SEVENTH FRAMEWORK PROGRAMME OF THE EUROPEAN COMMUNITY
(EC GRANT AGREEMENT N° 288819)

INTELLIGENT SERVICES FOR ENERGY-EFFICIENT DESIGN AND LIFE CYCLE SIMULATION

ISES

Deliverable D4.1:

Technical specification of the overall framework and the principal energy profile and consumption patterns

Responsible Authors:

Gudni Gudnason, Costas Balaras

Co-Authors:

Elena Dascalaki, Peter Katranuschkov, Tuomas Laine, John Grunewald,
Byron Protopsaltis, Tobias Mansperger

Due date: 30.11.2012
Issue date: 21.12.2012
Nature: Public

Start date of project: **01.12.2011** Duration: **36 months**

Organisation name of lead contractor for this deliverable:

Innovation Center Iceland (NMI)

History

Version	Description	Lead Author	Date
0.1	First Draft	NMI	04.09.2012
0.2	Second Draft	NMI, NOA	18.11.2012
0.3	Third Draft	NOA, NMI	21.11.2012
0.4	Pre-final	NMI	30.11.2012
0.9	Pre-final consolidated	NMI, All partners	12.12.2012
1.0	**Checked and approved final version**	TUD_CIB	21.12.2012

Copyright

Citation

Gudnasson G., Balaras C., Dascalaki E., Katranuschkov P., Laine T., Grunewald J., Protopsaltis B., Mansperger T., (2012): ISES Deliverable D4.1: Technical specification of the overall framework and the principal energy profile and consumption patterns, © ISES Consortium, Brussels.

Acknowledgements

The work presented in this document has been conducted in the context of the seventh framework programme of the European community project ISES (n° 288819). ISES is a 36-month project that started in December 2011 and is funded by the European Commission as well as by the industrial partners. Their support is gratefully appreciated.

The partners in the project are TECHNISCHE UNIVERSITÄT DRESDEN (Germany, Coordinator), GRANLUND Oy (Finland), UNIVERZA V LJUBLJANI (Slovenia), SOFISTIK HELLAS A.E. (Greece), NYSKOPUNARMIDSTOD ISLANDS (Iceland), NATIONAL OBSERVATORY OF ATHENS (Greece), LEONHARDT ANDRÄ UND PARTNER BERATENDE INGENIEURE VBI GMBH (Germany) and TRIMO INZENIRING IN PROIZVODNJA MONTAZNIH OBJEKTOV, D.D. (Slovenia). This report was created through the joint effort of the above organisations.

Project of the SEVENTH FRAMEWORK PROGRAMME OF THE EUROPEAN COMMUNITY		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

TABLE OF CONTENTS

Executive Summary

This document reports on the results from the work carried out in WP 4 "Energy Profile and Consumption Patterns for Built Facilities and their Components" within task T4.1 "Framework and stochastic templates for product life-cycle".

The main **objectives of WP 4** are to (1) develop the necessary service framework for the search and retrieval of data held as distributed, semi- or non-harmonised IT resources so that to enable their integration into the ISES Virtual Energy Lab platform, and (2) structure and organise resource templates that can be arranged and used for analyses in different combination and contexts for given facility and component configurations.

The main **results of Deliverable D4.1** are (1) the specification of the overall external data framework including weather profiles, occupancy profiles and energy related product and material data, and (2) the specification of stochastic templates (eeTemplates) capturing both deterministic and stochastic external data sets pertinent in energy simulation.

The deliverable is planned in **two versions**. It is the agreement of the consortium that the start of the WP was initially scheduled too early in the overall project plan ahead of or in parallel with other related WPs with deliverables, on which this deliverable relies on for its completion. Secondly a request has been granted to extend the consortium where new partner(s) will support the development of the stochastic methods fundamental to this deliverable.

This first version of the deliverable provides the conceptual basis, specific requirements and specification of the overall data and services framework and the stochastic templates capturing principal weather, occupancy and energy load profiles as baseline for the prototypes to be developed within tasks T4.2 and T4.3. The performed work draws on the results from the detailed surveys on the state-of-the-art and gap analysis reported in WP 1 on energy related ICT resources for energy performance simulation, occupancy patterns and occupant behaviour in buildings and weather data, along with the work presently available from WP2 and WP3 to facilitate the framework integration into the overall ISES Virtual Energy Lab platform.

