving parties. The pulse splitting can be done with various off-the-shelve commercial products. They also do not require any configuration. Instead they just take every input pulse and split it to its two outputs.

3.1.2 Convert analogue to digital signal

Analogue meters send pulses whenever a defined consumption threshold is met. Pulse transmissions are sensitive to interference and provide no historic readings. To counteract this deficit, pulses are converted from analogue meters to the M-Bus protocol. It minimises potential interference on the wire and allows the indirect participation of the meter in a digital bus system. The downside of the converter units is that they require parameterisation after their installation. This ensures that the converter learns the unit and value of a pulse. Configuration is usually done with vendor specific tools. The commercial market offers several cost efficient devices to convert a pulse signal to M-Bus. These get configured with an analogue meter's attributes:

- Measured medium
- Transformer ratio
- Measured unit
- Meter reading

This configuration enables the converter to represent the analogue meter on the digital bus. For other devices on the bus, the converter acts like a meter directly connected to it.

After its configuration, the converter unit acts as a full member on the bus. Since it participates in a bus network, the same applies as for the digital meters: incorrect set up of the converter unit can affect existing devices and their communication.

An additional source for mistakes is the incorrect set up of the meters attributes. For example, wrong transformer ratios will report consumption which is off by a factor.

3.1.3 Covering long distances

The monitoring of buildings and its systems gets complicated when (i) a large number of meters need to be monitored, and (ii) when these meters are located at distant or hard to reach locations. The difficulty of wiring installations could increase further when e.g. drilling of holes through fire barriers is required. One solution for this scenario is the utilisation of multiple data loggers. This would separate the building in logical groups. A downside to this approach is that it further increases complexity and cost.

A solution to this is to use existing LAN technology which often spans across a building and is existent in most rooms. Instead of having multiple data loggers which acquire meter readings individually, gateways will be utilised. These devices read the meter information and send them over the LAN to a single data logger. The advantage is that existing wiring can be used and only a single internet connection for the data logger is needed. It therefore minimises operational and installation cost. A downside is that the installation co-exists with the existing infrastructure, occupying the same network.

In order to exchange information through the LAN, a TCP/IP transport protocol needs to be utilised. Suitable gateways offer the conversion of M-Bus information to Modbus TCP. The latter is a standardised solution to encapsulate Modbus traffic into TCP packets. The Modbus specification is open and free of charge.

If, at any LAN-enabled location only a single meter is required, it is possible to install a meter that is capable of communicating via Modbus TCP directly. On the one hand, this type of meter is capable of measuring consumption. On the other hand, it has Ethernet connectivity and can communicate its readings via Modbus TCP. It is therefore a single-device driven solution which minimises installation efforts. However, this still requires a data logger capable of receiving and processing the information.

However, this needs to be decided with foresight, as any further meter will likely require a gateway device to consolidate the meter readings.

3.1.4 Data Transformation

Engelberg claimed that the choice of the right data logger is essential for a standardised procedure. It is a critical factor to have a device that can be easily configured in order to keep installation costs to a minimum (Engelberg, 2007).

Ultimately, the choice of the data logger is up to the implementer, but it is highly important to have a consistent installation across multiple buildings. Consistency can be defined as:

- M-Bus is the chosen field bus protocol.
- Analogue meters get their pulses converted to M-Bus.
- Data loggers collect all meter signals through M-Bus exclusively.
- Data logger storage format for signals should be unified.

Systems and installations which ignore defined consistencies still qualify for an EnMS. However, if certain standards are not followed, it becomes more difficult to (a) understand a custom installation, (b) train personnel, and (c) fix errors (e.g. lost signal from a meter or acquired incorrect meter readings). Additionally, a wider range of used components increases the management of replacement processes. In the end, minor changes in data loggers could easily affect both the field layer and the database layer.

3.1.5 Data communication and data transfer

All acquired readings are stored in the data logger’s local file system. On a broader scale, the acquired readings from many data loggers can be seen as a federated system. However, due to limited storage space and processing power, it is desirable to consolidate all readings in a database system. This also enables sophisticated analysis covered in chapter 5.

An Internet connection is used to transfer data collected in a data logger into the central Data Warehouse. Information captured with the data logger is accessible e.g. through the FTP protocol. Using a standard FTP program, meter readings can be accessed using the IP address of the data logger and a valid username/password combination.

To secure tenants’ information, it is recommended to use a secure tunnel protocol like a Virtual Private Network (VPN) for secure data transmission. From a facility managers perspective it is highly important not to cause any interference with the building operation or the locally installed IT equipment. Therefore, a dedicated line is recommended to guarantee that the monitoring equipment is installed as a stand-alone solution.

In order to receive standardised information from different buildings, a data exchange format has been defined. All acquired readings are stored in daily files. These have the advantage that the transferred information is only a few kilobytes per file. Always appending readings to the same file without creating new ones (no rotation) would result in a huge transmission overhead as previous readings would be retransmitted every time.

Another advantage of the file based approach is the tangibility which simplifies the communication across systems. It may also help in auditing errors or maintaining the data logger as it is human interpretable.

3.1.6 Security for data transfer

In IT security, an internationally recognised security standard has been established by ISO 27001 (ISO 27001, 2013). The standard outlines the concept of an information security management system (ISMS). Similarly to ISO 50001, an audit may be undertaken to gain ISO certification. In contrast to ISO 50001, there are no legal requirements in place which require such certification. The standard contains three protective goals as a fundamental concept to maintain security:

a) Confidentiality: Any information may only be read and processed by authorised users.

b) Integrity: Data may not be changed unknowingly

c) Availability: Data access must be guaranteed at a given time

Because of their starting letters, these three goals are often also called the CIA triad (Chia, 2012).

For metered data, an evaluation regarding the three protective goals has been undertaken by the author in Table 13.

Table 13 IT security principles

Protective Goal	Assessment of meter data
Confidentiality	The confidentiality of metered consumption data can be classified with medium importance. A malicious attacker who obtains the metered data is able to draw conclusions about the building and its usage. One could e.g. derive opening times and personal comfort data from obtaining the data.
Integrity	False readings are common and do not have any significant impact. Handling errors in data is a common task among energy analysts. Maliciously changed data can be identified when comparing recorded consumption to the utilities bill. Personnel familiar with the source can identify the data is implausible.
Availability	Availability of the data has low significance. Analysis is usually done on historic data. The absence of most current data can easily be ignored as long as the data eventually gets recorded in the central analysis platform. Downtimes of the storage platform would only affect analysts which have to postpone their tasks.

The security of the data exchanged between data logger and central analysis platform largely depends on the implementation. Common protocols found in data loggers for data transmission are shown in Table 14.

Table 14 Common data transfer protocols

Protocol	Security	Description
FTP	Unencrypted by default	Data gets transferred in files, e.g. CSV or XML files. Data is either pushed or pulled
Email	Unencrypted by default	Data gets sent by email from the data logger. CSV or XML formatted files are mostly used.
Web services	Unencrypted by default	Data is provided to a Web service which processes the information and stores it in the central analysis platform. Data is transferred via an API. As there is no standard API, data logger and Web service need to match
IEC 60870-5-104	Security is broken (Maynard et. al, 2014)	The IEC 60870 standard is a protocol often found in the energy sector. It does not utilise files or an API. Instead, it is packet-based. Amendment 5-104 enabled communication via TCP/IP networks (IEC, 2016)

As Table 14 outlines, security is not enabled by default. However, by adding transport layer security (TLS), data can be transferred in a secure way. TLS is a cryptographic protocol capable of encapsulating communication traffic (Dierks & Rescorla, 2008).

Another option is the utilisation of a virtual private network (VPN). A key benefit of using a VPN is that all information is sent through an encrypted channel. The information itself does not need to be encrypted which allows conventional devices which lack security to benefit from an additional security layer.

3.1.7 Classification of monitoring types

It is envisaged to cover building installations with four different monitoring types. These types are visualised in Figure 3-2. Types A and B are connected directly via M- Bus to the data logger. Types C and D have the same structure; however, the meters are not connected to a data logger. Instead, they connect to a gateway which transmits the received data to the data logger through Modbus/TCP. Therefore, type C is an extension of type A and type D is an extension of type B. Types C and D are only necessary in buildings where large distances need to be covered.

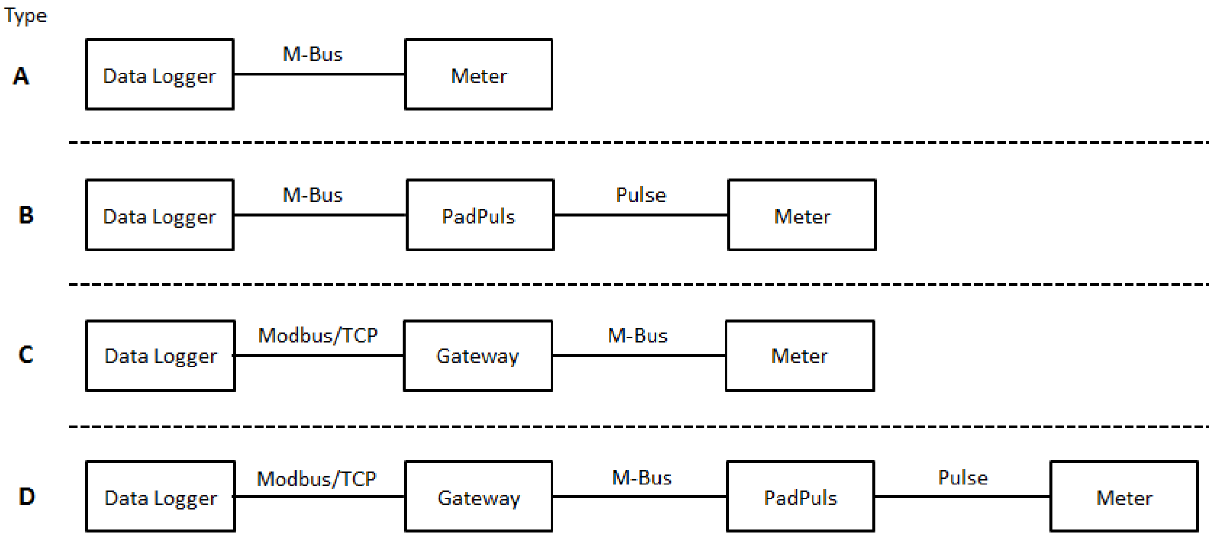


Figure 3-2 Different monitoring types

3.1.8 Sequence of activities

The chart depicted in Figure 3-3 illustrates chronologically the individual steps which need to be undertaken for conducting an installation. The last step "fixing bugs" contains time saving potential as standard processes should minimise bug fixing. Additional installations in different buildings should further increase routine and effectiveness.

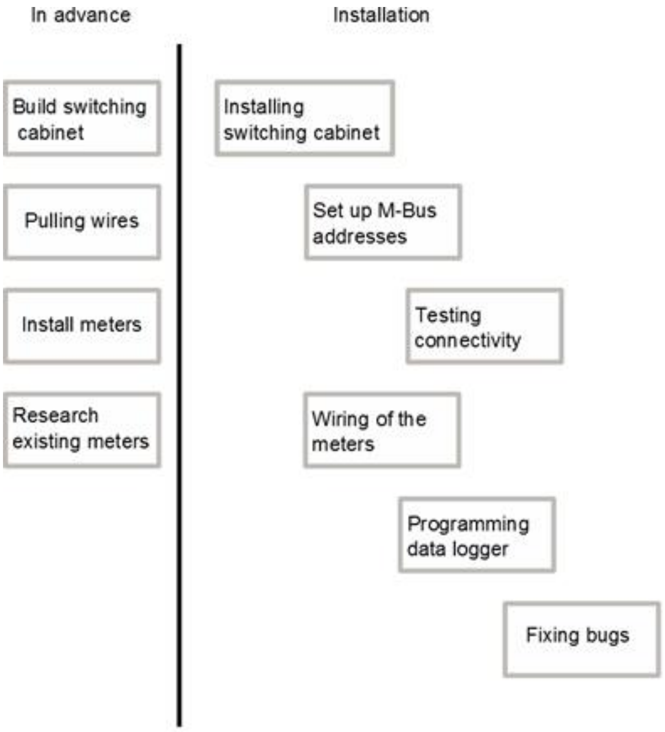


Figure 3-3 Sequence of activities

3.1.9 Potential error categories

This section highlights possible errors which should be avoided when conducting an installation of monitoring equipment. Table 15 categorises these errors and provides an explanation. It shows how versatile the issues are and that installations need to be undertaken with great care.

Table 15 Error categories for equipment installation

Error category	Error description
Pulse converter configuration	Wrong set transformer ratios will result in reported meter readings that deviate by a multitude. Wrong set time and signal type will report incorrect figures.
Analogue meters	Meters of unknown type (e.g. due to missing labelling) need to be either replaced or their technical parameters examined and verified by an electrician.
Digital Meters	Digital meters have multiple registers. Reading the wrong register retrieves incorrect values.
Data logger	A consistency for labelling the meters needs to be enforced as not to confuse data sources. Wrong parameters for meters will result in falsely reported values.
Bad reception	Bad signal reception can disturb wireless signals. The signal strength in installation areas needs to be examined during the site inspection.
Bad preparation	After a site inspection has taken place, the engineer installing the technical equipment should be aware of all meter locations and the envisaged location for the data logger. Main meters are often owned by utility providers and for the installation, the allowance to access their meter should be available.
Bad teamwork	When working with subcontractors, their coordination needs to be organized and structured.
Bad hardware	Data loggers should be tested and configured with an initial configuration before entering the building site.
Handling of values	When converting meter readings between different formats, it needs to be assured that the conversion is executed in a clear and precise manner.

It is noteworthy that these errors are generic and not just specific to a single installation. It is therefore beneficial to be familiar with these before any installation to minimise the occurrence of errors during installation.

3.1.10 Event notification between data logger and central analysis platform

Besides the periodic transmission of metered consumption data, additional information could be exchanged between data logger and central analysis platform. This information may be events, e.g.

- alarm functionality for exceeding defined thresholds
- data loss from meter(s)
- critical battery states from wireless equipment

While the before mentioned events are triggered by a behaviour on the data logger site, there can also be events originating from the central analysis platform, e.g.

- the definition or change of thresholds
- remote configuration of further meters
- data logger maintenance

Some modern data loggers can communicate via BACnet. This creates further areas of application for events. Once a data logger is connected to a BMS, certain control operations become possible. While this is a highly customised set up, it is still of great interest. Approaches like e.g. remote building operational control centres (RBOCC) are a specific component of current research and will likely see more attention in the future.

The requirement for real-time event communications is a dedicated line between the participating systems. This could be realised through IEC 60870-5-104 or a VPN connection. Additionally, the communication of events is not standardised and therefore must be implemented for each type of data logger.

3.1.11 Auto configuration of meters

After the wiring and installation of monitoring devices, a configuration of a data logger is necessary to enable communication. This is a repetitive step needed to finalise every installation. Ideally, a freshly installed data logger would scan the devices on its network and automatically configures itself to be able to communicate with every single device. Current research focuses on similar scenarios to mitigate this issue (IEEE, 2012, Nthontho et. al, 2011).

However, the ongoing research is only applicable to selected meters which communicate digitally. In contrast, impulse meters only transmit an analogue signal. They do not share any information about the medium they measure. Therefore, these cannot be automatically configured. An exception to this would be the case where only one identical type of analogue meter would be installed everywhere.

Automatic configuration of digital meters is possible within limits for certain types of meters. Research has shown that it cannot be used generically for all types of digital meters (Matt et. al, 2015). This is because the underlying protocol definitions are vague and vendors have interpreted the specifications differently. The different implementations of the specification would lead to false readings if an automatism would be utilised.

The applicability of an automated configuration depends on the use case. For utility providers, automated configuration is a useful feature as they work closely with selected meter vendors. For FM providers, installations cannot be done automatically as each building has a different setup and all possible types of meters may be found on site.

Furthermore, meter replacements for meters compatible with automated configuration are not feasible due to the installation cost. Additionally, meters usually do not belong to the FM provider, so owners' needs to be consulted first. This would create additional bureaucracy which is preferably avoided.

3.1.12 Added value of the metering concept

The standardised approach outlined in this chapter allows the planning and installation of monitoring equipment in a structured way. The method outlines routines to handle different scenarios which may be found on site upon inspection. By being aware of possible scenarios which could be identified, engineers can rely on the outlined methods to provide a robust solution to electricians, who handle the installation process. By following this chapter's definition of the metering concept, building installations may be conducted more organised in a shorter length of time. The awareness of the possible scenarios may lead to fewer surprises and also may positively affect the confidence of the executing personnel.

3.2 Data cleansing

Load curve data retrieved from meters is often inconsistent and not free of faults (Chen et al., 2010). This results in tedious efforts to cleanse data in order to evaluate and analyse. Ideally, this cleansing is done automatically. This research aims to eliminate the need for manual cleansing by introducing an algorithm which can be applied automatically on metered consumption data.

Since the gas consumption of a building strongly relates to its outside temperature, it can be used to interpolate missing or bad readings. Through a combination of gap detection and outlier identification, faulty data gets flagged. These occurrences get interpolated through a correlation between building heat consumption and outside temperature. The required weather data can usually be acquired from nearby weather stations.

The consumption of gas typically relates to heating purposes, e.g. rooms and domestic hot water. Therefore, its consumption can be related to the outside temperature. This allows deriving a building-specific equation to replace erroneous data.

3.2.1 Visual inspection of a faulty data set

Faulty readings often result in large spikes in the consumption data (see Figure 3-4 for an example). This data is taken from an office building which was equipped with a monitoring solution for its main meters. It seems impossible from this perspective to draw any meaningful conclusion with regards to the building's consumption behaviours. Note that the consumption has a spike of almost 15,000 kWh for a single reading. This would be roughly equivalent to the yearly consumption of a terraced house.

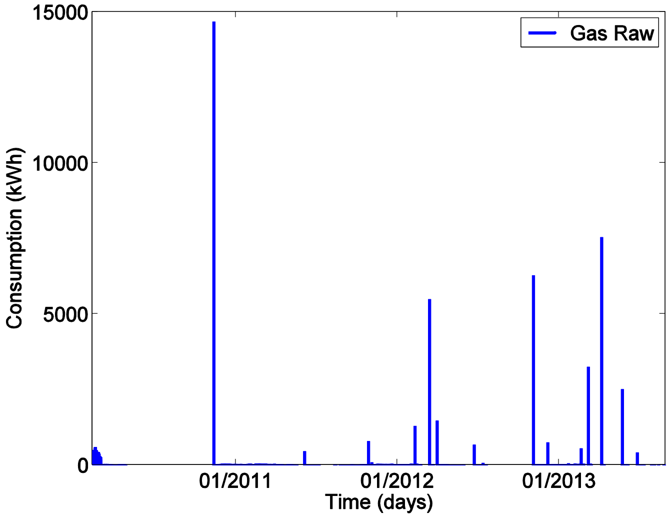


Figure 3-4 Raw values of gas consumption

While it seems that most values are 0, the largely magnified graph seen in Figure 3-5 reveals more typical consumption patterns. The areas with zero consumption have decreased significantly, yet there are still gaps with apparently no data available. In this representation, it seems more likely that the consumption peaks at less than 100 kWh with all further spikes classified as outliers. At this stage, one can only speculate about the reasons why the data is in such a bad shape. For example, there is a large spike after every gap which indicates that consumption during a gap is not “lost”. Instead, it is reported as single figure after an outage. While this might be even in line with the actual consumption, it is not distributed in the time interval as it occurred.

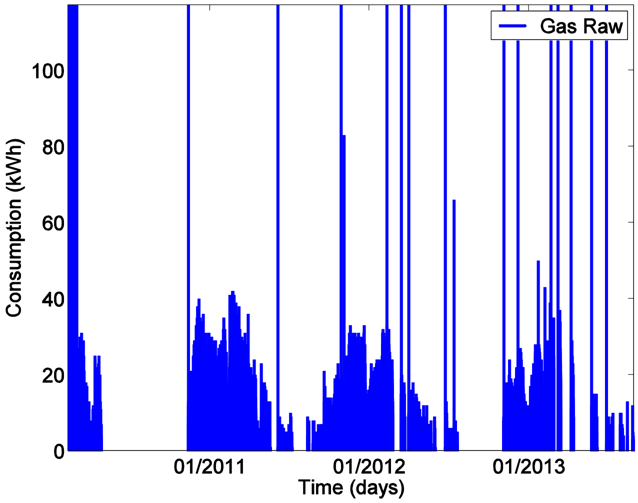


Figure 3-5 Magnified raw values of gas consumption

3.2.2 Cleansing algorithm

This methodology aims to flag all occurrences of either missing data or bad data and interpolate these with replacement values. The algorithm has four input variables:

(i) gas energy consumption

(ii) timestamp corresponding to gas energy consumption

(iii) average daily temperatures

(iv) date of average daily temperatures

Instead of gas energy consumption, a different energy carrier, e.g. electricity could be cleansed. However, the methodology is not fully applicable to cleanse electricity data since lighting and plug load is not related to the outside temperature. It may be related to sunshine duration for buildings with high lighting consumption but this was not further inspected as part of this work. The methodology could still be utilised for the cleansing of electricity data once a significant amount of consumption relates to heating or cooling efforts. If, through metering of sub systems, the non-temperature dependent consumers can be filtered out, the methodology may be applicable for other mediums. This possibility was not further explained as part of this thesis due to the lack of the necessary sub metering.

As a first step, a regression analysis is performed on the input data, by fitting a curve of the form as in equation 5.

$$W(T) = W_0 + \max(w(T_0 - T), 0) \tag{5}$$

Here, T is the outside daily average temperature, W_0 the base load, T_0 the heating threshold and w the temperature coefficient.

The heating threshold is defined as the temperature up to which gas is only consumed for the base load of a building. If it gets any cooler, gas consumption increases. This can be defined as the temperature dependent consumption.

For example, in the curve shown in Figure 3-6 the heating threshold T_0 is approximately 16 °C and the base load W_0 is 20 kWh. The temperature coefficient w equals the value of the slope. Here, the gas consumption at e.g. T = 0 °C would be around 350 kWh.

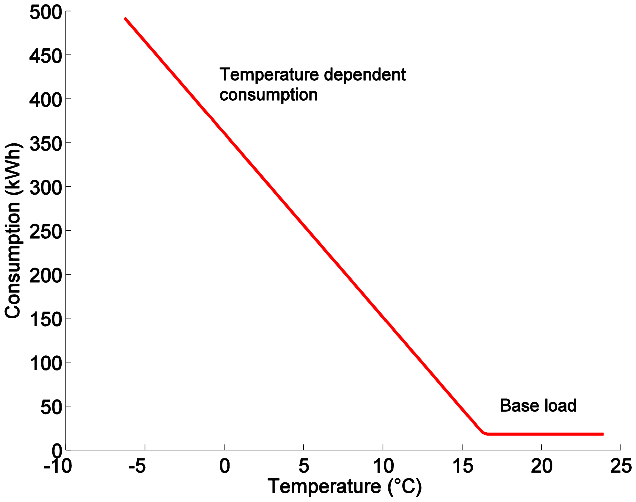


Figure 3-6 Ideal result of a regression analysis

Algorithmically, for fitting the above curve to the data and identifying outliers a standard RANSAC scheme (Fischler and Bolles, 1981) is employed. RANSAC can be used to fit a curve and detect outliers. In this case, RANSAC is utilised to construct the equation for the heat curve. Using the heat curve, missing values and outliers detected by RANSAC can be replaced by estimated values. RANSAC has been proven to be very robust in a wide area of fields and applications, and works also well in this application.

In pseudo code the algorithm is as follows, where it is assumed that the data points (temperature and gas consumption) are $(T_j, W_j),\ j = 1, \dots, N$.

```
Initiate outliers: $o = \emptyset$
Do
Find $W_0$, $T_0$, $w$ minimising
        $\sum_{j=1}^{N} |W_j - W(T_j)|$        (7)
where $j \notin o$
Median Error:
        $err = median(\{|W_j - W(T_j)|\ j = 1, \dots, N\})$        (8)
Update outliers:
        $o = \{j\ |W_j - W(T_j)|\ > 8\ err\}$        (9)
until convergence.
```

The nonlinear optimisation problem in the first step of the loop is solved by the simplex algorithm (Lagarias et al).

Once the heat curve is fitted, the following is known: (i) the parameters W_0, T_0 and w, and (ii) which data points are outliers. Next, the missing and/or faulty data points may be estimated to be $W(T)$, where T is the daily average temperature and $W(T)$ is as in equation 5.

Consider the data plotted in Figure 3-7 together with the heat curve estimated by the above algorithm. The data consists of some serious outliers, resulting in a fitted heat curve that is not meaningful in its graphical representation. Here, the heat curve is represented as a straight line instead of the expected shape as depicted in Figure 3-6.

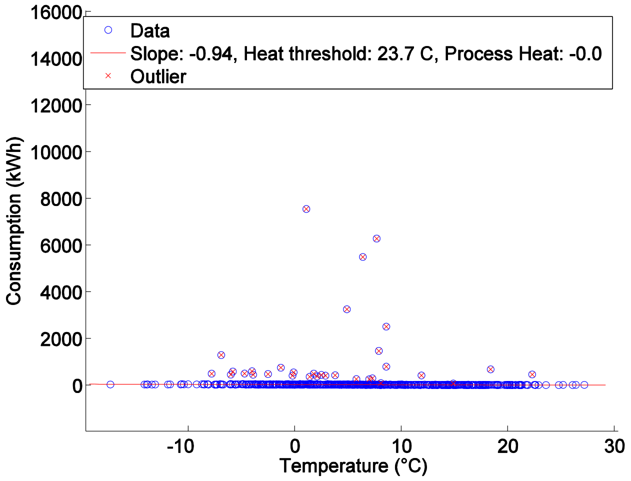


Figure 3-7 Scatter plot with daily consumption

In Figure 3-8 the missing or bad data was estimated, resulting in values placed directly on the red slope. The output resembles to what has been illustrated in Figure 3-6. After the cleansing process, the heat curve has again a shape as expected.

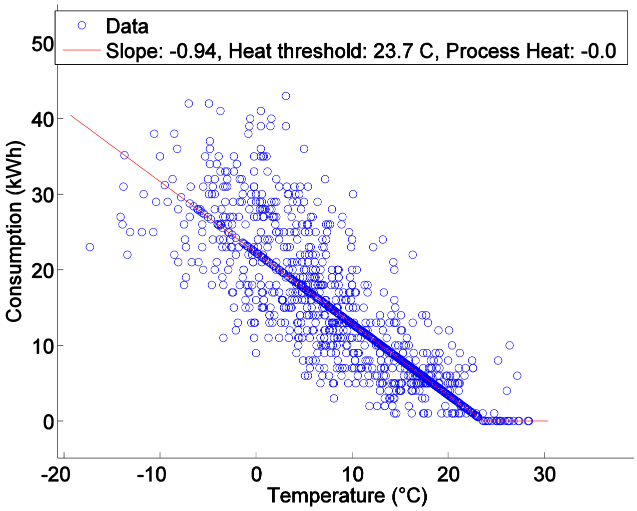


Figure 3-8 Scatter plot with bad readings replaced by calculated readings

The fitted heat curve can now be used to calculate W_T for each flagged data point. Afterwards, a direct comparison between raw and cleansed data can be examined. Figure 3-9 illustrates a magnified example of gas readings after applying the outlined methodology. In this example, the outlier (coloured blue) is replaced with a corrected consumption value (coloured green).

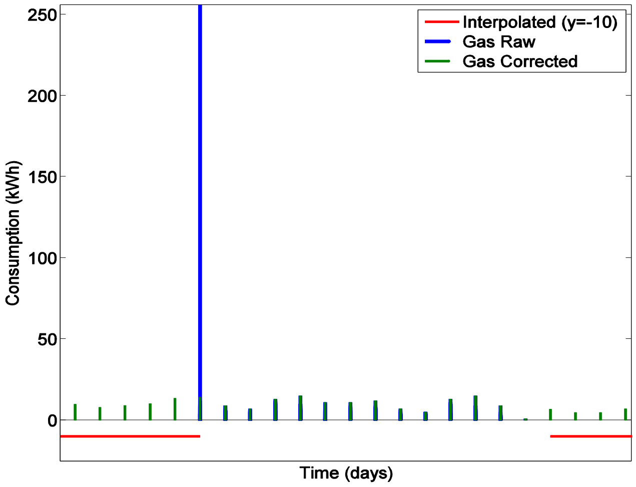


Figure 3-9 Magnified comparisons between raw and cleansed gas consumption

Figure 3-10 depicts the complete data set for the overall time period. All outliers and gaps which were present in Figure 3-4 were cleansed and interpolated.

Note: In both figures (Figure 3-9 and Figure 3-10), the interpolated and cleansed sections are highlighted in red, placed below the x-axis at -10 for a better readability.

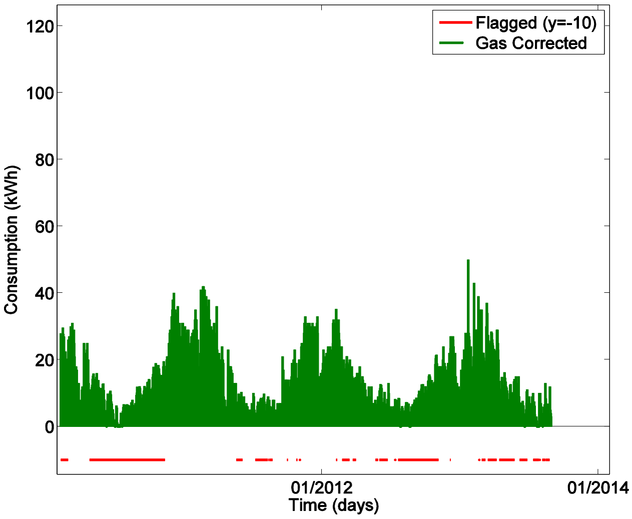


Figure 3-10 Cleansed data output after applying the outlined methodology

A data summary for the example used to discuss this methodology can be seen in Table 16. It is interesting to note that only few outliers were responsible for the high spikes in a graphical presentation.

Table 16 Data quality summary

Criteria	*Days*
Period	1295
Good data	702
Missing data	560
Outliers	33

As the proposed methodology interpolates bad or missing data from a load curve through corresponding temperature readings, it may be exploited to forecast gas consumption for the same period where forecasted weather data is available. When temperature data from a weather forecast is fed into the algorithm, it will output the estimated gas consumption for the corresponding period. This becomes particularly interesting for a demand based procurement of energy or load balancing algorithms (Simonis, 2013). Additionally, it may support building operators and facility managers in maintaining sustainable systems operation.

3.2.3 Accuracy of the cleansing algorithm

The actual accuracy of the algorithm was tested with a complete data set where sections of data were removed and then interpolated. The complete data set can be seen in Figure 3-11. Here, the recorded gas consumption is blue and the corresponding outside temperature is green.

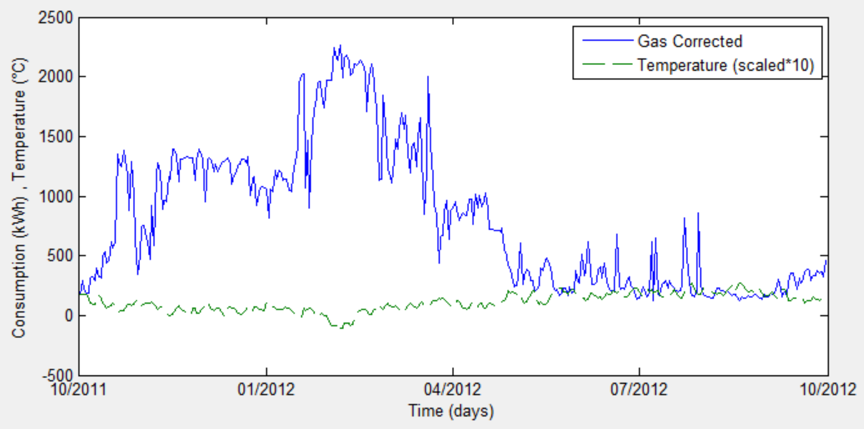


Figure 3-11 100% complete data set

The completeness of the data was reduced significantly to evaluate the efficiency of the interpolation algorithm. The outcome of the regression analysis is shown in Table 17. With a data completeness of just 1%, minor deviations can be noticed.

Table 17 Interpolation accuracy

Completeness [%]	Slope	Heat threshold [°C]	Base load [kWh]
100	-81.52	16.4	150.0
10	-72.62	16.4	201.0
1	-66.77	15.4	145.0
0.1	-106.36	15.2	0

Figure 3-12 provides a graphical comparison for the different levels of completeness. In this figure, the blue curves represent the interpolated gas consumption. The percentages next to the graph indicate the level of completeness in percent. With 10% of data available, the load curve shows strong similarities to the complete curve from Figure 3-11. With 1% of data available, especially the lower consumption areas lose their resemblance.

When only 0.1% of data is available, most similarities to Figure 3-11 are gone. Instead, the curve looks very similar to the (mirrored) outside temperature. The similarity to the outside temperature increases with less data available. As replacement values are calculated based on the heat curve, this resemblance can be explained.

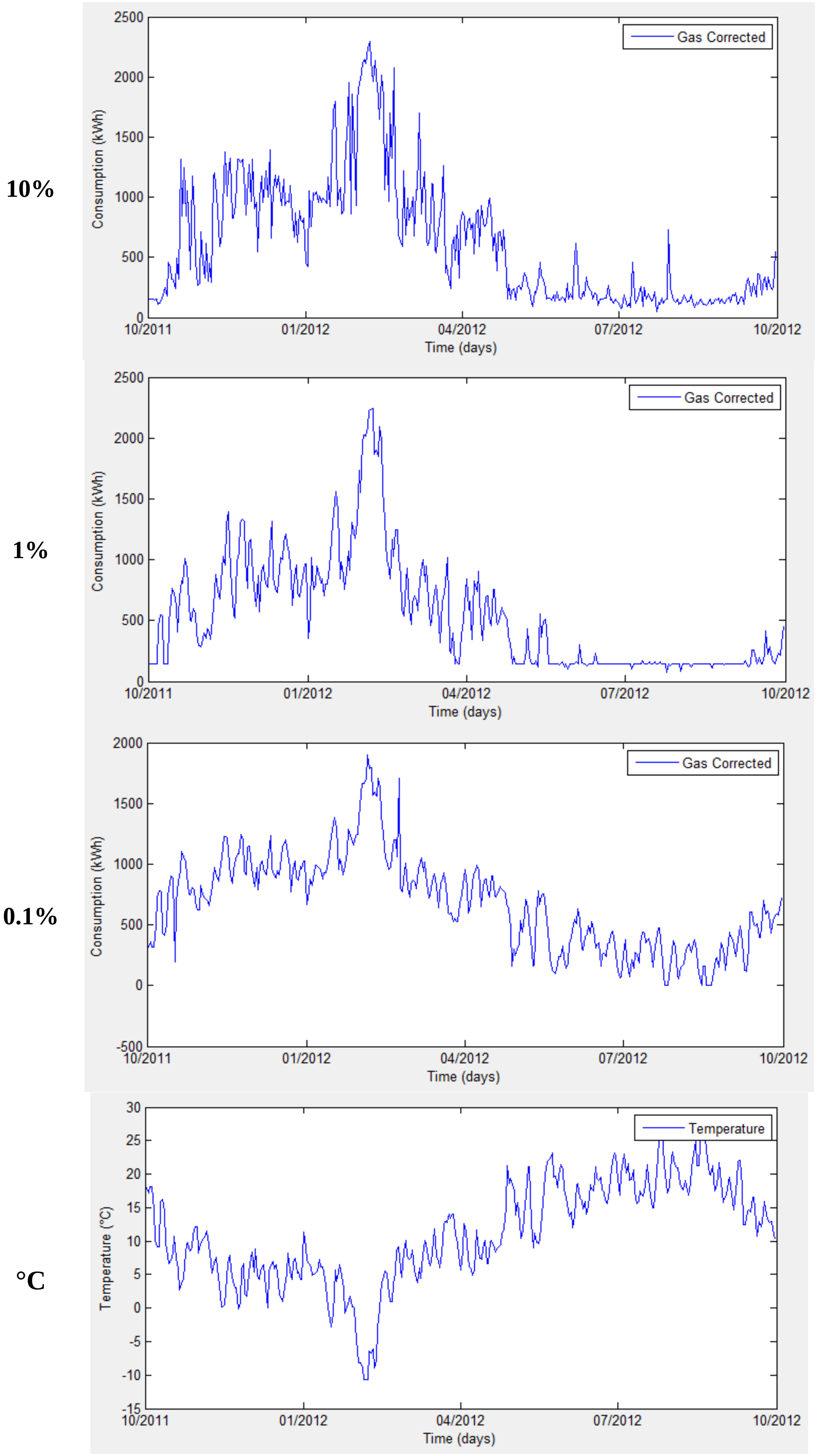


Figure 3-12 Graphical completeness comparison

3.2.4 Added value of data cleansing

The first step in undertaking an energy analysis is usually a verification of the data and its quality. Unfortunately, data is often flawed and needs manual cleansing. This kind of cleansing is tedious and time consuming. Ignoring the defects in the data leads to false analysis results, therefore the cleansing is a necessary step for any energy analysis. With the unique algorithm introduced in this chapter, energy experts can start their analysis immediately. This is because flaws and gaps are identified and corrected with automatic replacement values. By providing this methodology, energy analysis can be done more efficient in less time. This is especially important for energy analysis which is done on a frequent base. The availability of an automated cleansing algorithm provides a key benefit for competitiveness.

3.3 BIM data acquisition

This part of the research demonstrates how to extract descriptive information from BIM models in order to classify and structure monitored data. This is achieved through the exploitation of the open BIM standard IFC. The approach in this thesis extracts relevant information from the open BIM format IFC. The benefit of reusing selected objects from BIM models over existing solutions is the elimination of repetitive work. Currently, descriptive data often is fed into multiple systems manually (Willocks et al., 2015). This work demonstrates that data extracted from BIM models can enrich data analysis. It therefore eliminates the need to populate static information in a DWH through other means.

In order to enrich building performance data, only a manageable subset of elements described in the IFC metadata model is extracted. Four domains were identified in (Menzel et al., 2014) that can greatly increase interpretability of historical readings such as (i) spatial data, (ii) organizational data, (iii) building services systems and (iv) time data:

3.3.1 Spatial

The spatial dimension allows a lookup of spaces, floors, buildings and sites (i.e. groups of buildings). It will maintain relations and hierarchies, e.g. which room belongs to what building and what building floor. From a facility management perspective, this dimension will be helpful e.g. to identify which room temperature violates a SLA agreement negotiated with FM operators or to calculate the energy consumption per m².

3.3.2 Organization

The organization dimension enables the allocation of KPIs to groups of users. A large company can be broken down into a group, organization, business unit or single department. Combined with energy consumption, the organization dimension allows allocating consumption cost to aforementioned hierarchically structured units. It can also help the tenant to allocate consumption to its cost centres. Moreover, building owners could quantify cost of their assets. For building operators, the combination with the following dimension (System) could estimate the quality of the building operation per organization.

3.3.3 System

A system can be a network of elements from mechanical, electrical, and plumbing (MEP). It may be used to supply and control a distribution medium within a building. The system dimension can be used to allocate energy consumption to individual consumers, e.g. a gas consumer or a heating circuit for an area. Once a building has multiple heating circuits, this dimension allows analysing their performance individually. In combination with the previous dimension (Organization), energy cost can be mapped to individual tenants or business units.

3.3.4 Time

The time dimension is already existent in standard EnMS. When monitored data is visualised, it is usually realised in form of a graph were the x-axis represents time. By combining the first three dimensions with the time dimension, further analysis of the same data set becomes available.

These four domains were identified as foundation for KPI’s used for building performance evaluation. Specifically, the IFC objects enumerated in Table 18 will be of interest.

Table 18 IFC objects of interest

Domain	IFC Objects	IFC Relationship Objects
Spatial	IfcSpace, IfcBuildingStorey, IfcBuilding, IfcSite,	IfcRelAggregates, IfcRelContainedInSpatialStructure
Organization	IfcOrganization,	IfcOrganizationRelationship
System	IfcDistributionSystem, IfcDistributionCircuit, IfcDistributionFlowElement	IfcRelAggregates, IfcRelAssignsToGroup IfcRelFlowControlElement
Time	IfcDateTimeResource entities, IfcTimeSeries	IfcResourceLevelRelationship

While BIM provides functionality to model and export only certain building domains, it is not required to perform filtering for objects in BIM. Instead, filtering can be achieved with the following method.

To obtain the objects of interest, multiple parsing algorithms were designed to extract selected objects from IFC STEP files (Stapleton, 2014), (Hoerster, 2015). As IFC is standardised, the algorithm will extract the required objects from any IFC model saved in a STEP file format. Extracting the IFC objects from Table 18 will shrink down the content of the STEP file significantly. In this work standard UNIX shell commands for parsing and filtering were used. Through these shell commands, all objects of the four domains as in Table 18 can be filtered.

The feasibility of the outlined approach is evaluated with a building model created using a commercial BIM tool. An IFC STEP file can be generated through any BIM modelling software that supports the IFC standard. Popular software suitable for the export is e.g. Autodesk Revit. A sample BIM model from a university building exported to STEP revealed around 240.000 IFC objects in a total file size of 110 MB. The extracted objects from this file results in a file size less than 1 MB. The following figures illustrate the extraction process. Figure 3-13 depicts a floor plan of a building model. Figure 3-14 lists the first lines of the corresponding STEP file. Lastly, Listing 5 highlights a few filtered objects.

