DEVELOPMENT OF AN IFC-COMPATIBLE DATA WAREHOUSE FOR BUILDING PERFORMANCE ANALYSIS

Kai Mo, Researcher, kmo@ucc.ie
Karsten Menzel, Professor, k.menzel@ucc.ie
Stephan Hoerster, PhD Candidate, s.c.hoerster@umail.ucc.ie
IRUSE, School of Engineering, University College Cork, Ireland

ABSTRACT

This paper demonstrates how to apply a standardized Building Information Model (BIM) specified in IFC (Industry Foundation Classes) for Building Performance Monitoring. It describes how the IFC meta model is used to implement a Data Warehouse (DW) solution allowing the management and analysis of 'dynamic' buildings' performance data (compiled from sensors and meters) and 'static' descriptive data (product model). The consolidated data can then be shared in a structured way amongst different software applications, such as design software, energy simulation packages, building energy management tools, or facilities management software.

This standardized BIM provides an integrated data repository which supports the analysis and management of building performance data. To ensure that the benefits from this data source are maximized, it is essential that such a BIM provides reliable storage, effective sharing, and efficient retrieval of information. The success of this integrated DW-environment is based on an open and transparent shared knowledge source to support common language, communication protocols and standards. By supporting the IFC-standard (ISO PAS 16739), it greatly reduces the risk of miscommunication and improves data quality.

The research achievements presented in this paper build upon results associated with the EU-FP7 projects CAMPUS 21 and BaaS. The paper provides an overview of how bulk data compiled from two demonstration sites is managed and proposes a new approach on how the data should be structured within the DW based on the IFC standard. The study discusses the integration of a BIM solution with an IFC-compatible DW and outlines how existing BIM based IFC objects can be further enhanced through a synchronous relationship with an IFC structured DW.

Finally this paper illustrates the capability of using the IFC standard for data mapping and implementation of an IFC-compatible DW. Information modeling scenarios are presented and their applicability is demonstrated.

Keywords: building information modeling, building performance management, data warehouse technology

1. INTRODUCTION

Work on BIM and Data Warehouse Technology is in the heart of research projects of the Informatics Research Unit for Sustainable Engineering (IRUSE). Currently, tens of millions of data sets compiled from sensors, meters and actuators installed in living laboratories operated by University College Cork (UCC) are managed and analyzed to better understand the performance of these buildings. The various living laboratories provide a valuable reservoir of building performance data upon which to conduct many forms of analysis. An initial DW-schema was developed by IRUSE researchers for the Science Foundation Ireland (SFI) research project ITOBO (Information and Communication Technology for Sustainable and Optimised Building Operation – see Menzel et al, 2010).

Efficiency is essential when handling and managing large volumes of buildings' performance data. Current data management strategies over utilize a single table concept to store diverse categories of data. It is out of sync with peripheral applications such as support for BIM data, leading to inefficiencies in extracting a specific piece of information (Y. –S. Jeong et al. 2009). Furthermore, adding new devices to, and retrieving relevant data from DW requires a certain level of manual interaction within existing DW schema. The motivation of this research is to address these deficits and produce an unified database schema that can be used to share performance data in a structured way amongst different software applications.

In this paper, a new concept of a scalable schema for a data warehouse, focused on Building Performance Analysis, is being proposed. Its aim is formulated:

- to improve the compatibility with the well-established BIM applications,
- to improve the scalability of the DW to efficiently accommodate the exponentially increasing quantity of data and
- to provide flexible mechanisms for multi-dimensional data analysis on different levels of granularity.

The major elements of the newly proposed DW schema will be discussed based on the existing IFC standard (ISO/PAS 16739:2005). The paper will also discuss partitioning and decentralization strategies in the schema implementation. Finally, a DW cube is demonstrated to allows efficient acquisition of structured data.

2. EXISTING DATA WAREHOUSE

The existing DW schema consists two different database schemas, each of them dedicated to the management of building performance data compiled from a specific building.

The ERI_DW schema manages data from the ERI building, compiled from a commercial Building Management System (BMS), a Building Information Model (BIM), a Maintenance Management System (MMS) and a Wireless Sensor Network (WSN). The CEE_DW schema manages data from CEE-building which is compiled from a BMS and a BIM (More greater details explained in A. Ahmed, 2010).

The structure of these two schema is similar due to the reason they are all developed by the same research group. The number of database objects in each schema varies due to their differing purposes and level of data objects as can be seen from the summary Table 1. Commonly used objects belong to category 1 and 2. They are categorized as dimensional data and fact data. Attributes and semantic context of dimensional data is comparable to IFC objects. However, even in these cases neither the names of the schema objects nor the names of their attributes are compatible to IFC data model as defined in IOS16739.