The second version of the deliverable will include detailed methodology and principles for the stochastic approach with refinements based on the final results from WP2 and WP3. It is planned for delivery in project month 20.

The deliverable is structured in four parts:

Part 1 captures the overall requirements for the framework and stochastic templates

Part 2 describes the specification of the data framework

Part 3 describes the specification, data model and ICT support for stochastic templates

Part 4 describes the integration with the ISES Energy Virtual Lab platform.

A Conclusions chapter and a References section wrap up the deliverable report.

The partners participating in the WP have been responsible of separate parts of this document, but have worked together under the coordination of the WP leader over the duration of the work task to harmonize and update the work, as well as considering information from partner inputs and through active discussions with all project partners during consortium meetings.

1. Basic Concepts and Requirements

1.1 Introduction

Building performance simulation software is now more commonly used by designers for analyses of energy performance of buildings and in architectural energy efficiency studies. With advances in BIM, CAD and MEP software, designers can accurately model buildings and create detailed information about the building envelope, building elements, and building systems during the design process in a way that may be directly used in energy performance simulation software via open standards such as IFC and gbXML. There are, however, other aspects to energy performance simulation that engineers and energy consultants need to consider that are not typically available from the design documents, but are critical in performing realistic energy simulations and design optimization. These are aspects that relate to the physical environment in which the building is located and how it influences the energy consumption as well as the actual functional use of the building and individual spaces. Similarly, how building occupant dynamic and unpredictable behaviour and interaction with the building systems affect the overall energy efficiency of the building. This type of information is typically not very detailed and transparently available to designers and engineers for energy simulations. The input information is represented using different data schemas that have been developed independently over time to meet specific input requirements of simulation tools. As a consequence, input data needs to be manually manipulated or reworked, which can have significant impact on the time requirements and cost of performing the simulations as well as the accuracy of the simulation outcome. These parameters are very voluminous. They are specific and technical and need to be interpreted, evaluated and analysed by simulation experts before being used for simulations.

Simulation input parameters can be broadly categorized into three main categories (Hopfe 2009; Bourdeau et al. 2011). Some researchers also prefer to include building technical and control systems as a separate category. This is well justifiable based on the variety of such systems for energy generation, distribution and storage and the sophisticated control systems that are now being installed in new buildings. Such systems are becoming ever more technically complex and more expensive. As the level of complexity increases, more supervision, control, service and maintenance is required. Risks of failure or inadequate performance and uncertainty associated with their design and operation also become higher.

The currently recognised three main input **parameter categories** are**:**

Environmental and contextual variables: These variables, some of which can be beyond our control, have great influence on the building loads. For example, climate variables or site micro-climate, geographic location and the building surroundings, like woodlands, hills, building orientation are influencing daylight-shadowing and primary energy benefits.

Physical construction: This type of information is represented by the building envelope and electro-mechanical design documentation. It includes:

- Building geometry and planning of zones and spaces and their internal relation
- Building elements, products and materials, construction layers and material properties
- Heating, cooling, ventilation, air conditioning and control systems.

Functional properties of buildings and spaces and operational characteristics: These parameters relate to the actual operation of the facility and function of individual zones and spaces in the building e.g. occupant density and distribution, occupant behavioural aspects and social-cultural characteristics as they relate to the operation of the building and building zones. Examples include temperature set points, operation of blinds, lights, ventilation, infiltration, use of equipment etc.

The categories "Environmental and contextual variables" and "Functional properties of buildings and spaces and operational characteristics" are rarely modelled in the design tools and therefore not available as part of the design documents. The same is true for product and material energy related properties. As such, these parameters are often referred to as being **external data.** These categories on the other hand have significant impact on energy performance and energy consumption of buildings as illustrated in figure 1.