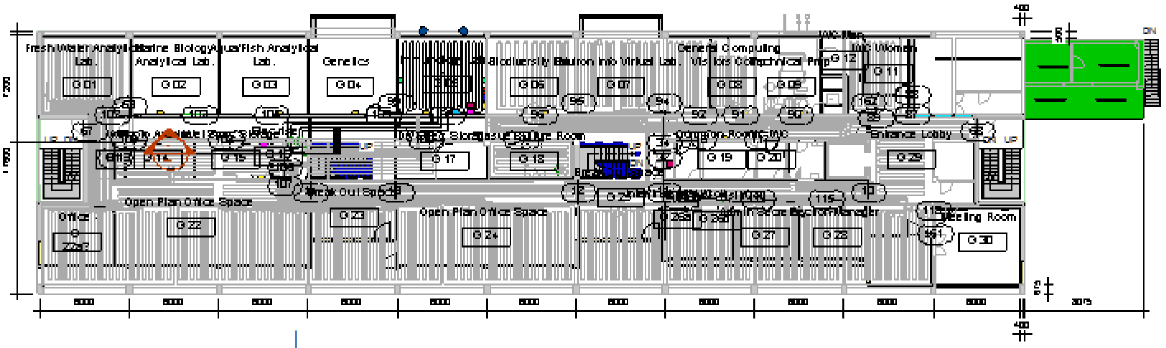
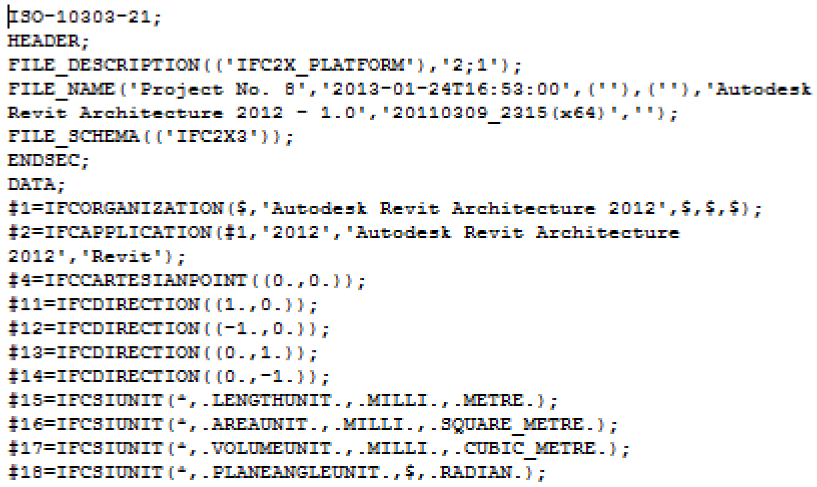


Figure 3-13 Floor plan as part of a BIM model in Autodesk Revit



```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('IFC2X_PLATFORM'),'2;1');
FILE_NAME('Project No. 8','2013-01-24T16:53:00',(''),(''),'Autodesk
Revit Architecture 2012 - 1.0','20110309_2315(x64)','');
FILE_SCHEMA(('IFC2X3'));
ENDSEC;
DATA;
#1=IFCORGANIZATION($,'Autodesk Revit Architecture 2012',$,$,$);
#2=IFCAPPLICATION(#1,'2012','Autodesk Revit Architecture
2012','Revit');
#4=IFCCARTESIANPOINT((0.,0.));
#11=IFCDIRECTION((1.,0.));
#12=IFCDIRECTION((-1.,0.));
#13=IFCDIRECTION((0.,1.));
#14=IFCDIRECTION((0.,-1.));
#15=IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.);
#16=IFCSIUNIT(*,.AREAUNIT.,.MILLI.,.SQUARE_METRE.);
#17=IFCSIUNIT(*,.VOLUMEUNIT.,.MILLI.,.CUBIC_METRE.);
#18=IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
```

Figure 3-14 Revit Export to IFC STEP

Listing 5 Parsed IFC file

```
#674=IFCSPACE('0L7_BX_o9EAuBLIe8WOWq$',#31,'LG. 6','',$,#663,#673,'Plant
      Room',.ELEMENT.,.INTERNAL.,$);
#44=IFCBUILDINGSTOREY('1e$PJwy2v56gfAEvTJTTFd',#33,'Level 00
      FFL',$,$,#43,$,$,.ELEMENT.,9100.000000000446);
#48=IFCBUILDINGSTOREY('1e$PJwy2v56gfAEvTJT7Hb',#33,'Level 01
      FFL',$,$,#47,$,$,.ELEMENT.,12700.);
#42338=IFCRELAGGREGATES('1bqwiPKMrFwRuKoLUI5fMl',#33,$,$,#48,(#803));
#803=IFCSPACE('0L7_BX_o9EAuBLIe8WOWqD',#33,'1.23','',$,#785,#802,'Office
      ',.ELEMENT.,.INTERNAL.,$);
```

In this example, the line starts with #674 as it is the 674th object in the STEP file. IFCSPACE is the type of the object. All attributes of the selected IFC objects are extracted. Attributes visible in the BIM model can be found again, e.g. the GUID. After exporting the BIM model and parsing the IFC file, objects of interest, for example room "LG. 6" can be found in the filtered output. Its attributes are explained in Table 19.

Table 19 IfcSpace attributes explained

STEP value	IFC attribute name	Explanation
'0L7_BX_o9EAuBLIe8WOWq$'	GlobalId	A unique ID to distinguish IFC objects
#31	OwnerHistory	A reference to IFC object 31
'LG. 6'	Name	The name of the room
''	Description	No further description has been entered by the building model creator
$	ObjectType	The object type has not been defined by the creator
#663	ObjectPlacement	The placement of the object is linked to another IFC object
#673	Representation	The representation is also linked to a different object
'Plant Room'	LongName	A long name for the room
.ELEMENT.	CompositionType	An enumeration type to describe the composition of the IFC object
.INTERNAL.	InteriorOrExteriorSpace	Another enumeration type to further define the physical location of the object
$	ElevationWithFlooring	Another unset attribute that could be used to set the elevation of the room

3.3.5 Added value of the BIM data acquisition

The purpose of the introduced methodology for data acquisition is to eliminate repetitive steps and to provide data of accurate quality. By utilising BIM, large amounts of information are stored within the model. In comparison, CAD provided very little extra information. By extracting data from BIM, digitally available information can be reused. The tedious work to reacquire specifics to aid in data analysis is therefore eradicated. Therefore, this method saves time that is usually spent on acquisition of the necessary information. Additionally, the data provided by BIM can be considered of high quality as BIM acts as source for many professions. The method from this chapter provides automated extraction of selected information useful for data analysis. By employing the open BIM standard IFC, all necessary information gets processed in a consistent way.

3.4 Database and Data Warehouse design

To maintain a maximum compatibility with the IFC file standard, it was decided to design a database schema which follows the IFC object definitions (Cahill, 2012) (Flynn, 2012). In this approach, tables are named after their corresponding IFC objects and the table columns match the attributes of the IFC objects. Instances of individual IFC objects are stored in dedicated tables (Hoerster et al., 2012).

Similar to relations in IFC, relations between tables are realised through relationship tables which follow the definition of IFC. Relationship objects of same type are consolidated in a single relationship table. The feasibility of this approach was evaluated as part of this thesis.

Figure 3-15 depicts how objects are linked to each other following this concept. Through resolving their hierarchical relationship, matching of a room to a floor and to a building becomes possible. Figure 3-16 highlights that in a database approach all instances of IfcRelAggregates are merged into a single table which maintains the relation of all related IFC entities.

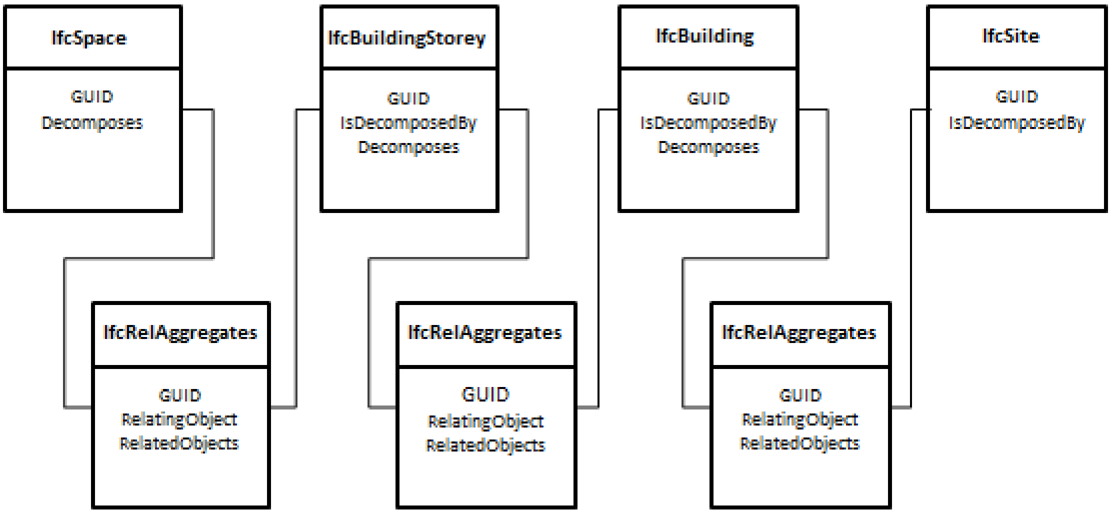


Figure 3-15 File based relation

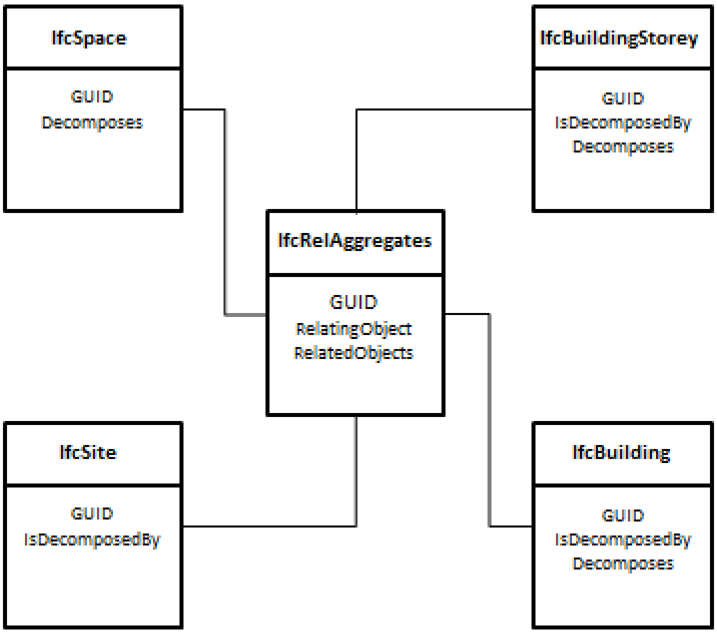


Figure 3-16 Database relation

In order to maximise compatibility between individual IFC elements and a database schema, it needs to support the attribute types found in the IFC meta data schema. Table 20 identifies the various data types available in the EXPRESS meta model.

Table 20 Overview of EXPRESS data types

EXPRESS attribute type	Description	Example
“Linked IFC object”	Relationship to another instance of an IFC object	#1337
STRING	Sequence of up to 255 characters	‘Boiler’
REAL	Decimal number	14.74
ENUMERATION	Select matching attributes from a list	‘red’, ’green’, blue’
SELECT	Select matching objects from a list	‘IfcSpace’, IfcBuilding’
INTEGER	A whole number without fraction	187
NUMBER	Could be either an integer or a real	1405
BOOLEAN	Either 1 or 0	1
LISTS	A collection of attributes, e.g. measurements	13, 11, 82

As Table 21 illustrates, database vendors have their own implementation of data types. This makes a generic mapping from STEP to SQL impossible. Therefore, it shows how the STEP data types could be mapped in three exemplary database environments.

Table 21 STEP attribute mapping in different databases

EXPRESS/STEP attribute type	*Microsoft SQL*	*Oracle*	*MySQL*
"Linked IFC object"	Int	Number	Int
STRING	Varchar	Varchar	Varchar
REAL	Float	Number	Float
ENUMERATION	Lookup table	Enum	Enum
SELECT	Lookup table	Enum	Enum
INTEGER	Int	Number	Int
NUMBER	Float	Number	Float
BOOLEAN	Bit/Int	Byte/Number	Int
LISTS	Lookup table	Lookup table	Lookup table

All cells which contain "Lookup table" require a solution where a separate lookup table with possible values is designed. Their values are primary keys which are accessed through foreign keys. For the STEP attributes ENUMERATION and SELECT, the selectable elements are predefined. This functionality is supported in Oracle and MySQL. However, the STEP attribute type LISTS is a random collection of attributes which cannot be foreseen. Therefore, all database systems will need to handle this attribute type dynamically through lookup tables.

The adaption of IFC relationship objects into a database schema is a challenge as IFC realises relationships between its entities through relationship objects. These objects maintain how objects are referenced by others. The IFC metadata model allows an unlimited number of referenced objects. This flexibility becomes problematic in a database schema where numbers of columns are static. During this research, four possibilities for implementation were identified but only one is suitable for implementation. The following section discusses these possibilities:

Option A:

Relating Object	Related Objects
PK_1	FK_1, FK_2, FK_3,

N objects get saved in a column holding all referencing objects. The fact that a set of elements shares a single column violates the first normal form of a relational database. As a result, no foreign key – primary key relationships may be created as the column holding the related object may contain more than one element. Additionally, parsing of the information stored in the column proves itself to be difficult.

Option B:

Relating Object	Related Object 1	Related Object 2	…	Related Object N
PK_1	$FK_{1,1}$	$FK_{1,2}$	…	$FK_{1,N}$
PK_2	$FK_{2,1}$	$FK_{2,2}$	…	$FK_{2,N}$

Create N columns for N objects. Since the actual number N is unknown, a database schema has to dynamically adapt to whatever information comes from the IFC file. A mutating schema is hard to optimise as Data Warehouse techniques or data indexing require predefined table layouts. Moreover, the schema is challenging to administrate as maintenance, debugging and optimising need constant adoption.

Option C:

Relating Object	Related Objects
PK_1	$FK_{1,1}$
PK_1	$FK_{1,2}$
PK_1	$FK_{1,3}$
PK_2	$FK_{2,1}$
PK_2	$FK_{2,2}$

Storing each related object in a separate row along with its relating object. This approach would allow the implementation of foreign keys for all related objects. This type of database design is often called Entity-Attribute-Value (EAV) which is used for highly heterogeneous data. Nadkarni et al. state that EAV schemas are less efficient and require significant custom programming (Nadkarni, 1999).

Option D:

IFC GUID	Relating Object	Related Objects
$GUID_1$	PK_1	$FK_{1,1}$
$GUID_2$	PK_1	$FK_{1,2}$
$GUID_3$	PK_1	$FK_{1,3}$
$GUID_4$	PK_2	$FK_{2,1}$
$GUID_5$	PK_2	$FK_{2,2}$

This solution adds the IFC GUID to the database schema. As a GUID is unique, it can serve as the primary key of the table. Most importantly, the GUID enables bi-directional lookups since it enables related objects to find their relating objects. This is a principle of the IFC standard and with this option, this principle can be maintained. Figure 3-17 provides a graphical explanation for the bi-directional relationship of option D.

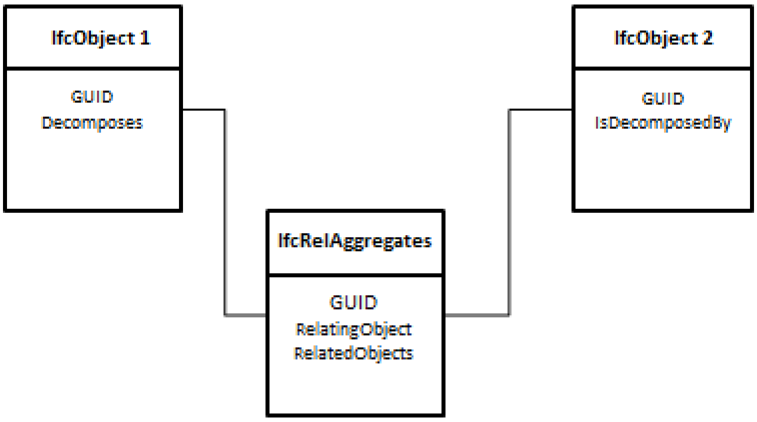


Figure 3-17 Bi-directional relationships in database design

The only viable solutions for a relational database schema are options C and D. Options A and B do not meet the requirements as they are not in the third normal form which insures integrity in table relationships. While option C would fulfil the requirements of the relational model, its relationship table does not support modelling of bi-directional relationships. By maintaining a GUID defined in an IfcRelAggregates object as seen in option D, database constraints between tables can be set up. Therefore, option D is the only viable solution.

In IFC, relationship objects are inherited from IfcRoot, therefore they already possess a GUID. This GUID may then be reused to retain consistency between the IFC file and the IFC database schema. An additional benefit from the inherited IfcRoot is, that besides the GUID also the OwnerHistory can be maintained. This allows to log for each IFC object, who has committed any changes.

In some cases, this 1:1 mapping of elements into a database schema is not possible. For example, some IFC attribute names are keywords in SQL and cannot be used as column names, e.g. “precision” or “outer”. Furthermore, some IFC attributes have longer names than the defined maximum character count for a column name, e.g. Oracle has defined a maximum of 30 characters for column names (Oracle, 2015). For these cases, workarounds need to be defined. This could be either renaming or shortening offending column names.

The overall database schema design was implemented under my supervision in a recent Master thesis where it is explained in detail (Mo, 2012). The aim of the student’s thesis was the development of a database schema which contains entities fully compatible with IFC objects. Besides afore mentioned naming convention constraints, the new schema unified object attributes and names.

The following section will detail the dimension creation process in order to support the introduced four domains.

3.4.1 Data Warehouse Design

The DWH acts as a central data repository. It is an essential part of this thesis. It consolidates information spread across multiple data sources and continuously keeps KPI’s up to date. For this to work, a subset of elements described in the BIM model is required. Data extracted from IFC metadata objects is categorised in dimensions, while sensed and metered data is considered as fact data.

The consolidated information per dimension will be stored in Materialized Views (MV). Their advantage is that they can be populated with data stored across multiple tables while excess information from the source tables can be omitted. In addition, MV’s support automated refreshes of their aggregated data to ensure that content stays in sync with the individual source tables it references. Moreover, during its processing, selected pre-calculations can be made which minimise processing time when accessing the data. This is especially helpful when the source tables contain millions of data sets. The schematic in Figure 3-18 illustrates an example for a MV.

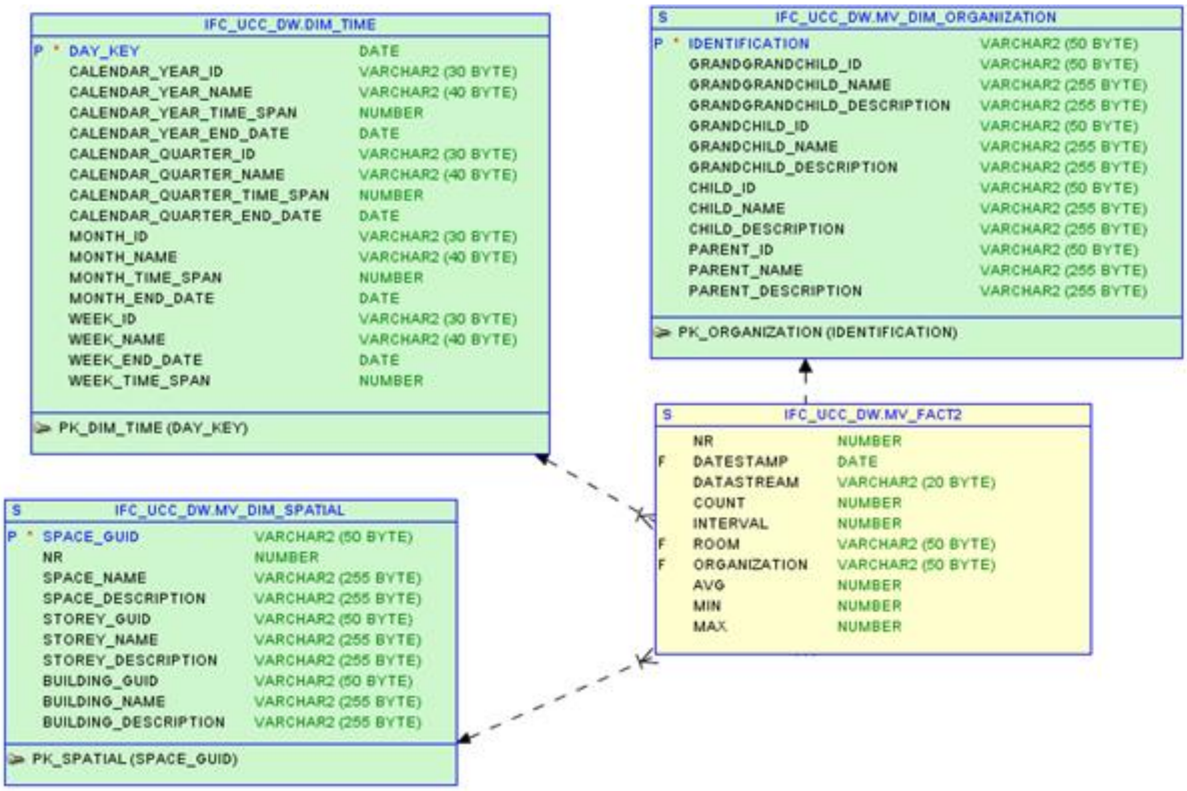

IFC_UCC_DW.DIM_TIME
P * DAY_KEY DATE
CALENDAR_YEAR_ID VARCHAR2 (30 BYTE)
CALENDAR_YEAR_NAME VARCHAR2 (40 BYTE)
CALENDAR_YEAR_TIME_SPAN NUMBER
CALENDAR_YEAR_END_DATE DATE
CALENDAR_QUARTER_ID VARCHAR2 (30 BYTE)
CALENDAR_QUARTER_NAME VARCHAR2 (40 BYTE)
CALENDAR_QUARTER_TIME_SPAN NUMBER
CALENDAR_QUARTER_END_DATE DATE
MONTH_ID VARCHAR2 (30 BYTE)
MONTH_NAME VARCHAR2 (40 BYTE)
MONTH_TIME_SPAN NUMBER
MONTH_END_DATE DATE
WEEK_ID VARCHAR2 (30 BYTE)
WEEK_NAME VARCHAR2 (40 BYTE)
WEEK_END_DATE DATE
WEEK_TIME_SPAN NUMBER
PK_DIM_TIME (DAY_KEY)
S IFC_UCC_DW.MV_DIM_ORGANIZATION
P * IDENTIFICATION VARCHAR2 (50 BYTE)
GRANDGRANDCHILD_ID VARCHAR2 (50 BYTE)
GRANDGRANDCHILD_NAME VARCHAR2 (255 BYTE)
GRANDGRANDCHILD_DESCRIPTION VARCHAR2 (255 BYTE)
GRANDCHILD_ID VARCHAR2 (50 BYTE)
GRANDCHILD_NAME VARCHAR2 (255 BYTE)
GRANDCHILD_DESCRIPTION VARCHAR2 (255 BYTE)
CHILD_ID VARCHAR2 (50 BYTE)
CHILD_NAME VARCHAR2 (255 BYTE)
CHILD_DESCRIPTION VARCHAR2 (255 BYTE)
PARENT_ID VARCHAR2 (50 BYTE)
PARENT_NAME VARCHAR2 (255 BYTE)
PARENT_DESCRIPTION VARCHAR2 (255 BYTE)
PK_ORGANIZATION (IDENTIFICATION)
S IFC_UCC_DW.MV_FACT2
NR NUMBER
F DATESTAMP DATE
DATASTREAM VARCHAR2 (20 BYTE)
COUNT NUMBER
INTERVAL NUMBER
F ROOM VARCHAR2 (50 BYTE)
F ORGANIZATION VARCHAR2 (50 BYTE)
AVG NUMBER
MIN NUMBER
MAX NUMBER
S IFC_UCC_DW.MV_DIM_SPATIAL
P * SPACE_GUID VARCHAR2 (50 BYTE)
NR NUMBER
SPACE_NAME VARCHAR2 (255 BYTE)
SPACE_DESCRIPTION VARCHAR2 (255 BYTE)
STOREY_GUID VARCHAR2 (50 BYTE)
STOREY_NAME VARCHAR2 (255 BYTE)
STOREY_DESCRIPTION VARCHAR2 (255 BYTE)
BUILDING_GUID VARCHAR2 (50 BYTE)
BUILDING_NAME VARCHAR2 (255 BYTE)
BUILDING_DESCRIPTION VARCHAR2 (255 BYTE)
PK_SPATIAL (SPACE_GUID)

Figure 3-18 Schematic example for a Materialized View

In this figure, selected data from three different tables (green) is consolidated in a single MV (yellow). The MV has three foreign keys (F), each pointing to a green table’s primary key (P). During the MV creation, these raw readings got aggregated using standard database calculations: COUNT, AVG, MIN, MAX. These functions determine the number of elements, their total average, the minimum and maximum value. Additionally, an INTERVAL was calculated which estimates the sending interval of the meter/sensor in question. Information from three, previously independent tables, form together the content of the MV coloured in yellow. The MV from this example is therefore an aggregated fact table with pre-calculated values.

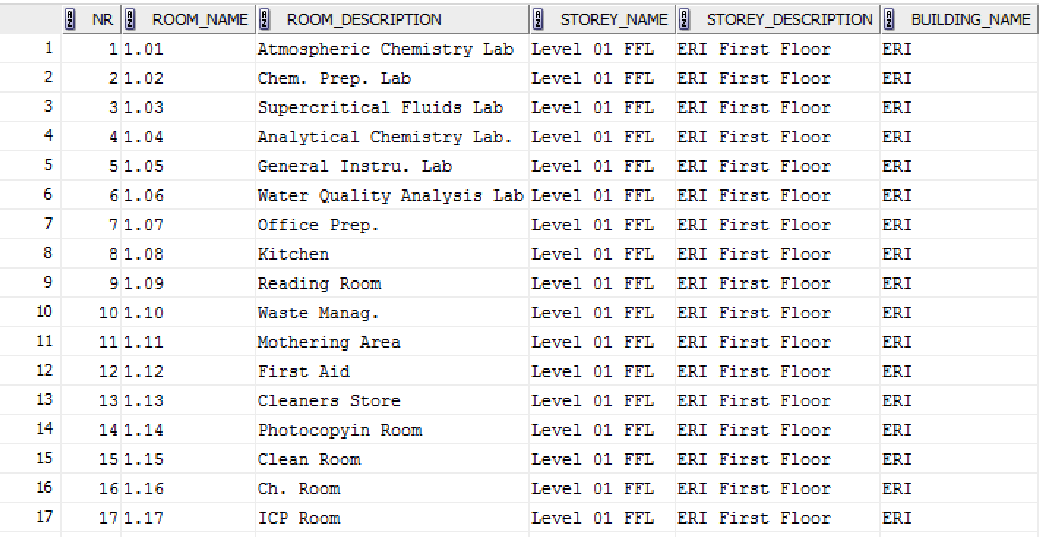
3.4.2 Data Warehouse dimensions

The development of the four introduced domains for building analysis will be discussed in this chapter. All four domains will be built on top of the IFC database schema. Their implementation will allow OLAP cubes to slice and dice through the fact data. Due to the architecture of IFC, the objects required for each dimension are split across several tables in the database schema. Therefore, the dimensions also consolidate this information in a single table (or view) which significantly increases readability. It was chosen to store this consolidated dimensional information in Materialized Views (MV). MV’s can be set up inside the DWH to refresh automatically. This ensures that information is always up to date. One requirement for the MV's to refresh is that the individual tables remain free of errors. A duplicate entry for example would break the MV refresh interval. These inconsistencies can be avoided through conscientious database schema design and usage of primary and foreign key relationships.

a) Spatial Dimension

This dimension consolidates information stored in the tables IfcSpace, IfcBuildingStorey, IfcBuilding, IfcSite and IfcRelAggregates.

This MV is realised through nested SQL commands. It builds the relationships between sites, buildings, storeys and spaces by resolving the IFC relationship tables IfcRelAggregates. Its output is shown in Figure 3-19.

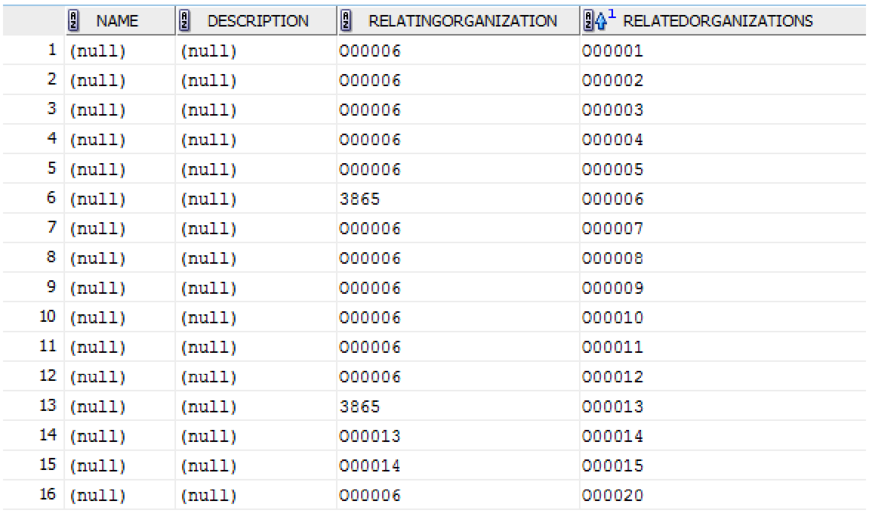


	NR	ROOM_NAME	ROOM_DESCRIPTION	STOREY_NAME	STOREY_DESCRIPTION	BUILDING_NAME
1	1	1.01	Atmospheric Chemistry Lab	Level 01 FFL	ERI First Floor	ERI
2	2	1.02	Chem. Prep. Lab	Level 01 FFL	ERI First Floor	ERI
3	3	1.03	Supercritical Fluids Lab	Level 01 FFL	ERI First Floor	ERI
4	4	1.04	Analytical Chemistry Lab.	Level 01 FFL	ERI First Floor	ERI
5	5	1.05	General Instru. Lab	Level 01 FFL	ERI First Floor	ERI
6	6	1.06	Water Quality Analysis Lab	Level 01 FFL	ERI First Floor	ERI
7	7	1.07	Office Prep.	Level 01 FFL	ERI First Floor	ERI
8	8	1.08	Kitchen	Level 01 FFL	ERI First Floor	ERI
9	9	1.09	Reading Room	Level 01 FFL	ERI First Floor	ERI
10	10	1.10	Waste Manag.	Level 01 FFL	ERI First Floor	ERI
11	11	1.11	Mothering Area	Level 01 FFL	ERI First Floor	ERI
12	12	1.12	First Aid	Level 01 FFL	ERI First Floor	ERI
13	13	1.13	Cleaners Store	Level 01 FFL	ERI First Floor	ERI
14	14	1.14	Photocopyin Room	Level 01 FFL	ERI First Floor	ERI
15	15	1.15	Clean Room	Level 01 FFL	ERI First Floor	ERI
16	16	1.16	Ch. Room	Level 01 FFL	ERI First Floor	ERI
17	17	1.17	ICP Room	Level 01 FFL	ERI First Floor	ERI

Figure 3-19 Materialized View of the spatial dimension

b) Organization Dimension

The implementation of the organization dimension involves only two IFC entities, namely IfcOrganization and IfcOrganizationRelationship. The realisation of the database dimension is challenging as it requires utilisation of recursive SQL. Specifically, the single table IfcOrganizationRelationship maintains parent-child relationships. Each child, however, could be a parent itself. Figure 3-20 shows an example of an IfcOrganizationRelationship table. Unlike in the spatial dimension, IFC does not hierarchically order parent-child relations in these two objects.



	NAME	DESCRIPTION	RELATINGORGANIZATION	RELATEDORGANIZATIONS
1	(null)	(null)	000006	000001
2	(null)	(null)	000006	000002
3	(null)	(null)	000006	000003
4	(null)	(null)	000006	000004
5	(null)	(null)	000006	000005
6	(null)	(null)	3865	000006
7	(null)	(null)	000006	000007
8	(null)	(null)	000006	000008
9	(null)	(null)	000006	000009
10	(null)	(null)	000006	000010
11	(null)	(null)	000006	000011
12	(null)	(null)	000006	000012
13	(null)	(null)	3865	000013
14	(null)	(null)	000013	000014
15	(null)	(null)	000014	000015
16	(null)	(null)	000006	000020

Figure 3-20 Exemplary IfcOrganizationRelationship table

A graphical representation for data sets of this type is illustrated in Figure 3-21. Each branch may or may not have further children.

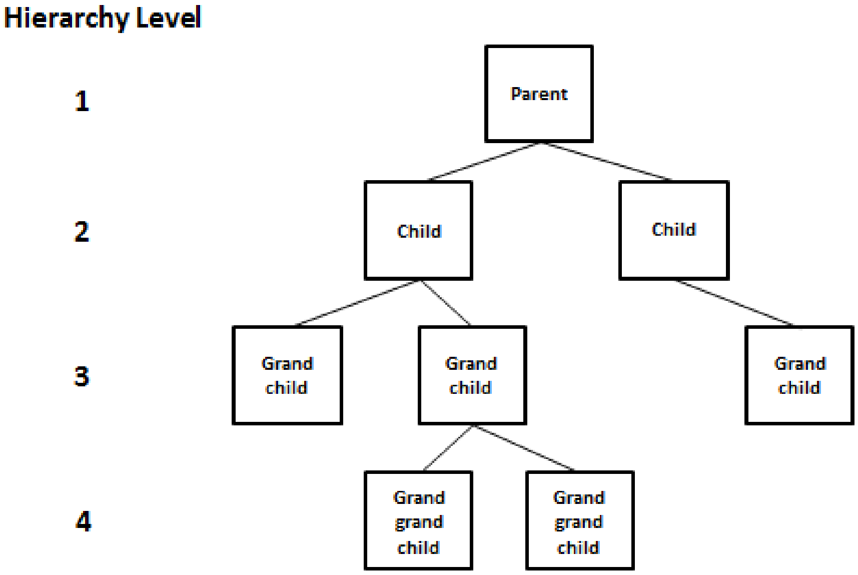


Figure 3-21 Hierarchy tree of a parent-child relationship

Recursive SQL allows the processing of a multi-tier hierarchy. Without resolving recursive relationships, any hierarchy would be limited to only two levels. The downside of recursive SQL functions is that its implementation requires vendor specific SQL statements. This means that a solution developed e.g. in Microsoft SQL will not work without modification in Oracle.

The following algorithm is implemented in Oracle. Adaptions will be required if porting to different vendors becomes necessary. The main principle, however, should stay the same.

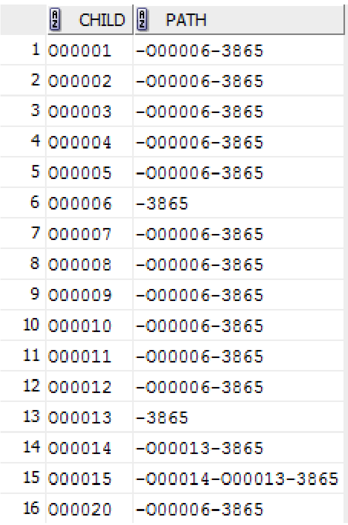
Description of the recursive algorithm:

1. The recursive algorithm works against the GUIDs stored in the table IfcOrganizationRelationship. This table stores the parent as RELATINGORGANIZATION. Each child is stored in a column called RELATEDORGANIZATIONS.

2. The system function CONNECT BY PRIOR is used in combination with CONNECT_BY_ROOT to identify all parents for each child. This identifies all leaf objects.

3. The system function CONNECT_BY_ISLEAF is utilised to filter out records that have further child records. It outputs a pseudo column that tags each row with binary 0 or 1. Through this functionality, it is possible to identify the youngest element in each chain. Objects, which have both parent and child are intermediate leaf objects and can therefore be filtered out.

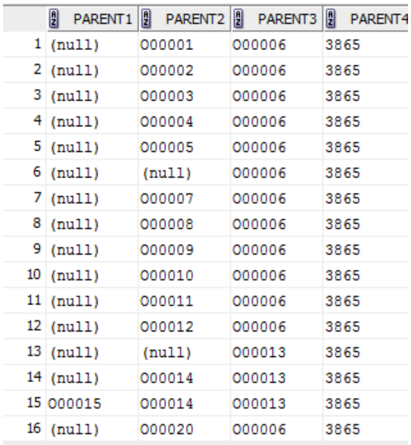
4. The system function SYS_CONNECT_BY_PATH spans for each child the full hierarchical path. Therefore, for each leaf objects the path of inheritance is determined. The output is stored in a single column with values separated by definable character. Figure 3-22 shows an excerpt where the youngest child is placed in the first column and all ancestors are stored sequentially in column two. In this representation, the hierarchy of the organizations can already be seen. The figure reveals that e.g. O00013 has only one parent (3865), while O00015 has three parents (O00014, O00013, 3865).



	CHILD	PATH
1	O00001	-O00006-3865
2	O00002	-O00006-3865
3	O00003	-O00006-3865
4	O00004	-O00006-3865
5	O00005	-O00006-3865
6	O00006	-3865
7	O00007	-O00006-3865
8	O00008	-O00006-3865
9	O00009	-O00006-3865
10	O00010	-O00006-3865
11	O00011	-O00006-3865
12	O00012	-O00006-3865
13	O00013	-3865
14	O00014	-O00013-3865
15	O00015	-O00014-O00013-3865
16	O00020	-O00006-3865

Figure 3-22 Intermediate step in creating an organization dimension

5. It is desired to not have multiple values in a single column; therefore each hierarchy level is stored in its own column. The data stored by SYS_CONNECT_BY_PATH is therefore split into multiple columns. This is realised through regular expressions nested in SQL commands which split the string at each position where the defined character appears. The regular expression functions used are REGEXP_SUBSTR to filter for substrings, and REGEXP_COUNT to count string occurrences. The intermediate output is given in Figure 3-23. Here, PARENT1 represents the youngest child and PARENT4 is the oldest parent. In this example, only row 15 consists out of four family members.

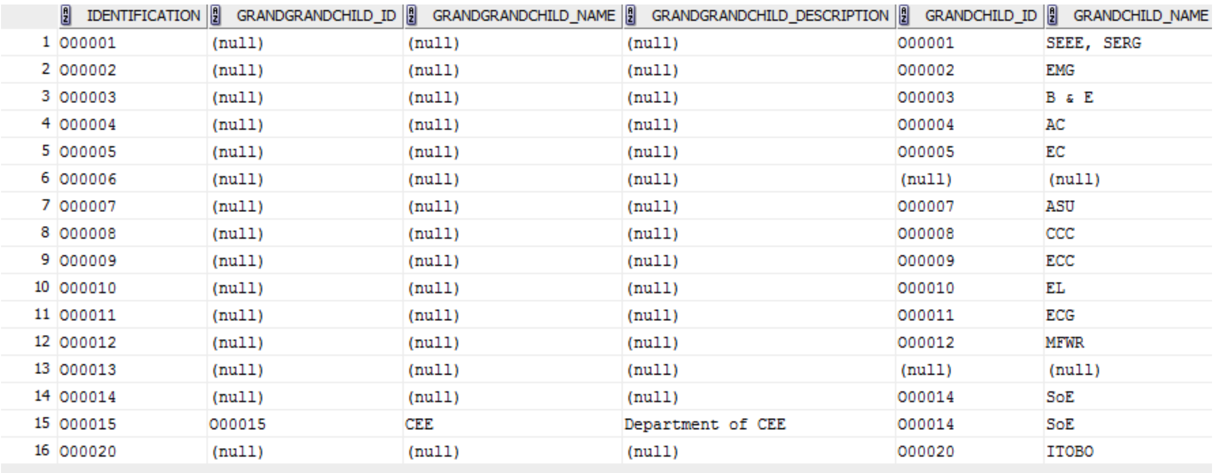


	PARENT1	PARENT2	PARENT3	PARENT4
1	(null)	000001	000006	3865
2	(null)	000002	000006	3865
3	(null)	000003	000006	3865
4	(null)	000004	000006	3865
5	(null)	000005	000006	3865
6	(null)	(null)	000006	3865
7	(null)	000007	000006	3865
8	(null)	000008	000006	3865
9	(null)	000009	000006	3865
10	(null)	000010	000006	3865
11	(null)	000011	000006	3865
12	(null)	000012	000006	3865
13	(null)	(null)	000013	3865
14	(null)	000014	000013	3865
15	000015	000014	000013	3865
16	(null)	000020	000006	3865

Figure 3-23 Organization dimension categorised by parent hierarchy

6. While not all parent-child hierarchies might consist out of all layers found, any record with less than the maximum hierarchy level will still have the additional columns. Cubes are not compatible with NULL values in MV's (Alexei, 2006). Therefore, the database keyword NULL is replaced with "N/A" to avoid incompatibility.

7. Finally, the output gets left joined with the information stored in IfcOrganization. This populates the output with all information needed for the organization dimension. The final output will look similar to Figure 3-24. Similarly to the previous dimensions, redundant information is stored in the dimension (see chapter 2.4.4)



	IDENTIFICATION	GRANDGRANDCHILD_ID	GRANDGRANDCHILD_NAME	GRANDGRANDCHILD_DESCRIPTION	GRANDCHILD_ID	GRANDCHILD_NAME
1	000001	(null)	(null)	(null)	000001	SEEE, SERG
2	000002	(null)	(null)	(null)	000002	EMG
3	000003	(null)	(null)	(null)	000003	B & E
4	000004	(null)	(null)	(null)	000004	AC
5	000005	(null)	(null)	(null)	000005	EC
6	000006	(null)	(null)	(null)	(null)	(null)
7	000007	(null)	(null)	(null)	000007	ASU
8	000008	(null)	(null)	(null)	000008	CCC
9	000009	(null)	(null)	(null)	000009	ECC
10	000010	(null)	(null)	(null)	000010	EL
11	000011	(null)	(null)	(null)	000011	ECG
12	000012	(null)	(null)	(null)	000012	MFWR
13	000013	(null)	(null)	(null)	(null)	(null)
14	000014	(null)	(null)	(null)	000014	SoE
15	000015	000015	CEE	Department of CEE	000014	SoE
16	000020	(null)	(null)	(null)	000020	ITOBO

Figure 3-24 Organization Dimension

c) Systems Dimension

In this dimension, elements from the following IFC entities are combined: IfcDistributionSystem, IfcDistributionCircuit, IfcDistributionFlowElement, IfcRelAggregates, and IfcRelAssignsToGroup. Figure 3-25 shows schematically how these entities interact with each other in the IFC meta model. One IfcDistributionSystem can have multiple IfcDistributionCircuit and these can be assembled out of a group of IfcDistributionFlowElement's. The relationship between IfcDistributionCircuit and IfcDistributionFlowElement is realised by the relational object IfcRelAssignsToGroup. This might look contrary to previous figures but is accounted to their inheritances. IfcDistributionFlowElement is a sub class if IfcProduct while IfcDistributionSystem and IfcDistributionCircuit are sub classes of IfcObject. Objects are related by IfcRelAggregates and products are linked to objects through IfcRelAssignsToGroup.