Table 1: Overview of total numbers of schema objects

NO.	**CATEGORY**	**ERI_DW**	**CEE_DW**
1	Dimensional Data (compiled from BIM)	8 tables	9 tables
2	Fact Data (sensor &meter readings)	3 tables	3 tables
3	Tables for Building Control and Maintenance Management.	6 tables	6 tables
4	Views for DW-Performance Optimisation	4 views	
5	Views for Management	1 views	

The major deficit of the existing DW-meta model is its inflexibility for modeling interdependencies between information objects, since the principle of objectified relationships is not implemented. This leads to a limited compatibility with the IFC standard and the lack for extendibility and customization of the existing meta data model. Furthermore, the accessibility of building performance data is limited, since there exists no clearly defined mechanism to subscribe to services which would allow easy access to building performance data. Therefore, our re-engineering activities are guided to ensure semantic compatibility with a leading BIM-standard – IFC - and to ensure that the analytical functionality meets industry needs.

3. DATA MODELLING

In this chapter the development process for the re-engineered meta-model of the 2nd generation of our DW application is discussed. We highlight the major changes implemented during the re-engineering process.

3.1 Semantic Richness

The major elements of the newly proposed data warehouse schema follow the specifications given in the open IFC-standard (BuildingSmart, 2013). The meta model of IFC 4 offers a rich body of semantic information covering most BIM disciplines such as architecture, structural engineering, building services engineering, building automation, construction management, and facilities management (BimServer 2012). The initial purpose of using IFC 4 was to improve the efficiency of data exchange between CAFM software, design software and the newly introduced DW. Main features and elements of this schema are explained in Menzel et al. 2013a.

The ‘IfcSensor’ object and ‘IfcSensorType’ object are introduced by BuildingSmart in IFC 4, which are used to describe the occurrence of any single sensor within the IFC domain (BuildingSmart 2012). Both information objects are defined as a subclass of IfcDistributionControlElement or ifcDistributionControlElementType. By using the ‘standard inheritance tree structure’ many relationship definitions for BIM-objects are becoming part of the definition of the ‘ifcSensor’ information object which simplifies the implementation of such a model.

The sensors used within the ITOBO project are capable of measuring multiple data streams, e.g. temperature, humidity, or lux level etc. IfcSensor offers a property called IfcSensorTypeEnum, which allows the definition of 22 different sensor types and covers almost generic measurable mediums. For each sensor type, a dedicated property set is defined. The mechanism provides a very rich, and commonly agreed, set of attribute definitions to modelers.

Table 2: Schema Mapping for the ifcSensor Information Object

Existing DW	**New Meta Model (ifc 2.x4)**	
Attribute Name	*Attribute Name*	*Attribute Type (or required relationship)*
	Inherited from entity ifcRoot	
Node_Type_id	GlobalId	IfcGloballyUniqueId
Name	Name	OPTIONAL IfcLabel
Description	Description	OPTIONAL IfcText
	Inherited from entity IfcObject	
Status	ifcSensorPHistory: **Status**	Accessible through objectified relationship ifcRelDefinesByProperties: **RelatingPropertyDefinition**
Interval	ifcRegularTimeSeries: **Timestep**	Accessible through objectified relationship ifcRelDefinesByProperties: **RelatingPropertyDefinition**
	Inherited from entity IfcProduct	
Node_id	ifcAsset: **GlobalId** and **Name**	Accessible through objectified relationship ifcRelAssignsToGroup: **RelatedObjects**
	Inherited from entity IfcElement	
Tag	**Tag**	OPTIONAL IfcIdentifier
Location_id (FK)	ifcSpatialStructureElement: **GlobalId** and **Name**	Accessible through objectified relationship ifcRelContainedInSpatialStructure: **RelatingStructure**
	Inherited from entity IfcDistributionElement	
IPaddressOrDataLog	ifcDistributionPort: **Name**	Accessible through objectified relationship ifcRelconnectsPortToElement: **RelatingPort**
	Inherited from entity IfcDistributionControlElement	
Equipment_id (FK)	ifcDistributionFlowElement: **GlobalId** and **Name**	Accessible through objectified relationship ifcRelFlowControlElements: **RelatingFlowElement**
	entity IfcSensor	
Sensortype	PredefinedType	OPTIONAL IfcSensorTypeEnum

3.2 BIM-Compatibility

Table 2 (overleaf) illustrates the differences between the sensor definition of the existing DW schema and IFC sensor definition. All objects supported in the old database can be re-engineered to support an IFC naming conventions and object extensions in line with inherited attributes.