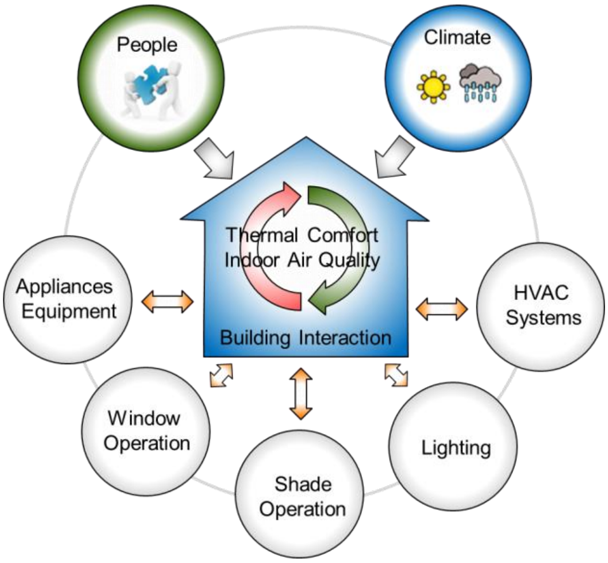


Figure 1 Climate conditions, microclimate, occupant activity and behaviour have a significant influence on building energy performance and total energy consumption

In ISES each of the parameter categories will be addressed within the scope of WP4, namely:

- Weather profiles
- Building products and material and their energy related properties
- Occupancy related profiles

Studies have shown that frequently, the calculated energy performance using simulation tools does not correlate with results from field studies using actual measured performance data. There is a number of plausible causes that can account for this deviation including building modelling details or level of detail (LoD), approximation in calculation methods, parameter estimation, and availability of applicable and appropriate data at the time of the simulation. We can, however, draw a general conclusion with reasonable likelihood that this can be attributed to the growing complexity and significant boost in capabilities of simulation tools and consequently in more complex configuration files. For end-users, especially non-experts, this presents an escalating challenge and effort of identifying the most appropriate data as well as acquiring the sufficient knowledge on the specifics of the simulation software and its configuration files. Furthermore, during the early design phases it is commonly the case that not all facts about the building are fixed, known or decided upon. This may sway end-users to take shortcuts, for example, using default values and pre-compiled typical data sets as provided with the software, which may or may not be appropriate for the context of the simulation (Hand et al. 2008), and use reference data, do rough estimations, apply best educated guesses or use subjective judgement in parameter value selection. On the other hand, many of the default parameter values have been empirically verified and libraries of typical data based on extensive data have been collected from different buildings types. However, these values and data sets are representative, statistical averages and dependent on discreteness, non-linearity, uncertainty and variability (Hopfe 2009). They may contribute to significant errors in simulation outcome when used without proper knowledge and evaluation.

For example, the specific weather data parameters and even the desirable time step depends on the simulation or calculation tool to be used and the methodology that these tools are based on, and of course availability of suitable local (micro) weather and climate data. Sparse locations with limited available weather data require interpolating between different locations in order to obtain some representative weather data. However, even for locations with available weather data, one has to take into account that these data are usually measured at airports, away from the urban environment that usually has different prevailing conditions. At present, there is little harmonisation and standardisation available for the industry practice. Using different sets of input weather data for the simulations may influence the estimated energy consumption by as much as +7.0%/-11.0% from long-term average weather patterns and the annual energy consumption by 5%, based on simulation results and analysis for an office building (Drury 1997). Annual variation in weather mostly affects energy consumption in heating-dominated locations and has lesser impact on energy consumption in cooling-dominated locations. The impact is more variable at locations where heating and cooling loads are more balanced. Accordingly, to capture more than the average conditions and provide simulation results that identify some of the uncertainty and variability inherent in weather it is recommended to use Typical Meteorological Years (TMY) that are based on a monthly composite weighting of solar radiation, dry bulb temperature, dew point temperature and wind velocity as compared to the long term distribution of those values. Months that are closest to the long term distribution are selected and thus each resulting TMY data file contains months from different years. Alternatively, it is possible to create a typical weather file that has three years: typical (average), cold/cloudy, and hot/sunny. This is to emphasise that different statistical calculation methods used for producing typical weather data files may produce different results keeping in mind that they are in fact statistically produced averages of variable number of years. It must be noted however, that there are regular version updates in weather profiles, improving their accuracy using more advanced numerical and statistical methods.