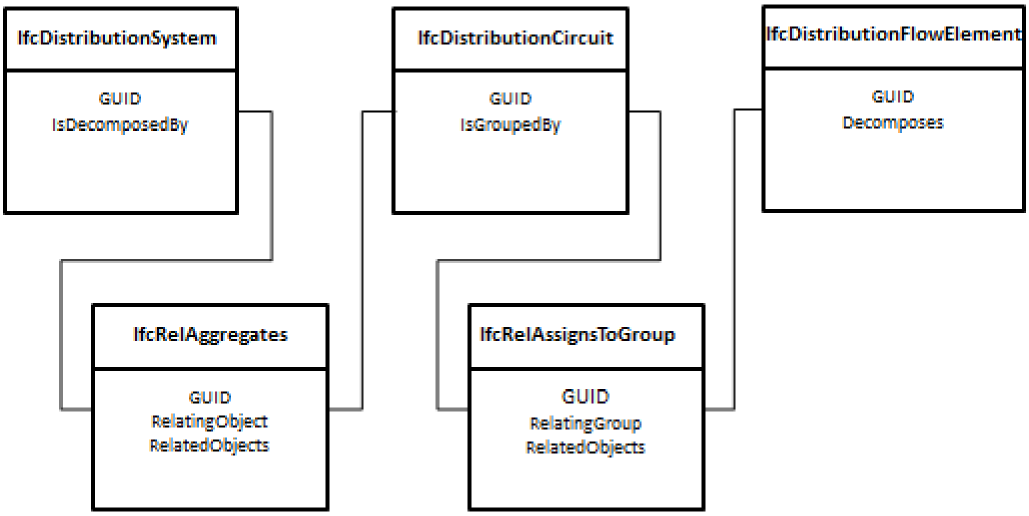
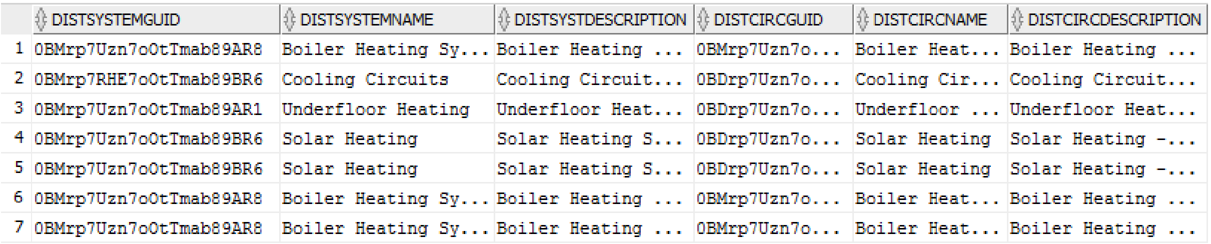


Figure 3-25 Systems hierarchy in IFC

The process of building a MV is similar to the spatial dimension. Again, relations between the instances are resolved and the output is stored in a single dimensional table. Exemplary output can be seen in Figure 3-26.



	DISTSYSTEMGUID	DISTSYSTEMNAME	DISTSYSTDESCRIPTION	DISTCIRCGUID	DISTCIRCNAME	DISTCIRCDESCRIPTION
1	0BMrp7Uzn7o0tTmab89AR8	Boiler Heating Sy...	Boiler Heating ...	0BMrp7Uzn7o...	Boiler Heat...	Boiler Heating ...
2	0BMrp7RHE7o0tTmab89BR6	Cooling Circuits	Cooling Circuit...	0BDrp7Uzn7o...	Cooling Cir...	Cooling Circuit...
3	0BMrp7Uzn7o0tTmab89AR1	Underfloor Heating	Underfloor Heat...	0BDrp7Uzn7o...	Underfloor ...	Underfloor Heat...
4	0BMrp7Uzn7o0tTmab89BR6	Solar Heating	Solar Heating S...	0BDrp7Uzn7o...	Solar Heating	Solar Heating -...
5	0BMrp7Uzn7o0tTmab89BR6	Solar Heating	Solar Heating S...	0BDrp7Uzn7o...	Solar Heating	Solar Heating -...
6	0BMrp7Uzn7o0tTmab89AR8	Boiler Heating Sy...	Boiler Heating ...	0BMrp7Uzn7o...	Boiler Heat...	Boiler Heating ...
7	0BMrp7Uzn7o0tTmab89AR8	Boiler Heating Sy...	Boiler Heating ...	0BMrp7Uzn7o...	Boiler Heat...	Boiler Heating ...

Figure 3-26 Materialized View system dimension

d) Time Dimension

Historical fact data is collected from meters and sensors throughout buildings and stored inside the DWH. IFC is supporting fact data through its IfcTimeSeries entity. Each record consists of a value, a timestamp and meter/sensor identification (ID). Through these attributes, it is possible to identify data source and reading at any time.

In order for a cube to operate time based slicing of performance data, a time dimension needs to be in place. A time dimension is a special kind of dimension that defines time periods of different intervals. Each defined interval will be a selectable granularity for DWH operations. During implementation it is important to define the intervals that should be available. So in order to e.g. select data by year and month, both yearly and monthly information has to be implemented in the time table. For each interval the table also needs to hold the end date of the interval and the time span in days. A detailed approach for the implementation of a time dimension can be found in (Mo et al., 2013).

As outlined in Table 18, IFC provides support for time elements through its IfcDateTimeResource entities. However, the BIM models available to the author did not store any time information. Therefore, no IFC time entities could be extracted. The lack of time information in the available models provides opportunities for future research where BIM models have been enriched with IFC time objects in afore mentioned objects.

The design frame for a time dimension is defined by the DWH vendors. The implementation for this methodology was realised through a PHP script which can be found in appendix 5.

3.4.3 Fact table

Building performance data (or fact data in database terms) is collected from monitoring devices throughout buildings and stored inside the DWH. IFC is supporting fact data through its IfcTimeSeries entity which is a supertype of IfcIrregularTimeSeries and IfcRegularTimeSeries. Here, IFC distinguishes between IfcTimeSeriesValue and IfcIrregularTimeSeriesValue. The first entity is based on readings which are acquired in a regular interval while the latter focuses on readings acquired irregularly. Regular readings could originate from devices which are pulled in predefined intervals while irregular readings could be from devices which push information on certain events. Each record in the IFC object consists out of a value, a timestamp and an unique identifier of the monitoring devices.

Table 22 presents the structure of the table IfcTimeSeriesValue based on the IFC object definition.

Table 22 Database table derived from the IfcTimeSeriesValue entity

Attribute Name	Attribute Type	Comment
READINGID	NUMBER	Unique reading ID
TIMESTAMP	TIMESTAMP(6)	Timestamp of measurement
VALUE	NUMBER	Recorded value
DIRECTION	VARCHAR2(20 BYTE)	
QUALITY	VARCHAR2(20 BYTE)	
STATUS	VARCHAR2(20 BYTE)	
ID	VARCHAR2(20 BYTE)	FK for monitoring device ID

While this way of storage is sufficient for a database and also is in line with the IFC standard, it is not feasible for implementing database cubes in a DWH. The reason for this is that the fact table is lacking any additional columns that could be linked through a foreign key relationship to dimensional objects. For example, this work envisages to provide a spatial dimension in the DWH. This requires that for each reading in the fact table, the location of the reading is also stored in the fact table.

Nonetheless, an IFC compatible fact table can be built which is compatible to Data Warehousing by introducing an additional intermediate database table to the schema which will be called surrogate table. Adding a surrogate table ensures that the underlying database schema is still compatible with the IFC definition. Changing the table layout of the fact table would allow the table to be used for the building of cubes. However, the approach of this work is to maintain compatibility between the IFC definition and the database schema. Therefore, the surrogate table must be built. A benefit from adding the surrogate table is the possibility to implement data quality checks at this level of implementation. These routines can be used to retrospectively verify and correct data before it populates the surrogate table.

This surrogate table holds the information from the fact table, but is also enriched with additional columns that allow the creation of relationships to dimensional objects. The surrogate table can be realised again through utilisation of MVs. A further benefit from building the surrogate table is that KPI’s can be calculated during the table’s creation process. Otherwise, KPI’s would need to be calculated separately. This effectively reduces further processing time. The process for creating the surrogate fact table is as outlined in the following six steps:

1. IFC relationships need to be resolved in order to link a record from the fact table to dimensional objects. All records in the fact table will be created by sensors. Therefore, the sensor is the mutual information which needs to be linked to the dimensional objects. The following section handles the linking of sensors to selected IFC objects, namely rooms, organizations and systems.

   For linking a sensor to a room it is necessary to resolve the relationships between IfcSpace, IfcRelContainedInSpatialStructure and IfcSensor, see Figure 3-27.

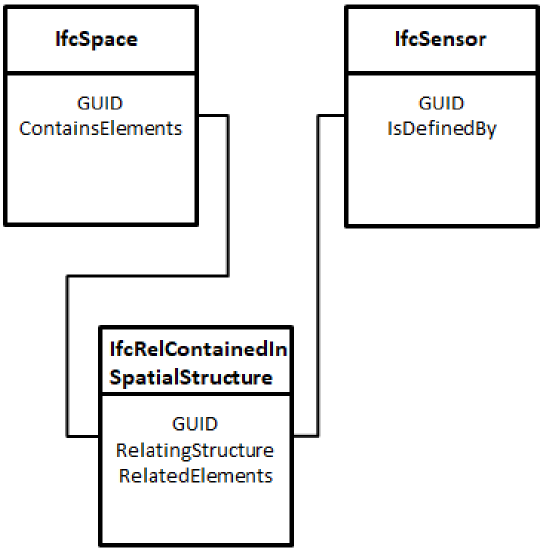


Figure 3-27 Relation between IfcSpace and IfcSensor

For linking organizations, it is necessary to resolve the relationship between IfcOrganization, IfcPersonAndOrganization, IfcOwnerHistory, IfcSpace, IfcRelContainedInSpatialStructure and IfcSensor. By extending the previous figure with an additional relation, the organization can be linked to the sensor. This is demonstrated in Figure 3-28. Here, the previous relation is framed grey.

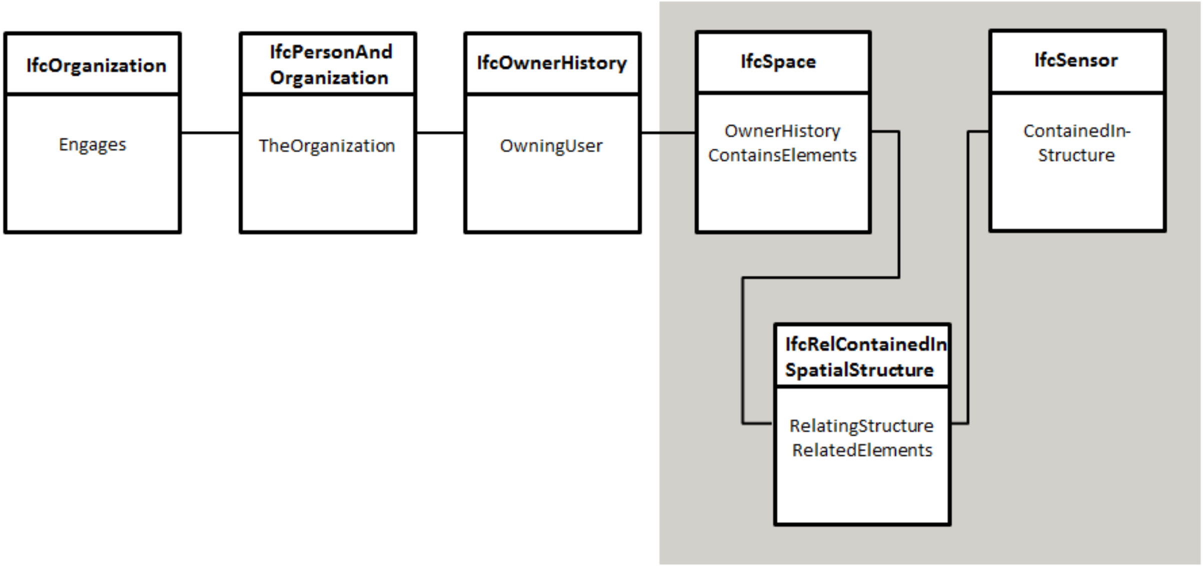


Figure 3-28 Relation between IfcOrganization and IfcSensor

To link building systems, it is required to resolve the relationship between IfcSensor, IfcRelContainedInSpatialStructure and IfcDistributionFlowElement. Their relation in the IFC meta model is sketched in Figure 3-29.

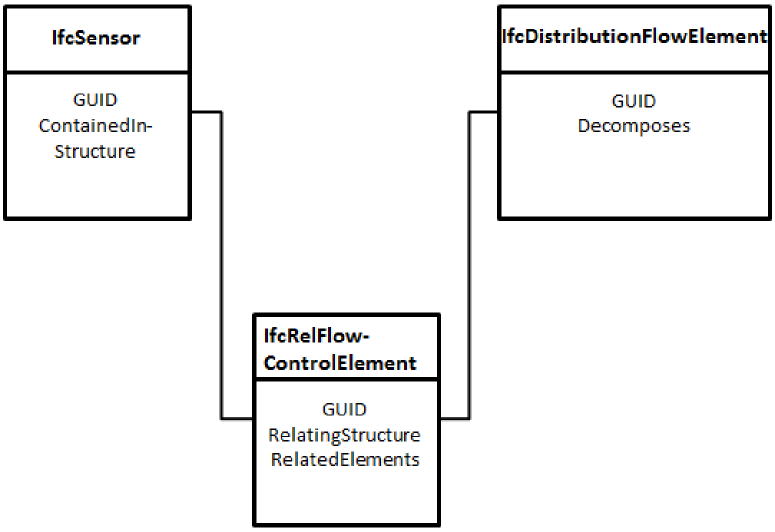


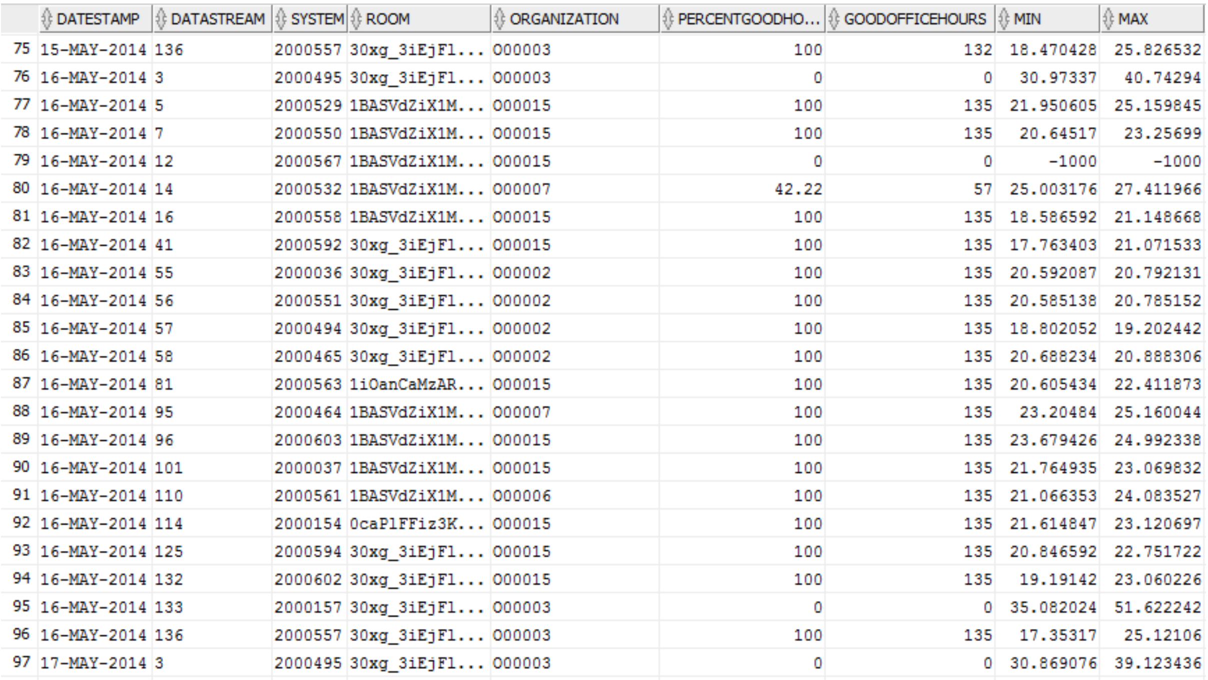
Figure 3-29 Relation between IfcDistributionFlowElement and IfcSensor

2. As sensed time series data may be acquired at irregular intervals, it gets aggregated from the IfcIrregularTimeSeriesValue table and grouped to common minimum time intervals. This time interval has to be the smallest interval defined in the time dimension.

3. Individual KPI may be calculated to be part of the surrogate table. These are discussed in the following chapter.

4. Foreign key IDs are created for linking to objects in the dimensional tables. For each record in the surrogate table, the sensing device is connected to a system, a room and an organization. For any record which cannot be linked to all dimensions, the record is discarded. This is a database design requirement as OLAP cubes cannot be populated with incomplete sets of fact data. Failures to link records could happen e.g. if a sensor is placed in a staircase which is not allocated to one specific building storey. Another reason could be that a sensor is attached to a system not represented in the MEP model.

5. The result of this operation is stored in a MV which fulfils all requirements for the implementation of OLAP cubes. The SQL script used to transform the outlined procedure can be found in appendix 6.



	DATESTAMP	DATASTREAM	SYSTEM	ROOM	ORGANIZATION	PERCENTGOODHO...	GOODOFFICEHOURS	MIN	MAX
75	15-MAY-2014	136	2000557	30xg_3iEjFl...	000003	100	132	18.470428	25.826532
76	16-MAY-2014	3	2000495	30xg_3iEjFl...	000003	0	0	30.97337	40.74294
77	16-MAY-2014	5	2000529	1BASVdZiX1M...	000015	100	135	21.950605	25.159845
78	16-MAY-2014	7	2000550	1BASVdZiX1M...	000015	100	135	20.64517	23.25699
79	16-MAY-2014	12	2000567	1BASVdZiX1M...	000015	0	0	-1000	-1000
80	16-MAY-2014	14	2000532	1BASVdZiX1M...	000007	42.22	57	25.003176	27.411966
81	16-MAY-2014	16	2000558	1BASVdZiX1M...	000015	100	135	18.586592	21.148668
82	16-MAY-2014	41	2000592	30xg_3iEjFl...	000015	100	135	17.763403	21.071533
83	16-MAY-2014	55	2000036	30xg_3iEjFl...	000002	100	135	20.592087	20.792131
84	16-MAY-2014	56	2000551	30xg_3iEjFl...	000002	100	135	20.585138	20.785152
85	16-MAY-2014	57	2000494	30xg_3iEjFl...	000002	100	135	18.802052	19.202442
86	16-MAY-2014	58	2000465	30xg_3iEjFl...	000002	100	135	20.688234	20.888306
87	16-MAY-2014	81	2000563	1iOanCaMzAR...	000015	100	135	20.605434	22.411873
88	16-MAY-2014	95	2000464	1BASVdZiX1M...	000007	100	135	23.20484	25.160044
89	16-MAY-2014	96	2000603	1BASVdZiX1M...	000015	100	135	23.679426	24.992338
90	16-MAY-2014	101	2000037	1BASVdZiX1M...	000015	100	135	21.764935	23.069832
91	16-MAY-2014	110	2000561	1BASVdZiX1M...	000006	100	135	21.066353	24.083527
92	16-MAY-2014	114	2000154	0caPlFFiz3K...	000015	100	135	21.614847	23.120697
93	16-MAY-2014	125	2000594	30xg_3iEjFl...	000015	100	135	20.846592	22.751722
94	16-MAY-2014	132	2000602	30xg_3iEjFl...	000015	100	135	19.19142	23.060226
95	16-MAY-2014	133	2000157	30xg_3iEjFl...	000003	0	0	35.082024	51.622242
96	16-MAY-2014	136	2000557	30xg_3iEjFl...	000003	100	135	17.35317	25.12106
97	17-MAY-2014	3	2000495	30xg_3iEjFl...	000003	0	0	30.869076	39.123436

Figure 3-30 Materialized View surrogate table

An example for a surrogate table is given in Figure 3-30. The table contains per record foreign keys to dimensional objects as well as a few aggregated values which could be used for KPI calculations. More specifically, the columns “datestamp”, “system”, “room” and “organization” are foreign keys to their corresponding dimensions. “Datastream” is a foreign key to a device table which identifies the sensor/meter. The columns “percentgoodhours”, “goodofficehours”, “min” and “max” are custom KPI which were calculated and added to the surrogate table during its generation. These pre-calculated KPI can be queried by the cube.

It is noteworthy to highlight the importance of a consistent database schema. During this research, several issues were encountered that failed the creation and/or refresh process of the surrogate fact table. These issues were caused by changes in the database schema which were not enforced by primary and foreign key relationships. The following list enumerates these occurrences:

- ID’s in the surrogate table which link to non-existing dimensional objects. This could happen e.g. after a change in the dimension.
- Improperly set up dimensional entries, e.g. a space that is not linked to a single building storey as it appears on multiple floors (e.g. a staircase room)
- Duplicate entries in the dimensions.

In addition to missing relationships, the fact table did not only store numeric readings as some data sources provided error strings as values. This has caused the following issue:

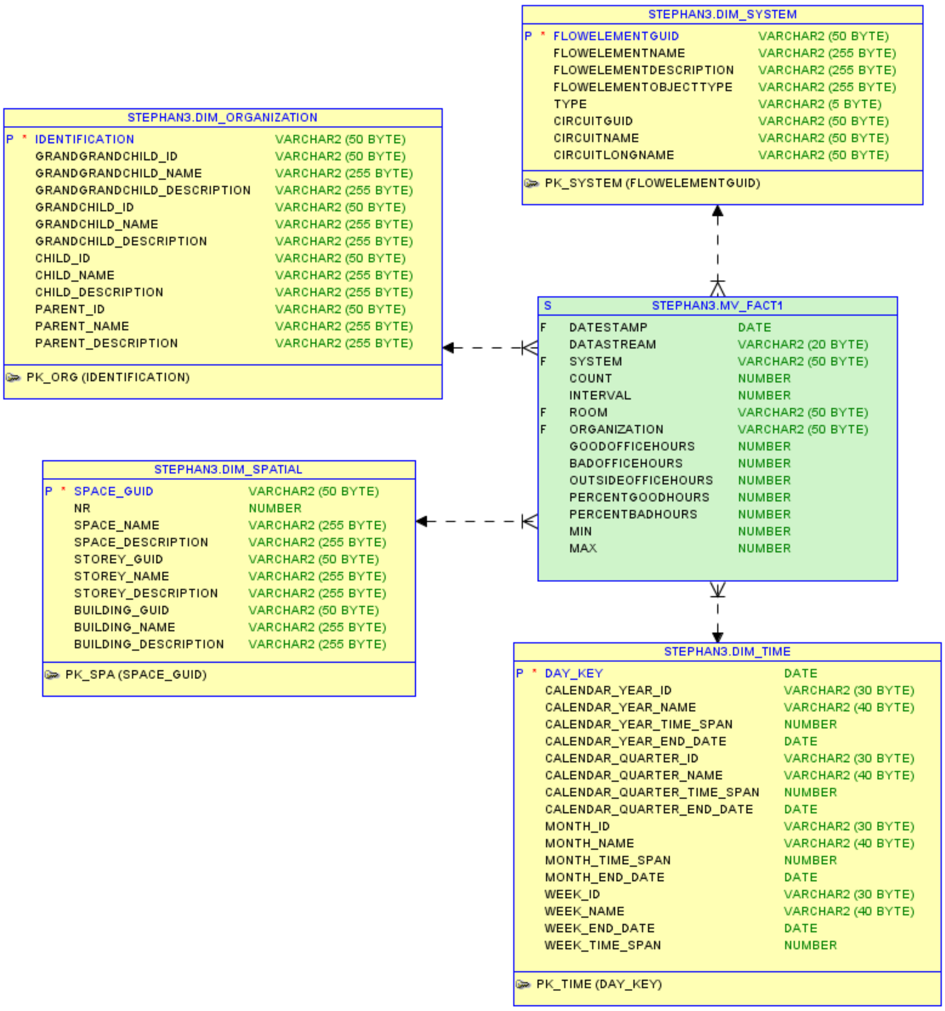
- Acquired fact readings which are not numeric. These fail the MV creation process if any KPI’s are calculated.

All these issues can be resolved by a proper DWH design. Duplicates or missing links cannot appear when columns are consistently linked through primary and foreign keys. Non numeric values can either be handled by a database trigger or simply discarded. Discarding would happen automatically if the column holding the values is of a numeric data type.

3.4.4 Database model

Through the implementation of aforementioned dimensional tables and the generation of a suitable fact table, the framework for a data cube is set. For the cube to function, the individual tables need to be linked to each other through database relations. Specifically, each foreign key in the surrogate fact table needs to be linked to its corresponding dimensional table. Each dimension needs to define a primary key which is acting as partner in the database constraint. This primary key needs to be appointed to the smallest element in the dimensional hierarchy.

As a result of this, a star schema (Figure 3-31) is created. Here, all five depicted tables are MVs. The four yellow tables are dimensional tables which hold selected information from IFC objects of a specific domain. The green table is the surrogate fact table which relates to the dimensions and which holds pre-calculated information.



STEPHAN3.DIM_SYSTEM
P * FLOWELEMENTGUID VARCHAR2 (50 BYTE)
FLOWELEMENTNAME VARCHAR2 (255 BYTE)
FLOWELEMENTDESCRIPTION VARCHAR2 (255 BYTE)
FLOWELEMENTOBJECTTYPE VARCHAR2 (255 BYTE)
TYPE VARCHAR2 (5 BYTE)
CIRCUITGUID VARCHAR2 (50 BYTE)
CIRCUITNAME VARCHAR2 (50 BYTE)
CIRCUITLONGNAME VARCHAR2 (50 BYTE)
PK_SYSTEM (FLOWELEMENTGUID)
STEPHAN3.DIM_ORGANIZATION
P * IDENTIFICATION VARCHAR2 (50 BYTE)
GRANDGRANDCHILD_ID VARCHAR2 (50 BYTE)
GRANDGRANDCHILD_NAME VARCHAR2 (255 BYTE)
GRANDGRANDCHILD_DESCRIPTION VARCHAR2 (255 BYTE)
GRANDCHILD_ID VARCHAR2 (50 BYTE)
GRANDCHILD_NAME VARCHAR2 (255 BYTE)
GRANDCHILD_DESCRIPTION VARCHAR2 (255 BYTE)
CHILD_ID VARCHAR2 (50 BYTE)
CHILD_NAME VARCHAR2 (255 BYTE)
CHILD_DESCRIPTION VARCHAR2 (255 BYTE)
PARENT_ID VARCHAR2 (50 BYTE)
PARENT_NAME VARCHAR2 (255 BYTE)
PARENT_DESCRIPTION VARCHAR2 (255 BYTE)
PK_ORG (IDENTIFICATION)
S STEPHAN3.MV_FACT1
F DATESTAMP DATE
DATASTREAM VARCHAR2 (20 BYTE)
F SYSTEM VARCHAR2 (50 BYTE)
COUNT NUMBER
INTERVAL NUMBER
F ROOM VARCHAR2 (50 BYTE)
F ORGANIZATION VARCHAR2 (50 BYTE)
GOODOFFICEHOURS NUMBER
BADOFFICEHOURS NUMBER
OUTSIDEOFFICEHOURS NUMBER
PERCENTGOODHOURS NUMBER
PERCENTBADHOURS NUMBER
MIN NUMBER
MAX NUMBER
STEPHAN3.DIM_SPATIAL
P * SPACE_GUID VARCHAR2 (50 BYTE)
NR NUMBER
SPACE_NAME VARCHAR2 (255 BYTE)
SPACE_DESCRIPTION VARCHAR2 (255 BYTE)
STOREY_GUID VARCHAR2 (50 BYTE)
STOREY_NAME VARCHAR2 (255 BYTE)
STOREY_DESCRIPTION VARCHAR2 (255 BYTE)
BUILDING_GUID VARCHAR2 (50 BYTE)
BUILDING_NAME VARCHAR2 (255 BYTE)
BUILDING_DESCRIPTION VARCHAR2 (255 BYTE)
PK_SPA (SPACE_GUID)
STEPHAN3.DIM_TIME
P * DAY_KEY DATE
CALENDAR_YEAR_ID VARCHAR2 (30 BYTE)
CALENDAR_YEAR_NAME VARCHAR2 (40 BYTE)
CALENDAR_YEAR_TIME_SPAN NUMBER
CALENDAR_YEAR_END_DATE DATE
CALENDAR_QUARTER_ID VARCHAR2 (30 BYTE)
CALENDAR_QUARTER_NAME VARCHAR2 (40 BYTE)
CALENDAR_QUARTER_TIME_SPAN NUMBER
CALENDAR_QUARTER_END_DATE DATE
MONTH_ID VARCHAR2 (30 BYTE)
MONTH_NAME VARCHAR2 (40 BYTE)
MONTH_TIME_SPAN NUMBER
MONTH_END_DATE DATE
WEEK_ID VARCHAR2 (30 BYTE)
WEEK_NAME VARCHAR2 (40 BYTE)
WEEK_END_DATE DATE
WEEK_TIME_SPAN NUMBER
PK_TIME (DAY_KEY)

Figure 3-31 Database model in a star schema

The following short SQL script automatically creates primary keys for all introduced dimensional tables. Additionally, it links the primary keys through constraints with the surrogate fact table.

Due to the characteristic of the database schema, which follows IFC nomenclature, the above script should be easily applicable with minimum changes to any IFC database schema. Since IFC is specified as meta-model, it does not specifically envisage its implementation in a database environment. Therefore, the IFC standard does not define the naming for dimensions, and primary/foreign keys. This is where minor changes might be needed for the script to run.

Listing 6 Create database constraints

```
ALTER TABLE DIM_ORGANIZATION ADD CONSTRAINT PK_ORG PRIMARY KEY
(IDENTIFICATION);
ALTER TABLE DIM_SPATIAL ADD CONSTRAINT PK_SPA PRIMARY KEY (SPACE_GUID);
ALTER TABLE DIM_TIME ADD CONSTRAINT PK_TIME PRIMARY KEY (HOUR_KEY);

ALTER TABLE MV_FACT ADD CONSTRAINT FK_ORG FOREIGN KEY (ORGANIZATION)
REFERENCES DIM_ORGANIZATION(IDENTIFICATION);
ALTER TABLE MV_FACT ADD CONSTRAINT FK_SPA FOREIGN KEY (ROOM) REFERENCES
DIM_SPATIAL(SPACE_GUID);
ALTER TABLE MV_FACT ADD CONSTRAINT FK_TIME FOREIGN KEY (TIME) REFERENCES
DIM_TIME(HOUR_KEY);
```

3.4.5 Added value of the Data Warehouse

The benefit of the Data Warehouse approach presented in this work is that it provides a central repository for both fact and dimensional data. It maintains compatibility with BIM through utilisation of the IFC standard. By implementing a database schema which is derived from IFC, it ensures that data may be exchanged in both directions. The adaption of IFC to a database schema allows room for future tools which do certain BIM operations on a database level. To avoid any increase in processing time, a DWH is built on top of the IFC database schema. The DWH and its technologies allow instant access to aggregated information and KPI's. The benefit of having both fact and dimensional data in a central data schema results in a standardised process to access information.

3.5 Data Analysis

This section outlines two methodologies used in this thesis for data analysis. One is the selection and implementation of KPI's, the other is the clustering of load curve data to reveal different patterns of building behaviours.

3.5.1 Key Performance Indicators

An additional aim of the DWH implementation is to provide KPI to facility managers. These figures should be obtainable with minimum efforts in order to support efficient data analysis.

The availability of the four introduced domains allows the definition of KPI. Various KPI's are of interest for a system deployed in the FM domain. Popular examples are

(i) average temperature across rooms facing one direction,

(ii) energy consumption per square footage,

(iii) efficiency of a system or a set of systems.

Further KPI’s were introduced by our research group and discussed in detail in (Menzel et al., 2013). The following KPI’s were identified within the EU FP7 project Campus21 as beneficial for the analysis of building performance data (Menzel et al., 2014).

KPI’s to aid in energy consumption analysis

a. Highest value of the day

b. Lowest value of the day

c. Sum of all values

d .Average value

KPI’s to aid in evaluating thermal comfort

e. Underperformance Time (UPT)

f. Underperformance Ratio (UPR)

g. Average duration of working hours

The UPT and UPR both measure performance based on defined parameters. Both UPT and UPR are percentage values while UPT is calculated from time values. For these KPI’s, an evaluation criterion and its allowed spread needs to be defined. Additionally, applicability may be constrained by time. This could be e.g. temperature which is allowed to vary between 18 °C and 26 °C during office hours.

Previous research (Menzel et al., 2014) clustered the individual KPI into four pillars, as seen in Figure 3-32

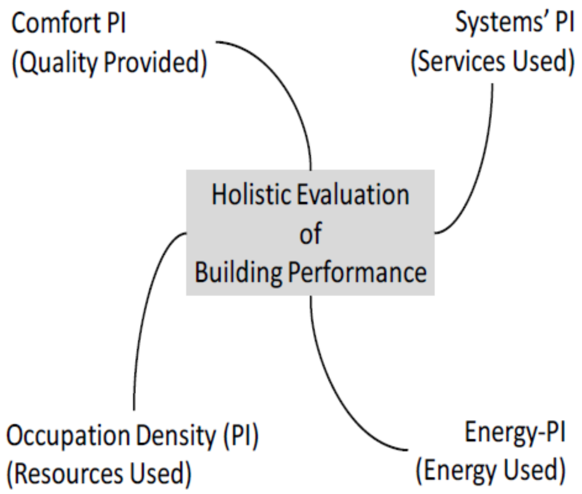


Figure 3-32 Categories of Key Performance Indicators as identified in (Menzel et al., 2014)

In addition to these, facility managers often show interest in an energy related subset of KPI's which are sometimes called Energy Performance Indicators (EnPI). Their amalgamation was executed in a recent Masters project (Arnold, 2014). Its findings are summarised here:

EnPI's applicable to a building:

a) Energy consumption per m^2 gross (total floor space)

b) Energy consumption per m^2 net (usable floor space)

c) Energy consumption per office spaces/storeys (for tenant benchmarking)

The following EnPI's are only applicable for certain building categories:

d) Energy consumption per employee/visitor/resident/patient

e) Energy consumption per produced piece/growth/workload

f) Energy consumption per meal, bed, seat, event

In more generalised terms, energy consumption may be put in relation with anything quantifiable to provide an EnPI which is specific to a sector or industry.

The combination of KPI's with the four introduced domains is what makes these KPI's so meaningful. In this thesis, only a selected number of KPI's will be implemented as their calculation within a DWH is a straight forward process. Two examples for calculating KPI within a DWH are given below.

Example 1: EnPI per gross square meter of a building on a weekly basis

Listing 7 Calculate KPI / m^2

```
define sqm = 3000

SELECT  id as ELECTRICITY_METER,
          TO_CHAR(timestamp, 'YYYY-MM') as MONTH,
          round(sum(value) / &&sqm) as EnPI_sqm,
FROM      IFCITSV
WHERE     id = 143
GROUP BY  id, TO_CHAR(timestamp, 'YYYY-MM')
ORDER BY  id;
```

In this example, the total square meters of the building were given as 3000. The electricity meter(s) are accessible through ID number 143. The KPI is calculated on a monthly basis by grouping timestamps with “YYYY-MM”. Data from a table called “IFCITSV” is processed and the actual readings are stored in a column named “values”.

Example 2: Underperformance Ratio (UPR) of rooms. The UPR is the amount of time in percent where a room is not performing within given thresholds. For office rooms these thresholds are defined as 18 to 26 degrees Celsius between 9am and 5pm.

Listing 8 Calculate UPR KPI

```
define max_t = 26
define min_t = 18
define openingtime = 09
define closingtime = 17

SELECT
      datestamp,
      datastream,
      round(badofficehours  /  (goodofficehours  -  badofficehours  )  *
      100,2) as percentbadhours
FROM
(
SELECT
      TO_date(trunc(timestamp),'DD-MM-YY') AS Datestamp,
      id AS datastream,
      count(case  when  value  <=  &&max_t  and  value  >=  &&min_t  and
      TO_char(timestamp, 'HH24') > &&openingtime and TO_char(timestamp,
      'HH24') <= &&closingtime then 1 end) as goodofficehours,
      count(case  when  (value  >  &&max_t  or  value  <  &&min_t)  and
      TO_char(timestamp, 'HH24') > &&openingtime and TO_char(timestamp,
      'HH24') <= &&closingtime then 1 end) as badofficehours
FROM IFCITSV
GROUP BY  TO_date(trunc(timestamp),'DD-MM-YY') ,id
)
ORDER BY datestamp;
```

In this example, the inner SELECT statement counts good and bad intervals as per given thresholds. This KPI is calculated on a daily basis by grouping timestamps with “DD-MM-YY”. Added time intervals are stored in “goodofficehours” and “badofficehours”. The outer routine then calculates a percentage value per day. These percentage values are rounded to numbers with 2 decimal places.

These and more KPI’s can be calculated during the processing of the before discussed surrogate table. This is an improvement as it eliminates the need to calculate KPI’s in a separate step. Any changes, addition or removal of KPI’s requires DWH engineers to work in a single location. The combined process is sketched in Figure 3-33. This figure should be read from inner to outside SELECT statement. The inner SELECT statement acquires the fact data, groups this data per timestamp and calculates KPI’s. The middle SELECT statement resolves the relations to the dimensional tables and establishes foreign keys to link the surrogate table to the dimensions. The outer SELECT statement may be used to process further KPI’s. This could be used e.g. if information from the dimensional tables needs to be consulted for calculation. The full script which created the environment for this research can be found in appendix 6.

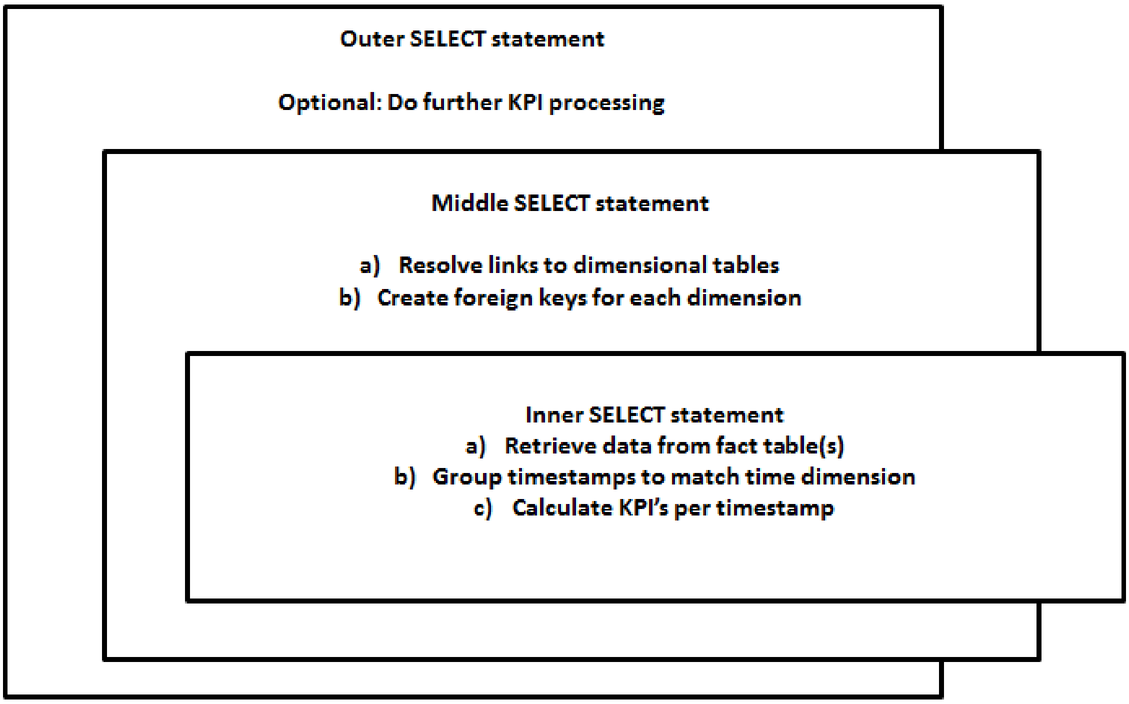

Outer SELECT statement
Optional: Do further KPI processing
Middle SELECT statement
a) Resolve links to dimensional tables
b) Create foreign keys for each dimension
Inner SELECT statement
a) Retrieve data from fact table(s)
b) Group timestamps to match time dimension
c) Calculate KPI’s per timestamp

Figure 3-33 Sketched generation of surrogate table with KPI’s

3.5.2 Data clustering

This section combines machine learning algorithms with performance data to cluster data into groups that share similar usage patterns. Based on the state of the art analysis conducted in chapter 2, the standard k-means algorithm was chosen to be utilised. This is because it has proven to be very robust in related fields and applications (Hoerster et al., 2015).

k-means minimises the sum of distances of each object to its centroid (centre). The algorithm continues until sums cannot be decreased any further. The result is a number of k clusters. MATLAB allows specifying the number of envisaged clusters, initial values for a centroid and the numbers of iteration runs to minimise the sum of clusters (Mathworks, 2015).

For choosing the initialisation values (also: seed), the deterministic PCA-Part (Principal component analysis) algorithm (Su and Dy, 2004) is employed. The aim of this work is to enable users without much acquaintance with data mining principles to make use of the functionality. Requirements to the algorithm are therefore:

- The clustering algorithm must be robust and be able to operate on a variety of datasets of different sources in different dimensions, with minimal user interaction.
- The algorithm must be deterministic, as users would be confused by being presented varying results for the same data set.