The existing DW is implemented as an entity-relationship model. Relations between database objects are modeled using the concepts of primary and foreign keys (e.g. LocationId and EquipmentID as foreign keys to tables containing rooms or building services components). However, the IFC meta model introduces the concept of objectified relationships to manage entities' relationships. Relationship objects are categorized as either assignments, associations, aggregations, declarations, or decompositions.

3.3 Extraction, transformation and loading

In the old database schema, BIM data are normally filled manually according to research reports by surveyors. Different reports can easily generate duplicated data leading to replicated records in the database. To avoid this from happening and also to improve the interoperability, transforming the existing database schema into IFC-compatible schema is inevitable. Table 2 gives a good view where an existing object can be beneficial by mapping to an IFC object. Entities in the new DW schema are fully compatible to IFC objects. The new DW schema not only unified the object names and attributes, but also greatly improved the efficiency of data exchange between BIM and DW as shown in Figure1. Multiple steps to transfer data can be eliminated.

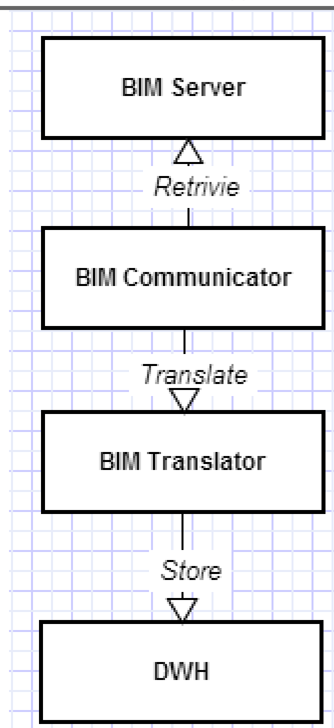
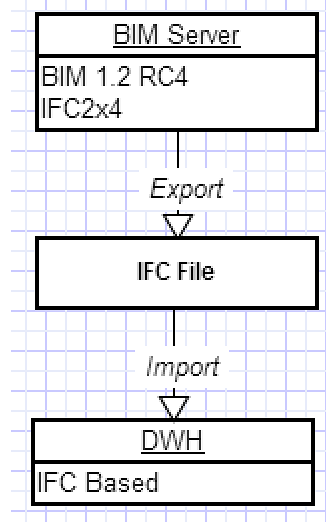
Existing Data Exchange Process between BIM and DW	**New Data Exchange Process between BIM and DW**
	

Figure 1: Data Exchange Process between BIM and DW

3.4 Multi-dimensional Data Analysis

Key performance indicators in facilities and energy management can be calculated against multiple comparison criteria. For example, a space-related KPI can determine max, min, or average temperature in all rooms of the first floor. A component-related KPI provides the supply temperature for the underfloor heating.

Space related KPI can be calculated if sensors are linked to space objects. In order to assign sensors to a corresponding room, a connection between entities needs to be defined. IFC introduces the IfcRelContainnedInSpatialStructure object to manage the relationship of sensor objects to a spatial objects such as building, space, and zone. The relationship is shown in Figure 2.

Other types of KPI and how these can be used to evaluate the performance of buildings' services systems is explained in greater detail in Menzel et al. 2013b.