Among the research challenges and needs is the issue of how different building designs or existing buildings may perform in the future under different weather conditions, especially in urban environments. Given the ongoing debate about global warming and possible future climate changes, there are ongoing efforts to develop future typical meteorological years (FTMY). In the UK, climate projections are used to develop such data for future years up to the end of the 21st century and for various predictions of climate change using mathematical transformations of observed weather (morphing) or the use of a synthetic weather generator (Eames et al. 2012).

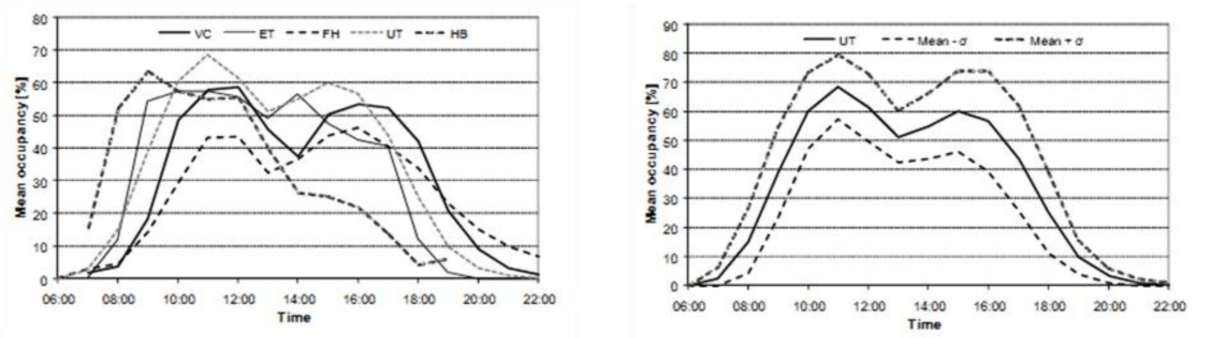
As noted earlier, occupant presence and behaviour in a building has significant impact on energy performance including:

- occupant density at any given time and location in the building which increases internal heat gains
- occupant interaction with the building systems to set thermal comfort and indoor air quality through operation of lights, shades, desirable indoor temperature and ventilation rates
- occupant activity during their presence in the building, which impacts the use of lighting and equipment resulting in increased energy loads, and along with physical activity, increased internal heat gains.

Accordingly, occupancy has a direct impact on the building's energy consumption since it determines the number of operating hours and the use of systems (i.e. heating, cooling, mechanical ventilation, domestic hot water, lighting and other equipment or appliances) during the day and week. The density of people and their activity influences internal heat gains, in terms of number of people present in a space or the building and heat gains from use of artificial lighting, equipment and appliances. In addition, occupancy also unavoidably controls the energy demand stemming from use of equipment (plug loads), lighting, domestic hot-water and heating, cooling, ventilation for maintaining a comfortable indoor environment and air quality.

Different end use buildings have distinctly different operating hours. Not all employees show up at the same time in the morning or leave at the same time in the end of the day, they take lunches at different hours, attend meetings and have engagements out of the office, take holidays at different times and so on. This contributes to variability in energy loads. Similarly people of different size and weight with different clothing contribute differently to internal heat gains, and people have different preferences for personal comfort e.g. optimal room temperatures and ventilation requirements. This preference depends on many factors, e.g. social status and income, lifestyle, cultural background and individual preferences. Many of the processes that individuals perform within buildings are highly unpredictable and fuzzy in terms of how occupants perceive indoor conditions like comfort and what actions they take to control and influence the indoor environment.

As illustrated in the following figures, similar office buildings have variable occupancy profiles and occupancy for the same office building shows high variance in occupancy throughout its operating hours.



(Mahdavi A., 2009) Mean occupancy level for a reference day for 5 office buildings

(Mahdavi A., 2009) Mean occupancy level and standard deviation for a reference day in office building UT

Figure 2. Occupancy profiles [as taken from (Mahdavi 2009)]

Mahdavi (2009) and other researchers (Page et al. 2007; Widen et al. 2009; Davis & Nutter 2010) agree that using deterministic modelling techniques greatly influences the accuracy of the simulation outcome. The inherent uncertainty in the simulation parameter space underlines the necessity to introduce stochastic and probabilistic methods in building energy simulation. However, at present, stochastic approaches in building energy simulation are still rarely used and mainly in academic and research studies. No main stream approaches have emerged as yet.