The implementation of the algorithm is as follows:

Given a set of data points $p_1 \dots p_n$ and a number k, the k-means algorithm searches for a set of cluster centres $P_1 \dots P_k$ minimising the squared distance

$$\sum_{j=1}^{N} \frac{min}{\alpha} \parallel p_j - P_\alpha \parallel^2 \tag{6}$$

It does so by alternatively finding for each data point j the cluster index α_j such that $\left\|p_j - P_\alpha\right\|$ is minimal, and the centre P_α minimising the sum of squared distances from the points in the cluster α.

Specifically, the algorithm used reads as follows.

1. Initialise the cluster centre P_α, $\alpha = 1, \dots, k$ and the cluster indices $\alpha_j \in \{1, \dots, k\}$, $j = 1, \dots n$ by using the PCA-Part algorithm

2. For each α set $P_\alpha = \frac{1}{|C_\alpha|} \sum_{j \in C_\alpha} p_j$. Here C_α is the set of indices in the cluster α, i.e., $C_\alpha = \{ j \mid a_j = \alpha \}$.

3. For each j, find the cluster index a_j minimising $\left\| p_j - P_{\alpha_j} \right\|$

4. Continue with 2. until convergence is reached.

The algorithm was implemented in MATLAB and can be found in appendix 7.

In order to apply the algorithm, the data is beforehand split into daily intervals. By default, a user needs to define how many clusters k he is expecting to see. The algorithm then clusters all data into the given number. This limitation is extended to automatically estimate the number of clusters so as to minimise the required user interaction. For the estimation of the cluster number the following heuristic is used:

1. Run the k-means algorithm for $k = 1, \dots, 10$ clusters and record the corresponding sums of distances S_k , $k = 1, \dots, 10$, of the data points from their cluster centres.

2. The lowest k is chosen as $S_k < \frac{(S_1 + 2S_{10})}{3}$. This approximation worked well with the data available for this research. If required, further research may substitute this with a more complex algorithm to identify the inflection point.

In this heuristic, S_k is the error for choosing k clusters. When plotting S_k, one can see a decreasing curve, see Figure 3-34.

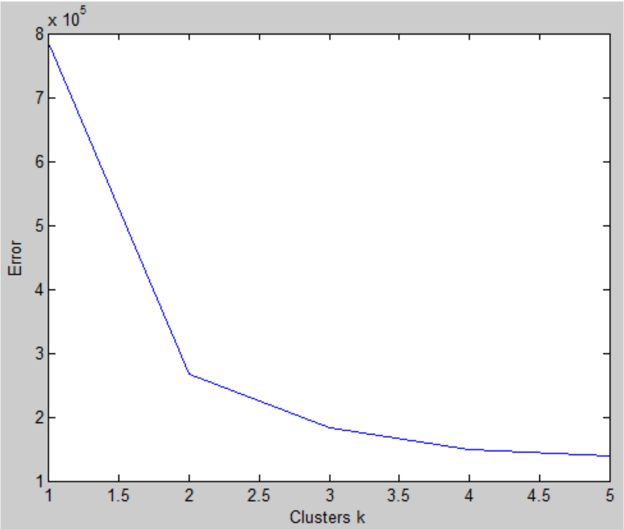


Figure 3-34 Error for automatic cluster detection

Upon inspection of the curve, the number of clusters k is the significant around the area where the inflection point in the curve sits. This section represents the best trade-off between the least number of clusters and the highest approximation of the cluster's data. In the evaluated data sets, the error S_k dropped quickly for about $\frac{2}{3}$ and then slowly for the remaining $\frac{1}{3}$. Therefore, this heuristic was chosen to include all clusters k that are with the drop of the first $\frac{2}{3}$. In Figure 3-34, the preferred number of clusters k is 2.

One example for data clustering can be seen in Figure 3-35 and Figure 3-36. Figure 3-35 depicts the monitored electricity consumption from a factory building. One year of data is evaluated in this example. Figure 3-36 shows the clustering results from this data set. Here, one can see that the algorithm has detected $k = 3$ centroids. The blue cluster permanently shows high load. The red and green clusters are similar in the first time period, however they significantly change after 20 readings. These three patterns are not identifiable when looking at Figure 3-35. With further knowledge about the building and its operational patterns, it is likely that these behavioural patterns can be related to different processes in the factory.

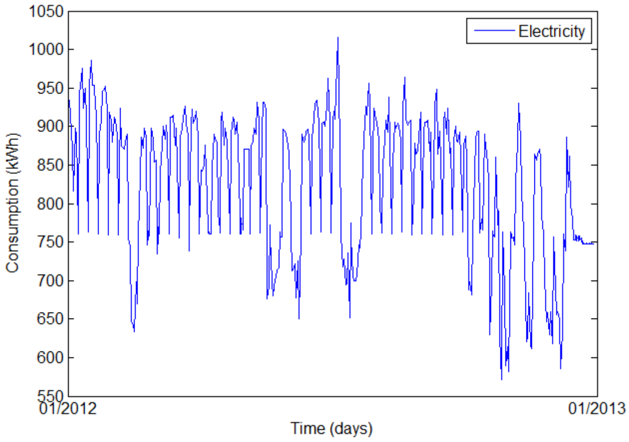


Figure 3-35 Raw electricity consumption

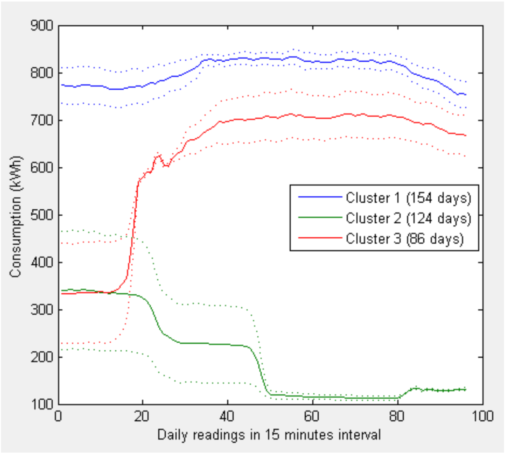


Figure 3-36 Clustering results

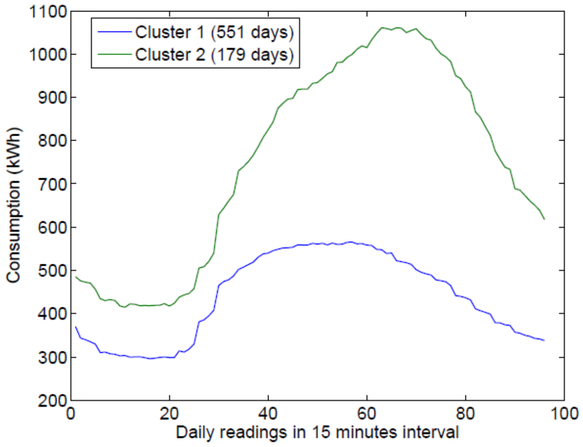
Note: The clustered data is in this figure, and in the following analysis is sampled in 15 minute intervals. Therefore, a daily consumption pattern consists of 96 values. To increase readability, Table 23 provides a conversion to the reader.

Table 23 Conversion of readings to time

Number of 15 minute sample	Time in 24 hour format
20	05:00
40	10:00
60	15:00
80	20:00

In a different example, if the number of patterns was defined by the user as 2, the output would look like in Figure 3-37. The lines represent 2 possible modes of operation for the building.

In this case the algorithm is enabled to identify the number of behavioural patterns automatically, it comes up with three individual patterns, as seen in Figure 3-38. When comparing both figures, a third, large power consumer is revealed. This building behaviour is not obvious from the diagram where k has been defined as $k = 2$. The high power consumption only occurs on 46 days which was not obvious when k was set to 2.



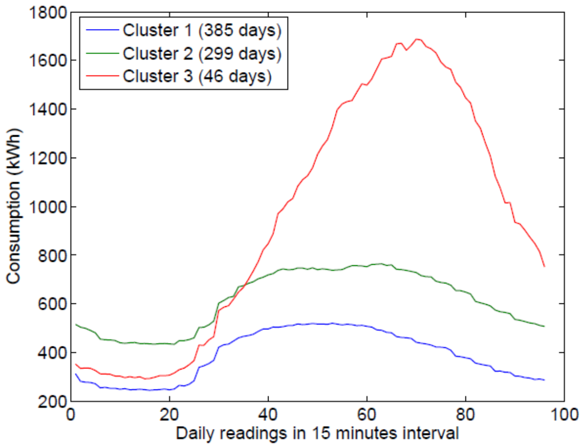


Figure 3-37 k-means clustering with k=2

Figure 3-38 k-means clustering with k=3

Finally, a principal component analysis on each cluster found is performed in order to identify the leading principal component. Additionally, the standard deviation in this direction within the cluster is revealed. This data is then used to indicate to the user the principal intra-cluster variation, see Figure 3-39. For example, the dataset shown is for a football stadium, the big peak corresponds to days with football matches taking place. The match can take place in the afternoon or in the evening, and this variation is distinctly indicated.

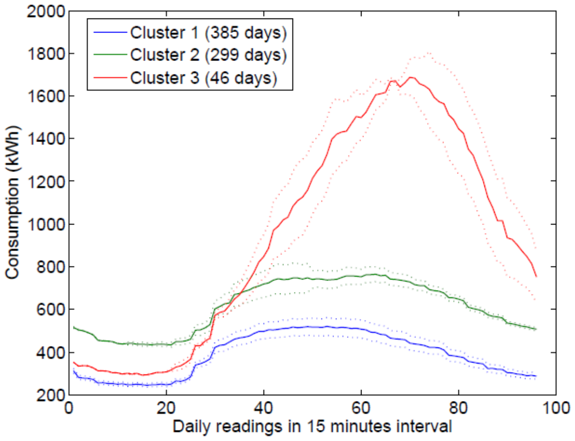


Figure 3-39 Clustering with standard deviation

Further examples of clustering will be discussed in chapter 5. Here, data from three pilot buildings will be evaluated and other forms of energy will be tested with the proposed clustering algorithm.

3.5.3 Added value to data analysis

This chapter introduced clustering to load curve analysis. Upon clustering a load curve, the typical behaviour patterns of a building are revealed. This allows investigating the individual clusters and their number of occurrence. These patterns are not visible through a regular load curve analysis and therefore enhance the output gained through data analysis. A limitation of clustering is that a user needs to know the number of clusters k. For energy analysts, the number of clusters k is often not known as they are working from remote and often not visit the building they have to analyse. Therefore, this work developed a unique, heuristic approach which identifies the number of clusters k. With the knowledge of the right number of clusters and a visual of their behaviour, future load curve analysis can result in better analysis.

3.6 Summary

Chapter 3 has introduced the methodologies developed as part of this thesis. Their common objective is the efficient processing of building data. Data gets acquired from a building, interpolated and cleansed, enriched with information extracted from BIM models, stored in a database, aggregated into a DWH and finally analysed. Following these methodologies, the procedure for building data acquisition, processing and analysis can be standardised.

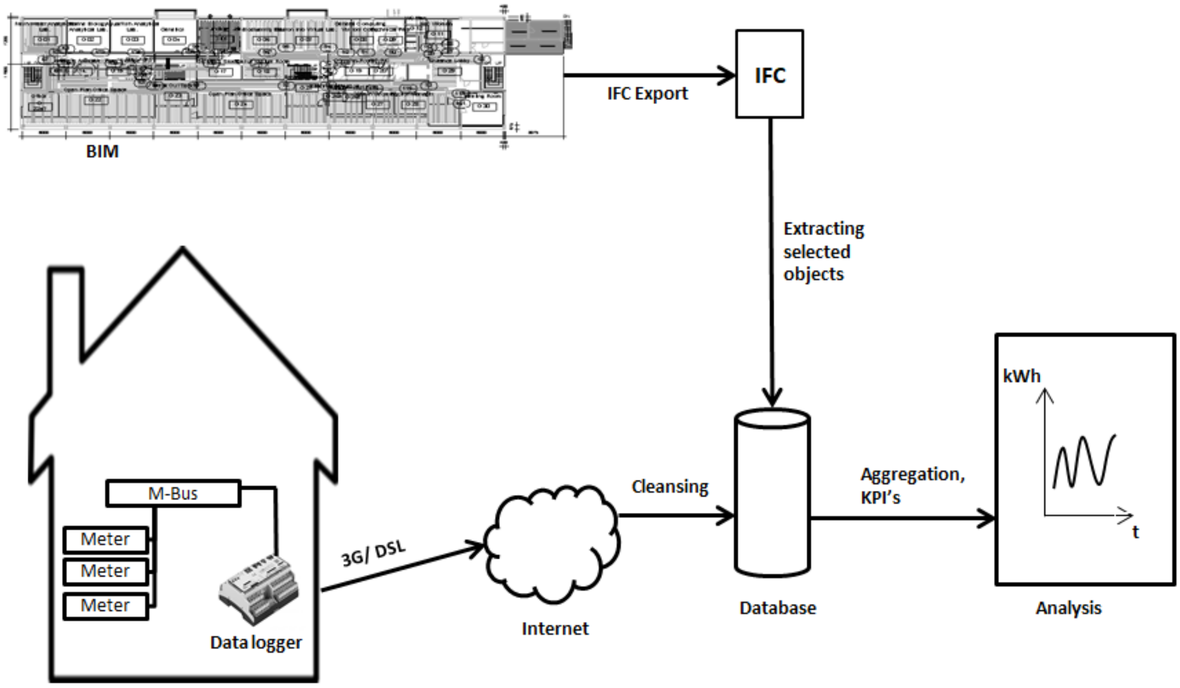


Figure 3-40 Computing architecture

Figure 3-40 provides a schematic overview of the computing architecture utilised in this work. It demonstrates that building data, stored in the database, is acquired through two different approaches. On the one hand, there is BIM which gets exported to IFC and parsed for selected objects. These objects then are stored as dimensional data in the database. On the other hand, data gets acquired through a data logger installed in the building. The data logger is connected via M-Bus protocol to individual meters and frequently queries their readings. The recorded readings are subsequently transferred via Internet. This connectivity is provided to the data logger via 3G or DSL modem. Before writing the data into the database, the cleansing routine corrects anomalies. On the database level, gathered information gets combined and aggregated for analysis. The whole process from acquisition to analysis can be automated which allows users to focus on the analysis of data.

The next chapter will outline the implementations conducted in the pilot buildings and chapter 5 will present the achieved results based on chapter 4.

4 Evaluation of the Monitoring Concept

In this chapter, three pilot buildings have been selected in order to verify the methodologies introduced in the previous chapter. Each building will be introduced to the reader and its monitoring concept will be discussed. The aim of the installations is a flawless and comprehensive data feed of meter information which ultimately may be displayed in an EnMS.

The three selected buildings are very different in size, usage and monitoring detail. The buildings in question are:

- A university research building,
- A public hospital,
- A multi-purpose sports arena.

The demonstration buildings discussed in this chapter are presented by their monitoring complexity, starting with the least complex (university building) and ending with the most complex (sports arena).

The data cleansing and interpolation algorithm is applied to the data acquired from the pilot buildings and selected information and its results are discussed at the end of this chapter. The cleansed data from these pilot buildings will be the foundation of the analysis discussed in chapter 5.

4.1 Pilot buildings

This chapter introduces the three pilot buildings. It provides a general overview about the building and summarises the meters already installed. The chosen buildings are unique to each other in their usage. This also reflects the individual expectations towards an EnMS.

4.1.1 University building

This building is a three storey research building located in Cork, Ireland. It is mainly occupied by post-doctoral researchers. The building has a gross size of approximately 3.000 square meters housing about 90 researchers. The rooms are mainly offices, meetings rooms and laboratories.

The building is considered a “living laboratory”, where many environmental research projects take place. The building is already used as demonstration site for renewable energy sources such as geothermal and solar systems. It is supplied with gas for its hot water boilers and grid electricity to drive building systems, lighting and equipment. Furthermore, the building is equipped with its own locally installed weather station. This station is able to measure: wind speed, wind direction, total and diffuse solar radiation, humidity, light level, sunshine hours and temperature. The building is controlled with a Building Management System (BMS).

4.1.1.1 Site inspection

A site inspection was conducted to identify meters to monitor. It revealed one meter for each medium (gas, electricity and water). Gas and electricity meters are installed in the same room, while the electricity meter is located in close proximity.

The electricity meter is a “Diris Ap” from Socomec (Socomec, 2013). The meter is of a modular type which provides connection ports for up to four add-on modules. Its data sheet reveals that pulse outputs can be retrieved through an add-on module installed at one of the connection ports (Electrocomponents, 2013). The emitted pulse interval can be set with a programmable value (0.1, 1, 10, 100) in kWh. It was found that this specific meter was pre-configured with a set interval of 1 pulse per kWh.

The gas meter is an Actaris G16 (Actaris, 2013). The meter has a single pulse output. The meter itself provides the technical information as exemplary shown in Figure 4-1. It is usually the case that technical details are printed on the meter. Each pulse output equals the gas consumption of 0.1 m^3 which is approximately 1 kWh.

The installed water meter could not be identified as there are no manufacturing labels or technical details printed on the meter. The meter also has one pulse output. Its non-identifiability is undesired as an EnMS needs to know the correct value and unit of each pulse. This work will later discuss how to deal with this issue.

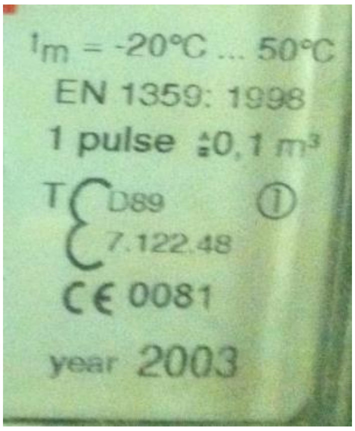


Figure 4-1 Technical description from the Actaris G16 meter

Table 24 summarises the findings from the site inspection.

Table 24 University building site inspection

	Electricity Meter	*Gas Meter*	*Water Meter*
Manufacturer	Socomec	Actaris	unknown
Type	Diris AP	G16	unknown
Connection port	yes	yes	yes
Communication method	pulse	pulse	pulse
Pulse value	1 pulse = 1 kWh	1 pulse = 0.1m^3	**unknown**

4.1.2 Public hospital

This demonstration building is a public hospital located near Cologne, Germany. The main building hosting patients dates back to 1977 and is built as 10 storey complex. The adjacent office building has 3 floors. Total net square footage of the site is almost 12.000 square meters. The hospital has a 310 bed capacity for patients. Its building systems have not been renewed since the construction of the building. Among the technical systems are two gas boilers and two steam boilers. Before the installation of any monitoring equipment, the only available consumption data was derived from its monthly utility bills. The building tenant placed an order to renew and optimise its building systems. This renewal involved two Combined Heat and Power Plants (CHP). The reason for installing two plants was the redundant supply to cover power outages. As part of the installation, the monitoring approach from this thesis was introduced to the building as well. Its aim is to document the reduction in energy consumption and to verify the effectiveness of the upgraded building systems. The FM company in charge of the installation may then use the monitoring data to justify the need for the refurbishment.

4.1.2.1 Main meters

The electricity meter is an ISKRA MT581 (ISKRA emeco) owned by the local utility supplier. The meter can be classified as an analogue meter as it is able to output a pulse for every consumed kWh.

The locally installed gas meter is an Iltron Delta G250 (Delta). Similar to the electricity meter, it is also owned by the utility supplier. The meter is an analogue meter supporting pulse outputs. Each pulse equals 1 m^3 gas consumed.

The main water meter was identified as Sensus Meitwin 100 (Sensus Meitwin). It is also an analogue meter outputting pulses for each 0.1 m^3 water used.

4.1.2.2 Heat meters and sub meters

The two existing gas boilers and one steam boiler are equipped with separate gas sub meters. All three meters are of the same type. They are RMG Terz 91 (RMG Terz). These are analogue meters outputting pulses for each m^3 of consumed gas.

For the CHP plant, additional meters were installed to support monitoring of the CHP. This selective metering enables to determine the actual coefficient of performance (COP). They may also be used to validate operational behaviours of the CHP plants.

The meters are MWK multidata meters which meter heating and cooling in kWh. These meters can be classified as digital meters as they support the M-Bus technology. In addition to their field bus interface, they also provide an interface for pulse outputs. Nonetheless, for this work the outputted pulses were not considered, as the digital communication should always be preferred. Due to their internal memory, the meters keep the meter readings even after a power outage. This further increases the robustness of the measurement concept.

Table 25 and Table 26 summarise the findings from the site survey:

Table 25 Public hospital site inspection main meters

	Electricity Meter	*Gas Meter*	*Water Meter*
Manufacturer	ISKRA	Iltron	Sensus
Type	MT581	G250	meitwin
Connection port	Yes	Yes	Yes
Communication method	Pulse	Pulse	Pulse
Pulse value	1 pulse = 1 kWh	1 pulse = 1m^3	1 pulse = 0.1m^3

Table 26 Public hospital site inspection heat meters

	Existing sub gas meters	*CHP heat meters*
Manufacturer	RMG	MWK
Type	Terz 91	multidata
Connection port	Yes	Yes
Communication method	Pulse	Digital
Pulse value	1 pulse = 1m^3	-
Measured unit	-	kWh
Bus	-	M-Bus
Quantity	3	2

4.1.3 Sports arena

The last pilot building is a multi-function outdoor sport arena located in Frankfurt, Germany. The building and its building systems were rebuilt and renewed in 2006. It has a total square footage of approximately 110,000 m^2. More than half of this space is reserved for underground parking. The outdoor pitch area has approximately 7,000 m^2. Other big areas include office spaces, hallways and staircases, lodges and plant rooms. The arena hosts approximately 300 events per year. 10% are considered major events i.e. the arena is used at its full capacity. These events can be concerts or soccer matches. The total capacity of the site is about 55,000 visitors. In addition to that, the arena has more than 80 VIP lounges which are usually rented for a full year by larger companies. The business area inside the building has 5,700 m^2. The adjacent kitchen area totals to about 800 m^2. It may be used for e.g. seminars, meetings and conventions. About 40 members of staff are working in the building. Their tasks include among others: organizing events, maintaining building systems and to ensure proper building operation. The arena is controlled by a BMS which has access to almost 10,000 data points throughout the building. However, these data points do not monitor any energy consumption. The building has no system in place to monitor energy consumption and the only viable source for consumption data are the monthly utility bills. Additionally, members of staff occasionally tour through the building and read selected meters manually.

Electricity is provided to each area through three technical distribution rooms located in the basement. Three electricity transformers per room are able to handle a total input of up to 10 MW. The highest consumption peak was in 2008 at 2.3 MW. In addition to that, two emergency diesel generators are capable of providing up to 560 kW on site. During major events these diesel generators are operational to cover any outages instantaneously. Any excess energy is fed into the power grid of the electricity provider. In the first three years of operation the power consumption totalled up to 4,500 MWh, 5,200 MWh and 5,500 MWh. This consumption led to operational costs (solely for electricity) of more than half a million Euros per year.

The heat supply is maintained through a double boiler plant serving up to 2,400 kW. The base load is approximately 200 kW. The yearly gas consumption is roughly 5,000 MWh. The main consumer of this huge energy amount is a grass heating system which starts operating whenever the outside temperature falls below a defined temperature threshold set at the BMS. The operation of the grass heating ensures that the pitch remains at healthy quality. Without a grass heating system, the pitch would need to be replaced more frequently. This would not be economically feasible. Besides the grass heating, large consumers of heat are radiators, lounges, kitchen areas and changing areas.

The water circuit supplies hot water to the kitchen area, changing areas, for the permanent staff and other miscellaneous consumers.

4.1.3.1 Site Inspection

A site inspection was conducted to identify the main buildings systems and to locate and determine if they are already monitored by existing meters. Due to its large footprint, the individual plant rooms and the energy transfer points are geographically distributed. This becomes problematic as networks installed for monitoring equipment will not cover long distances. The reach of the network primarily depends on the way of transmission, whereas analogue signals are only certified for up to 10 meters distance. Even if higher distances were available, installation of wiring throughout the building would create enormous investment costs.

Each meter discussed in the next sections will be given a short name for identification purposes.

4.1.3.2 Electricity Meters

Three main electricity meters measuring the supply for the arena were identified. All three meters are owned by the same utility provider.

EL01 and EL02 are of the type EMH LZKJ (EMH) while EL03 is a Landis + Gyr ZMD 410 (Landis + Gyr). Both types of meters support the output of pulses after a connection is made by an electrician of the utilities company. Both types support the configuration of the pulse value within a given range. Therefore, the pulse value cannot be just read from a technical data sheet. Investigating their configuration revealed that each pulse equals 1 kWh.

In addition to the main electricity meters, it is envisaged to also retrieve data from certain sub meters. These can be categorised as follows:

- Meters monitoring the floodlight system used to illuminate events.
- Meters monitoring the energy generation of the emergency generators.
- Meters monitoring the transformers which distribute the energy inside the building.
- Meters monitoring the electricity consumption of the arena’s chillers.

These systems were selected for sub metering as it is believed that these are the main electricity consumers in the arena. This opinion is shared by the operational staff of the building (Sirr, 2012).

An overview about these building systems is given in Table 27.

Table 27 Sports arena electricity sub metering overview

Sub system	*Metering status*
Flood light	The flood light is provided by four different energy circuits from four different rooms. No existing meters were found.
Emergency diesel generators	The two diesel generators used during major events in the arena are both located in the same basement room. No existing meters were found
Transformators	The transformators are located in three different rooms called NSHV1 to NSHV3. Each room is already equipped with Janitza UMG 96 meters (Janitza UMG 96). In total, 9 meters are already installed for monitoring the transformators. These meters are classified as analogue meters.
Chiller	The arena has a chiller system for the business areas and VIP lodges during warm days. Its electricity consumption is currently not measured. In NSHV3, one outgoing circuit is dedicated to the cooling machine. Therefore, it is feasible to install the electricity meter for the chiller within the NSHV3.

4.1.3.3 Gas meter

The main gas meter is owned by the utility supplier. The gas supplier will be required to provide an output of this meter. The meter is an Elster EK 260 (Elster EK 260). Its operation manual reveals that pulse connectivity is available and that 1 pulse = 1 m^3 (Elster operation manual). This meter will be called GA01.

4.1.3.4 Heating and cooling meters

The arena has several areas which consume large amounts of heat. These areas can be described as follows:

- The grass heating system of the arena.
- Kitchen area which caters for events.
- Domestic hot water.
- Cooling load generated by the chiller.

Additionally, the produced heat of the main heat boiler is of interest in order to quantify the remainder which is not covered by the above mentioned large consumers. Table 28 outlines the mentioned areas and their metering status.

Table 28 Sports arena heating and cooling sub metering overview

Sub system	*Metering status*
Grass heating	It is assumed that the biggest consumer is the grass heating. It is also the only system currently equipped with a heat meter. The existing heat meter of the grass heating is a Sensus WP Dynamic 125 (Sensus WP Dynamic). It belongs to the class of analogue meters and is able to output pulses. 1 pulse = 0.01 MWh. The meter will be called WMZ06.
Kitchen area	The kitchen is used to cater for events of all kinds. The heat consumption is currently not monitored.
Domestic hot water	The heat used to provide domestic hot water is currently not metered.
Main heat boiler	The boiler needs to be equipped with a meter. This will allow quantifying the amount of heat which is consumed unmetered when subtracting all metered heat from the boiler readings.
Chiller	The chiller is not metered yet. Metering will allow estimating the load required for cooling during hot periods.

4.1.3.5 Water meters

The main water meter of the arena is a Zenner WPH-N (Zenner WPH). The meter is an analogue meter outputting pulses. 1 pulse = 1 m^3. The meter will be called WA01.

The arena has a couple of high water consumers. These are:

- Domestic hot water supply for the kitchen.
- Domestic hot water supply for the sanitary facilities.
- Collection of rain water.
- Irrigation of the pitch.

The site inspection identified that some of these high water consumers are already equipped with a dedicated meter. These locations and (if available) their meters are discussed in Table 29.

Table 29 Sports arena water sub meters

Sub system	*Metering status*
Domestic hot water supply for the kitchen	The amount of water used to heat the domestic water supply is currently measured with a basic meter. This meter has no connectivity. It needs to be replaced in order to enable monitoring of the consumption
Domestic hot water supply for the sanitary facilities	The consumption of hot water for the sanitary facilities is not monitored. A water meter needs to be installed.
Collection of rain water	The collection of rain water is already metered. The meter is an Actaris Cyble Sensor V2 (Actaris Cyble Sensor) which can be classified as an analogue meter. It has a pulse output where 1 pulse = 1 m^3. It will be called WA04.
Irrigation of the pitch	The amount of water used to irrigate the grass pitch is also metered. It is again an Actaris Cyble Sensor V2 with the same properties as the previous water meter. Its name will be WA05.

Table 30, Table 31 and Table 32 summarise the existing meters in the sports arena categorised by their respective mediums.

Table 30 Sports arena existing electricity meters

	Main electricity meter	*Main electricity meter*	*Transformer meter*	*Transformer meter*	*Transformer meter*
Internal name	EL01, EL02	EL03	EL19 – EL21	EL22 – EL24	EL25 – EL27
Manufacturer	EMH	Landis + Gyr	Janitza	Janitza	Janitza
Type	LZKJ	ZMD 410	UMG 96	UMG 96	UMG 96
Connection port	Yes	Yes	Yes	Yes	Yes
Communication method	Pulse	Pulse	Pulse	Pulse	Pulse
Pulse value	1 pulse = 1 kWh	1 pulse = 1 kWh	1 pulse = 1 kWh	1 pulse = 1 kWh	1 pulse = 1 kWh

Table 31 Sports arena existing gas and heat meters

	Main gas meter	*Grass heating meter*
Internal name	GA01	WMZ06
Manufacturer	Elster	Sensus
Type	EK 260	Dynamic 125
Connection port	Yes	Yes
Communication method	Pulse	Pulse
Pulse value	1 pulse = 1 m^3	1 pulse = 0.01 MWh

Table 32 Sports arena existing water meters

	Main water meter	*Rain water meter*	*Irrigation meter*
Internal name	WA01	WA04	WA05
Manufacturer	Zenner	Actaris	Actaris
Type	WPH-N	Cyble Sensor V2	Cyble Sensor V2
Connection port	Yes	Yes	Yes
Communication method	Pulse	Pulse	Pulse
Pulse value	1 pulse = 1 m^3	1 pulse = 1 m^3	1 pulse = 1 m^3

4.1.4 Building comparison

The site inspection reveals that the installations and requirements for the demonstration buildings are versatile. The university building has only small demands to an EnMS with only three meters installed. The public hospital has eight meters installed which could be considered to be of medium complexity. The sports arena has a total of 17 meters already installed. Additionally, several sub systems show potential for monitoring which will require the installation of further meters. The scope of the monitoring concept for this building is large. Table 33 summarises the findings from the three pilot buildings

Table 33 Pilot building comparison

	University building	*Public hospital*	*Sports arena*
Number of meters	3	8	17
Further meters required	No	No	Yes
Complexity	small	medium	large
Meter communication methods	Pulse	Pulse, M-Bus	Pulse

4.2 Application of the metering concept

This section covers how the methodical concepts described in chapter 3.1 are implemented. It outlines the standard approach used in all three demonstration buildings.

4.2.1 Splitting pulses

In this work, the splitting of pulses was realised with KV001A devices from Relay (Relay KV001A). Through splitting, the original pulse gets duplicated. The first pulse will still be sent to its original destination. The second pulse can be utilised for other purposes. To be compliant with the methodology outlined in chapter 3, the split pulse will be next to M-Bus protocol using a converter module.

4.2.2 Converting pulses

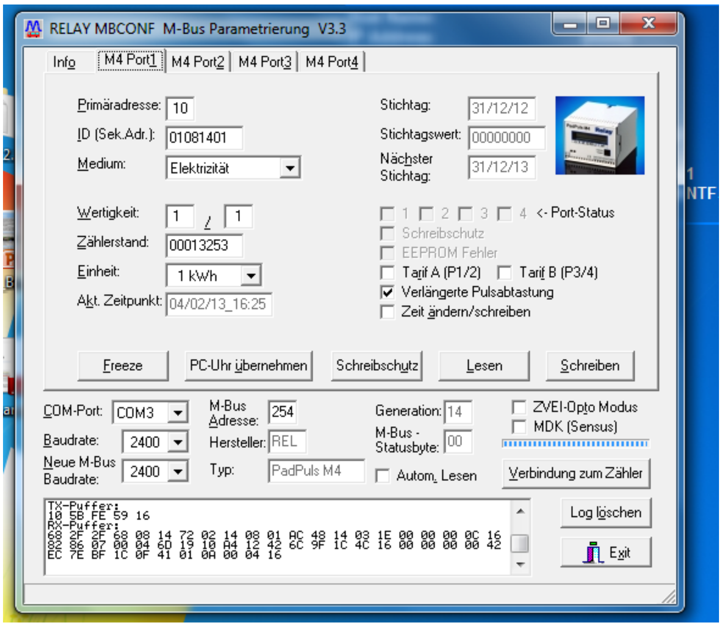
This conversion of analogue pulses is achieved through the PadPuls M4L modules from Relay (Relay PadPuls). These modules have four inputs for pulse signals which can be converted to M-Bus. The PadPuls M4L requires an initial configuration which can be done with a GUI tool written for Windows operating systems.

As an example, the configuration of a meter in the PadPuls converter unit is discussed below. Here, the main electricity meter from the university building is presented. Table 34 summarises its attributes needed for configuration. It also provides a translation from German for the relevant terms.

Table 34 Sample electricity meter attributes

German	*English translation*	*Meter 1 (Electricity)*
Primäradresse	Primary address	10
Sekundäradresse	Secondary address	01081401
Medium	Medium	Electricity
Wertigkeit	Transformer ratio	1 / 1
Zählerstand	Meter reading	13253
Einheit	Meter unit	1 kWh
Akt.Zeitpunkt	Timestamp	04/02/13_16:25

Figure 4-2 depicts the configuration in the PadPuls converter module. The electricity meter was configured with the primary M-Bus address 10 and the secondary address 01081401. The medium was specified as electricity. The transformer value was set to 1/1 and the unit to 1 kWh. Therefore, each pulse will equal 1 kWh. The local time was saved so that meter readings passed onto the M-Bus can be time stamped correctly.



Primäradresse: 10
ID (Sek.Adr.): 01081401
Medium: Elektrizität
Wertigkeit: 1 / 1
Zählerstand: 00013253
Einheit: 1 kWh
Akt. Zeitpunkt: 04/02/13_16:25

Stichtag: 31/12/12
Stichtagswert: 00000000
Nächster Stichtag: 31/12/13

□ 1 □ 2 □ 3 □ 4 <- Port-Status
□ Schreibschutz
□ EEPROM Fehler
□ Tarif A (P1/2) □ Tarif B (P3/4)
☑ Verlängerte Pulsabtastung
□ Zeit ändern/schreiben

COM-Port: COM3
Baudrate: 2400
Neue M-Bus Baudrate: 2400
M-Bus Adresse: 254
Hersteller: REL
Typ: PadPuls M4
Generation: 14
M-Bus - Statusbyte: 00
□ Autom. Lesen
□ ZVEI-Opto Modus
□ MDK (Sensus)

```
TX-Puffer:
10 5B FE 59 16
RX-Puffer:
68 2F 2F 68 08 14 72 02 14 08 01 AC 48 14 03 1E 00 00 00 0C 16
82 86 07 00 04 6D 19 10 A4 12 42 6C 9F 1C 4C 16 00 00 00 00 42
EC 7E BF 1C 0F 41 01 0A 00 04 16
```

Figure 4-2 Configuration of an electricity meter in the PadPuls converter module

4.2.3 Covering long distances

For establishing connectivity across large distances, off-the-shelve commercial solutions exist. For this thesis, one standard device was chosen. This device is able to convert up to 20 connected M-Bus meters to ModBus and transmit these readings over TCP/IP. If required, there are versions with support for up to 254 meters. A free software package which allows the configuration of the gateway is provided by the vendor (Wachendorff ADFWeb).

Figure 4-3 from the user manual provides an overview of its functionality. It depicts the gateway as a central element for the communication. Multiple meters are connected to it and there readings are converted to Modbus TCP and communicated to various destinations, even via Internet.

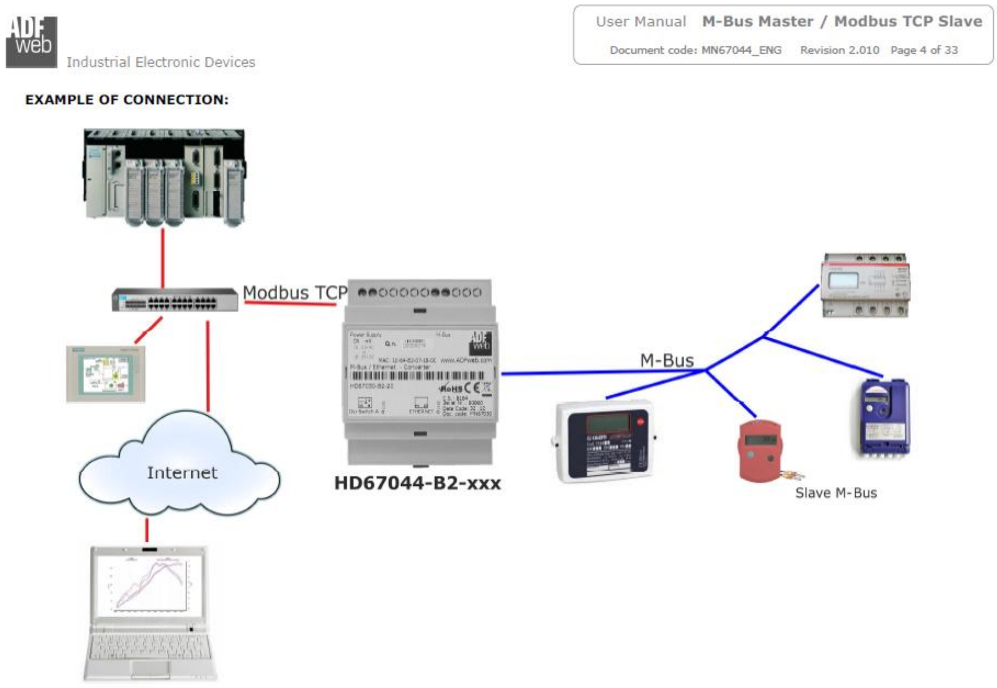

ADF web
Industrial Electronic Devices
User Manual M-Bus Master / Modbus TCP Slave
Document code: MN67044_ENG Revision 2.010 Page 4 of 33
EXAMPLE OF CONNECTION:
Modbus TCP
Internet
HD67044-B2-xxx
M-Bus
Slave M-Bus

Figure 4-3 Wachendorff ADFWeb example from its user manual, (Wachendorff ADFWeb)

4.2.4 Data logger configuration

In this PhD thesis, an off-the shelve data logger was utilised. This data logger is a Saia PCD1 which has been chosen due to its variety of protocols supported. Additionally, the data logger is programmable through so-called FUPLA diagrams. FUPLA is an abbreviation for functional plan. FUPLA allows programming the data logger in a graphical way. This is done by selecting and connecting pre-defined components available from a library. No source code needs to be developed to customise the data logger. Configuration of the data logger is required to obtain and relay correct information (Saia Burgess, 2016). The graphical programming is considered faster and easier to learn compared to text-based programming (Rogers & McVay, 2012).

4.2.4.1 Meter configuration

An example configuration of an electricity meter can be seen in the FUPLA diagram in Figure 4-4. Here, two components were selected, namely the "PadPuls M1-4" and the "Custom Counter" components. The first component depicts a pulse to M-Bus converter module while the latter component is a software meter counting and storing the acquired information. The "PadPuls M1-4" is one of many products which can be chosen from the built-in library. During configuration, one needs to assure that the library is selected which corresponds to the meter. Misconfiguration is likely to result in a failure to communicate with the meter.

Components have their available inputs on their left side while their outputs are on the right. It is not required to use any outputs if it is not required by the desired functionality. However, the software requires that each input is wired. Unrequired inputs can be fed with "0" to disable them. In the example seen in Figure 4-4, only the input labelled "Count1" is needed, therefore any further inputs are wired with "0". The signal connected to "Count1" has a tag "LG09_EL1" which is a name tag that gets attached to the acquired data. It is used to relate the readings to their source. The "H" on each component's input side is short for "High" and expresses that the component is enabled.

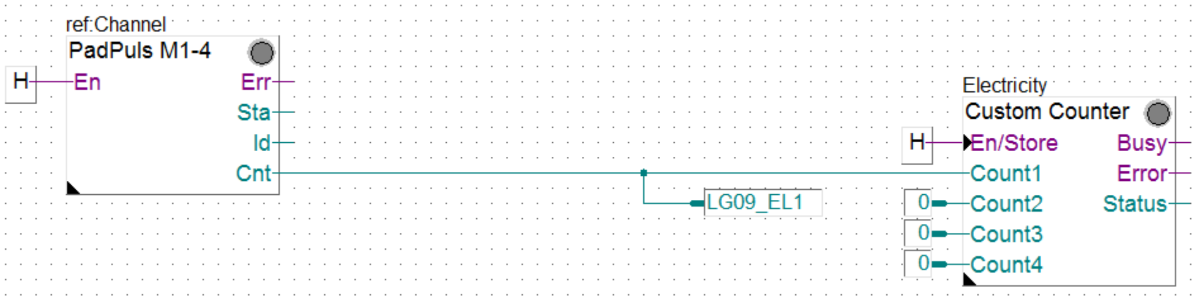
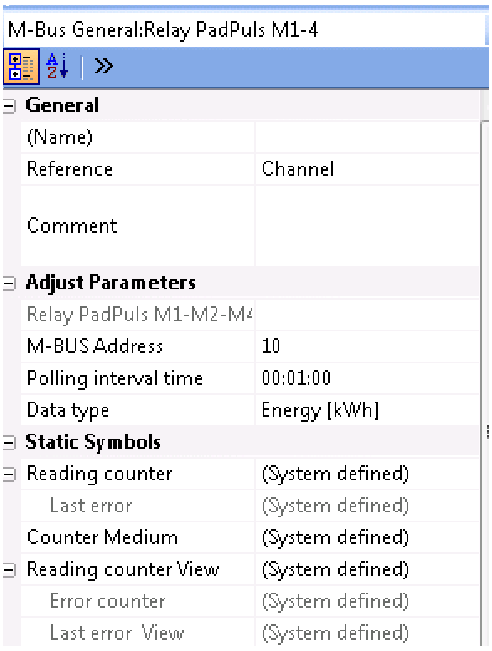


Figure 4-4 FUPLA sample meter configuration

The properties dialog of the Custom Counter can be seen in Figure 4-5. It enables the configuration of the primary M-Bus address (10). In addition, the polling interval (1 minute) and the data type (kWh) can be set.