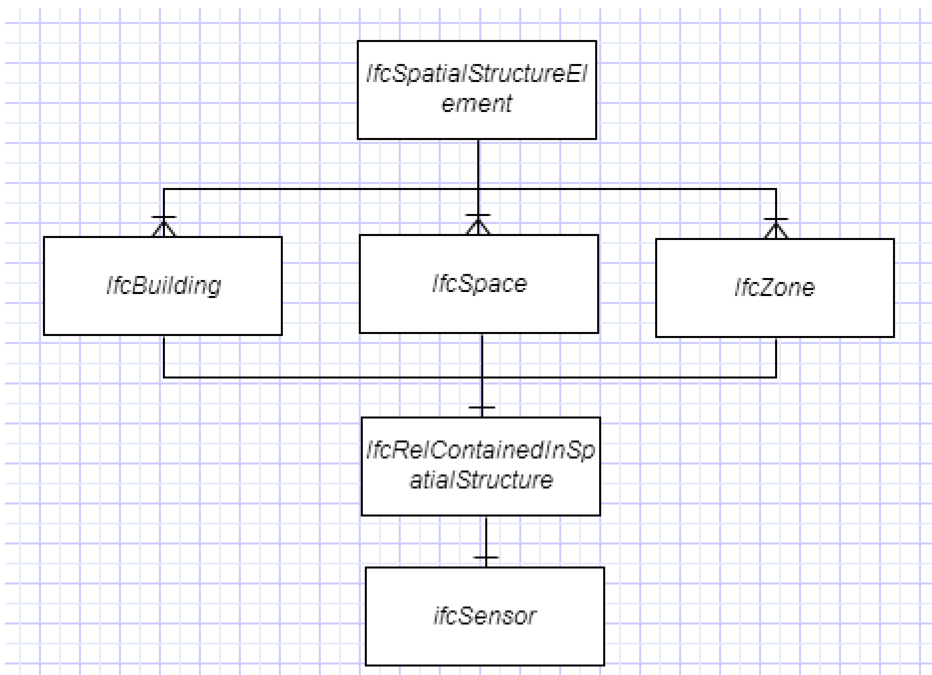


Figure 2: Class Hierarchy of IfcSensor and IfcSpatialStructureElement

Benchmarking is the process of comparing one's business processes and performance metrics to industry bests or best practices from other industries (Wikipedia, "Benchmark"). A benchmark criteria for building performance, exploiting space related KPI, is the comparison of temperatures in rooms with comparable features. For example, when all offices south facing, identify rooms with extreme deviations from the mean value, indicating that either system components are mal functioning or the users interfere with the systems, e.g. by forgetting to close a window in a room. Also, benchmarking of a specific room's performance is possible by comparing the relative humidity (RH), the temperature (T) against defined industry standards, norms and regulations. Finally, the performance of building services components becomes possible by comparing the thermal comfort (RH and T) against the status of valves and the windows indicating what amount of resources was required to achieve a certain thermal comfort.

3.5 DW Performance Optimization

Trends for KPI and benchmarks are calculated either over a long timeframe (trends) or over a broad variety of objects with comparable features (benchmarks). Therefore, complex retrieval patterns need to be executed. Partitioning, in combination with indexing and materialized views, is one way to optimize the execution.

Partitioning is a tool to optimize the performance of database and data warehouses from multiple perspectives, such as optimization for extraction, transformation and loading (ETL) process; or optimization for execution of search algorithms. In general, one can distinguish between vertical and horizontal partitioning.

Horizontal partitioning means that data will be split into individually accessible blocks and thus, the execution time for the calculation of "sub-totals" will decrease. Vertical partitioning means that the size of a tuple is reduced by dividing the columns into domain-specific "blocks". Again, the read time is decreased.

3.6 DW Cube

An OLAP (online analytical processing) cube is a technology that stores data in an optimized way to provide quick response to queries by dimension and measure (Wikipedia, "OLAP Cube"). A cube's structure and pre-aggregation allows it to provide very fast responses to queries that would have required reading, grouping and summarizing millions of rows of relational star-schema data. In order to implement a cube, several data warehouse technologies need to be in place as they represent the foundation of the cube.

The following listed requirements are necessary to implement the cube in an IFC-compatible database.

- Fact table needs to have a primary key (PK) set and for each dimension a foreign key (FK) which pointing to the dimension table is required.
- Each dimensional table needs a primary key.
- A time dimension has to exist and linked by PK/FK to fact table.

An example for the creation of a time dimensional cube is given in the following section. For simplicity, only the objects and columns with data of interest are explained below. Table 3 (below) presents the structure of the Fact Data Table Ifcits_SenorPHistory. This fact table has a primary key (reading ID) and a foreign key (ID) linking it to a second table called ifc_IrregularTimeSeriesValue.

Table 3: Fact table, for Irregular IFCits_SensorPHistory

Attribute Name	**Attribute Type**	**Comment**
READINGID	NUMBER	Unique reading ID
TIMESTAMP	TIMESTAMP(6)	Timestamp of measurement
VALUE	NUMBER	Recorded value
DIRECTION	VARCHAR2(20 BYTE)	Not needed
QUALITY	VARCHAR2(20 BYTE)	
STATUS	VARCHAR2(20 BYTE)	
ID	VARCHAR2(20 BYTE)	FK for Sensor/Meter ID

Table 4 (below) represents the ifcIrregularTimeSeriesValue-object which can be used to manage multiple partitions of sensor readings.