This deliverable report presents a general purpose framework for various data types, often referred to as external data, commonly used in simulation software that are in many cases only available to energy and energy simulation experts. The objective is to make this type of data more accessible to engineers and energy consultants in a more transparent and integrated way, in use with ISES simulation software tools.

Benefits of using a generic framework will encourage and support:

- Uniform organization of external data and definition of terms
- Productivity by separation of work based on expertise and reducing unnecessary duplication of effort.

The proposed framework and software suggested in this document, further defines stochastic templates to model, using probabilistic techniques, the inherent uncertainty and variability in external data e.g. weather patterns, user activity patterns and energy-relevant material properties.

1.2 General Requirements

Figure 3 below depicts a high level conceptual representation of the overall simulation process and integration of external data in to the simulation model. The Simulation Model Configurator prepares the simulation software input files. First, it filters and augments a building model (BIM) for geometric, space and building element information. Next it extends the BIM stored data, by various additional energy related information hosted in different external networked data sources. At last, it transforms, configures and formats the various data models into properly configured simulation input files as required by the simulation software.

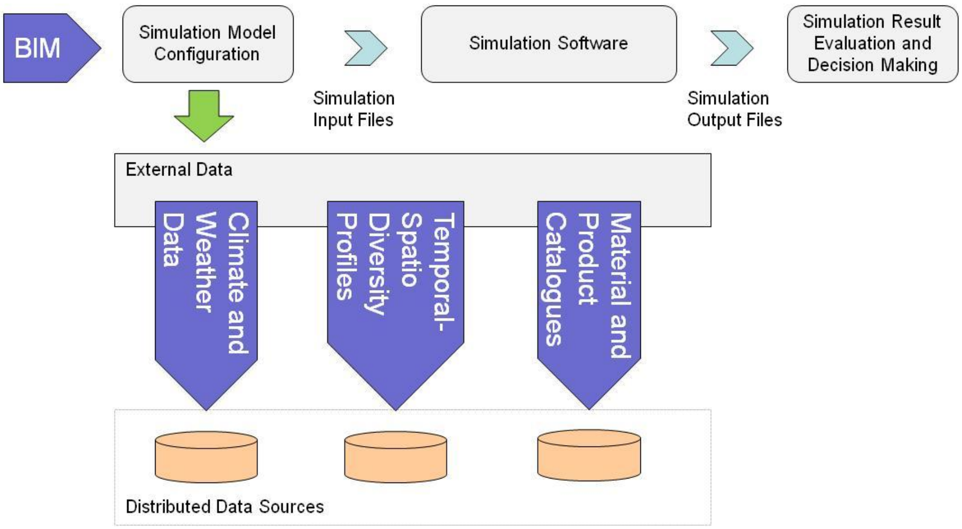


Figure 3. Information flow of external data.

The BIM data only contains part of the necessary information to perform energy performance simulations. A framework specification, as provided by the HESMOS energy enhanced Building Information Model, eeBIM (Liebich et al., 2011) is a coherent approach to enhance the BIM data with energy information from heterogeneous data sources. The eeBIM Framework facilitates the integration of diverse multi-model schemas in a consistent modelling framework.

At the center of the eeBIM is the BIM/IFC schema to which all other information data sources are linked. This approach is outlined in principle by the following scenario:

- BIM -> energy enhanced BIM (eeBIM): The CAD-BIM exported IFC instance model is transformed to an energy specific model. Missing energy specific details in the BIM are added like e.g. enhancements of space boundaries.
- eeBIM -> Climate data: Local weather profiles are linked to the eeBIM via the IFC objects ifcSite, IfcBuilding

- eeBIM -> Occupancy data: Occupancy profiles and energy load profiles are linked to the eeBIM via the IFC objects ifcBuilding, ifcZone and ifcSpace
- eeBIM -> Energy related product and material data: Product and material properties are linked to the eeBIM via the IFC objects ifcBuildingElement and ifcMaterial

Matching the diverse external data sources to the technical and functional specifications of a building requires searching and accessing many different sources of information to include in the eeBIM. Parametric values need to be obtained from BIM objects with addition of end user provided criteria and boundary conditions to retrieve the appropriate level of information matching the building type and technical and functional requirements. To accomplish this, the ISES approach proposes the use of energy enrichment Templates (eeTemplates) that are specifically tailored to be representative for specific building types, space types and material types and/or specific schema types (weather data, occupancy data). In this view, an eeTemplate contains a mesh-up of external data elements related to a specific view specification and usage scenario.