M-Bus General:Relay PadPuls M1-4

General	
(Name)	
Reference	Channel
Comment	
Adjust Parameters	
Relay PadPuls M1-M2-M4	
M-BUS Address	10
Polling interval time	00:01:00
Data type	Energy [kWh]
Static Symbols	
Reading counter	(System defined)
Last error	(System defined)
Counter Medium	(System defined)
Reading counter View	(System defined)
Error counter	(System defined)
Last error View	(System defined)

Figure 4-5 PadPuls properties in FUPLA

4.2.4.2 Storage of meter readings

The aim of the data logger is to store acquired meter readings in its file system. This functionality is provided by modules of the data logger. The configuration is similar for each type of energy that is being monitored.

The acquired meter readings are stored locally in .CSV files. These are maintained by a FUPLA module called HDLog. Their setup is the same for each meter. Exemplary, the setup for the electricity meter is shown in Figure 4-6.

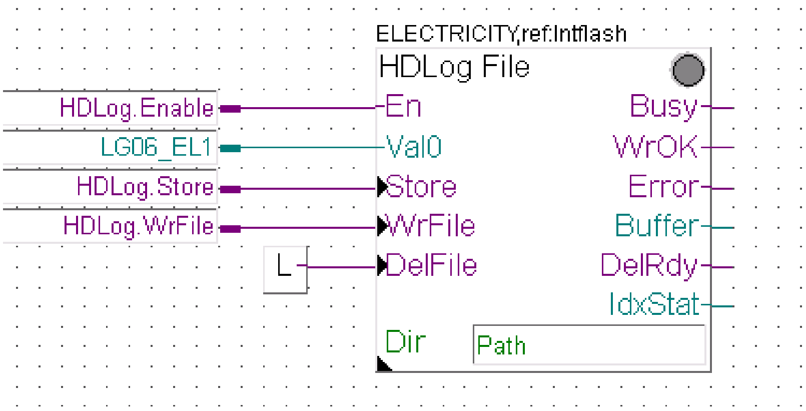
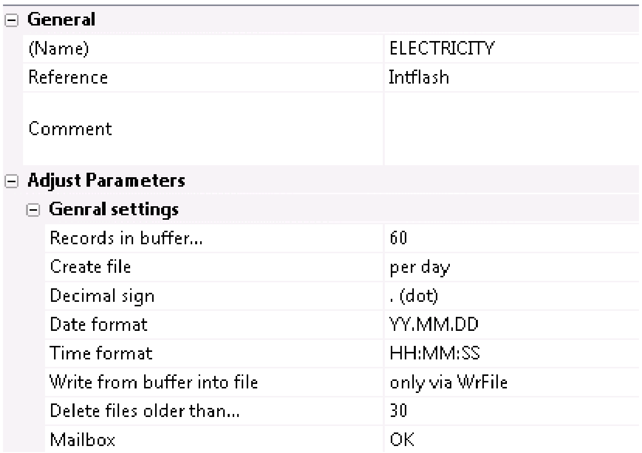


Figure 4-6 Configuration for storing meter readings in FUPLA

As an example, the electricity meter is set up as illustrated in Figure 4-7 where all properties are enlisted. The important setting in this dialog is the name, as this will become part of the filename. Furthermore, it is defined that each day a new file is created, and the date and time format is specified.



⊟ **General**	
(Name)	ELECTRICITY
Reference	Intflash
Comment	
⊟ **Adjust Parameters**	
⊟ **Genral settings**	
Records in buffer...	60
Create file	per day
Decimal sign	. (dot)
Date format	YY.MM.DD
Time format	HH:MM:SS
Write from buffer into file	only via WrFile
Delete files older than...	30
Mailbox	OK

Figure 4-7 FUPLA HDLog properties

It is also noteworthy that in this setup, files older than 30 days are automatically deleted. This ensures that the data logger never runs out of free space to hold new meter readings. To limit the read/write access to the file system, a buffer (minimum value: 60) is in place. This can be overridden by defining the "Write from buffer into file" directive with "only via WrFile". This assures that acquired readings are saved instantaneously. Thus, data can be provided quicker to the communication layer. The buffer is only needed for applications with high read/write needs.

4.2.4.3 Synchronize timing

For later analysis of the captured meter data, it is important that all data is captured in the same interval and also at the same time. If for example one main and one sub electricity meter are recorded in a 15 minute interval, but their recordings are shifted by 5 minutes, the consumptions cannot be directly compared without certain inaccurateness. To illustrate, one may think of the following scenario: A main meter indicates a peak in consumption but the corresponding sub meter indicates the corresponding peak at a slightly off-set time. In a set up where only two meters are monitored, it might be easy to consider the time shift when analysing data. However, with increased monitoring complexity, one loses the ability to accurately allocate the consumption any longer.

Timing is realised through the data logger's internal clock. It is battery buffered similar to PCs so that it does not lose its clock settings after a power outage. To enforce readings at the same time, additional FUPLA logic is required which can be seen in Figure 4-8. The "Div" module divides the minutes (provided through input A) passed by 15 and outputs the remainder on its “A” labelled exit. The remainder is always zero every 15 minutes. A simple “Cmp” module compares the output of “Div” to 0 to verify if another 15 minute interval needs to be saved to file. A limitation of this concept is that the timing is done through the internal clock and not synchronized centrally. The benefit of central syncing would be a synchronisation across many data loggers and consistent timestamps in the database.

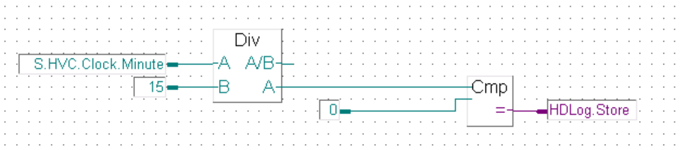


Figure 4-8 FUPLA clock programming

4.3 Installation concepts

This chapter will give a schematic overview about each installation steps executed in the demonstration buildings. Table 35 highlights the differences between the three buildings.

Table 35 Demonstration building differences

	University building	*Hospital building*	*Sports arena*
Splitting required	Yes	No	Yes
Further meters installed	No	Yes	Yes
Data transfer via	FTP over Internet	FTP over 3G	FTP over 3G
Information exchange through local LAN needed	No	No	Yes
Total number of meters	3	8	29

4.3.1 Installation in the university building

The installation in the university building is solely of the monitoring type where pulses are converted to M-Bus. The data logger is directly connected via M-Bus protocol. The monitoring type is depicted in Figure 4-9. This schematic is applicable to all three university meters.

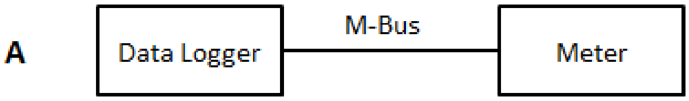


Figure 4-9 University building - Monitoring concept

4.3.2 Installation in the public hospital

For this pilot building, the data logger will be connected to a total of 8 meters. Out of these, 6 meters were already installed previously. The newly installed meters monitor the two CHP units. They will be connected directly using M-Bus, while the existing analogue meters get their pulses converted to M-Bus through the PadPuls converter unit. No splitting of impulses is needed for this building. The required monitoring types are visualised in Figure 4-10. Two meters were connected through type A, six meters through type B.

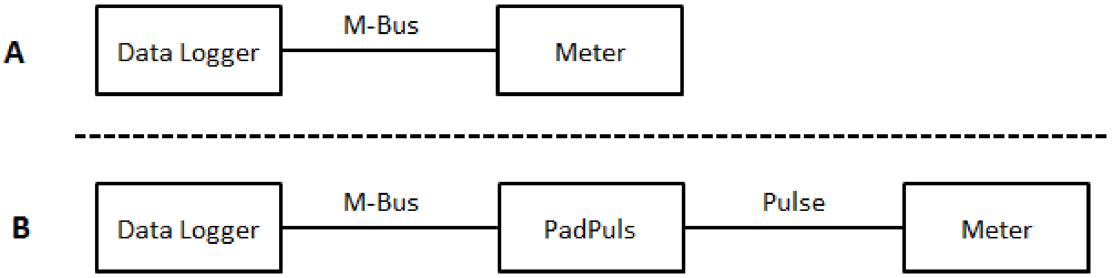


Figure 4-10 Public hospital – Monitoring concept

4.3.3 Installation in the sports arena

Due to its size, it was not feasible to interconnect all meters through a single data logger. Instead, the building's LAN is utilised to communicate meter readings via Modbus TCP. In most rooms, multiple meters are present. These are bundled and their readings are collected by a gateway device. Four rooms only required a single meter, in this case a meter capable to communicate via Modbus TCP was chosen.

The monitoring concept is outlined in Figure 4-11. A total of 29 meters are included in the monitoring solution. Out of these, no meter was connected directly to the data logger by M-Bus. Instead, all readings get converted to Modbus/TCP and send through the LAN. Type C was implemented 12 times and type D 17 times.

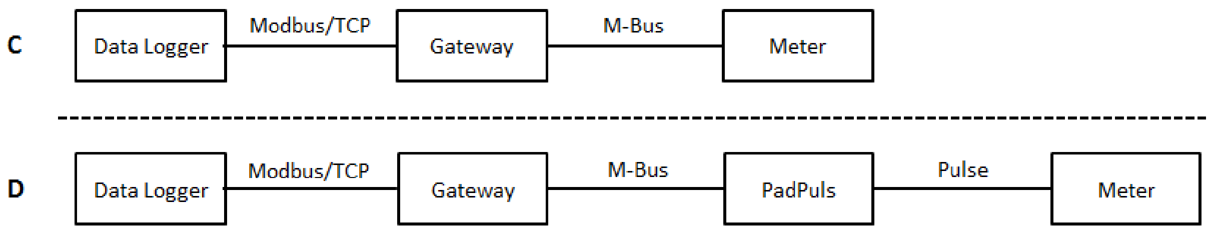


Figure 4-11 Sports arena LAN utilisation

Table 36 summarises how often each monitoring type has been implemented in each pilot building.

Table 36 Monitoring type occurrences

	Type A	Type B	Type C	Type D
University Building		3x		
Public hospital	2x	6x		
Sports arena			12x	17x
Total	**2x**	**9x**	**12x**	**17x**

4.4 Post installation diagnosis

This section summarises faults which occurred during the installation of monitoring equipment. It relates these to the definition of the Potential error categories from chapter 3.1.9 which should be avoided during the installation of monitoring equipment. During the installations for this research, several faults happened which can be classified by the before discussed error categories.

Table 37 discusses the occurred errors.

Table 37 Occurred errors during implementation

Error category	Fault description
PadPuls configuration	The PadPuls converter unit transmits converted values per default in the BCD format (Binary coded decimal). The data logger falsely interpreted these as integers. Additionally, transmitted values were set incorrectly. The application of factors at PadPuls level should be avoided. This was done to unify the pulse output with other meters. While this might have made sense at the time of the installation, it turned out to be a bad decision as it effectively removes detail from data analysis.
Analogue meters	One meter without vender labelling and printing had to be characterized by an electrician in order to set its proper transformer ratio and unit.
Digital Meters	Incorrect registers were read from M-Bus meters, e.g. volume flow instead of heat measured. Some M-Bus meters transmit their readings without decimal point. E.g. 18.7 °C is transmitted as 187. These factors need to be amended in order to relay correct information.
Data logger	Incorrect labelling in FUPLA led to misinterpretation.
Bad reception	In one instance, an external antenna had to be installed in order to provide a stable Internet connection.
Bad preparation	Faulty wiring in a cabinet led to a power failure at the data logger.
Bad hardware	In one installation a power supply unit failed shortly after installation. As no spare unit was available, it delayed the whole configuration.
Handling of values	Some meters measure consumption in Wh. High consumers can easily have readings larger than 4,294,967,296 – which is the maximum for a 32 bit integer. This leads to an overflow and therefore results in values far smaller than the actual meter reading. A solution for this is to change the measurement to kWh. This results in figures smaller by a factor of 1000.

4.5 Data cleansing and anomaly identification

This section discusses results to the application of the gas interpolation algorithm introduced in chapter three. It is found that the data analysis from the university building and the sports arena will deliver good results while the data from the hospital building demonstrates the limitations of the algorithm. Interestingly, missing data is less relevant, instead data of low granularity badly affects analysis.

4.5.1 Example 1 – university building

Figure 4-12 visualises a load curve of gas consumption from the university building. Data of the university building is available from February 2013 to January 2014. In the plot, one can see that sections of data are missing.

The reason for the data loss was a power outage at the data logger. Due to no alarm functionality, this was only noticed three weeks after the outage. All data from that period is lost. With alarm functionality in place, the gap from outage to detection could be minimised. Sections, in which data is missing, are marked in red on the ordinate. The visual of the load curve is unsatisfactory due to a very high consumption spike just before the 01/2014 mark.

This high spike is the result of missing meter readings over a longer period of time. After this issue was identified and fixed, the difference between the last recorded meter reading and the new meter reading was about 330 kWh. The consumption of the building usually varies between 1 kWh and 5 kWh. A difference of 330 kWh between meter readings therefore results in this huge spike.

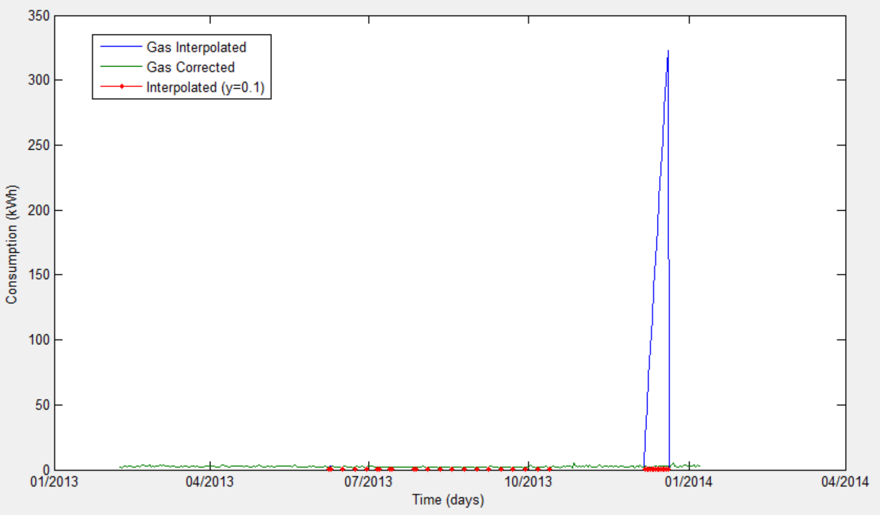


Figure 4-12 Interpolation of gas consumption data

After the initial visualisation, the interpolation algorithm introduced in chapter 3 is applied to the data. Its result can be seen in Figure 4-13. Here, all sections which required interpolation are marked with green on the ordinate. The load curve now has a clear and readable visual. It confirms that the building consumption fluctuates between 1 kWh and 5 kWh. One can see that gas consumption is lower in the summer months than in the winter months.

The graph also illustrates that consumption is only measured at a 1 kWh resolution. This can be seen by the stepped appearance of the load curve. Fractional values only appear where interpolation has taken place. This can be seen especially at the position where the huge spike was located before interpolation.

The 1 kWh resolution is a limitation to deeper analysis. It is not possible to clearly identify any differences in gas consumption for weekdays and weekends or daytime and night time. A higher monitoring resolution, e.g. 0.1 kWh would result in a more detailed consumption curve. This low granularity results from the PadPuls configuration done during the installation of the monitoring equipment. A modification of the monitoring parameters as outlined in the previous chapter would allow the acquisition of data in a higher granularity.

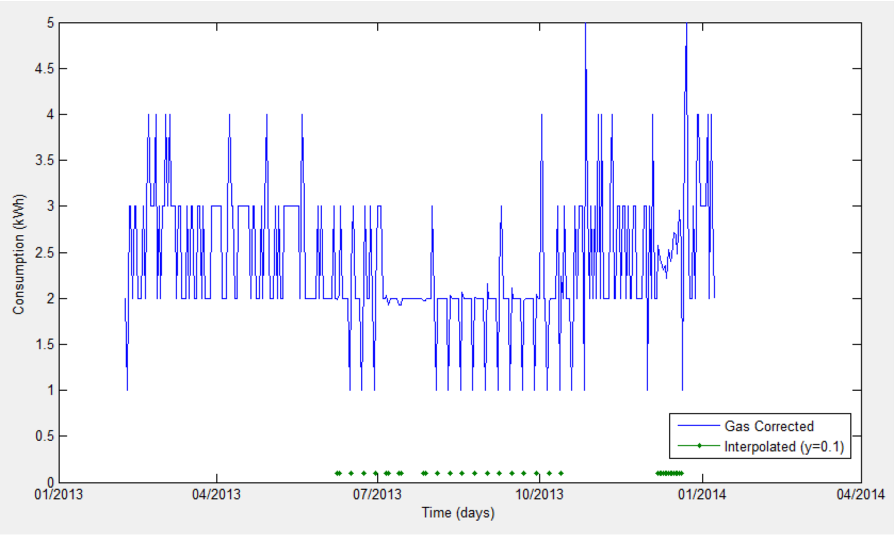


Figure 4-13 Corrected gas consumption

4.5.2 Example 2 – sports arena

The arena features high gas consumption with peaks up to 2,300 kWh. Its load curve can be seen in Figure 4-14. Here, data from autumn 2011 to spring 2014 is graphed.

The visual of the raw gas consumption has the typical layout of a heat consumer where summer months are significantly lower in consumption than winter months. Gaps and outliers are barely visible in the figure. Upon closer inspection of the data, it is revealed that consumption is measured in 1 kWh steps. While technically at the same granularity as in the previous example, the strong deviation of the consumption (between 0 kWh and 2500 kWh) allows analyses at a much more detailed level.

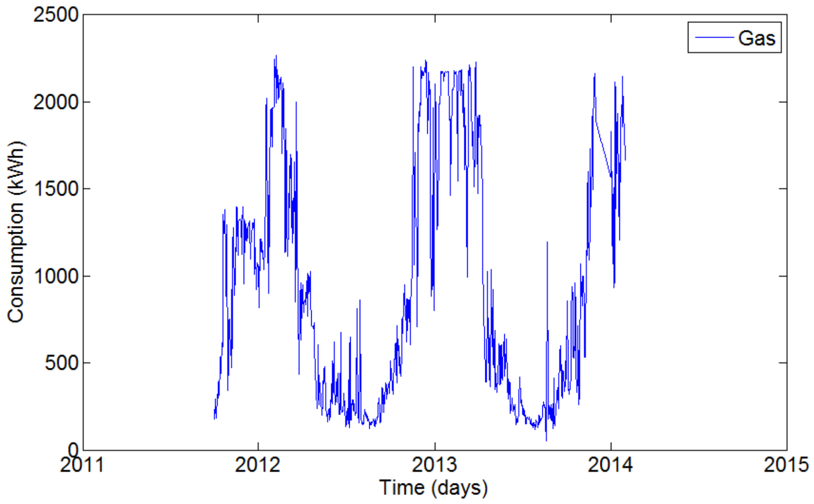


Figure 4-14 Arena gas consumption

The good data quality delivers a heat curve which is suitable for the interpolation of the few bad data sets, see Figure 4-15. The heat curve is calculated over the entire data set of the arena.

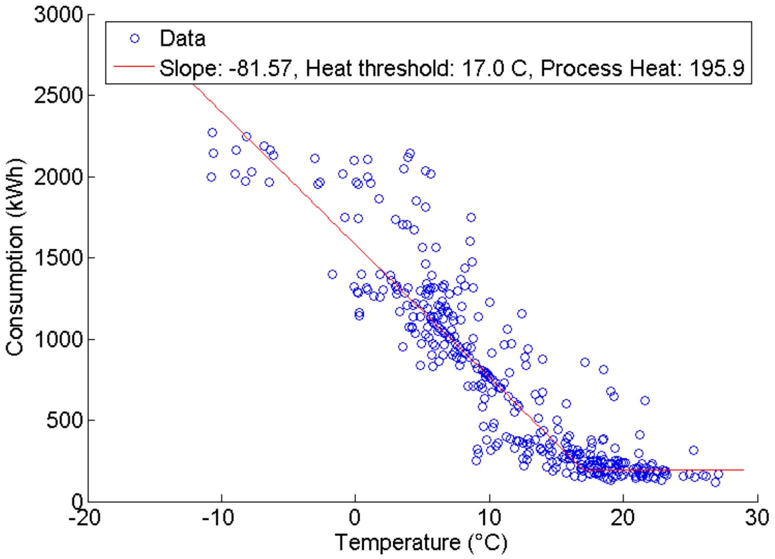


Figure 4-15 Heat curve of the sports arena

Areas of the load curve highlighting measurement gaps can be seen in Figure 4-16. In Figure 4-17, these gaps were filled through application of the interpolation algorithm. A section of magnified data and the application of the interpolation algorithm can be seen in Figure 4-18 and Figure 4-19. Gaps present in Figure 4-18 are no longer present in Figure 4-19. Instead, replacement values provided by the interpolation algorithm and derived from the heat curve function were inserted into the data.

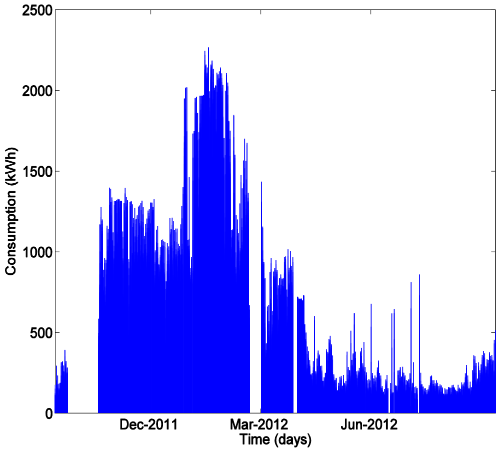


Figure 4-16 Raw gas consumption

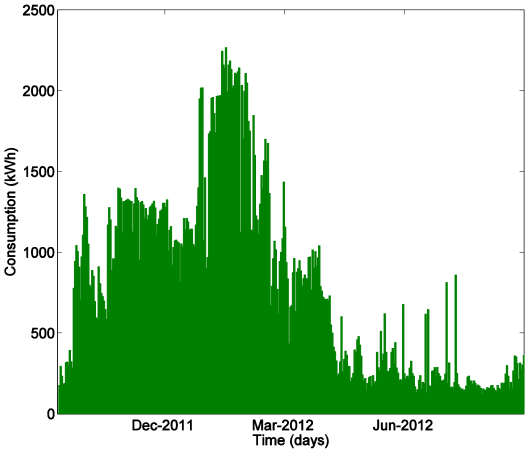


Figure 4-17 Corrected gas consumption

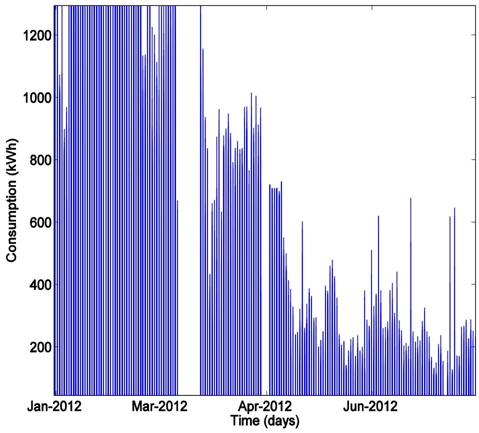


Figure 4-18 Magnified raw gas consumption

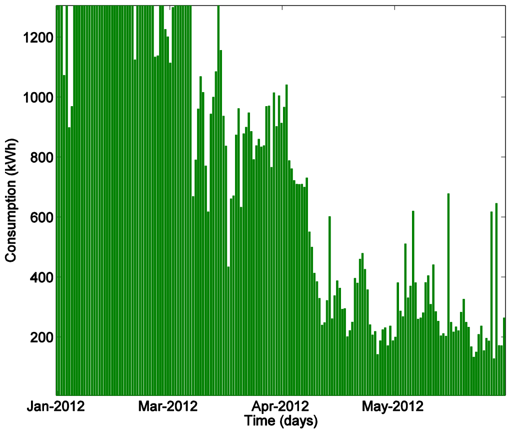


Figure 4-19 Magnified corrected gas consumption

4.5.3 Limitations of the interpolation algorithm

Figure 4-20 and Figure 4-21 show the gas consumptions of the hospital building. In this example, data is available from August 2013 to February 2014. In Figure 4-20, few large spikes exist which limit the readability of the consumption data. In Figure 4-21, these large outliers were interpolated. The locations where the interpolation took place are marked in green on the ordinate. The graph reveals a monitoring solution of 100 kWh. This is a hundred times lower than the two previous examples.

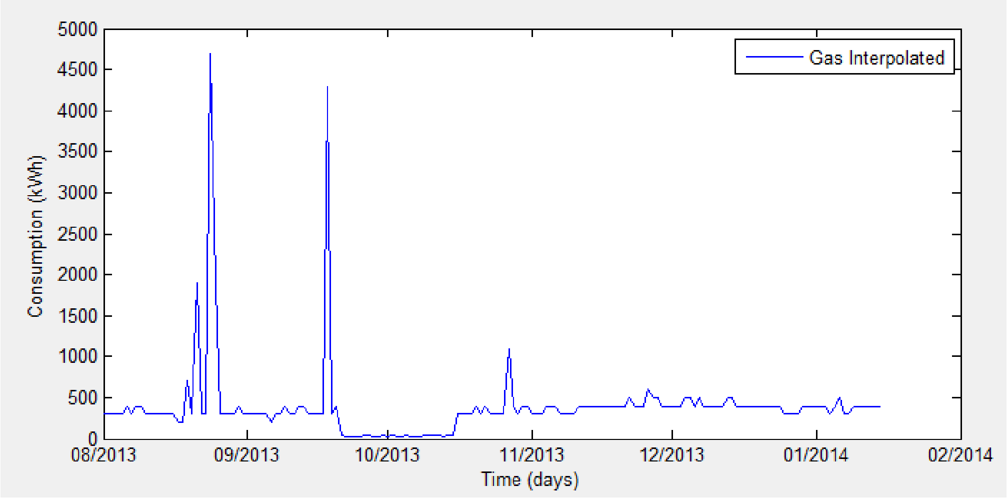


Figure 4-20 Raw gas consumption

Due to the extremely low monitoring granularity, the interpolation of missing data and outliers are beyond meaningful values. The algorithm fails to detect a proper heat curve. This can be seen in Figure 4-22 where values are placed at 100 kWh apart. Yet, the interpolation algorithm still calculates a slope and replaces outliers with values derived from the heat curve function. However, its accuracy is questionable.

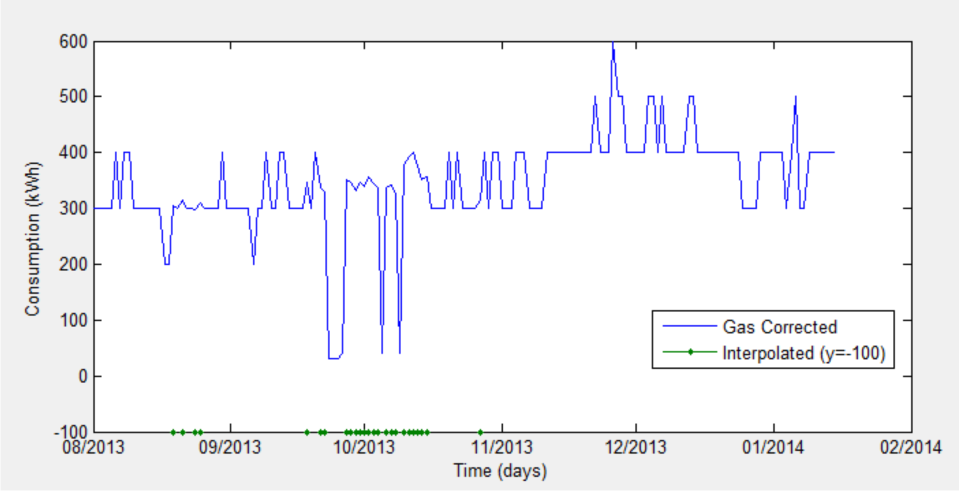


Figure 4-21 Cleansed raw consumption

While this result is not desired for analysis purposes, it succeeds in demonstrating a limitation of the introduced methodology. Whenever the monitoring granularity is too low, a plausible heat curve function cannot be produced by the interpolation algorithm.

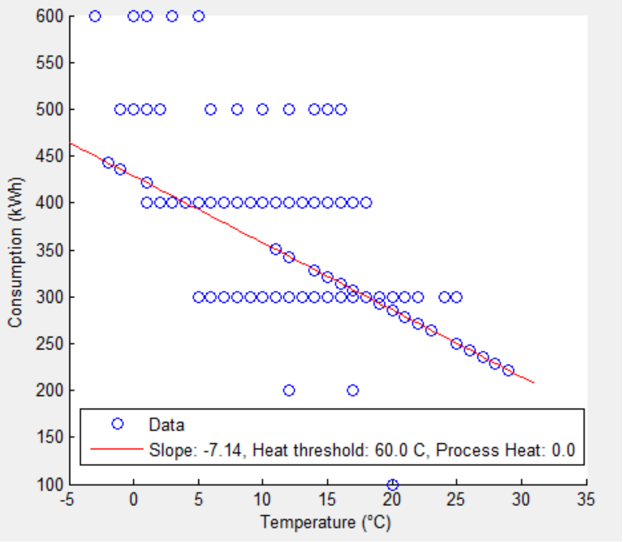


Figure 4-22 Failed heat curve

As seen in Figure 4-22, a heat curve similar to Figure 3-6 cannot be produced. Due to the low resolution of the monitoring, values are apart by 100 kWh steps. Some outliers were still identified and replaced with approximated values. These can be easily spotted as they are located on top of the red slope.

5 Data Analysis

This chapter highlights selected analyses from the data retrieved by the experimental installations in three pilot buildings. These findings are based on the methodologies introduced in chapter 3.

The structure of this chapter is as follows:

- A short introduction to a MATLAB analysis tool and its functionality to support initial visual data analysis.
- The clustering of load curve data from each pilot building is presented and its output is analysed.
- The KPI analysis of selected data aggregated in the DWH is presented and discussed.

The SQL scripts developed to create and populate all dimensions and facts for the DWH are in appendix 2 to 6.

The source code for the MATLAB applications can be found in appendix 8.

This work promotes new analysis methodologies which are applicable to any building fitted with monitoring equipment. It extends existing energy analysis methods.

5.1 Analytical Functions

The method and related tool introduced in this section were developed for initial visual analysis. Due to its flexibility, it can import data from various sources, such as a DWH, and visualise analysis results.

The tool was equipped with two different ways to acquire data.

a) The preferred way is direct connectivity to the DWH system. This can be achieved through the MATLAB Database Toolbox. Using this add-on, the program reads selected data from the DWH into the MATLAB workspace to enable any kind of analysis.

b) As a second option, the tool supports the import of CSV files. This file based approach does not require the MATLAB database toolbox. It can be considered as a stand-alone solution which may be used when database connectivity is not available.

The following MATLAB snippet handles the data import from the DWH:

Listing 9 Create database connection

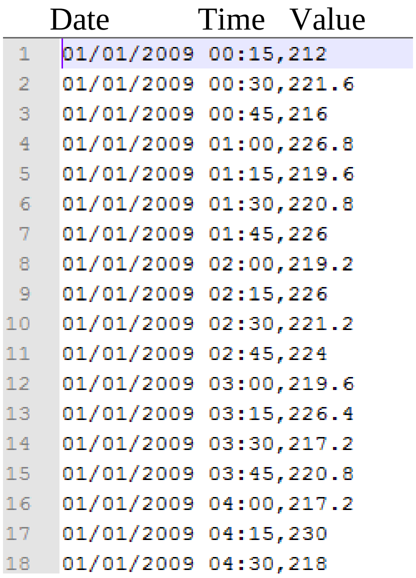
```
% connect to database
conn = database('orcl','USER','PW','oracle.jdbc.driver.OracleDriver',
'jdbc:oracle:thin:@zuse3.ucc.ie:1521:');

% retrieve data
curs3 = exec(conn,'SELECT * FROM ERI_ELEC WHERE ELEC < 150 ORDER BY time
ASC');
curs3 = fetch(curs3);
data3 = curs3.data;

% save readings and timestamps in two variables
RAW_ELEC = data3(:,2);
RAW_ELEC = cell2mat(RAW_ELEC);
RAW_ELEC_T = data3(:,1);
RAW_ELEC_T = datevec(RAW_ ELEC_T,'yyyy-mm-dd HH:MM:SS.FFF');
```

This source code is divided into three sections; (i) establish connection with the DWH, (ii) retrieve data and (iii) format data and save into workspace.

MATLAB's CSV import can be configured to meet the requirements for virtually any CSV file. The above code snippet can read a CSV file i.e. of the type seen in Figure 5-1. The code allows specification of the used delimiter and the formatting of the date string.



```
01/01/2009 00:15,212
01/01/2009 00:30,221.6
01/01/2009 00:45,216
01/01/2009 01:00,226.8
01/01/2009 01:15,219.6
01/01/2009 01:30,220.8
01/01/2009 01:45,226
01/01/2009 02:00,219.2
01/01/2009 02:15,226
01/01/2009 02:30,221.2
01/01/2009 02:45,224
01/01/2009 03:00,219.6
01/01/2009 03:15,226.4
01/01/2009 03:30,217.2
01/01/2009 03:45,220.8
01/01/2009 04:00,217.2
01/01/2009 04:15,230
01/01/2009 04:30,218
```

Figure 5-1 Sample CSV file for MATLAB import

Listing 10 Read CSV files

```
% define CSV layout
delimiter = ',';
dateformat = 'dd/mm/yyyy HH:MM';
csvfile = './loadcurve.csv';

% read CSV into workspace
fid = fopen(csvfile);
DATA = textscan(fid,'%s %f', 'delimiter', delimiter);
fclose(fid);

% save readings and timestamps in two variables
RAW_ELEC = DATA{2};
RAW_ELEC_T = datevec(DATA{1}, dateformat);
```

This source code is again divided into three section; (i) defines the layout of the timestamp and the type of delimiter, (ii) opens and reads the CSV file and (iii) saves each parsed column into the workspace.

The returned data from both acquisition options is in the same structure, allowing working with a unified data format.

A selection of analysis methods was implemented. These will be briefly introduced in the next section.

5.1.1 Visualisation of the load curve

A load curve is the chronological representation of consumption values. Load curves can be visualised for both meters and sensors. It is a basic visualisation which provides a first glance at the data quality and consumption peaks.

An example of the arena's load curve is depicted in Figure 5-2. This plot depicts the load curve of a single meter. In this data representation the x-axis depicts all samples taken at all given timestamps. With data recorded in 15 minute intervals, one gets 35,040 readings per year. In this example, we see $11 * 10^4 = 110{,}000$ values, which is roughly three years of data.

This data representation can be useful for a first inspection of data. Figure 5-2 e.g. reveals consumption peaks. Obvious gaps in the data cannot be seen in this example.

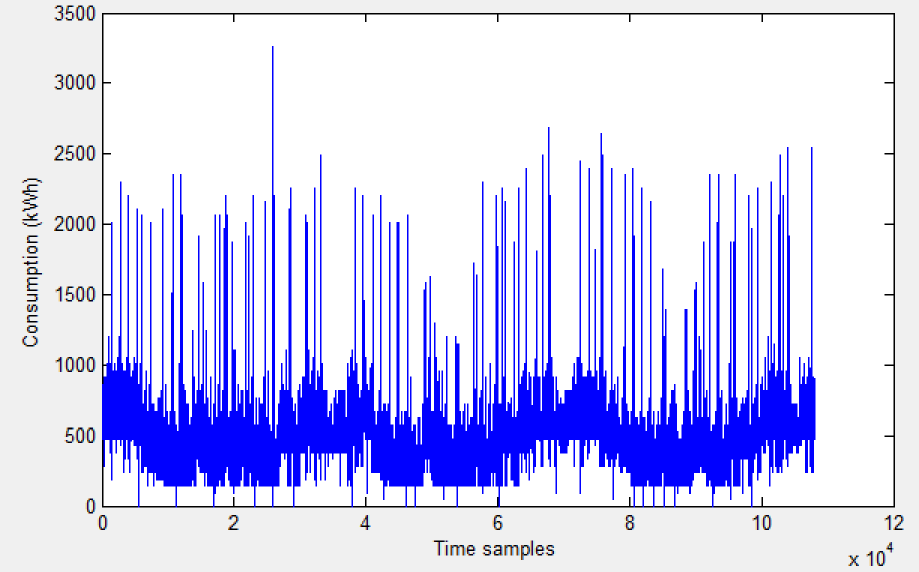


Figure 5-2 Electricity load curve of the arena

Daily consumption varies roughly between 400 kWh and 1.200 kWh daily. However, many peaks even higher than 2.000 kWh exist. Due to the nature of the building's usage, these peaks can be linked to the major sport events that are hosted in the arena throughout the year. During these events, the arena's flood lighting is switched on, which most likely causes the peaks. The corresponding source code is given in the listing below.

Listing 11 Plot load curve

```
% plotting all values over time
plot(RAW_ELEC);
ylabel('Consumption (kWh)');
xlabel('Time samples');
```

5.1.2 Determination of daily maxima's

This method plots for each day only the highest recorded value. This data representation aids in a better identification of daily peaks. It may also be used for energy procurement purposes where the definition of the maximum load is a price component.

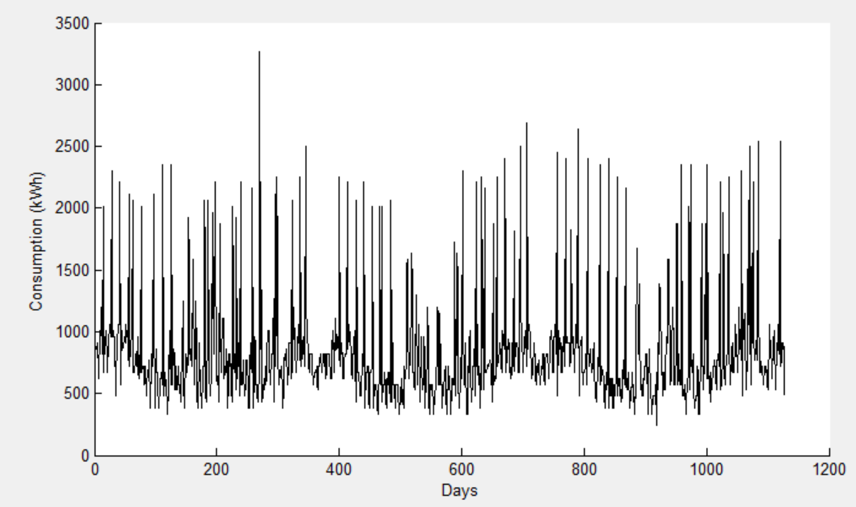


Figure 5-3 Plot of daily maxima's

When comparing the example in Figure 5-3 with Figure 5-2, one can easier identify the daily peaks, especially days with lower consumption. Figure 5-3 reveals that daily peaks are at least 400 kWh. This information cannot be derived from the data representation given in Figure 5-2 due to all values being plotted and not just the maxima's.

This plot is realised through MATLAB's statistic functions. All recorded time series values are separated on a daily basis and then the highest value gets selected through MATLAB's "grpstats" function for each day.

Listing 12 Determine daily maxima's

```
% identify the highest value per day using MATLABs statistic functions
elec_day = datenum(elec(:,1:3));
daily_max = grpstats(elec(:,8), elec_day, {'max'});

% plot and label axes
plot(daily_max, 'color', 'black');
ylabel('Consumption (kWh)');
xlabel('Days');
```

5.1.3 Total daily consumption analysis

Unlike the previous method, all readings from a day are summarised and their total consumption is visualised. Therefore, the graph in Figure 5-4 identifies the total daily consumptions which could help e.g. in energy planning. The total daily consumption is also a key price component in energy procurement. Lastly, the spread between the daily base load and the daily peak can be easily determined by visually comparing minima to maxima.

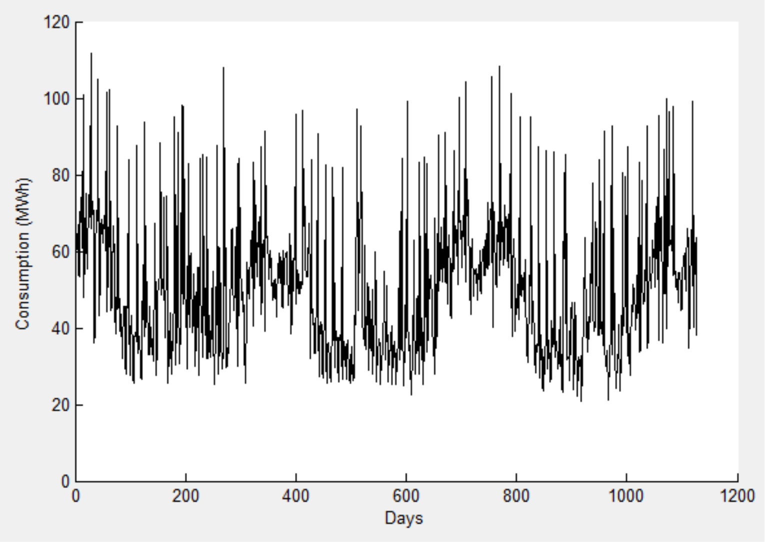


Figure 5-4 Daily total consumption

In Figure 5-4, the daily base load can be roughly derived as 20-50 MWh, while the peak load fluctuates between 50 MWh and 110 MWh. The spread between daily base load and daily peak varies between 20 MWh and 80 MWh.

Similar to the previous analysis method, the MATLAB function “grpstats” is invoked to summarise all values for each individual day.