Table 4: Table ifcIrregularTimeSeriesValue managing partitions of the Fact Tables

Attribute Name	**Attribute Type**	**Comment**
ID (PK)	VARCHAR2(20 BYTE)	Sensor/Meter ID
DESCRIPTION	VARCHAR2(255 BYTE)	Description of the Sensor/Meter
STARTTIME	TIMESTAMP(6)	Time of first reading
ENDTIME	TIMESTAMP(6)	Time of last reading
TIMESERIESDATATYPE	VARCHAR2(50 BYTE)	
DATAORIGIN	VARCHAR2(50 BYTE)	measured
USAGEDEFINEDDATAORIGIN	VARCHAR2(20 BYTE)	
UNIT	VARCHAR2(50 BYTE)	
VALUE	VARCHAR2(255 BYTE)	Name of "Partition" (instance of table 3)

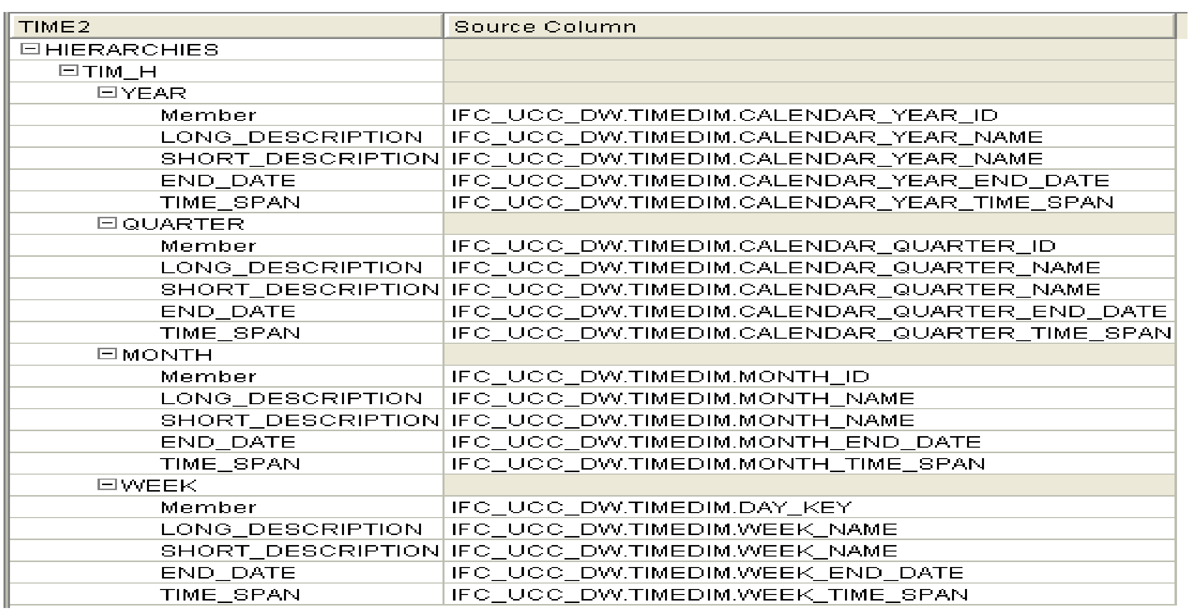
The Time Dimension is usually considered as a “standard dimension” which needs to be included in each DW-application. Therefore, an information object for this Dimension needs to be added. This information object is not part of the IFC-schema. It needs to be created in order to provide the required information to support “drill-down or roll-up operations” within the cube.

The DW developers need to define to up to which granularity the cube supports a drill down of data. For each period of time, the time table also needs to hold the start and the end. The Time Dimension table is created as presented in Table 5. Day_KEY is the primary key in the Time Dimension table.

Table 5: Dimensional Table (Time Dimension)

Column Name	**Column Data Type**	**Stored Information**
DAY_KEY (PK)	DATE	Unique Day String
CALENDAR_YEAR_ID	VARCHAR2(30 BYTE)	String describing the year
CALENDAR_YEAR_NAME	VARCHAR2(40 BYTE)	String describing the year
CALENDAR_YEAR_TIME_SPAN	NUMBER	Time span of the year in days
CALENDAR_YEAR_END_DATE	DATE	Last day of the year
CALENDAR_QUARTER_ID	VARCHAR2(30 BYTE)	String describing the quarter
CALENDAR_QUARTER_NAME	VARCHAR2(40 BYTE)	String describing the quarter
CALENDAR_QUARTER_TIME_SPAN	NUMBER	Time span of the quarter in days
CALENDAR_QUARTER_END_DATE	DATE	Last day of the quarter
MONTH_ID	VARCHAR2(30 BYTE)	String describing the month
MONTH_NAME	VARCHAR2(40 BYTE)	String describing the month
MONTH_TIME_SPAN	NUMBER	Time span of the month in days
MONTH_END_DATE	DATE	Last day of the month
WEEK_ID	VARCHAR2(30 BYTE)	String describing the week
WEEK_NAME	VARCHAR2(40 BYTE)	String describing the week
WEEK_END_DATE	DATE	Time span of the week in days
WEEK_TIME_SPAN	NUMBER	Last day of the week

A hierarchy has to be defined for each dimension (the time dimension contains Year, Quarter, Month or Week). Figure 3 presents how the time dimension is modeled in Oracle’s Analytical Workspace Manager (AWM).