Figure 4 shows a typical workflow for generating the eeBIM data using eeTemplates. Users construct queries to the eeTemplate store using the available data from the BIM and at the same time provide additional information pertaining to the building to fill in information gaps. Several matching templates may be found that users will need to filter through and select those that best matched the search criteria and finally add the selected templates to the eeBIM. This workflow may need to be repeated several times before all the required eeTemplates have been added to the eeBIM.

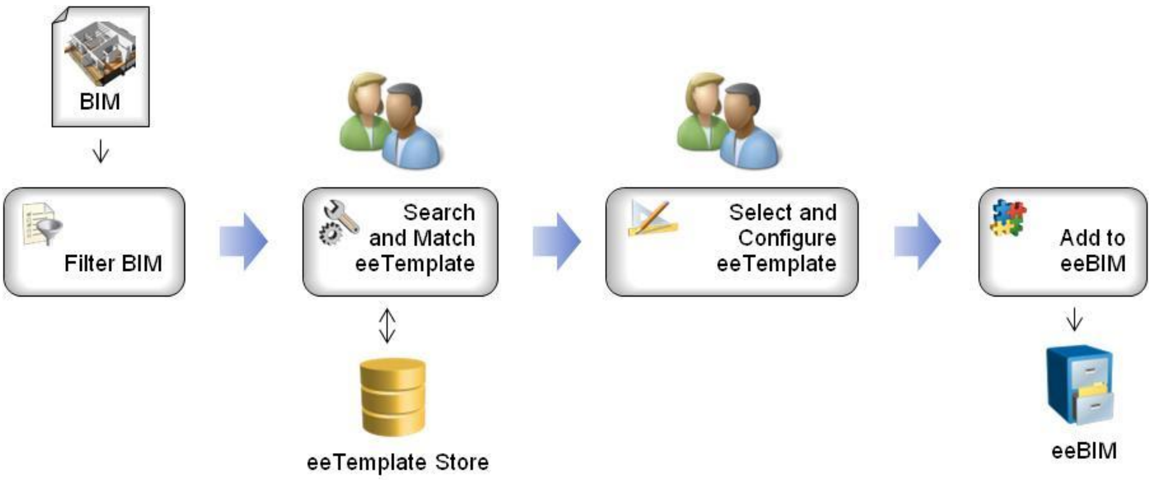


Figure 4. Generic workflow process to include eeTemplates in the eeBIM

The ISES Virtual Energy Laboratory (VEL) platform supports integration of different simulation software products that depend on a variety and different types of information and data for energy calculations and simulations. To manage and collate these diverse data models, ISES suggests a data framework that provides a common foundation and an approach to describe this heterogeneous data - currently existing in different data models and hosted by different technologies -, in a consistent and standardized manner, independent of particular simulation and calculation tools.

The ISES data framework shall act as a generic container for eeTemplates, climate data, and diversity profiles (occupancy, lighting, equipment etc.) as well as energy related building, product and material information.

1.3 Integration Requirements

The ISES VEL delivers a cloud-based platform for integrating different energy analyses and simulation software applications. Figure 5 shows a conceptual view of the principal components of the VEL platform (Baumgärtel et. al. 2012). The architecture shows the end-user applications, cloud-based energy solvers, the core modules and services and finally the cloud-based data repository inscribed with the red rectangle labelled “Data Repository”. This data repository essentially contains the varied data sets descried in the ISES data framework e.g. climate data, occupancy and related energy load profiles, energy product and material catalogues containing energy properties of products and materials. The VEL CORE functionality is realized using web-services orchestrated by SOA principles. In the architecture the VEL CORE is the middleware that handles the integration of the data repository and the navigation in constructing project-based eeBIMs. The nD Navigator facilitates the access and retrieval of data for inclusion in the eeBIM from the data repository, linking it in the eeBIM and enabling the processing of stochastic methods embedded within eeTemplates.