Listing 13 Calculate total consumption

```
% calculate total consumptions per day using MATLABs statistic functions
elec_day = datenum(elec(:,1:3));
daily_total = grpstats(elec(:,8), elec_day, {'sum'});

% plot and label axes
plot(daily_total, 'color', 'black');
ylabel('Consumption (MWh)');
xlabel('Days');
```

5.1.4 Annual load curve analysis

The annual load curve is a plot of the same data as for the regular load curve. However, for an annual load curve, all consumption readings are sorted descending by their size.

The plot of an annual load curve can be used e.g. to easily identify the base load of a building. Energy consultants often use this kind of data representation to plan combined heat and power plants (CHP).

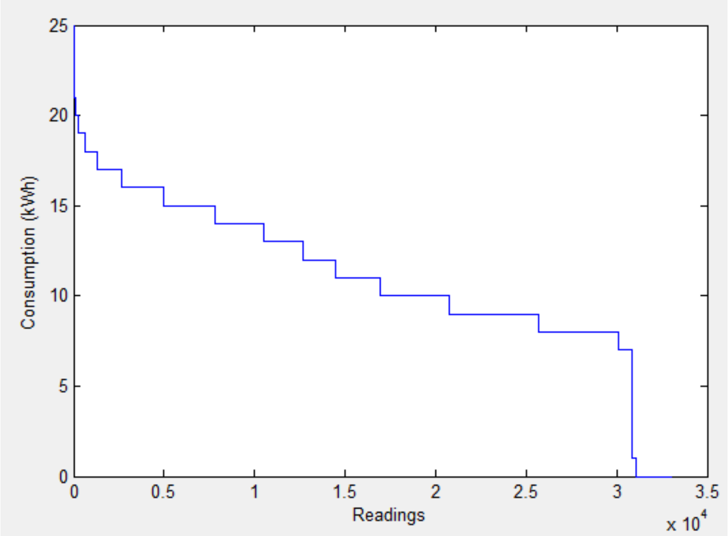


Figure 5-5 Yearly load curve

An exemplary output of the electricity consumption in the university building can be seen in Figure 5-5. The base load in this figure can be estimated as 8 kWh. The figure reveals a significant number of missed readings as one year should have 35,040 readings. Additionally, the steps in the graph confirm that consumption is measured in a 1 kWh interval. Since consumption fluctuates only between 8 kWh and 25 kWh, a higher monitoring granularity, e.g. 0.1 kWh, would have minimised the steps.

An annual load curve is realised in MATLAB using its “sort” function. By sorting all values from large to small, any load curve can be converted to an annual load curve.

Listing 14 Create yearly load curve

```
% plot data sorted descending
plot(sort(RAW_ELEC,'descend'));
ylabel('Consumption (kWh)');
xlabel('Readings');
```

5.1.5 Determination of day with highest consumption

In this approach the daily readings of the single day with the highest total consumption is identified. For large electricity consuming sites, the day with the highest consumption dictates the price for each kWh. This is widely known as the demand rate of a building. This data representation can be useful to retrospectively investigate into the day which caused the high consumption.

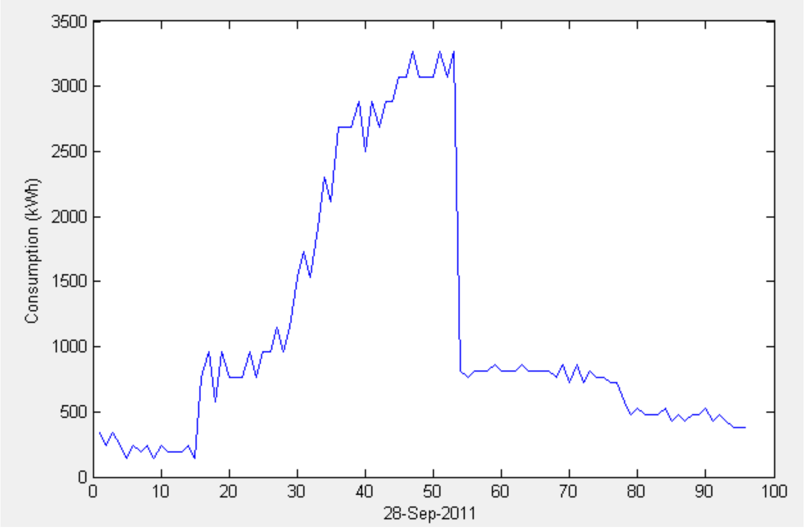


Figure 5-6 Day with highest consumption

A sports arena example can be seen in Figure 5-6. Here, the x-axis contains 96 samples, which indicates 15 minute reading intervals. Furthermore, it labels the graph with the day of the highest consumption. The y-axis reveals that the peak consumption on this day was roughly 3300 kWh. In the given example, the consumption significantly rose between sample 32 and 55 which corresponds to 8 am and 1:45 pm. Given that this data is from the sports arena, one may speculate that a test of the flood light system has taken place.

This analysis is realised by identifying the highest readings and corresponding timestamp in the dataset. Afterwards, all values for the day in question are selected and visualised.

Listing 15 Identify day of highest consumption

```
% find maximum value and time occurrence
[maxnum, maxind] = max(elec(:,8));
[row, col] = ind2sub(size(elec), maxind);
elec(row,:)
totalmax = maxnum;
totalmaxtime = elec(row,1:3);

% plot all readings for the identified day
z = elec(elec(:,9) < addtodate(totalmaxtime, 1, 'day'), 1:9);
z = z(z(:,9) >= totalmaxtime , 1:9);
plot(z(:,8));
ylabel('Consumption (kWh)');
xlabel(datestr(totalmaxtime));
```

5.1.6 Weekly data analysis

A more sophisticated analysis is the overlay of data sets. A weekly overlay of a year's data plots 52 curves on top of each other. The expected result is a clear trend with the majority of measurements in close proximity to the trend. This trend is the consumption for a normal daily operation. Anything outside this trend can be considered as unusual building operation. The benefit of this analysis is a) to identify the building trend and b) to detect the anomalies which indicate an unusual building operation.

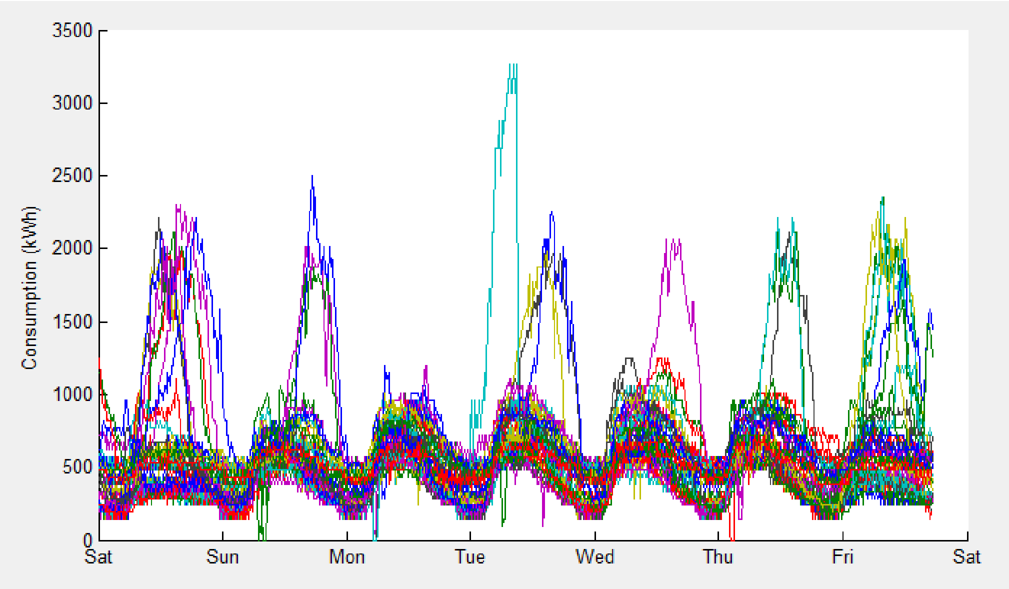


Figure 5-7 Weekly overlay plot of the arena

In Figure 5-7 one can easily identify weekdays and weekends. The example is taken from the arena where sport events stand out as peaks. Upon inspecting one can see that Monday is the only day in the selected 52-week period where no event has been hosted. All other days feature peaks with the majority happening on Fridays and Saturdays. The Tuesday features the peak with the highest consumption. It is significantly higher than other peaks. Large energy consumers should avoid this kind of peak as it dictates the price for energy procurement.

Listing 16 Perform weekly analysis

```
% assuming 15 minute intervals, we need 672 readings per week
M = RAW_ELEC;
weeks = floor(size(M)/672);
weeks = weeks(:,1);
M2 = M(1:weeks*672);

% reshape data so that we have one array with one line per week
M_analysis = reshape(M2',[672,weeks])';
plot(M_analysis(1:52,:)');
ylabel('Consumption (kWh)');
set(gca,'XTickLabel',{y1 y2 y3 y4 y5 y6 y7});
```

This data representation is realised by grouping the data into weekly blocks. Assuming 15 minute reading interval, one week consists of 672 individual readings. Afterwards, a new array is generated which holds the individual weeks data in a separate row. This array is then plotted into the existing figure to create the overlay.

5.1.7 Same weekday analysis

In this approach, all days of a selected day, e.g. all Tuesdays are plotted as an overlay. The expected output is a trend typical for the chosen day. To further increase its usefulness, all minima and maxima are highlighted in blue. Finally, a total average is superimposed in black. The aim of this analysis is to visualise the deviation occurring in repeating time patterns. Unexpected behaviour clearly stands out and can be interpreted.

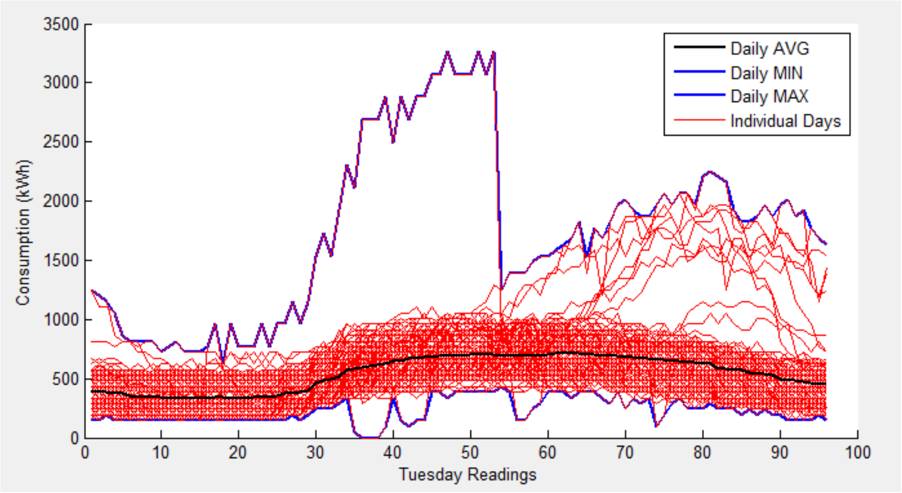


Figure 5-8 Tuesday consumption in the arena

Figure 5-8 shows an example plot for all Tuesdays from the sports arena. It is apparent which times of the day the building was the busiest. The previously identified high peak can be identified again (see section 5.1.5). Interestingly, it is the only major anomaly in the early Tuesday hours. Other anomalies are reoccurring in accordance at later hours.

When comparing the Tuesdays to Saturdays (see Figure 5-9), it is revealed that major events tend to happen at weekends. On a Tuesday, fewer peaks exist. Additionally, Tuesday peaks occur at a later time of the day indicating that weekend events start earlier.

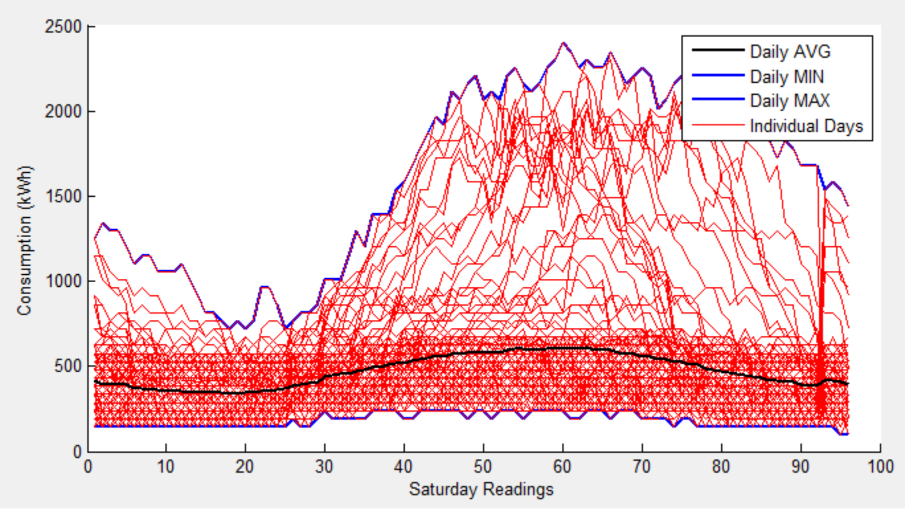


Figure 5-9 Saturday consumption in the arena

This analysis is realised by selecting all data for the chosen day first.

Listing 17 Select weekday

```
% Sunday =1, Saturday = 7
day = 3;
[x y1] = weekday(day,'long');

% filter only desired days
elec2 = elec( (elec(:,7) == day)  , 1:8);
elec3 = elec2(:,8);
```

Next, the total number of available days in the data set needs to be evaluated. Similarly to the previous analysis, all data for each available day gets stored in a single row in a matrix. Three further data sets are calculated and stored in individual arrays: the minimum, maximum and average values.

Listing 18 Calculate MIN, MAX, AVG

```
% identify total number of available days and prepare matrix
numbdays = floor(size(elec3)/96);
numbdays = numbdays(:,1);
elec4 = elec3(1:numbdays*96);
elec5 = reshape(elec4',[96,numbdays(:,1)]);

% calculate minima, maxima and average
yearly_avg = [];
yearly_min = [];
yearly_max = [];
for i =1:96
    yearly_avg(i) = mean(elec5(i:96:end));
    yearly_min(i) = min(elec5(i:96:end));
    yearly_max(i) = max(elec5(i:96:end));
end
```

Finally, all data sets get visualised in a single graph. The three additionally calculated arrays will be visually highlighted through thicker line width and different colour. The daily average is plotted last in order to superimpose all other data sets.

Listing 19 Create plot

```
% plot all figures, label axes and print legend
hold on;
plot(yearly_min, 'color', 'blue', 'Linewidth', 2);
plot(yearly_max, 'color', 'blue', 'Linewidth', 2);
plot(elec5, 'color', 'red');
plot(yearly_avg, 'color', 'black', 'Linewidth', 2);
legend('Daily AVG', 'Daily MIN', 'Daily MAX', 'Individual Days');
ylabel('Consumption (kWh)');
xlabel(strcat(y1 , ' Readings'));
hold off;
```

5.1.8 Yearly data analysis with base load detection

This method can be seen as a combination from the weekly overlay plot and the statistical accumulations overlaid in the previous section. Specifically, the overall daily minima and maxima in the analysed data set are detected. The average consumption of the year as well as from the first and second half of the year is revealed. Additionally the total overall base load and the total overall off-peak base load that occurred in the building are included. Both are represented with a horizontal line.

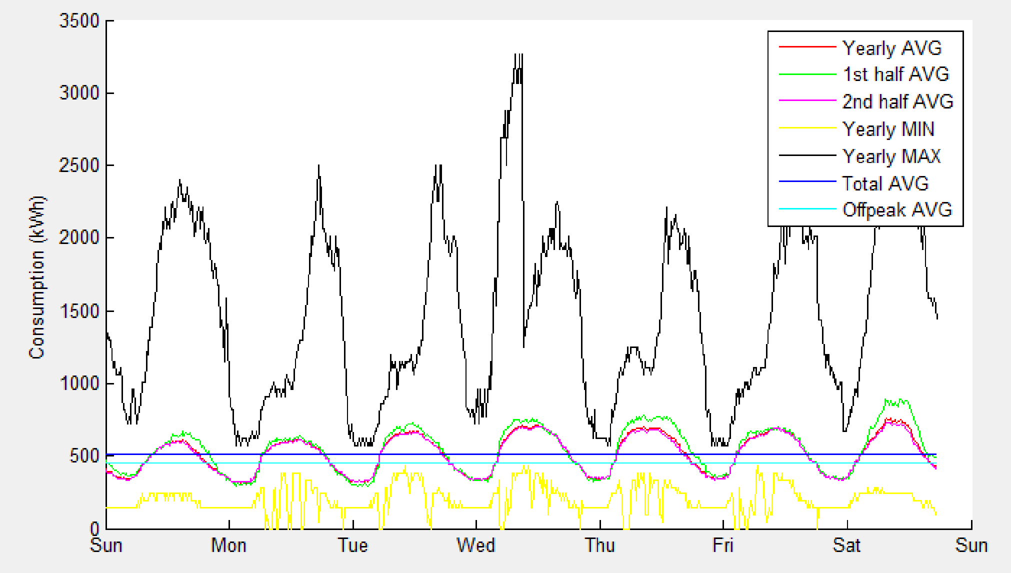


Figure 5-10 Yearly data analysis with base load detection

The benefit of this data representation is the variety of information which can be learned through the combination of analysis methods.

Figure 5-10 provides various interpretations of consumption data from the arena. It shows a huge difference between minimum and maximum consumption. While this was already concluded while inspecting the general load curve, this graph enriches knowledge by revealing a yearly average consumption of only 500 kWh. One could have expected a higher average, given the occurrence of events in the arena. There are neglectable differences between consumption during the first and second half of the year as indicated by the green and red lines.

The outside office hours were specified in MATLAB as all readings monitored before 8am and after 8pm at work days. Additionally, both Saturday and Sunday are considered outside office hours. These hardcoded thresholds might need adoption when applied to specific sites as their office hours are likely different. This adoption could be realised by maintaining this information in BIM and subsequent extraction into the DWH as additional building information.

This data analysis is produced in several steps. First, the data is numbered as per MATLAB's internal date scheme. Here, Sundays = 1 and Saturdays = 7. Next, all off-peak readings are accumulated. In this snippet, the off-peak consumption is defined as all Saturday and Sunday readings as well as all readings before 8 am and after 8 pm. Lastly, the average off peak consumption is calculated.

Listing 20 Select day and define off-peak period

```
% Sunday =1, Saturday = 7
DayNumber = weekday(datestr(RAW_ELEC_Tnum));
elec = [RAW_ELEC_Tvec DayNumber RAW_ELEC];

% filter offpeak values: all sat/sun + before 8am and starting from 8pm
offpeak_raw = elec(
(elec(:,7) == 1 | elec(:,7) == 7) | %filter weekends
(elec(:,4) < 8 | elec(:,4) > 19) , %filter before 8am and after 7:45pm
1:8 ); %select all remaining rows

% calculate offpeak
offpeakavg = mean(offpeak_raw(:,8));
```

In a second step, five further arrays are created which will hold the yearly average, the average for the first and second half of the year and the yearly minimum and maximum values. Afterwards, the overall average base load is calculated. Logically, the average base load should always be higher than the off-peak base load.

Listing 21 Calculate MIN, MAX, AVG and base load

```
% calculate minima, maxima and average
yearly_avg = [];
firsthalf_avg = [];
secondhalf_avg = [];
yearly_min = [];
yearly_max = [];
for i =1:672
    yearly_avg(i) = mean(RAW_ELEC(i:672:end));
    firsthalf_avg(i) = mean(RAW_ELEC(i:672:17520));
    secondhalf_avg(i) = mean(RAW_ELEC(i+17472:672:end));
    yearly_min(i) = min(RAW_ELEC(i:672:end));
    yearly_max(i) = max(RAW_ELEC(i:672:end));
end

% calculate baseloads
total_avg = mean(yearly_avg);
baseload = offpeakavg / total_avg * 100;
peakload = (total_avg - offpeakavg) / total_avg * 100;
```

Finally, all data sets get plotted into a single graph. For better interpretability, each data set is plotted in a different colour. As a last step, the x-axis is labelled with the name of the day.

Listing 22 Plot graph and format axes

```
% plot all figures, label axes and print legend
hold on
plot(yearly_avg, 'color', 'red');
plot(firsthalf_avg, 'color', 'green');
plot(secondhalf_avg, 'color', 'magenta');
plot(yearly_min, 'color', 'yellow');
plot(yearly_max, 'color', 'black');
plot([1,672],[total_avg,total_avg], 'color', 'blue');
plot([1,672],[offpeakavg,offpeakavg], 'color', 'cyan');
legend('Yearly AVG', '1st half AVG', '2nd half AVG', 'Yearly MIN',
'Yearly MAX', 'Total AVG','Offpeak AVG');
ylabel('Consumption (kWh)');
[x y1] = weekday(RAW_ELEC_Tnum(1+96*0));
[x y2] = weekday(RAW_ELEC_Tnum(1+96*1));
[x y3] = weekday(RAW_ELEC_Tnum(1+96*2));
[x y4] = weekday(RAW_ELEC_Tnum(1+96*3));
[x y5] = weekday(RAW_ELEC_Tnum(1+96*4));
[x y6] = weekday(RAW_ELEC_Tnum(1+96*5));
[x y7] = weekday(RAW_ELEC_Tnum(1+96*6));
set(gca,'XTickLabel',{y1 y2 y3 y4 y5 y6 y7});
 hold off;
```

5.1.9 Summary of the functionalities

The analysis methods introduced in this chapter range from the basic load curves to more sophisticated methods. These methods give an idea of what can be done with load curve data through analysis. Table 38 provides an overview about the discussed analysis methods and their applicability to data sets of different origins.

Table 38 Summary of demonstrated analysis methods

Analysis method	*Applicable for*
Visualisation of the load curve	Meter and sensor data
Determination of daily maxima's	Meter and sensor data
Total daily consumption analysis	Only meter data
Annual load curve analysis	Only meter data
Determination of day with highest consumption	Meter and sensor data
Weekly data analysis	Meter and sensor data
Same weekday analysis	Meter and sensor data
Yearly data analysis with base load detection	Only meter data

The following chapter transforms the load curve data and inspects it from a different perspective. It enables the detection of different building behaviour patterns, not through base loads and overlays, but through mathematical clustering.

5.2 Clustering

This chapter discusses results from clustering load curve data from the three pilot buildings. Through clustering, a load curve reveals building operational patterns not visible from a load curve inspection.

Example 1:

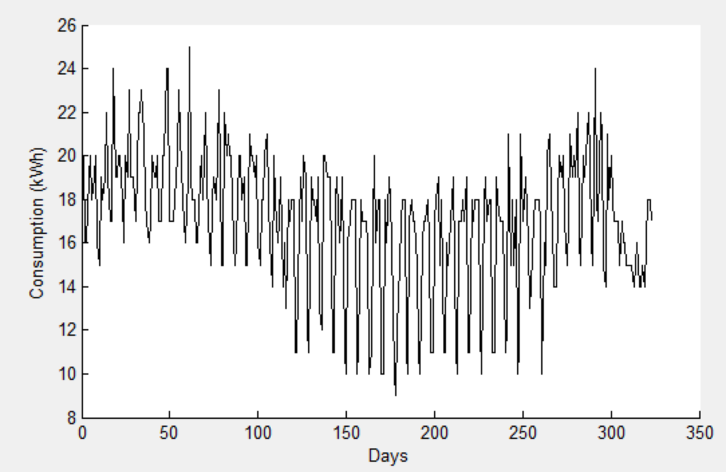


Figure 5-11 University building electricity consumption load curve

One year of electricity consumption from the university is provided in Figure 5-11. The electricity is monitored in a resolution of 1 kWh (no fractions of consumed kWh are monitored and recorded).

Through clustering the load curve, three individual behaviour patterns of the buildings are revealed. These are visualised in Figure 5-12. The blue and red cluster show a similar pattern shifted by about 4 kWh. The green pattern works contrary with higher consumption at the later stages of the day.

The electricity consumption from the blue and red line is getting lower during day time. This may sound controversial. However, in this case, gains from solar energy come into play around sample 40 (roughly 10 am) which result in a reduced grid electricity consumption. From analysing the load curve in the previous Figure 5-11, one cannot identify these separate building patterns.

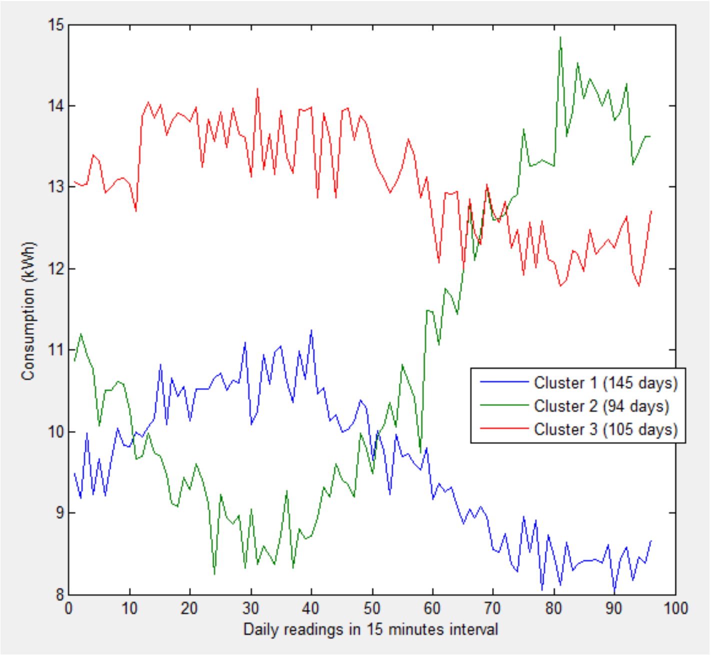


Figure 5-12 Clustered electricity consumption

Example 2:

The gas meter of the university building sends pulses for every 100 kWh gas consumed. This low resolution therefore results is a stepped load curve. Figure 5-13 shows two identified clusters and their usage behaviour throughout the day. The blue curve could potentially be the base load for the domestic hot water while the green curve may be related to the underfloor heating and geothermal heat pump.

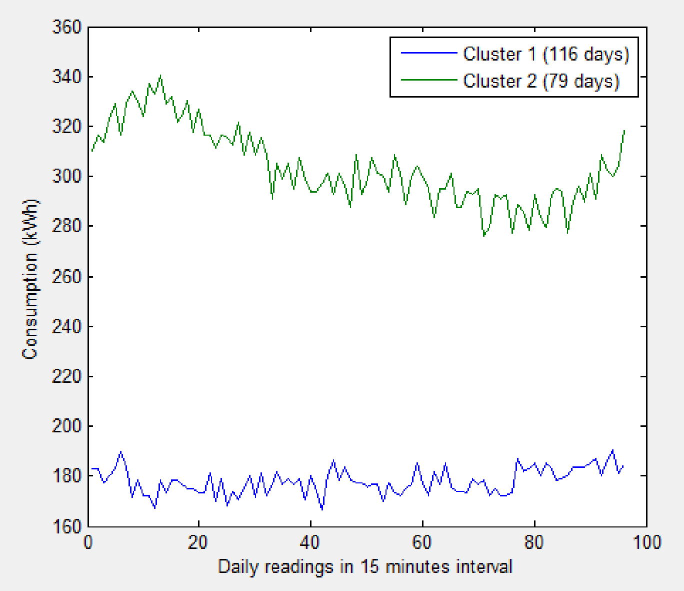


Figure 5-13 Clustering of the university gas consumption

Upon manual selection of 5 clusters as seen in Figure 5-14, distinguishing the systems becomes increasingly difficult. Without individual metering, one may only speculate which cluster represents which system.

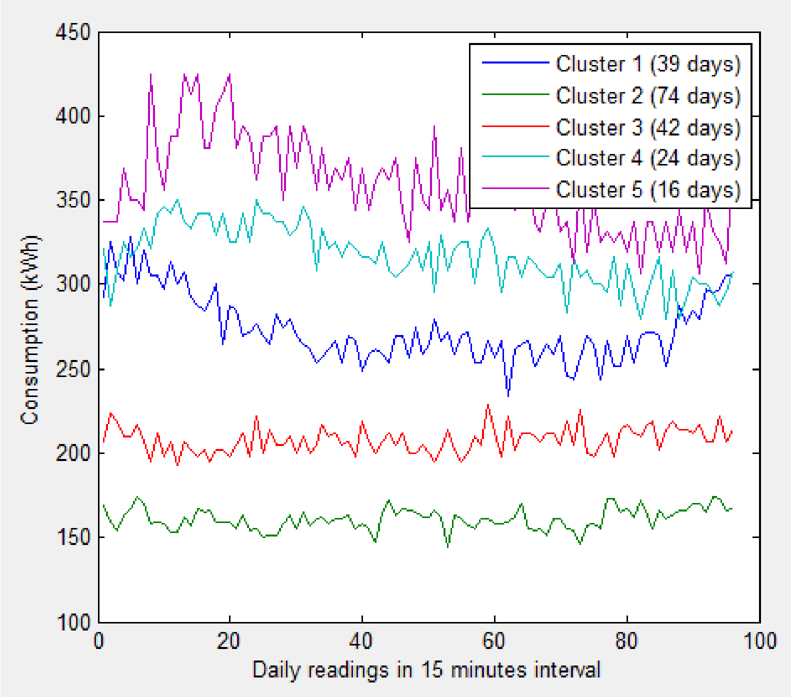


Figure 5-14 Clustering of gas consumption with k=5

Example 3:

The load curve of the arena (see Figure 5-15) already implies that during the year, few peaks exist. Clustering of the arenas electricity consumption reveals three clusters which can be explained through the building's usage behaviour (see Figure 5-16). Cluster 1 is related to non-event days and weekends, cluster 2 can be related to regular work days and small events. Cluster 3 can be linked to major events where the building's flood light system is used. The dotted lines around the clusters visualise the standard deviation within the cluster (see section 3.5.2).

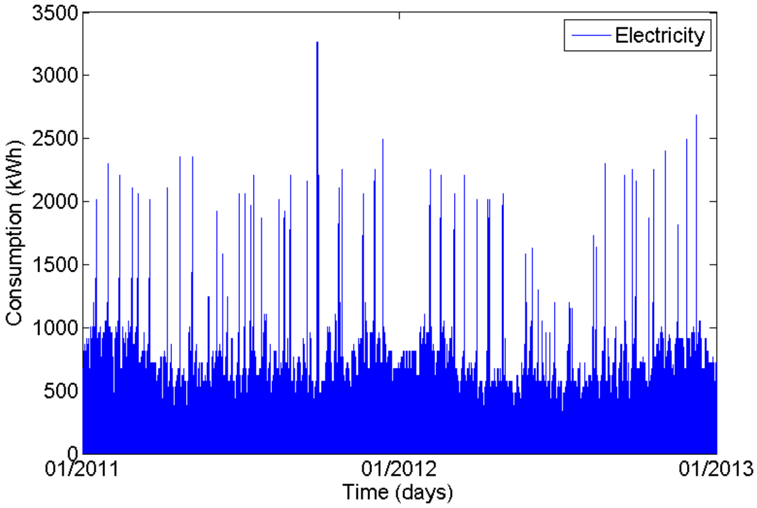


Figure 5-15 Raw electricity consumption

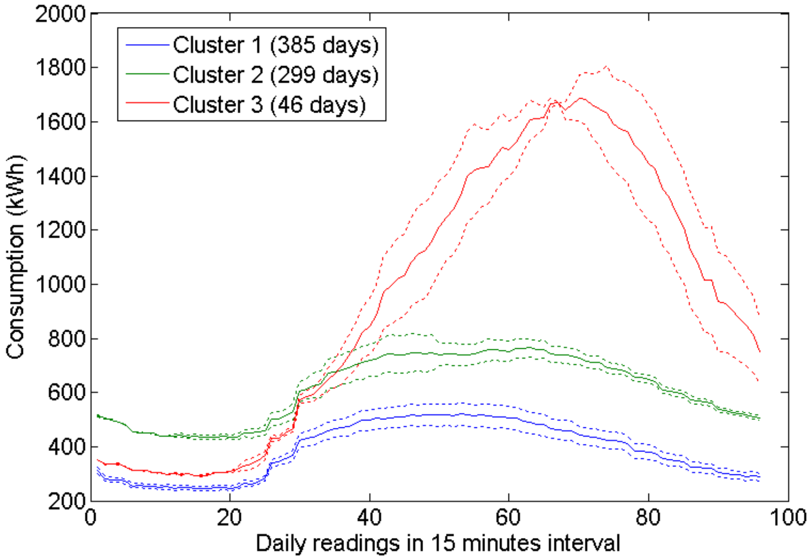


Figure 5-16 Clustered electricity consumption

The monitoring of the sport arena is with a resolution of 1 kWh identical to the university building. However, due to the wide spread in consumption (between 0 kWh and 1,800 kWh), the arena's data set can be considered to be of significantly higher quality. This highly detailed variation in consumption allows the successful application of the clustering methodology.

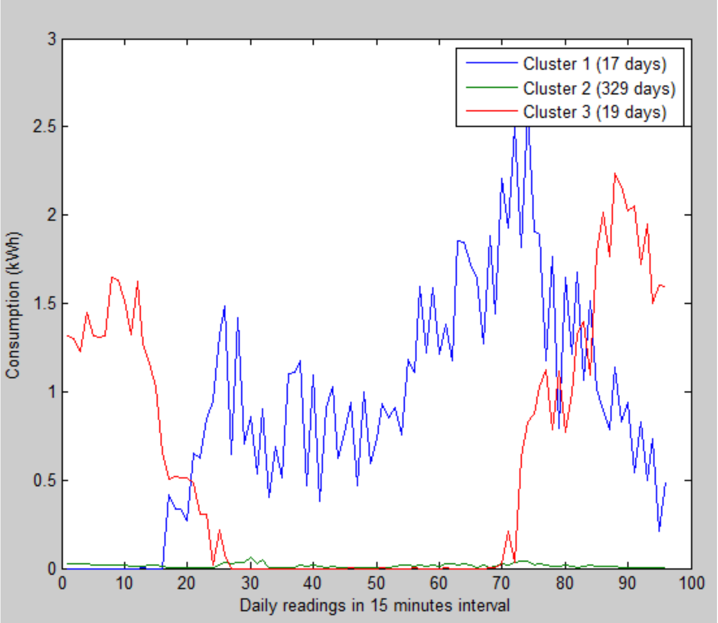


Figure 5-17 Clustered flood light consumption

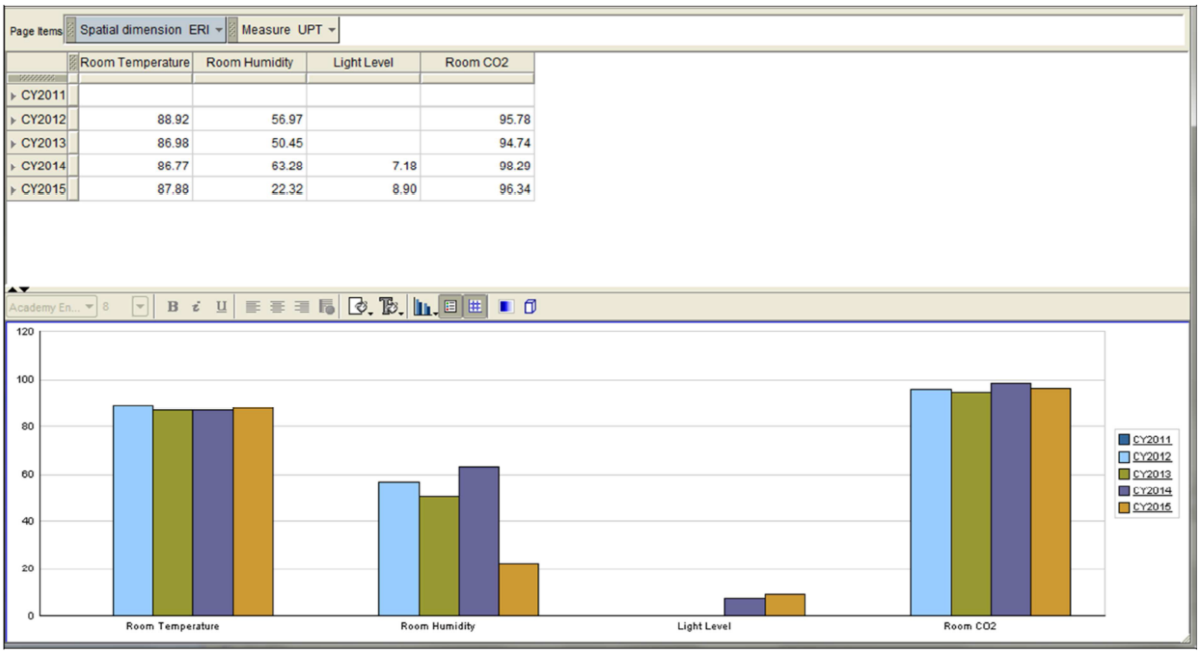
In Figure 5-17, the consumption of the flood lighting system is clustered. In contrary to the previous analysis, sub metering data of the floodlight system was available for 365 days. The data reveals that the floodlight is switched off most of the time (329 days) while two different usage behaviours were identified. One behaviour (red) starts in the evening time (after 70 reading intervals = 5:30 pm) and continues until the next morning around 5am. A second behaviour (blue) starts in the morning at around 20 reading intervals (5 am) and finishes before midnight. This behaviour peaks between 55 and 80 readings (1:45 pm to 8 pm). One can conclude from this that the flood light has two typical usage scenarios: Scenario 1 is an evening event which last until the early morning. Scenario 2 is a daytime event which peaks in the late afternoons.

5.3 Key Performance Indicator analysis

This chapter depicts various KPI’s resulting from aggregated data stored in DWH cubes. This kind of data analysis provides single key figures. This data representation is useful for management summaries where a single figure is more appropriate then load curve analyses.

5.3.1 Comfort KPI

The underperformance ratio KPI (UPR) for the calendar years 2011 to 2015 can be seen in Figure 5-18. While 100% is the ideal performance, the building underperforms in each of the four comfort disciplines: temperature, humidity, light level and CO_2.



	Room Temperature	Room Humidity	Light Level	Room CO2
CY2011				
CY2012	88.92	56.97		95.78
CY2013	86.98	50.45		94.74
CY2014	86.77	63.28	7.18	98.29
CY2015	87.88	22.32	8.90	96.34

Figure 5-18 Underperformance time KPI

While CO_2 and room temperature are close to the ideal value, humidity has a strong fluctuation. However, upon closer inspection, it is revealed in Figure 5-19 that the humidity KPI is very low in every first quarter of a year. The data set depicted in Figure 5-18 provides data only until March 2015. Therefore, the first quarter of 2015 determines an inaccurate yearly KPI result of 22.32%. Figure 5-19 provides a drill-down into a quarterly data representation of the humidity KPI. Here, In this case, the yearly total is coloured red; the quarterly KPI is coloured blue. One can nicely see how the bad performance in the first quarter of each year affects the overall yearly performance.

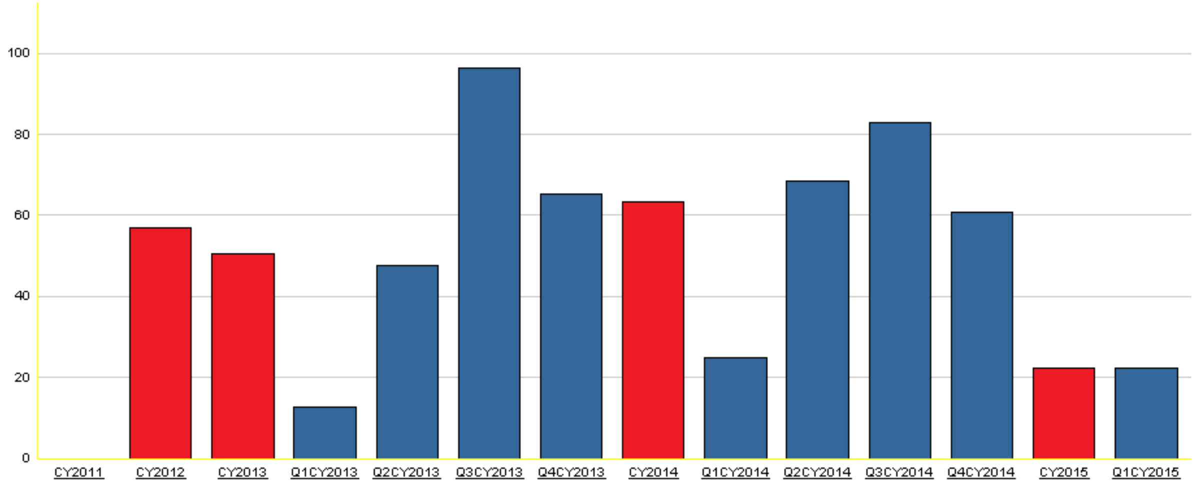
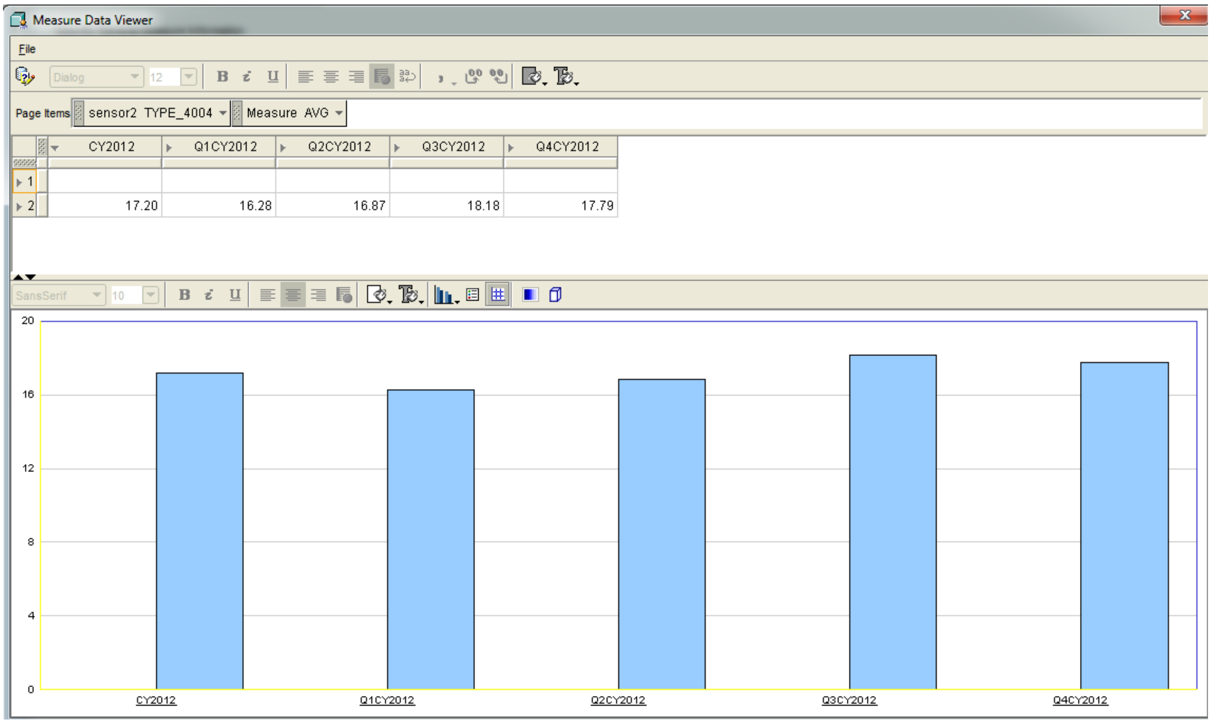


Figure 5-19 Humidity KPI quarterly drill-down

5.3.2 Temperature KPI

Figure 5-20 gives a summary of the total average temperature per quarter year in 2012. In this figure, the average temperature is calculated across all sensors. It suggests that temperate levels were in accordance with desired office temperatures. A drill-down into the data allows inspecting temperatures up to the individual room level.