TIME2	Source Column
⊟HIERARCHIES	
⊟TIM_H	
⊟YEAR	
Member	IFC_UCC_DW.TIMEDIM.CALENDAR_YEAR_ID
LONG_DESCRIPTION	IFC_UCC_DW.TIMEDIM.CALENDAR_YEAR_NAME
SHORT_DESCRIPTION	IFC_UCC_DW.TIMEDIM.CALENDAR_YEAR_NAME
END_DATE	IFC_UCC_DW.TIMEDIM.CALENDAR_YEAR_END_DATE
TIME_SPAN	IFC_UCC_DW.TIMEDIM.CALENDAR_YEAR_TIME_SPAN
⊟QUARTER	
Member	IFC_UCC_DW.TIMEDIM.CALENDAR_QUARTER_ID
LONG_DESCRIPTION	IFC_UCC_DW.TIMEDIM.CALENDAR_QUARTER_NAME
SHORT_DESCRIPTION	IFC_UCC_DW.TIMEDIM.CALENDAR_QUARTER_NAME
END_DATE	IFC_UCC_DW.TIMEDIM.CALENDAR_QUARTER_END_DATE
TIME_SPAN	IFC_UCC_DW.TIMEDIM.CALENDAR_QUARTER_TIME_SPAN
⊟MONTH	
Member	IFC_UCC_DW.TIMEDIM.MONTH_ID
LONG_DESCRIPTION	IFC_UCC_DW.TIMEDIM.MONTH_NAME
SHORT_DESCRIPTION	IFC_UCC_DW.TIMEDIM.MONTH_NAME
END_DATE	IFC_UCC_DW.TIMEDIM.MONTH_END_DATE
TIME_SPAN	IFC_UCC_DW.TIMEDIM.MONTH_TIME_SPAN
⊟WEEK	
Member	IFC_UCC_DW.TIMEDIM.DAY_KEY
LONG_DESCRIPTION	IFC_UCC_DW.TIMEDIM.WEEK_NAME
SHORT_DESCRIPTION	IFC_UCC_DW.TIMEDIM.WEEK_NAME
END_DATE	IFC_UCC_DW.TIMEDIM.WEEK_END_DATE
TIME_SPAN	IFC_UCC_DW.TIMEDIM.WEEK_TIME_SPAN

Figure 3: Define Cube Hierarchies in AWM

4. SIMPLE EXAMPLE FOR ANALYSIS

A DW-cube allows the user to quickly access results from complex retrieval patterns. Examples are the calculation of the total annual energy consumption from smart meter readings with a 15 minutes sampling rate or the determination or average temperatures for a given period of time or more complex KPI as discussed in Menzel et al 2013b. Alone the calculation of the annual energy consumption in the above case requires the reading and addition of 35,040 values. Reading all values will take between 8.76 s (flash with 250 10^{-6}s) and 5.8 min. (hard disc with 10 ms access time).

Figure 4 below shows an example of comparing average annual temperatures of 56 rooms of the ERI-building. The response time with DW and flash memory is 0.014 s compare to 8.176 minutes. This trivial example already explains the need for using DW-technology. Furthermore, one can see that the thermal comfort in two rooms can be quickly identified as "deviating from average". Further diagnostics can be triggered.