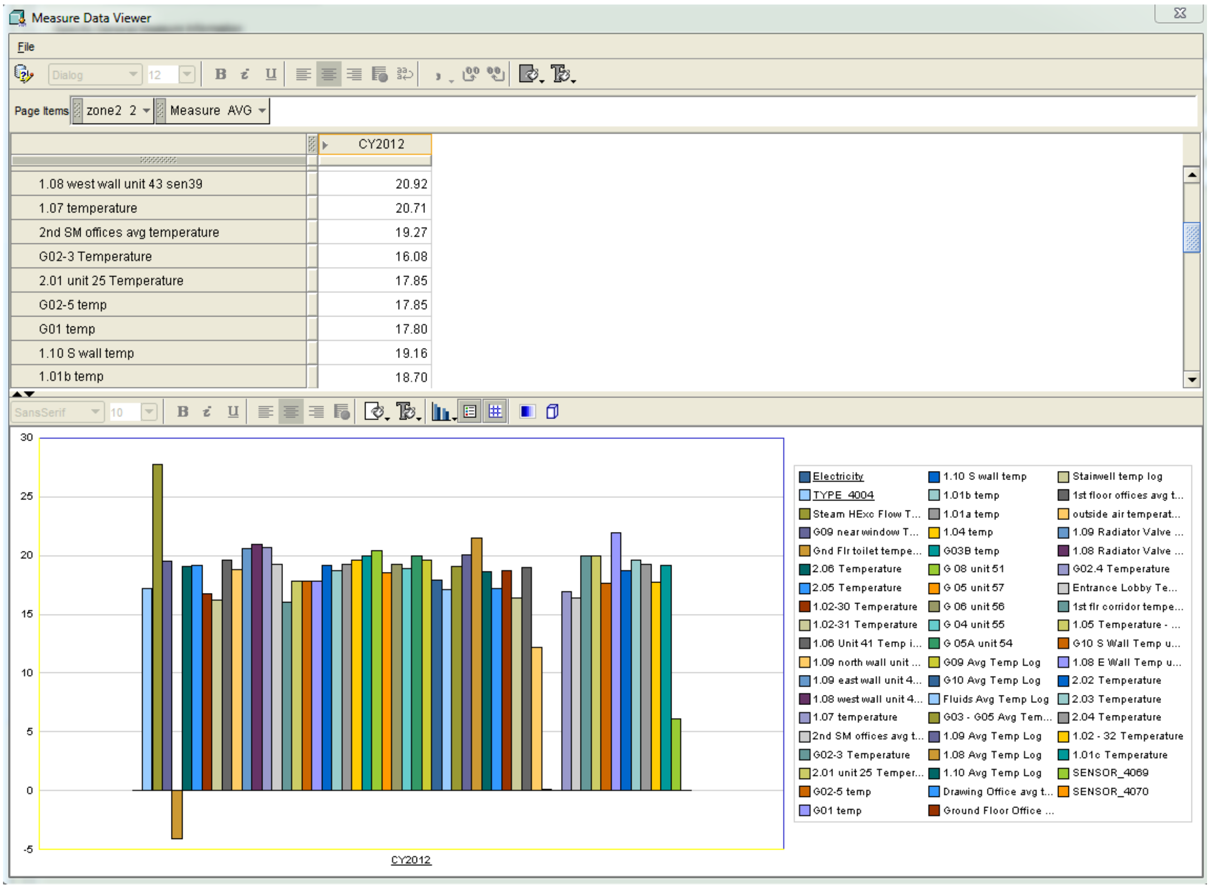
Page Items sensor2 TYPE_4004 Measure AVG

	CY2012	Q1CY2012	Q2CY2012	Q3CY2012	Q4CY2012
1					
2	17.20	16.28	16.87	18.18	17.79

Figure 5-20 Yearly average temperature

In Figure 5-21, the average temperature for each individual sensor in 2012 can be seen. This figure provides more detail than Figure 5-20 by breaking down the total averages into their individual data feeds. While most rooms have desirable average temperatures between 16 and

22 degrees, a few outliers exist. One room even sends negative average temperatures. From looking at the data, one cannot be certain of the reason for these outliers. But, they provide indications for further inspection. One could argue that a single temperature sensor cannot be of significance. In some cases, however, BMS rely on temperature values to drive systems. An improperly set up temperature sensor could generate system operation cost when there is no need.



Page Items: zone2 2 | Measure AVG

	CY2012
1.08 west wall unit 43 sen39	20.92
1.07 temperature	20.71
2nd SM offices avg temperature	19.27
G02-3 Temperature	16.08
2.01 unit 25 Temperature	17.85
G02-5 temp	17.85
G01 temp	17.80
1.10 S wall temp	19.16
1.01b temp	18.70

Figure 5-21 Average temperature drill-down

A drill-down into a quarterly data representation can be seen in Figure 5-22. Here, one can see that the negative value from Figure 5-21 occurred in the first two quarters only. One may conclude that this issue was fixed as quarter four reports values within the desired range. In quarter three, its value is a bit lower than the average value of the other sensors. This indicates that the sensor was fixed probably in the early weeks of quarter three. A further drill down would enable the viewer to identify the week and day where the sensor stopped reporting negative values.

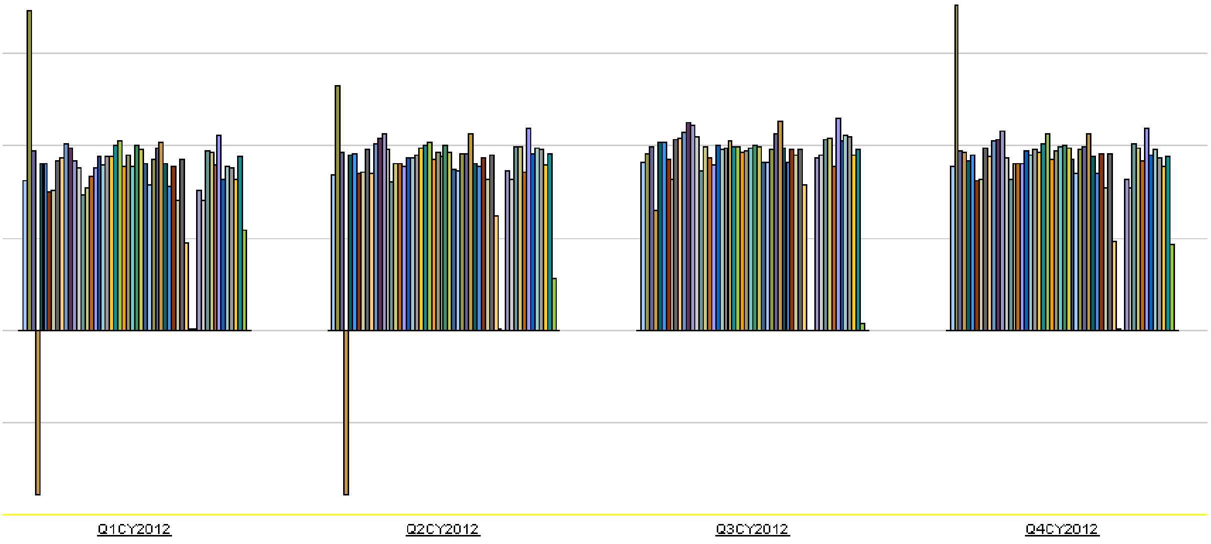
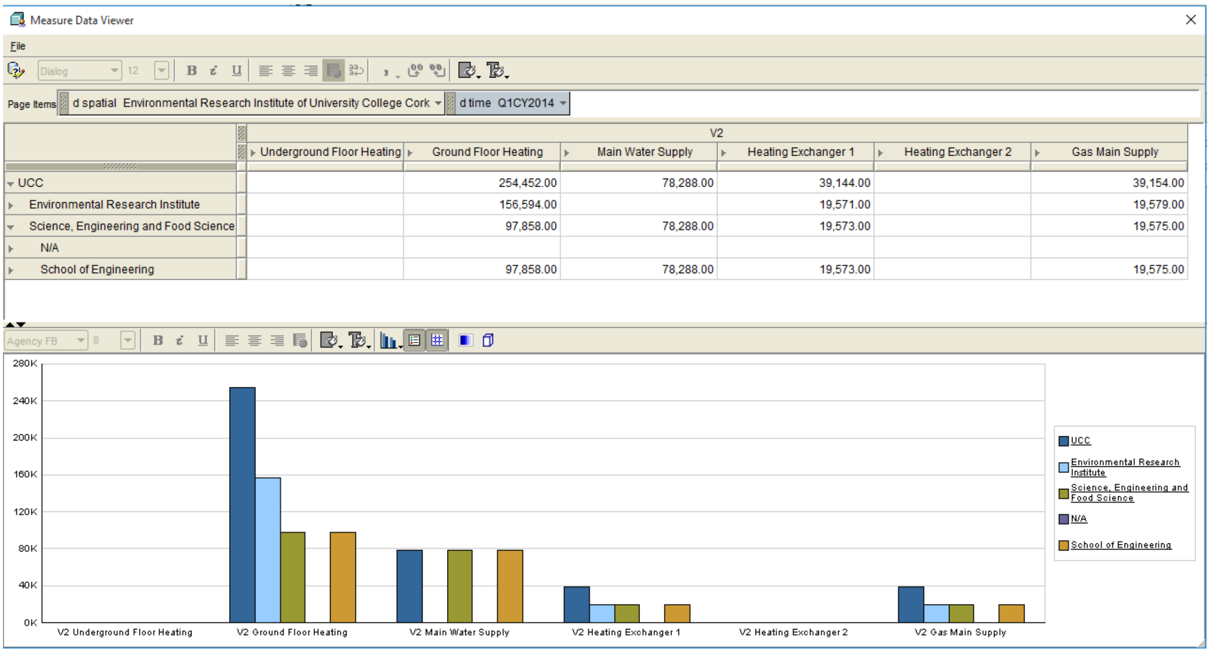


Figure 5-22 Quarterly representation of individual sensor temperatures

5.3.3 System and Organization KPI

In Figure 5-23, one can see the heat consumption per system and per organization. This figure concentrates exemplary on consumption in the first quarter of the year 2014. On the left side, the organization tree is opened up and shows the sub-organizations of an organization called "UCC".


Measure Data Viewer

Page Items: d spatial Environmental Research Institute of University College Cork; d time Q1CY2014

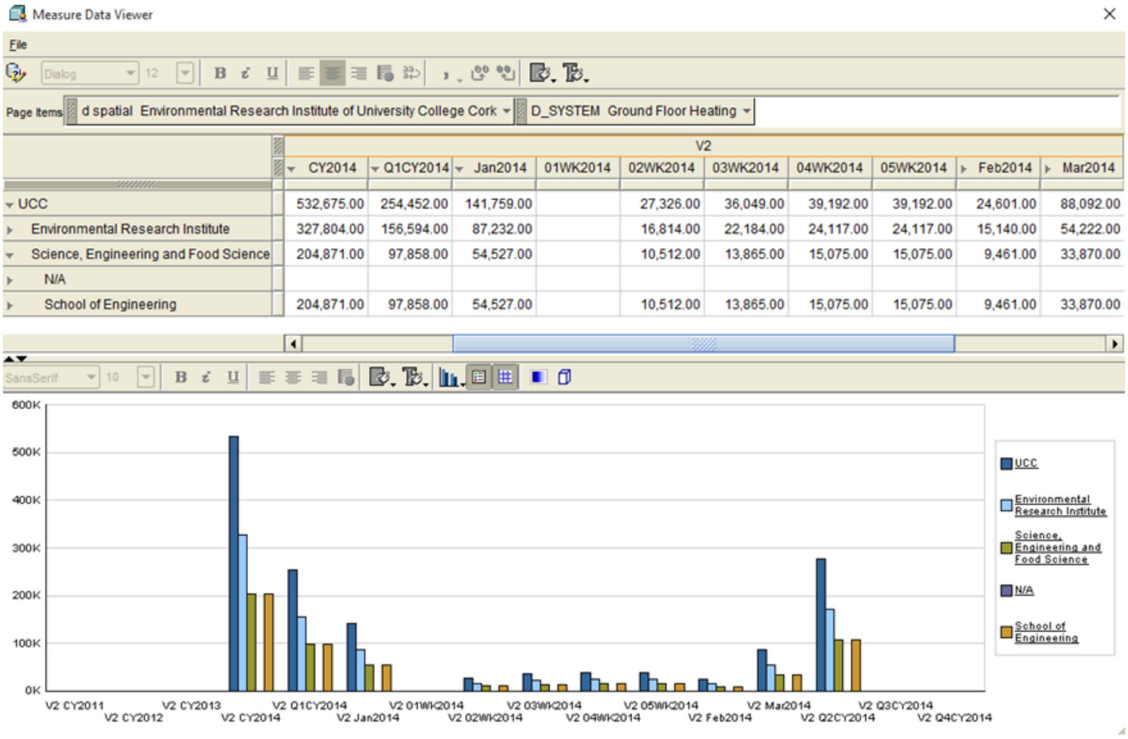
	V2					
	Underground Floor Heating	Ground Floor Heating	Main Water Supply	Heating Exchanger 1	Heating Exchanger 2	Gas Main Supply
UCC		254,452.00	78,288.00	39,144.00		39,154.00
Environmental Research Institute		156,594.00		19,571.00		19,579.00
Science, Engineering and Food Science		97,858.00	78,288.00	19,573.00		19,575.00
N/A						
School of Engineering		97,858.00	78,288.00	19,573.00		19,575.00

Figure 5-23 Heat consumption per system and organization

The total heat consumption of the ground floor heating is 254,452 kWh. The drill down enables the user to identify which sub-organization consumed how much energy. This becomes particularly interesting if, for example, departments are charged individually for their energy consumption. Here, the “Environmental Research Institute” consumed 156,594 kWh while the “Science, Engineering and Food Science” consumed only 97,858 kWh. Upon further drill-down into the organizational dimension, it is revealed that all energy was consumed by the “School of Engineering”.

Furthermore, one can see that the gas consumption (data is represented in m^3) is equally distributed among the two sub-organizations. The same applies to the heat exchanger 1. This highlights the possibility that further monitoring equipment is needed to accurately allocate consumption to the individual organizations.

A further drill-down into the first weeks in 2014 is presented in Figure 5-24. It is revealed that the first week is actually without any data. This can be misleading and potentially dangerous in KPI data representations as the aggregation of missing data still succeeds. From the previous Figure 5-23, it is not obvious that data may be incomplete. This deficit is worthy to be tackled in future work in order to deliver reliable KPI values.



Page Items: d spatial Environmental Research Institute of University College Cork | D_SYSTEM Ground Floor Heating

	V2									
	CY2014	Q1CY2014	Jan2014	01WK2014	02WK2014	03WK2014	04WK2014	05WK2014	Feb2014	Mar2014
UCC	532,675.00	254,452.00	141,759.00		27,326.00	36,049.00	39,192.00	39,192.00	24,601.00	88,092.00
Environmental Research Institute	327,804.00	156,594.00	87,232.00		16,814.00	22,184.00	24,117.00	24,117.00	15,140.00	54,222.00
Science, Engineering and Food Science	204,871.00	97,858.00	54,527.00		10,512.00	13,865.00	15,075.00	15,075.00	9,461.00	33,870.00
N/A										
School of Engineering	204,871.00	97,858.00	54,527.00		10,512.00	13,865.00	15,075.00	15,075.00	9,461.00	33,870.00

Figure 5-24 Weekly drill-down of system consumption

6 Conclusions

The topic of this thesis is the analysis of building performance data. The aim of this thesis was to optimise the acquisition, analysis, integration and retrieval of data to increase overall efficiency. Complex data analysis is problematic as data needs to be acquired and consolidated from multiple heterogeneous sources. This is time consuming as no common data source exists. Therefore, current efficiency in data analysis can be significantly increased.

This work provides a solution to this problem. It introduces methodologies to minimise consolidation efforts. The data acquisition process in buildings is optimised by the introduction of a monitoring concept. This concept covers reoccurring elements for the installation of monitoring equipment and provides a guideline for proper installations. The acquired data is examined for faults and replaced through a novel data cleansing and interpolation methodology.

Another source of information is the BIM model of a building. Descriptive Information from the model can be extracted to enrich the acquired meter data. By utilising BIM models to process and classify building performance data, a tedious yet necessary step to reacquire previously gathered information is eliminated. This was achieved by combining selected information from the open BIM standard IFC with DWH technology. This integrates data from various sources into a standardised data schema. Having data consolidated enables the development of more complex and intensive analysis methods. This unified data stored in the DWH provides a more comprehensive foundation for data analysis.

The introduced DWH dimensions allow the analysis of individual or grouped sensors/meters while guaranteeing efficient data lookups. This analysis is supported by the application of KPI’s. Analysis may be performed against arbitrary combinations of organization units, building systems, spatial zones over time periods. Even huge numbers of installed sensors can be analysed following the same principles. Another benefit of the combination of data sets in a DWH is that it allows before and after comparisons. After conducting an energetic measure, sensor data may be examined e.g. to verify that comfort KPI’s are still classed within defined thresholds.

The novel algorithm for the cleansing of metered gas data is suitable for large building pools without emphasising on manual data cleansing. The interpolation algorithm was tested against readings acquired from operational buildings which featured wide differences with regard to incomplete and accurate data. Results have shown that faulty consumption readings from gas meters can be cleansed in order to obtain replacement values of adequate quality. The interpolation algorithm delivered robust results even with large quantities of corrupt data. This algorithm is beneficial not only for Facility Management companies and utility providers, but also for energy analysts and building owners.

The clustering of performance data combines large monitored data sets with machine learning algorithms. Through clustering, it becomes possible to identify usage behaviours and partition them into k behavioural patterns. The different patterns represent the different types of usage occurrences that are happening inside a building. This enables a view at monitored data from a perspective not given by normal means. This method reveals building behaviours not visible from a regular load curve analysis, making this algorithm very powerful for daily building operations.

In the future the author expects to see a convergence between machine learning algorithms and smart metering technology. The recently introduced high sampling rate of meters, with up to 4000 measurements per second, allows the utilisation of complex stochastic functionality that can eventually lead to a better understanding of building performance data (smartB, 2016). Its full practicability is still under development, for this reason this type of meter was not covered in this thesis.

Ultimately, the aim of this thesis is that its proposed methodologies will eventually be adapted into energy monitoring systems. This provides additional analyses functionality while supporting the reduction of energy consumption.

6.1 Issues Identified

One obvious issue for meter installations is the cost factor. Installing monitoring equipment does not automatically reduce the energy consumption of the building. To rectify the investment costs, facility managers and tenants have to work with the acquired data and perform energy consumption analyses. However, cost can be reduced by following the monitoring concept of this thesis. This minimises extra efforts for correcting and adjusting on site configurations.

One issue related to the IFC based database schema is it requires both database and IFC knowledge. While IFC adds an overhead in structure and information to the database, this can be simplified again by the utilisation of database views.

This work successfully introduced four database dimensions based on data acquired from the open BIM standard IFC. The challenge to create these dimensions was of differing difficulty. It cannot be guaranteed that further dimensions can be created similarly without breaking compatibility to the IFC meta model.

Another issue identified is the limited availability of data. As smart meters are rolling out only in the recent years, comprehensive data sets are scarce. This is likely to change as a report of the European Commission states that 16 member states will roll out smart meters by 2020 (European Commission, 2015).

BIM modelling is not widely used yet (especially in industry) which makes it very difficult to obtain building models. This will also change over the course of the next few years as BIM becomes increasingly more accepted with stakeholders. The British Standard PAS 1192-2 defines BIM as an information management tool for construction projects which came into effect in 2016 (PAS 1192-2, 2013). Germany also announced that their infrastructure projects will require BIM from 2020 onwards (Dobrindt, 2015).

6.2 Future Work

The monitoring concept introduced in this work greatly reduces the amount of errors and time required to successfully install monitoring equipment in a building. This could be further simplified by a data logger that detects connected meters. For digital meters, their configuration could be automatic as they provide their details in registers. When following the monitoring concept, analogue meters are transformed to M-Bus, thus all meters provide digital information.

Future work in data cleansing could add support for sub meters. As the heat curve derived for cleansing is dependent on the outside temperature, the current algorithm will only produce meaningful values for readings collected from main meters. Sub meters have further dependencies which need to be considered for successful cleansing. Additionally, support for further mediums could be investigated. Table 39 outlines various mediums and estimates if the proposed algorithm can be utilised similarly. Their applicability was not evaluated in this work but may be evaluated in the future.

Table 39 Evaluation of different mediums for data cleansing

Medium	*Feasibility*
Gas	Application was evaluated as part of this thesis
Heat	Likely applicable
Electricity	No direct relation between electricity consumption and outside temperature. There are exceptions e.g. storage heating
Water	Unlikely
Inside temperature	Physical dependency exists. Most likely applicable

In addition to sub meters, the heat curve can be utilised to forecast the gas consumption. When temperature data from a weather forecast is fed into the algorithm, it will output the estimated gas consumption for the corresponding period. This becomes particularly interesting for a demand based procurement of energy or load balancing algorithms.

Another research could focus on increasing the accuracy of the cleansed data by differentiation between multiple building behaviours, e.g. by data collected during work hours and data collected off-peak. For each behavioural pattern, an individual heat curve would need to be calculated to increase the algorithm's efficiency.

Since the introduction of an IFC compatible DWH schema is new to the field, no third party tools exist that are specialised on manipulating data stored in an IFC database schema. This creates potential for software products to extend their compatibility by supporting a schema which is in sync with DWH schemas that comply with the IFC standard. Database tools customised to work against IFC schemas are potential products, as they would work out-of-the-box with any stored data following the IFC meta-data model.

7 References

A C Bogen, M Rashid, E W East, J Ross, 2013. Evaluating a Data Clustering Approach for Life-Cycle Facility Control, s.l.: ITcon.

AMEV (Arbeitskreis Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen), Energie 2010

AMEV, 2001. Messgeräte für Energie und Medien (EnMess 2001), Berlin: s.n.

Aalst, W., Process Cubes Slicing, Dicing, Rolling Up and Drilling Down Event Data for Process Mining, First Asia Pacific Conference, Beijing, China, 2013

Abraham, B. and Chuang, A. , Outlier detection and time series modelling. Technometrics, vol 31 no 2 pp 241–248, 1989

Actaris Cyble Sensor V2, https://www.itron.com/mxca/en/productsAndServices/Pages/Cyble%20Sensor.aspx, last accessed 23.08.2015

Actaris, G16 gas meter, https://www.itron.com/mxca/en/productsAndServices/Pages/EU%20Series.aspx, last ac-cessed 06-08-2015

Action Energy, Energy Consumption Guide 19: energy use in offices, Carbon Trust., Crown, 2003.

Adamson C., The Star Schema Handbook: The Complete Reference to Dimensional Data Warehouse Design, Wiley & Sons, 2009

Ahmed, A. Multi-criteria data analysis of building performance data, PhD thesis University College Cork, 2011

Alahakoon, D., Yu, X., Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey, IEEE Transactions on Industrial Informatics, Vol. 12, No. 1, February 2016

Alexei, F., 2006. Nulls and Zeroes in Data Warehousing. Online, accessed 30-JUN-14 http://www.inmentis.com/docs/research/NullsAndZeroesInDataWarehousing.htm

Arnold, M., IT for Energy in Buildings, Project Report, University College Cork, 2013

Augenbroe, G. and Park C-S., Quantification methods of technical building performance, Building Research & Information, vol 33 no 2, pp 159 - 172, 2005

BACnet interest group, http://www.big-eu.org/bacnet/steckbrief/, last accessed 22.08.2016

Baker, E., Open source data logger for low-cost environmental monitoring, Biodiversity Informatics Horizons conference, Rome, Italy 2013

Balachandran, K., Olsen, R.L., Pedersen, J. M., Bandwidth analysis of smart meter network infrastructure, Advanced Communication Technology (ICACT), 2014 16th International Con-ference, pp 928 - 933, 2014

Benzi, F., Anglani, N., Bassi, E., Frosini, L., Electricity Smart Meters Interfacing the Households, IEEE Transactions on Industrial Electronics, Vol 58, No 10, October 2011

BIM Task Group Newsletter 45th Edition, http://www.bimtaskgroup.org/wp-content/uploads/2015/07/UK-BIM-Task-Group-Newsletter-45.pdf , last accessed 11.11.2016

Bogen, A.C., Rashid, M., East, E.W., Ross, J., Evaluating a Data Clustering Approach for Life-Cycle Facility Control", Journal of Information Technology in Construction, 2013

BuildingSmart, available online: http://www.buildingsmart.org/, last accessed June 2015

Bundesministerium fuer Justiz und Verbraucherschutz - German electricity grid access ordi-nance, Electricity network access regulation § 8, available online: http://www.gesetze-im-internet.de/stromnzv/BJNR224300005.html, last accessed May 2015, 2005

Bundesregierung, 2010, Energy Concept for an Environmentally-Friendly, Reliable, and Affordable Energy Supply, http://www.bundesregierung.de/ContentArchiv/DE/Archiv17/_Anlagen/2012/02/energiekonzept-final.pdf?__blob=publicationFile&v=5, last accessed 09/11/2015

Bushby, S. T., BACnet - A standard communication infrastructure for intelligent buildings, Automation in Construction, Vol. 6 No. 5-6, 1997, p. 529-540

Cahil, B., K. Menzel & D. Flynn. 2012. BIM as a centre piece for optimised building operation. Pg. 549 – 555. London: Taylor & Francis Group.

Campus21, Deliverable 4.1, Specification of Integrated BMS Data Models and Protocols selection, 2011

Chaudhuri S., Dayal U., An overview of data warehousing and OLAP technology, ACM SIG-MOD Record, vol 26 no 1 pp 65-74, 1997

Chen, J., Li, W., Lau, A., Cao, J., Wang, K., Automated Load Curve Data Cleansing in Power Systems, IEEE Transactions on Smart Grid, Vol 1, No 2, September 2010

Chia, T., Confidentiality, Integrity, Availability: The three components of the CIA Triad, http://security.blogoverflow.com/2012/08/confidentiality-integrity-availability-the-three-components-of-the-cia-triad/, last accessed December 2017

Currie, J., Isakow, C., Kelly, S., 50001 Reasons to improve energy performance, Johnson Con-trols, 2012

Davenport, T., Dyche, J., Big Data in Big Companies, International Institute For Analytics, 2013

Davies, L. and Gather, U., The identification of multiple outliers. J. Amer. Statist. Assoc., vol. 88, no. 423, pp. 782–792, 1993

Dayama, R., Chatla, A., Shaikh, H., Kilkarni, M, Android Based Meter Reading Using OCR, International Journal of Computer Science and Mobile Computing, Vol. 3, Issue. 3, March 2014, pg.536 – 539

Desjardins, P.A., Integrated alarm, security, building management, and communications system, US Patent 4,375,637, 1983

Dierks, T., Rescorla, E.; "The Transport Layer Security (TLS) Protocol", RFC 5246, 2008

Dobrindt, A., German Federal Minister of Transport and Digital Infrastructure, Building Information Modeling (BIM) wird bis 2020 stufenweise eingeführt, https://www.bmvi.de/SharedDocs/DE/Pressemitteilungen/2015/152-dobrindt-stufenplan-bim.html, 2015

EMH, http://www.emh-metering.com/ , last accessed 17.08.2015

Eastman, C., Teicholz, P., Sacks, R., Liston, K., BIM Handbook: A Guide to Building Infor-mation Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd ed., John Wiley & Sons, 2011

Electrocomponents plc, Socomec Diris Ap technical datasheet, http://docs-europe.electrocomponents.com/webdocs/010f/0900766b8010f19d.pdf, last accessed 06-08-2015

Elmasri, R., Navarte, S.B., Fundamentals of Database Systems, 6th ed., Addison Wesley, 2011

Elster EK 260 operation manual, http://docuthek.kromschroeder.com/documents/download.php?lang=de&doc=14465 , last accessed 17.08.2015

Elster EK 260, http://www.elster.sk/en/362.html , last accessed 17.08.2015
Enphase MyEnlighten, https://enphase.com/en-us/products-and-services/enlighten-and-apps, last accessed 22.08.2016

Engelberg, S., Kaminsky, T., Horesh, M.. Instrumentation notes - A USB-Enabled, FLASH-Disk-Based Data Logger. Instrumentation & Measurement Magazine, IEEE, 10, 63-66, 2007

Erek, K., Drenkelfort, G., Proehl, T. State of the Art von Energiemonitoringsystemen. TU Ber-lin, 2013. ISBN 978-3-7983-2459-6

Ericsson, N., Lennvall, T., Akerberg, J., Bjorkman, M., A Flexible Communication Stack Design for Time Sensitive Embedded Systems, Industrial Technology (ICIT), 2017 IEEE International Conference

European Commission, 2013. A 2030 Framework for climate and energy policies, http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52013DC0169, last accessed 09/11/2015

European Commission, 2015, Staff Working Document, Cost-benefit analyses & state of play of smart metering deployment in the EU-27 Accompanying the document Report from the Commission Benchmarking smart metering deployment in the EU-27 with a focus on electricity, http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1403084595595&uri=SWD:2014:189:FIN, accessed 09/08/2016

European Union, 2002. DIRECTIVE 2002/91/EC, http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0091 &from=EN, last accessed 09/11/2015

European Union, 2010. DIRECTIVE 2010/31/EU, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do ?uri=OJ:L:2010:153:0013:0035:EN:PDF, last accessed 09/11/2015

European Commission, Recommendation on preparations for the roll-out of smart metering systems, (2012/148/EU), http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32012H0148, 2012

Ferguson, T.S.,On the rejection of outliers. Proc. 4th Berkeley Symp. Math. Statist. Probab, vol. 1, pp. 253–287, 1961

Fischler, M. A., and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model

Fischler, M. A., and Bolles, R. C., Random Sample Consensus A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, vol. 24, no. 6, 1981

Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, vol. 24, no. 6, pp. 381-395

Flynn, D., Data Warehouse Requirements and extended BIM specification, BaaS FP7 project deliverable D2.1, 2012

GEFMA 510, German Facility Management Association, Model contract facility management, 2014

Gupta A. and Mumick I.S., Maintenance of Materialized Views. Problems, Techniques and Applications, IEEE Data Eng. Bull. vol 18 no 2 pp 3-18, 1995

Høverstad, A., Tidemann, A. and Langseth, H., Effects of Data Cleansing on Load Prediction Algorithms, CIASG, IEEE, 2013

Hagemann, I., PV in buildings - the influence of PV on the design and planning process of a building, Renewable Energy, Vol 8, No 1-4, 1996, pg. 467-470

Heidarinejada, M., Dahlhausena, M., McMahonb, S., Pykeb, C., Srebric, J., Cluster Analysis of Simulated Energy use for LEED Certified U.S. Office Buildings, Energy and Buildings, Elsevier, 2014

Hoerster, S., Cahill, B., Menzel, K., Building Performance Analysis, Reengineering an existing Data Warehouse, Forum Bauinformatik, 2012

Hoerster, S. eBusiness in AEC, Materialized Views, lecture notes, 2013

Hoerster, S., Katzemich F., Menzel, K., A Methodology for Data Logging and Retrieval from Remote Sites. European Conference on Product & Process Modelling, Vienna, Austria, 2014

Hoerster, S., Menzel, K., BIM based Classification of Building Performance Data for Advanced Analysis, in proceedings of CISBAT 2015

Hoerster, S., Willwacher, T., Menzel, K., Algorithmic Cleansing of Metered Building Performance Data, CIBW78, 2015

Holness G., BIM Gaining Momentum, ASHRAE Journal, 2008

IEC (International Electrotechnical Commission), IEC 60870-5-104:2006 https://webstore.iec.ch/publication/25035, 2016, last accessed December 2017

IEEE Standard for Utility Industry Metering Communication Protocol Application Layer (End Device Data Tables)," in IEEE Std 1377-2012 (Revision of IEEE Std 1377-1997) , vol., no., pp.1-576, Aug. 10 2012

IFC 4 Addendum 2, http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/annex/annex-c/general-usage/all.htm , last accessed 12.11.2016

ISKRA emeco, MT581, http://www.iskraemeco.si/iskraemeco/products.nsf/%28product%29/301E5E9151959602C12569F20024F8F3?OpenDocument , last accessed 11-08-2015

ISO 10303-11, Description methods: The EXPRESS language reference manual, International Standards Organisation, 2004

ISO 10303-21, Implementation methods: Clear text encoding of the exchange structure, In-ternational Standards Organisation, 2002

ISO 10303-28, Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas, International Standards Organisation, 2007

ISO 16484-5, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44998, last accessed 22.08.2016

ISO 50001, Energy Management Systems, International Standards Organisation, 2011

ISO 27001, Information technology & Security techniques, ISO/IEC 27001:2013, https://www.iso.org/standard/54534.html, 2013, last accessed December 2017

Iltron Delta G250, https://www.itron.com/local/Indonesia%20Portfolio/Delta_brochure.pdf, last accessed 11-08-2015

Inmon, W.H., Building the Data Warehouse, 3rd ed., John Wiley & Sons, 2002

International Energy Association, 2008. Energy Technology Perspectives, pg99. Paris, France: IEA Publications.

International Standards Organisation, 2011. ISO 50001: Energy management systems.

Irish Government, 2012, Strategy for Renewable Energy: 2012 to 2020, http://www.teagasc.ie/energy/Policies/docs/RenewableEnergy_Strategy2012-2020.pdf, last accessed 09/11/2015

Janitza ProData 2 , http://www.janitza.com/prodata-2-en-downloads.html?file=files/download/manuals/ProData-2/Janitza-Manual-ProData-2-en.pdf, last accessed 20.08.2015

Janitza UMG 96 RM, http://www.janitza.com/manuals-current-devices.html?file=files/download/manuals/UMG96RM/Basic/Janitza-Manual-UMG96RM-20-250V-en.pdf, last accessed 20.08.2015

Janitza UMG 96 RM, http://www.janitza.com/umg-96rm-p-m-cbm-el-overview.html , last accessed 17.08.2015

Janitza UMG 96 RM-E, http://www.janitza.com/leaflets.html?file=files/download/leaflets/UMG-96RM-E/UMG96RME-EN.pdf, last accessed 20.08.2015

Janitza UMG 96, http://www.janitza.com/umg-96l-umg-96-en.html , last accessed 17.08.2015

Kamstrup Multical 602, http://products.kamstrup.com/ajax/downloadFile.php?uid=512b545d1c678&pid=143, last accessed 20.08.2015

Kamstrup Multical 62, http://products.kamstrup.com/ajax/downloadFile.php?uid=515d521115625&pid=252, last accessed 20.08.2015

Kastner, W., Neugschwandtner, G., Soucek, S., Newman, H.M., Communication Systems for Building Automation and Control, Proceedings of the IEEE, Vol. 93, No 6, June 2005

Kimball R. and Caserta J., The Data Warehouse ETL Toolkit, John Wiley & Sons, 2004

Kimball R. and Ross M., The Data Warehouse Toolkit, 2nd ed. John Wiley & Sons, 2002

Kimball, R., Reeves, L., Ross, M., Thornthwaite, W., The Data Warehouse Life Cycle Toolkit, John Wiley & Sons, 2008

Knibbe, E.J., Building management system, https://www.google.com/patents/US5565855, Google Patents, 1996

Knox, E.M. and Ng, R. T., Algorithms for mining distance-based outliers in large datasets. Proc. Int. Conf. Very Large Data Bases, 1998

Krippendorf, M., Song, I., The Translation of Star Schema into Entity-Relationship Diagrams, Database and Expert Systems Applications, 1997

Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright. (1998). Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal of Optimization, vol. 9 no. 1, pp. 112-147

Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright., Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal of Optimization, vol. 9 no. 1, pp. 112-147, 1998

Landis + Gyr, http://www.landisgyr.com/ , last accessed 17.08.2015

Lane, P, Data Warehousing Guide, 11g Release 2 (11.2), Oracle, 2013

Lehner, W., Datenbanktechnologie fuer Data Warehouse Systems, dpunkt Verlag, pg.64ff, 2003

Lee, G., Sacks, R., Eastman, C.M., Specifying parametric building object behavior (BOB) for a building information modeling system, Automation in Construction, vol 15 no 6, pp 758 - 776, 2005

Li, W., Risk Assessment of Power Systems Models, Methods, and Applications. IEEE Press—Wiley. New York, 2005

Ljung, G.M., On outlier detection in time series. J. R. Statist. Soc. Ser. B (Methodol.), pp. 559–567, 1993

M-Bus User group. M-Bus standard EN 13757-2, Rev 4.8, 1998, [Online]. Last accessed July 2015, Available: http://www.m-bus.com/.

Ma, Z., Cooper, P., Daly, D., Ledo, L., Existing building retrofits: Methodology and state-of-the-art, Energy and Buildings 55, 2012, pg. 889–902

MATLAB, 1-D data interpolation, http://uk.mathworks.com/help/matlab/ref/interp1.html, last accessed November, 2015

Matt T., Schappacher M. and Sikora, A., "Development of a web-based monitoring device for the wired Metering Bus (M-Bus) as defined in EN13757-3," 2015 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE), Offenburg, 2015, pp. 187-194

Maynard P., McLaughlin K. & Haberler B., Towards Understanding Man-In-The-Middle Attacks on IEC 60870-5-104 SCADA Networks, 2nd International Symposium for ICS & SCADA Cyber Security Research 2014, St Pölten, Austria, September 2014

MWK Systeme Messgeraete GmbH, www.mwksysteme.de/pdf/rechenwerk.pdf, last accessed 11-08-2015

Manning, C.D., Raghavan , P., Schütze, H., Introduction to Information Retrieval, Cambridge University Press, 2008

Maroušek, J., Hašková, S., Zeman, R. et al. Clean Techn Environ Policy (2015) 17: 549. doi:10.1007/s10098-014-0800-1

Marr, B., Key Performance Indicators (KPI): The 75 measures every manager needs to know, pg. xxv, Pearson, 2012

Martyn, T. Reconsidering Multi-Dimensional Schemas,ACM SIGMOD Record, vol 33 no 1 pp 83-88, 2004

Mathworks, k-Means Clustering, http://uk.mathworks.com/help/stats/k-means-clustering.html, last accessed October 2015, 2015

McGraw Hill Construction, SmartMarket Report: The business value of BIM for construction in major global markets: How contractors around the world are driving innovation with building information modelling. McGraw Hill Construction, 2014

McSwiney, S., Analysis of Monitoring Data Aiming to Identify Devices and User Behaviour including Web-based Representation of Data and Results, Master Thesis, University College Cork, 2015

Menzel, K., Browne, D., Deng, S., Performance Indicators to Evaluate Buildings' Systems' Performance. European Conference on Product & Process Modelling, Vienna, Austria, 2014

Menzel, K., Information Modelling & Retrieval, lecture notes, 2014

Menzel, K., Katzemich, F., Mahdavi, A., Impacts of building performance monitoring on integrated energy management, CESBP Building Physics, Vienna, 2013

Mo, K., Development of an IFC-Compatible Data Warehouse for Building Performance Analysis, Minor Thesis, University College Cork, 2012

Morré, E., Computer Aided Facility Management, lecture notes, 2014

Nadkarni P., Marenco L., Chen R., Skoufos E., Shepherd G., Miller P. , Organization of Heterogeneous Scientific Data Using the EAVCR Representation , American Medical Informatics Association, 1999

Nikolaou, T.G., Kolokotsa, D.S., Stavrakakis, G.S., Skias,I.D.,"On the Application of Clustering Techniques for Office Buildings' Energy and Thermal Comfort Classification", IEEE, 2012

Nisha and Puneet, J. K., A Survey of Clustering Techniques and Algorithms, Panjab University Chandigarh, IEEE (978-9-3805-4415-1), 2015

Nizar, G., Crucianu, M., Boujemaa, N., "Unsupervised and Semi-supervised Clustering: a Brief Survey", France, 2005

Nthontho M., Chowdhury S. P. and Winberg S., "Smart communication networks standards for smart energy management," 2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC), Amsterdam, 2011, pp. 1-9

Ofgem, Office of Gas and Electricity Markets, Fact Sheet - Meter Accuracy & Billing Disputes , https://www.ofgem.gov.uk/ofgem-publications/42361/5875-factsheetmeteraccuracy-and-billingdisputes.pdf, last accessed 22.08.2016

Oracle Database SQL Language Reference, Schema Object Names and Qualifiers, Online, last accessed July 2015, http://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements008.htm#SQLRF00223

Oracle Database SQL Reference, Hierarchical Query Pseudocolumns, Online, last accessed July 2015, http://docs.oracle.com/cd/B12037_01/server.101/b10759/pseudocolumns001.htm#i1007332

PAS 1192-2:2013, Publicly Available Specification, Specification for information management for the capital/delivery phase of construction projects using building information modelling, United Kingdom, 2013

Parmenter, D., Key Performance Indicators, John Wiley & Sons, 2007

Pedersen, T.B., Jensen C.S., Multidimensional database technology, Computer, vol 34 no 12 pp 40 - 46, 2001

Petrenko M., Rada A., Fitzsimons G., McCallig E., Zuzarte C., Physical database design for data warehouse environments, IBM DB2, 2012

RMG Messtechnik GmbH, Terz 94, http://www.rmg.com/produkte/messen/gas-volumenmessung/elektronischer-turbinenradgaszaehler-terz-94.html, last accessed 11-08-2015

Rahm, E., Hai Do, H., Data Cleaning Problems and Current Approaches, Bulletin of the Technical Committee on Data Engineering, 23, 4, 2000,

Ramaswamy, S., Rastogi, R., Shim, K., Efficient algorithms formining outliers from large data sets. ACM SIGMOD Rec., vol. 29, no. 2, pp. 427–438, 2000

Relay KV001A, http://relay.de/KVoo1A_e.htm , accessed 17.05.2013

Relay PadPuls, http://relay.de/PPM4L_e.htm , accessed 17.05.2013

Rob, P., Coronel, C., Database Systems. Design, Implementation and Management. 8th Ed., 2009

Rogers, J. R. and McVay, R. C., Graphical microcontroller programming, IEEE International Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, 2012, pp. 48-52

Saia Burgess, PG5 Fupla Editor, https://www.sbc-support.com/en/services/getting-started/introduction/programming/fupla-programming/, accessed 2016

Sandhu, M., Kaur, A., Kaur, R., Data Warehouse Schemas, International Journal of Innovative Research in Advanced Engineering, Vol 2, No 4, 2015

Sensus Meitwin 100, http://sensus.com/web/uk/product?division_id=water-and-heat&product_line_id=uk-commercial-industrial&product_id=meitwin , last accessed 11-08-2015

Sensus WP Dynamic, http://sensus.com/web/uk/water-and-heat/product-line/heating-and-cooling/product/wp-dynamic-flow-sensor-uk, last accessed 23.08.2015

Simonis, H., D6.1 Specification of Systems Architecture and Concept for Integration and Upscaling, Campus21, FP7, 2013

Sirr, S. D3.2 Advanced Use Case Specification and Installation Plan, Campus21, FP7, 2012

smartB, Product description, http://www.smartb.de/product/, last accessed 29.8.2016

Socomec, Single Circuit Multi Function Meters, Diris Overview, http://www.socomec.com/single-circuit-multifunction-meters_en.html, last accessed 06-08-2015

SolarEgde PV Monitoring, http://www.solaredge.com/products/pv-monitoring#/, last accessed 22.08.2016

Stapleton, M., Development and Implementation of an IFC4-Compatible, Tabular Modeller for Building Services and Building Automation Systems, Master Thesis, UCC, 2014

Steel, J., Drogemuller, R., Toth, B., Model interoperability in building information modelling, Software & Systems Modeling, February 2012, Vol 11, No 1, pp 99–109

Su, T., Dy, J., A deterministic method for initializing k-means clustering, Tools with Artificial Intelligence, ICTAI, 16th IEEE International Conference, pp 784-786, 2004.