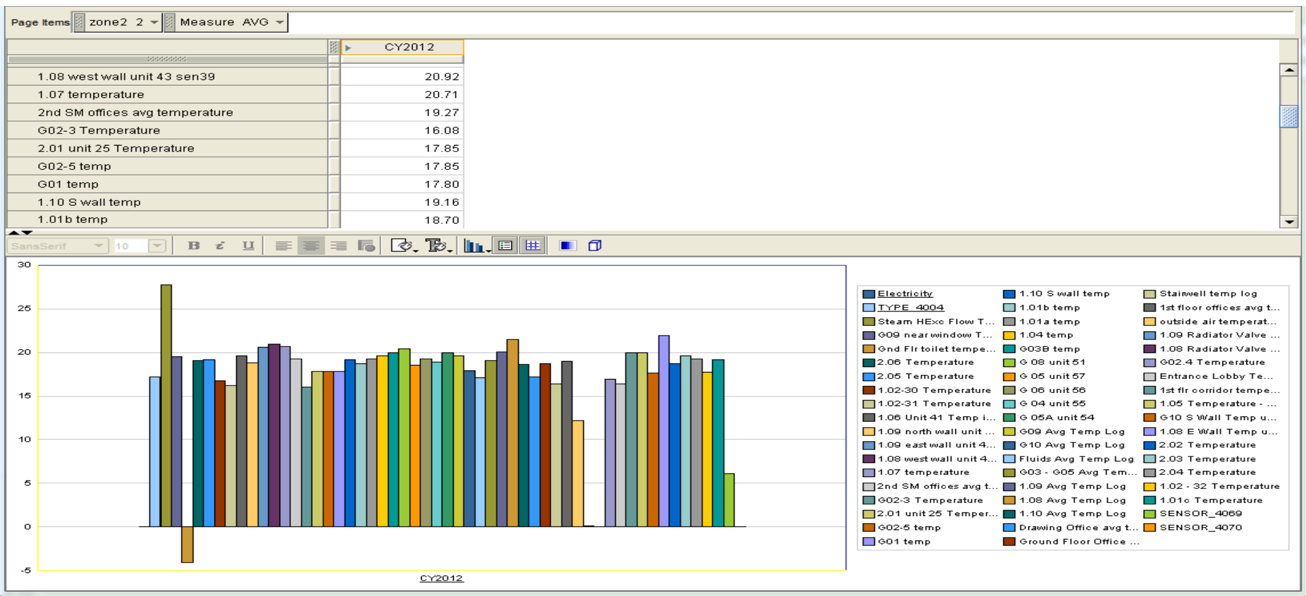


Figure 4: Cube Output(yearly averages)

4.1 Increasing Granularity

The output of the pre-calculate cube can be rearranged and displayed in various ways using an OnLine Analytical Processing (OLAP) tool. Those tools, such as Oracle's AWM allow quick navigation through the pre-calculated data cubes by changing the scale of one or multiple axis of the data cube.

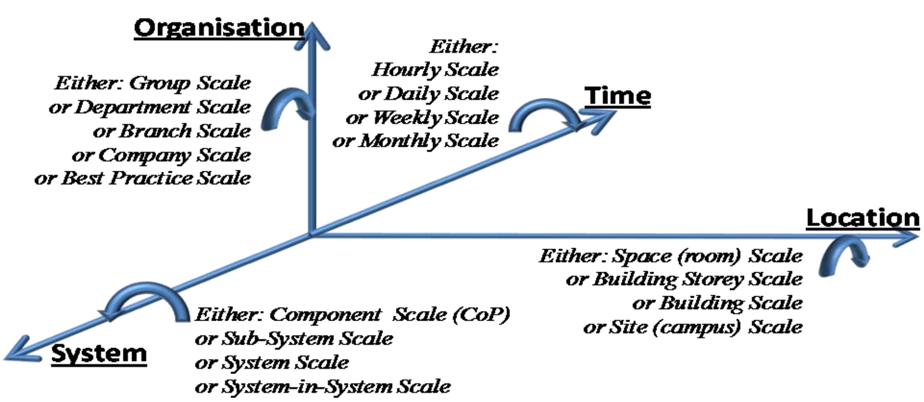


Figure 5: Cube with different granularity at each of the four dimensions

4.2 Example for Increased Granularity

By drilling down in the time dimension the scale of the time axis is now changed from yearly to quarterly. The output for quarterly average temperature data is presented in Figure 6. As one can see the two rooms with non-average temperature values can be clearly identified in Quarter 1 and Quarter 2. It appears, that temperatures for quarter 3 are "on average" for all rooms and the temperature in one room is "too high" again in one room in the 4th Quarter. A further breakdown on "Weekly", "Daily" or "Hourly" level might help to further diagnose the identified symptoms or anomalies.

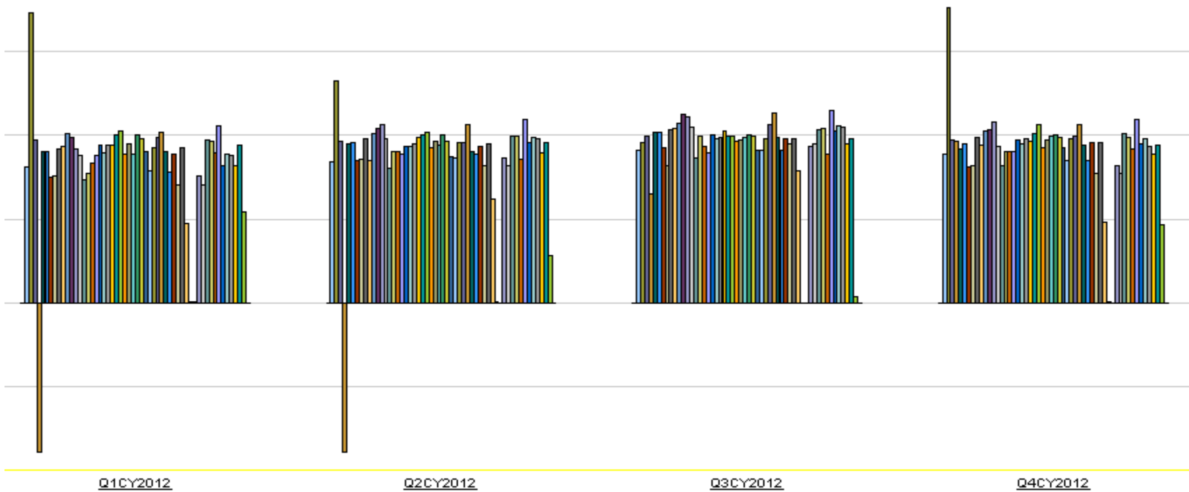


Figure 6: Cube Output (quarterly average temperatures for ERI-rooms)

5. CONCLUSION AND FURTUREWORK

This paper provides an overview of the major steps, which have to be considered for the re-engineering of an existing DW solution that aims to improve building performance analysis. It was demonstrated that multi-dimensional analysis of sensor readings can be improved by better exploiting the semantic richness of available, standard meta-models, such as IFC. Currently, a full re-design of an existing DW schema is executed by the authors, which will focus on developing a smart platform supporting advanced building performance analysis and control.

In a joint effort with representatives from different industry sectors the authors have compiled a set of approximately 15 generic, scalable KPI which can be used to evaluate the performance of buildings, building services systems or single components. The KPI are developed for both, the evaluation of comfort parameters and the resources consumed to provide certain comfort levels.

These KPI are described in one of the technical reports of the EU-FP7 project CAMPUS 21. We envisage to implement the algorithms for calculating these KPI in the second half of the year 2013. The availability of generic, scalable KPI is an essential input for the execution of benchmarks over large groups of buildings with comparable features.

An important pre-requisite for the efficient calculation of the above mentioned KPI is the availability of BIM-data for the categorization and classification (Dimensional Data) of monitoring data (Fact Data), since the "automatic", standardized import of these dimensional data speeds up the ETL and data pre-processing processes.

6. ACKNOWLEDGEMENT

This work has been funded by the European Commission (projects CAMPUS 21 and BaaS). We would like to thank our former colleagues Dr. Ammar Ahmed, Paul Stack and Brian Cahill who substantially contributed to the set-up of the initial DW-application compiling more than 20 million performance data sets from UCC's ERI-Building between 2008 and 2011.

REFERENCES

BimServer (2012). [Last accessed 29 May 2013]. Available from World Wide Web: http://bimserver.org/.

BuildingSmart (2013). IfcSensor definition [online].[Last accessed 29 May 2013]. Available from: http://www.buildingsmart-tech.org/ifc/IFC2x4/rc4/html/schema/ifcbuildingcontrolsdomain/lexical/ifcsensor.htm

K. Menzel, E. Freuder, K. Brown, G. Provan (2010). A Platform for the Optimisation of Building Operation. In: K. Menzel and R. Scherer (eds). *Proceedings of the 8th European Conference on Product and Process Modelling*. Cork, Ireland: CRC Press, pp.273 – 281.

Y. –S. Jeong, C. M. Eastman, R. Sacks, I. Kaner (2009). Benchmark tests for BIM data exchanges of precast concrete. *Automation in Construction,* Volume 18, Issue 4, July 2009, Pages 468-484.

A. Ahmed, J. Ploennigs, K. Menzel, B. Carhill (2010). Multi-dimensional Building Performance Data Management for continuous Commissioning. *Advanced Engineering Informatics,* Volume 24, Issue 4, November 2010, Pages 466-475.

K. Menzel, M. Weise, T. Liebich, C. Valmaseda (2013a): Capabilities of IFC 4 for Advanced Building Performance Management: in: *Proceedings of the 2nd Central European Symposium on Building Physics*: Vienna, 2013.

K. Menzel, A. Mahdavi, F. Katzemich (2013b): Impacts of Building Performance Monitoring on Integrated Energy Management in: Proceedings of the 2nd Central European Symposium on Building Physics: Vienna, 2013.

Wikipedia contributors. "Benchmarking." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 9 Jul. 2013. Web. 19 Jul. 2013.

Wikipedia contributors. "OLAP cube." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 13 Jul. 2013. Web. 19 Jul. 2013.