Sustainable Energy Authority of Ireland, Public Sector Energy Monitoring & Reporting Sys-tem, FAQ, 2015, pg 69ff. http://www.seai.ie/Your_Business/Public_Sector/Reporting/FAQ/PS_Energy_Monitoring_Reporting_System_FAQ.pdf, last accessed November 2015

Swales, A., Open Modbus TCP Specification, Schneider Electric, March 1999

Trianni, A., Cagno, E., Thollander, P., Backlund, S., Barriers to industrial energy efficiency in foundries: a European comparison, Journal of Cleaner Production 40 (2013)

W3C, Extensible Markup Language (XML), 5th ed., available online: http://www.w3.org/TR/REC-xml/ , last accessed June 2015, 2008

Wachendorff ADFWeb HD67044-B2-20, http://www.adfweb.com/download/filefold/MN67044_ENG.pdf, last accessed 20.08.2015

Wang, S., Yan, C., Xiao, F., Quantitative energy performance assessment methods for existing buildings, Quantitative energy performance assessment methods for existing buildings, Energy and Buildings 55, 2012

Webfactory, The Gap Detection and Interpolation Features, http://webfactory-support.de/knowledgebases/KB_i4ENERGY/Default.htm# Usage/GapDetectionAndInterpolation.htm%3FTocPath%3DUsage|_____14, last accessed No-vember 2015

Wei, C., Li, Y., Design of Energy Consumption Monitoring and En Electronics, Communications and Control (ICECC), 2011

Widom J., Research Problems in Data Warehousing, 4th Int'l Conference on Information and Knowledge Management (CIKM), 1995

Wikipedia , Binary coded decimal, https://en.wikipedia.org/wiki/Binary-coded_decimal, last accessed 23.08.2015

Wikipedia, Multimeter, https://en.wikipedia.org/wiki/Multimeter , last accessed 06-08-2015

Willcocks, L., Lacity, M. and Craig, A., Robotic Process Automation at Xchanging. The Outsourcing Unit Working Paper Series, June 2015

Zach, R. and Mahdavi, A. Monitoring for Simulation Validation. BauSim 2010 - Building Performance Simulation in a Changing Environment. Vienna, Austria, pp 190-195

Zhang, T., Song, X., Meng, L., Yu, J., Chen, X, Feasibility Analysis on Customized Home Evaluation for Alternate and Renewable Energy Systems, Applied Mechanics and Materials, Vols. 488-489, pp. 716-721, 2014

Zenner WPH-N, http://www.zenner.com/product_categories/category/products_bulk-water-meters/product/bulk-water-meter-woltman-wph_parallel.html, last accessed 23.08.2015

Appendix

Appendix 1

Questionnaire undertaken with Dr. Theis, department head of the Center of Competence for Energy & Sustainability at Bilfinger HSG FM on March, 1st, 2014.

General Energy Monitoring

1) Are you aware of BMS with basic energy monitoring features? If you are, do you believe these features are sufficient for sustainable building operations?

Yes they are an essential growing market but vendor independence is not given. Mid-term I believe that external energy control systems will emerge with capabilities to control regular BMS.

2) Some utility providers and manufactures of e.g. solar panels offer their customer's access to web based monitoring systems. Do you believe these systems are sufficient to monitor a building?

No. They exist to monitor their individual sub system. They do not have the overall picture as they lack information about other systems.

3) Can you highlight the most important features you expect in a monitoring system? Does the market offer an off-the-shelf monitoring system which fulfils your requirements?

For me, the most important feature is transparency of energy consumption in combination with handling instructions for operational staff to reduce energy sustainably. I am not aware of any system on the market which provides this.

Energy Monitoring Systems for FM operations

1) For which categories of buildings do you see high potentials for an energy monitoring system?

High potentials are hidden in all energy intensive building types, e.g. offices, production, shopping centres, hospitals, swimming pools

2) What outcome can be expected from using a monitoring system? (e.g. underperforming systems, leakage, consumption peaks..)

Faulty/unusual behaviour of technical systems, wrongly set operational parameters, incorrectly set run-times, low degrees of efficiency, low operational load

3) Will an energy monitoring system help you to easier reach agreed targets with a customer?

Yes most certainly. It helps us in documenting our undertaken measures.

4) For what duration is it worth storing historical meter and sensor readings from a customer?

2 years is sufficient for most tasks. More than 5 years is not necessary due to changes in buildings (e.g. replacement of technical systems, different building usage behaviours)

Customer Relationship

1) Why would a customer become interested in an energy monitoring system?

Mainly to save cost but also for creating transparency in energy consumption.

2) Have you experienced an increase in inquiries for monitoring systems?

Customers' awareness for monitoring systems is still very low.

3) How important is a monitoring system for you to win new customers?

A system alone does not help to win customers. This comes once combined with service and consultation.

4) Are you afraid that you could lose customers if you cannot provide a monitoring system in the near future?

This is likely to happen mid-term as added values can only be provided through holistic services.

5) Do you find it challenging to convince a customer of the added value he obtains when investing in an energy monitoring system?

Yes absolutely.

Appendix 2

The script creates and populates the organization dimension for the OLAP cube based on the IFC meta model definition.

```
create or replace view v_dim_organization as (
SELECT
-- stephan
-- find out the primary key for the MV
CASE WHEN org.identification IS NOT NULL THEN org.identification
     WHEN org2.identification IS NOT NULL THEN org2.identification
     WHEN org3.identification IS NOT NULL THEN org3.identification
     ELSE org4.identification
     END AS identification,

-- as NULL's are not supported by Cubes we replace all NULL by N/A
CASE WHEN org.identification IS NULL THEN 'N/A' ELSE org.identification END
AS grandgrandchild_id,
CASE WHEN org.name IS NULL THEN 'N/A' ELSE org.name END AS
grandgrandchild_name,
CASE WHEN org.description IS NULL THEN 'N/A' ELSE org.description END AS
grandgrandchild_description,
CASE WHEN org2.identification IS NULL THEN 'N/A' ELSE org2.identification
END AS grandchild_id,
CASE WHEN org2.name IS NULL THEN 'N/A' ELSE org2.name END AS
grandchild_name,
CASE WHEN org2.description IS NULL THEN 'N/A' ELSE org2.description END AS
grandchild_description,
CASE WHEN org3.identification IS NULL THEN 'N/A' ELSE org3.identification
END AS child_id,
CASE WHEN org3.name IS NULL THEN 'N/A' ELSE org3.name END AS child_name,
CASE WHEN org3.description IS NULL THEN 'N/A' ELSE org3.description END AS
child_description,
org4.identification AS parent_id,
org4.name AS parent_name,
org4.description AS parent_description
FROM (

-- cutting path and sorting
SELECT
       CASE WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is not null
THEN substr(regexp_substr(path, '-[^-]*', 1, 1),2)
            WHEN regexp_count(path,'-') < 3 THEN 'N/A'
            ELSE child
       END AS parent1,
       CASE WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is not null
THEN substr(regexp_substr(path, '-[^-]*', 1, 2),2)
            WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 3),2) is not null THEN
substr(regexp_substr(path, '-[^-]*', 1, 1),2)
            WHEN regexp_count(path,'-') < 2 THEN 'N/A'
            ELSE child
       END AS parent2,
       CASE WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is not null
THEN substr(regexp_substr(path, '-[^-]*', 1, 3),2)
            WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 3),2) is not null THEN
substr(regexp_substr(path, '-[^-]*', 1, 2),2)
            WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 3),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 2),2) is not null THEN
substr(regexp_substr(path, '-[^-]*', 1, 1),2)
            WHEN regexp_count(path,'-') < 1 THEN 'N/A'
            ELSE child
       END AS parent3,
       CASE WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is not null
THEN substr(regexp_substr(path, '-[^-]*', 1, 4),2)
            WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 3),2) is not null THEN
substr(regexp_substr(path, '-[^-]*', 1, 3),2)
            WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 3),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 2),2) is not null THEN
substr(regexp_substr(path, '-[^-]*', 1, 2),2)
            WHEN substr(regexp_substr(path, '-[^-]*', 1, 4),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 3),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 2),2) is null and
substr(regexp_substr(path, '-[^-]*', 1, 1),2) is not null THEN
substr(regexp_substr(path, '-[^-]*', 1, 1),2)
            ELSE child
       END AS parent4


from (
-- resolve hierarchy rescursively
SELECT CONNECT_BY_ROOT relatedorganizations as child,
sys_connect_by_path(relatingorganization,'-') path
FROM ifcorganizationrelationship where CONNECT_BY_ISLEAF = 1
CONNECT BY PRIOR relatingorganization = relatedorganizations
)
) rel
LEFT JOIN ifcorganization org ON ( rel.parent1 = org.identification)
LEFT JOIN ifcorganization org2 ON ( rel.parent2 = org2.identification)
LEFT JOIN ifcorganization org3 ON ( rel.parent3 = org3.identification)
LEFT JOIN ifcorganization org4 ON ( rel.parent4 = org4.identification)
);

create materialized view mv_dim_organization as select * from
v_dim_organization;
```

Appendix 3

The script creates and populates the system dimension for the OLAP cube based on the IFC meta model definition.

```
create or replace view v_dim_system as (
select
FlowElementGUID,
FlowElementName,
FlowElementDescription,
FlowElementObjecttype,
type,
CircuitGUID,
CircuitName,
CircuitLongName
from
(

select
  GLOBALID as FlowElementGUID,
  name as FlowElementName,
  description as FlowElementDescription,
  objecttype as FlowElementObjecttype,
--  'METER' as TYPE,

    CASE WHEN (select relatingobject from IFCRELAGG_CIRCUITANDMETER where
IFCRELAGG_CIRCUITANDMETER.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid) IS NULL
    THEN
      'meter'
    ELSE
      'pipe'
    END AS type,

  CASE WHEN (select relatingobject from IFCRELAGG_CIRCUITANDMETER where
IFCRELAGG_CIRCUITANDMETER.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid) IS NULL
    THEN
      (select globalid from IFCDISTRIBUTIONCIRCUIT where globalid =
        (select relatingobject from IFCRELAGG_CIRCUITANDPIPE where
IFCRELAGG_CIRCUITANDPIPE.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid))
    ELSE
      (select globalid from IFCDISTRIBUTIONCIRCUIT where globalid =
        (select relatingobject from IFCRELAGG_CIRCUITANDMETER where
IFCRELAGG_CIRCUITANDMETER.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid)) END AS CircuitGUID,

  CASE WHEN (select relatingobject from IFCRELAGG_CIRCUITANDMETER where
IFCRELAGG_CIRCUITANDMETER.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid) IS NULL
    THEN
      (select name from IFCDISTRIBUTIONCIRCUIT where globalid =
        (select relatingobject from IFCRELAGG_CIRCUITANDPIPE where
IFCRELAGG_CIRCUITANDPIPE.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid))
    ELSE
      (select name from IFCDISTRIBUTIONCIRCUIT where globalid =
        (select relatingobject from IFCRELAGG_CIRCUITANDMETER where
IFCRELAGG_CIRCUITANDMETER.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid)) END AS CircuitName,

  CASE WHEN (select relatingobject from IFCRELAGG_CIRCUITANDMETER where
IFCRELAGG_CIRCUITANDMETER.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid) IS NULL
    THEN
      (select longname from IFCDISTRIBUTIONCIRCUIT where globalid =
        (select relatingobject from IFCRELAGG_CIRCUITANDPIPE where
IFCRELAGG_CIRCUITANDPIPE.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid))
    ELSE
      (select longname from IFCDISTRIBUTIONCIRCUIT where globalid =
        (select relatingobject from IFCRELAGG_CIRCUITANDMETER where
IFCRELAGG_CIRCUITANDMETER.relatedobjects =
IFCDISTRIBUTIONFLOWELEMENT.globalid)) END AS CircuitLongName
```

```
  from IFCDISTRIBUTIONFLOWELEMENT
)
where CircuitGUID is not null
);

create materialized view mv_dim_system as select * from v_dim_system;
```

Appendix 4

The script creates and populates the spatial dimension for the OLAP cube based on the IFC meta model definition.

```
create or replace view v_dim_spatial as (
select
  GLOBALID as Space_GUID,
  row_number() OVER (ORDER BY  name) as nr,
  name as Space_Name,
  description as Space_Description,
  (select globalid from ifcbuildingstorey where globalid =
    (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects = ifcspace.globalid)) as Storey_GUID,
  (select name from ifcbuildingstorey where globalid =
    (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects = ifcspace.globalid)) as Storey_Name,
  (select description from ifcbuildingstorey where globalid =
    (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects = ifcspace.globalid)) as
Storey_Description,
  (select globalid from ifcbuilding where globalid =
    (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects =
      (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects = ifcspace.globalid))) as Building_GUID,
  (select name from ifcbuilding where globalid =
    (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects =
      (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects = ifcspace.globalid))) as Building_Name,
  (select description from ifcbuilding where globalid =
    (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects =
      (select relatingobject from ifcrelaggregates where
ifcrelaggregates.relatedobjects = ifcspace.globalid))) as
Building_Description
from ifcspace where exists (select * from ifcrelaggregates where
ifcspace.globalid = ifcrelaggregates.relatedobjects)
);

create materialized view mv_dim_spatial as select * from v_dim_spatial;
```

Appendix 5

This PHP code creates SQL statements for creating a time dimension in the DWH. Each year needs to be generated separately. They year can be set in the beginning of the script in the configuration section. The output is then used to create and populate the time dimension for the OLAP cube.

```
<?php
$firstpart = 'INSERT INTO TIMEDIM ( DAY_KEY, CALENDAR_YEAR_ID,
CALENDAR_YEAR_NAME, CALENDAR_YEAR_TIME_SPAN, CALENDAR_YEAR_END_DATE,
CALENDAR_QUARTER_ID, CALENDAR_QUARTER_NAME,
CALENDAR_QUARTER_TIME_SPAN, CALENDAR_QUARTER_END_DATE, MONTH_ID,
MONTH_NAME, MONTH_TIME_SPAN, MONTH_END_DATE, WEEK_ID, WEEK_NAME,
WEEK_TIME_SPAN, WEEK_END_DATE) VALUES';

// config start
$year = '2014';
// config end

$year_short = substr($year,2);
$day = new DateTime('01-JAN-' . $year_short);

if ($year % 4 == 0) //leap year
    $daysinyear = 366;
else
    $daysinyear = 365;

for ($i=0; $i<$daysinyear;$i++)
{
    echo $firstpart;

    $curMonth = date("m", $day->getTimestamp());
    $curWeek = date("W", $day->getTimestamp());
    $curQuarter = ceil($curMonth/3);
    $unixtime = getdate($day->getTimestamp());

    echo "( '" . $day->format('d-M-y');
    echo "', 'CY" . $year . "', 'CY" . $year . "', '";
    echo date("z", mktime(0,0,0,12,31,$year)) + 1;
    echo "', '31-DEC-" . $year_short . "', '";
    echo "Q" . $curQuarter . "CY" . $year . "', 'Q" . $curQuarter . "CY" .
$year . "', '";

    if ( ($unixtime['mon'] >= 1) && ($unixtime['mon'] <= 3) )
        if ($year % 4 == 0) //leap year
            echo "91', '31-MAR-" . $year_short . "', '";
        else
            echo "90', '31-MAR-" . $year_short . "', '";
    else if ( ($unixtime['mon'] >= 4) && ($unixtime['mon'] <= 6) )
        echo "91', '30-JUN-" . $year_short . "', '";
    else if ( ($unixtime['mon'] >= 7) && ($unixtime['mon'] <= 9) )
        echo "92', '30-SEP-" . $year_short . "', '";
    else if ( ($unixtime['mon'] >= 10) && ($unixtime['mon'] <= 12) )
        echo "92', '31-DEC-" . $year_short . "', '";
    else
        echo "XXXXX";

    echo substr($unixtime['month'],0,3) . $year . "', '" .
substr($unixtime['month'],0,3) . $year . "', ";
```

```
    echo "'" . cal_days_in_month(CAL_GREGORIAN, $unixtime['mon'], $year) .
"', '";
    echo gmdate("d-M-y", strtotime('last day of this month',$day-
>getTimestamp()));
    echo "', '" . $curWeek  . "WK" . $year . "', '" . $curWeek  . "WK" .
$year . "', '7',";

    $current_day = date("N", $day->getTimestamp());
    if ($current_day == 7) // 7 = Sunday. We assume that Sunday is the last
day in the week.
        echo "'" . gmdate("d-M-y", strtotime('This Sunday',$day-
>getTimestamp())) . "');";
    else
        echo "'" . gmdate("d-M-y", strtotime('Next Sunday',$day-
>getTimestamp())) . "');";

    echo "<br>";

    $day->modify('+1 day');
}

?>
```

Appendix 6

This SQL script generates the surrogate fact table which will be utilised as fact table for the OLAP cube.

```
create materialized view mv_fact1 as
SELECT

datestamp,
datastream,
system,
count,
interval,
room,
organization,
goodofficehours,
badofficehours,
-- more kpi
count - goodofficehours - badofficehours AS outsideofficehours,
round(goodofficehours / (goodofficehours + badofficehours +0.0001) * 100,2)
as percentgoodhours,
round(badofficehours / (goodofficehours - badofficehours +0.0001) * 100,2)
as percentbadhours,
min,
max
FROM
      (
      select

      datestamp,
      datastream,
      count,
      interval,
      goodofficehours,
      badofficehours,

      -- resolve system relationship
      CASE
      WHEN (select systemid from rel_system_datastream where datastream =
readings.datastream) IS NOT NULL THEN (select systemid from
rel_system_datastream where datastream = readings.datastream)
      ELSE null
      END AS system,

      -- resolve room relationship
      CASE
      -- eri rooms
      WHEN (select relatingstructure from ifcrelciss_node where
relatedelements = readings.datastream) IS NOT NULL THEN (select
relatingstructure from ifcrelciss_node where relatedelements =
readings.datastream)
      -- cee rooms
      WHEN (select relatingstructure from ifcrelciss_sensor where
relatedelements = (select relatingsensor from ifcrelagg_sensoranddatapoint
where relateddatapoint = readings.datastream)) IS NOT NULL THEN (select
relatingstructure from ifcrelciss_sensor where relatedelements = (select
relatingsensor from ifcrelagg_sensoranddatapoint where relateddatapoint =
readings.datastream))
      ELSE null
      END AS room,


      -- resolve organization relationship
      CASE
      -- eri organizations
      WHEN
             (select organization from ifcpersonandorganization where
ownerid =
               (select ownerhistory from ifcspace where globalid =
                 (select relatingstructure from ifcrelciss_node where
relatedelements = readings.datastream))) IS NOT NULL THEN
             (select organization from ifcpersonandorganization where
ownerid =
               (select ownerhistory from ifcspace where globalid =
                 (select relatingstructure from ifcrelciss_node where
relatedelements = readings.datastream)))
      -- cee organizations
      WHEN
             (select organization from ifcpersonandorganization where
ownerid =
               (select ownerhistory from ifcspace where globalid =
                 (select relatingstructure from ifcrelciss_sensor where
relatedelements =
                   (select relatingsensor from ifcrelagg_sensoranddatapoint
where relateddatapoint = readings.datastream)))) IS NOT NULL THEN
             (select organization from ifcpersonandorganization where
ownerid =
               (select ownerhistory from ifcspace where globalid =
                 (select relatingstructure from ifcrelciss_sensor where
relatedelements =
                   (select relatingsensor from ifcrelagg_sensoranddatapoint
where relateddatapoint = readings.datastream))))
      ELSE null
      END AS organization,
      max,
      min

      FROM
      (

          SELECT

            -- general data
            TO_date(trunc(timestamp),'DD-MM-YY') AS Datestamp,
            id AS datastream,
            count(value) AS count,
            round(1440 / (count(value)+1)) AS interval,

            -- kpi calculation
            count(case when value <= 26 and value >= 18 and
TO_char(timestamp, 'HH24:MI') > '09:00' and TO_char(timestamp, 'HH24:MI')
<= '17:00' then 1 end) as goodofficehours,
            count(case when (value > 26 or value < 18) and
TO_char(timestamp, 'HH24:MI') > '09:00' and TO_char(timestamp, 'HH24:MI')
<= '17:00' then 1 end) as badofficehours,

            -- general data
            max(value) AS max,
            min(value) AS min

            FROM FACT2014
            GROUP BY  TO_date(trunc(timestamp),'DD-MM-YY') ,id

      ) readings
)
WHERE room is not null AND organization is not null and system IS NOT NULL
ORDER BY datestamp
;
```

Appendix 7

This MATLAB code performs data clustering as introduced in chapter 3.

```
M = RAW_ELEC;
dailyval = 96;

days = floor(size(M)/dailyval);
days = days(:,1);
%M2 should contain a multiple of 96
M2 = M(1:days*dailyval);
%M_analysis will have 96 columns and rows will equal the days
M_analysis = reshape(M2',[dailyval,days])';
% easier readable?
M_analysis = reshape(M2,[dailyval,days]);
M_analysis = M_analysis' ;

x = 'Daily readings in 15 minutes interval';

weeks = floor(size(M)/(dailyval*7));
weeks = weeks(:,1);
M2 = M(1:weeks*(dailyval*7));
M_analysis = reshape(M2',[(dailyval*7),weeks])';
x = 'Time in a week';

d=1;

% for automated k detection
ds=[];
for k=1:5
    [ms, inds, mindists] = mykmeans_pca(M_analysis,k,1);
    ds(k)=sum(mindists);
end
thresh = (ds(1) + 2*ds(5))/3;
a=find(ds < thresh);
k = a(1);

% for manual k selection from a gui
k = 5;

% run k-means matlab algorithm
[ms, inds, mindists, us, vs, ss] = mykmeans_pca(M_analysis,k,d);


% plotting and formatting
theplot = plot(ms');
ylabel('Consumption (kWh)');
xlabel(x);

% make legend
the_legend=cell(k,1);
for kk=1:k
    the_legend{kk} = sprintf('Cluster %d (%d days)', kk, sum(inds==kk));
end
legend(the_legend);

% calculate deviation - to be shown in the plot

if get(chkDeviation,'Value')
    % plot variation in each cluster
    colors = get(theplot, 'Color');
    hold on
    for kk=1:k
        u = us{kk};
        v = vs{kk};
        s=ss{kk};
        m=ms(kk,:)';
        %ddd=1;
        % generate typical values in the cluster
        %msk = repmat(m,1,2^ddd) + u(:,1:ddd)*diag(s(1:ddd))*[-.5,-
.5,.5,.5;-.5,.5,-.5,.5]*mean(abs(v(:,1)));
        msk = repmat(m,1,2) + u(:,1)*diag(s(1))*[-
.5,+.5]*mean(abs(v(:,1)));
        plot(msk,':', 'Color', colors{kk})
    end
    hold off
end;
```

These helper functions are called from within the main script:

My K-Means Function

```
function [ms, inds, mindists,us,vs,ss] = mykmeans_pca(data, k, d)

[n,m] = size(data);

% pick k means at random
% ms = data(ceil(rand(1,k)*n),:);

% introduction of the "ste_randomizer" ;-)
% to achieve reproducable outputs
% ste = [];
% randval = 0.1;
% for i=1:k
%     ste = [ste randval];
%     randval = randval + 0.05;
% end;
% ms = data(ceil(ste*n),:);

ms = kmeans_seed_pcasplit(data,k);

dists = zeros(n,k);

% assign to closest mean
for l=1:k
  dists(:,l) = sum((data - repmat(ms(l,:),n,1)).^2,2);
end
[mindists , inds] = min(dists,[],2);

% compute new means and principal vectors
us = cell(k);
vs = cell(k);
ps = cell(k);
ss = cell(k);
for l=1:k
  [c,u,s,v]=centeredsvd(data(inds==l,:));
  ms(l,:) = c;
  us{l} = v(:,1:d);
  ps{l} = v(:, d+1:end) * v(:, d+1:end)';
end

for step=1:20
  % assign to closest mean
  for l=1:k
    diffs = data - repmat(ms(l,:),n,1);
    %%% Comment or uncomment the following line to "punish" principal
intra-cluster
    %%% variance or not
    %diffs = diffs * ps{l};
    dists(:,l) = sum(diffs.^2,2);
  end
  [mindists , inds] = min(dists,[],2);

  % compute new means and principal vectors
  for l=1:k
    [c,u,s,v]=centeredsvd(data(inds==l,:));
    ms(l,:) = c;
    us{l} = v(:,1:d);
    vs{l} = u(:,1:d);
    ps{l} = v(:, d+1:end) * v(:, d+1:end)';
    ss{l} = diag(s);
    ss{l} = ss{l}(1:d);
  end
end

function [c,u,s,v]=centeredsvd(A)

c = mean(A);
n=size(A,1);
B=A-repmat(c,n,1);
[u s v]=svd(B);


function [ ms ] = kmeans_seed_pcasplit( A, k )
%kmeans_seed_pcasplit Finds initial clusters for k means by pcasplit
%   algorithm
% Input:
%   A: n x m array of n data points
%   k: desired number of means
% Output:
%   ms: kxm array of means

ms=mean(A);
As{1} = A;
sses(1)= sse(A,ms);

for kk=2:k
   % find largest sse cluster
   [~,j] = max(sses);

   % pca
   [c,u,s,v]=centeredsvd(A);

   % split in the middle
   Aproj = As{j} * v(:,1);
   mm=mean(Aproj);
   A1 = A(Aproj >mm,:);
   A2 = A(Aproj <=mm,:);

   % replace split cluster
   As{j} = A1;
   ms(j,:) = mean(A1);
   sses(j) = sse(A1, ms(j,:));

   % create new cluster
   As{kk} = A2;
   ms(kk,:) = mean(A2);
   sses(kk) = sse(A2, ms(kk,:));

end

end

function [s] = sse(A,m)
    s = sum(sum((A-repmat(m,size(A,1),1)).^2));

end
```

Appendix 8

This appendix reflects the data analysis tool discussed in chapter 5. It is separated into three classes: gui, load_data_db and plot_data.

gui.m

```
fh = figure('Units', 'Pixels', 'OuterPosition', [100 100 800 500],
'Toolbar', 'none', 'Menu', 'none');
set(fh,'Name','Load Curve Analysis v1.0 (c) sho','NumberTitle','off')

% Tab Component
p = uiextras.TabPanel('Parent', fh, 'Padding', 5 );


% tab 1 - Load data
b1 = uiextras.VBox('Parent', p, 'Spacing', 5);

cmbFolder = uicontrol('Parent',
b1,'Style','popupmenu','String',{'Arena','Hospital','University'},'Value',1
);
b1top = uiextras.HBox('Parent', b1, 'Spacing', 5);

% tab 2 - Load Curve Analysis
b2main = uiextras.VBox('Parent', p,  'Spacing', 5);
b2 = uiextras.HBox('Parent', b2main,  'Spacing', 5);
a2 = axes('Parent', b2);

% tab 3 - All weeks plotted on top of each other
b3main = uiextras.VBox('Parent', p,  'Spacing', 5);
b3 = uiextras.HBox('Parent', b3main,  'Spacing', 5);
a3 = axes('Parent', b3);

% tab 4 - Load Curve Analysis
b4main = uiextras.VBox('Parent', p,  'Spacing', 5);
b4 = uiextras.HBox('Parent', b4main,  'Spacing', 5);
a4 = axes('Parent', b4);

% tab 5 - Daily Plots
b5 = uiextras.VBox('Parent', p,  'Spacing', 1);
cmbDaySelect=uicontrol('Parent',b5,'Style','popupmenu',
'String',{'Saturday','Sunday','Monday','Tuesday','Wednesday','Thursday','Fr
iday'});
b5top = uiextras.HBox('Parent', b5,  'Spacing', 1);
a5 = axes('Parent', b5);
set( b5, 'Sizes', [1 28 -1] );

% tab 6 - JDL
b6main = uiextras.VBox('Parent', p,  'Spacing', 5);
b6 = uiextras.HBox('Parent', b6main,  'Spacing', 5);
a6 = axes('Parent', b6);

%tab 7 - Simple Load curve with Min/Max Output
b7 = uiextras.HBox('Parent', p, 'Spacing', 5);
a7 = axes('Parent', b7);

b7r = uiextras.VBox('Parent', b7, 'Spacing', 5);
```

```
uicontrol('Parent', b7r, 'Style', 'text', 'String', 'Min');
txtTotalMin = uicontrol('Parent', b7r, 'Style', 'text', 'String', '0');
uicontrol('Parent', b7r, 'Style', 'text', 'String', 'Min Date');
txtTotalMinDate = uicontrol('Parent', b7r, 'Style', 'text', 'String', '0');
uicontrol('Parent', b7r, 'Style', 'text', 'String', 'Max');
txtTotalMax = uicontrol('Parent', b7r, 'Style', 'text', 'String', '0');
uicontrol('Parent', b7r, 'Style', 'text', 'String', 'Max Date');
txtTotalMaxDate = uicontrol('Parent', b7r, 'Style', 'text', 'String', '0');

set( b7, 'Sizes', [-1 105]);
set( b7r, 'Sizes', [20 20 20 28 20 20 20 28] );


% tab 8 - Day with highest consumption
b8main = uiextras.VBox('Parent', p,  'Spacing', 5);
b8 = uiextras.HBox('Parent', b8main,  'Spacing', 5);
a8 = axes('Parent', b8);

% tab 9 - Daily Peak Value (taegliches leistungsmaxima)
b9main = uiextras.VBox('Parent', p,  'Spacing', 5);
b9 = uiextras.HBox('Parent', b9main,  'Spacing', 5);
a9 = axes('Parent', b9);

% tab 10 - Daily Total Consumption
b10main = uiextras.VBox('Parent', p,  'Spacing', 5);
b10 = uiextras.HBox('Parent', b10main,  'Spacing', 5);
a10 = axes('Parent', b10);




% Set Tab Names - no blanks supported
set(p, 'TabNames', {'Load Data', 'LoadCurveAnalysis', 'AllWeeks',
'TotalWeeks',
'DailyPlots','JDL','AllValues','HighestDay','DailyMax','DailyTotals' });



% wire up callback
%tab1
set(cmbFolder,'Callback', 'load_data_db');
set(cmbFolder,'Value', 10);

%tab5
set(cmbDaySelect,'Value', 1);
set(cmbDaySelect,'Callback', 'plot_data');

%load data
load_data_db
```

load_data_db

```
% load the dataset

%connect to database
```

```
conn =
database('orcl','USER','PW','oracle.jdbc.driver.OracleDriver','jdbc:oracle:
thin:@zuse3.ucc.ie:1521:');

%retrieve data

% ORDER BY resolves some strange outputs - apparently readings were not
% always sequential. still wondering how that can be possible.
if (get(cmbFolder,'Value') == 1)
    curs3 = exec(conn,'SELECT * FROM ARENA_ALL_ELEC ORDER BY time ASC');
elseif (get(cmbFolder,'Value') == 4)
    curs3 = exec(conn,'SELECT * FROM KRANKENHAUS_ELEC WHERE ELEC < 150
ORDER BY time ASC');
elseif (get(cmbFolder,'Value') == 5)
    curs3 = exec(conn,'SELECT * FROM ERI_ELEC WHERE ELEC < 150 ORDER BY
time ASC');
else
    error('Invalid entry selected. Dying.');
end;

curs3 = fetch(curs3);
data3 = curs3.data;


%%% ELEC
RAW_ELEC = data3(:,2);
RAW_ELEC = cell2mat(RAW_ELEC);
RAW_ELEC_T = data3(:,1);
RAW_ELEC_T = datevec(RAW_ELEC_T);

 % kill first entries so file starts with 00:00
inds= find(RAW_ELEC_T(:,4)==0 & RAW_ELEC_T(:,5)==0);
ind = inds(1);
RAW_ELEC_T=RAW_ELEC_T(ind:end,:);
RAW_ELEC = RAW_ELEC(ind:end);
RAW_ELEC_T = datenum(RAW_ELEC_T);

RAW_ELEC_Tvec = datevec(RAW_ELEC_T);
RAW_ELEC_Tnum = datenum(RAW_ELEC_Tvec);


%elec = [RAW_ELEC_T RAW_ELEC];
elec = [round(datenum(RAW_ELEC_T)) RAW_ELEC];



plot_data
return;
```

plot_data

```
%%% tab 2
axes(a2);
cla(a2);

%Sunday =1, Saturday = 7;
DayNumber = weekday(datestr(RAW_ELEC_Tnum));
```

```
elec = [RAW_ELEC_Tvec DayNumber RAW_ELEC];
%filter offpeak values: all sat/sun + before 8am and after 8pm
offpeak_raw = elec( (elec(:,7) == 1 | elec(:,7) == 7) | (elec(:,7) > 1 &
elec(:,7) < 7) & (elec(:,4) < 8 | elec(:,4) > 19), 1:8 );
offpeaksum = sum(offpeak_raw(:,8));
offpeakavg = mean(offpeak_raw(:,8));
yearlypeak = elec(elec(:,8) == max(elec(:,8)) , 1:8);

yearly_avg = [];
firsthalf_avg = [];
secondhalf_avg = [];
yearly_min = [];
yearly_max = [];
for i =1:672
    yearly_avg(i) = mean(RAW_ELEC(i:672:end));
    firsthalf_avg(i) = mean(RAW_ELEC(i:672:17520));
    secondhalf_avg(i) = mean(RAW_ELEC(i+17472:672:end));
    yearly_min(i) = min(RAW_ELEC(i:672:end));
    yearly_max(i) = max(RAW_ELEC(i:672:end));
end

total_avg = mean(yearly_avg);
baseload = offpeakavg / total_avg * 100;
peakload = (total_avg - offpeakavg) / total_avg * 100;
totalkw = sum(RAW_ELEC);

hold on
plot(yearly_avg, 'color', 'red');
plot(firsthalf_avg, 'color', 'green');
plot(secondhalf_avg, 'color', 'magenta');
plot(yearly_min, 'color', 'yellow');
plot(yearly_max, 'color', 'black');
plot([1,672],[total_avg,total_avg], 'color', 'blue');
plot([1,672],[offpeakavg,offpeakavg], 'color', 'cyan');


legend('Yearly AVG', '1st half AVG', '2nd half AVG', 'Yearly MIN', 'Yearly
MAX', 'Total AVG','Offpeak AVG');

ylabel('Consumption (kWh)');
[x y1] = weekday(RAW_ELEC_Tnum(1+96*0));
[x y2] = weekday(RAW_ELEC_Tnum(1+96*1));
[x y3] = weekday(RAW_ELEC_Tnum(1+96*2));
[x y4] = weekday(RAW_ELEC_Tnum(1+96*3));
[x y5] = weekday(RAW_ELEC_Tnum(1+96*4));
[x y6] = weekday(RAW_ELEC_Tnum(1+96*5));
[x y7] = weekday(RAW_ELEC_Tnum(1+96*6));
set(gca,'XTickLabel',{y1 y2 y3 y4 y5 y6 y7});

hold off;


%%% tab 3
axes(a3);
cla(a3);

hold on;
M = RAW_ELEC;
weeks = floor(size(M)/672);
weeks = weeks(:,1);
M2 = M(1:weeks*672);
```

```
M_analysis = reshape(M2',[672,weeks])';

[x1,x2] = size(M_analysis);
if (x1 >= 52)
    plot(M_analysis(1:52,:)');
    ylabel('Consumption (kWh)');
    set(gca,'XTickLabel',{y7 y1 y2 y3 y4 y5 y6});
end;
hold off;

%%% tab 4
axes(a4);
cla(a4);

hold on;
if (x1 >= 52)
    plot(sum(M_analysis(1:52,:)'));
    ylabel('Consumption (kWh)');
    xlabel('Weeks');
end;
hold off;


%%% tab 5
axes(a5);
cla(a5);

day = get(cmbDaySelect,'Value');

elec2 = elec( (elec(:,7) == day)  , 1:8);
elec3 = elec2(:,8);
numbdays = floor(size(elec3)/96);
numbdays = numbdays(:,1);
elec4 = elec3(1:numbdays*96);
elec5 = reshape(elec4',[96,numbdays(:,1)]);

yearly_avg = [];
yearly_min = [];
yearly_max = [];
for i =1:96
    yearly_avg(i) = mean(elec5(i:96:end));
    yearly_min(i) = min(elec5(i:96:end));
    yearly_max(i) = max(elec5(i:96:end));
end

[x y1] = weekday(day,'long');

hold on;
plot(yearly_avg, 'color', 'black', 'Linewidth', 2);
plot(yearly_min, 'color', 'blue', 'Linewidth', 2);
plot(yearly_max, 'color', 'blue', 'Linewidth', 2);
plot(elec5, 'color', 'red');
plot(yearly_avg, 'color', 'black', 'Linewidth', 2);
legend('Daily AVG', 'Daily MIN', 'Daily MAX', 'Individual Days');
ylabel('Consumption (kWh)');
xlabel(strcat(y1 , ' Readings'));
hold off;


%%% tab 6 JDL
```

```
axes(a6);
cla(a6);

plot(sort(RAW_ELEC,'descend'));
ylabel('Consumption (kWh)');
xlabel('Readings');

%%% tab 7 All values
axes(a7);
cla(a7);


elec = [RAW_ELEC_Tvec DayNumber RAW_ELEC];
totalmin = elec( (elec(:,8) == min(elec(:,8)) ), 1:8);
totalmax = elec( (elec(:,8) == max(elec(:,8)) ), 1:8);
totalminval = totalmin(1:1,8);
totalmaxval = totalmax(1:1,8);
totalmintime = datenum(totalmin(1:1,1:6));
totalmaxtime = datenum(totalmax(1:1,1:6));

set(txtTotalMin, 'String', totalminval);
set(txtTotalMinDate, 'String', datestr(totalmintime));
set(txtTotalMax, 'String', totalmaxval);
set(txtTotalMaxDate, 'String', datestr(totalmaxtime));

plot(RAW_ELEC);
ylabel('Consumption (kWh)');
xlabel('Time samples');


%%% tab 8 day with highest load
axes(a8);
cla(a8);
elec = [RAW_ELEC_Tvec DayNumber RAW_ELEC RAW_ELEC_Tnum];
%sorted =  elec (elec(:,8) == sort(elec(:,8),'descend') , 1:8);
%highest = sorted(1,:);
%highestday = datenum(highest(1:3));

totalmax = elec( (elec(:,8) == max(elec(:,8)) ), 1:8);
totalmaxtime = datenum(totalmax(1:1,1:3));

z = elec(elec(:,9) < addtodate(totalmaxtime, 1, 'day'), 1:9);
z = z(z(:,9) >= totalmaxtime , 1:9);
plot(z(:,8));
ylabel('Consumption (kWh)');
xlabel(datestr(totalmaxtime));

%%% tab 9 Daily Load Maxima
axes(a9);
cla(a9);

elec = [RAW_ELEC_Tvec DayNumber RAW_ELEC];

elec_day = datenum(elec(:,1:3));
daily_max = grpstats(elec(:,8), elec_day, {'max'});

hold on
plot(daily_max, 'color', 'black');
ylabel('Consumption (kWh)');
xlabel('Days');
```

```
hold off;

%%% tab 10 Daily total consumption
axes(a10);
cla(a10);

elec = [RAW_ELEC_Tvec DayNumber RAW_ELEC/1000];

elec_day = datenum(elec(:,1:3));
daily_total = grpstats(elec(:,8), elec_day, {'sum'});

hold on
plot(daily_total, 'color', 'black');
ylabel('Consumption (MWh)');
xlabel('Days');
hold off;
```