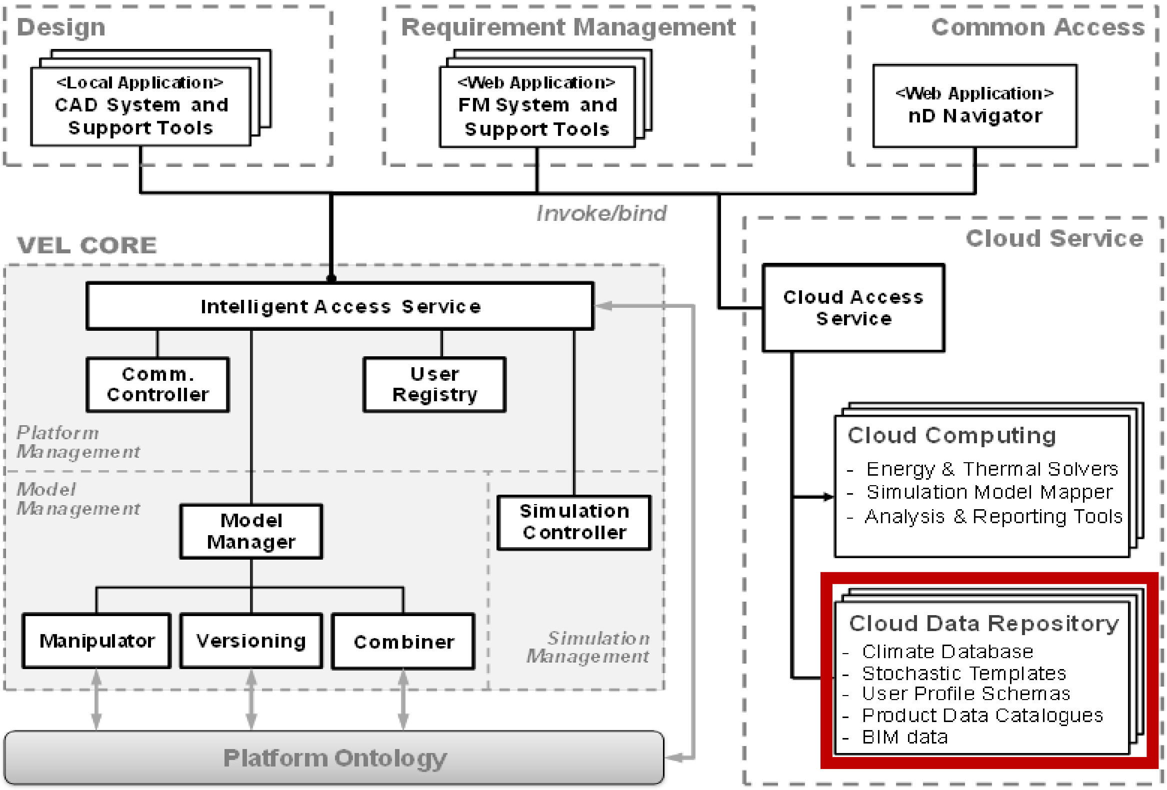


Figure 5. ISES VEL overall software architecture (Baumgärtel et. al. 2012)

To facilitate this, the following ICT requirements are envisioned:

- Application Interface (API) to the cloud-based data repository (files, data bases and BIM server)
- Web service exposing the cloud-based API with extension methods to realize searching and re-trieval of data from the repository accessed by the VEL CORE
- Web Service to upload external resources hosted by databases, web services, data files
- Web Service to handle the stochastic and probabilistic processing of the Stochastic Templates
- Specialised software tools for maintaining the cloud-based data repository.

1.4 Stochastic Approach

The ISES platform shall integrate a total of 6 energy solvers as outlined in ISES Deliverable D1.1 (Kavcic et al. 2012). Each of these tools has its own specific requirements for input in terms of data types and formats. ISES Deliverable D2.2 (Baumgärtel et al. 2012) gives an overview of input parameter requirements for each energy solver.

All these energy solvers, use as input deterministic values collated in text-based input files, the simulation configuration files. From the end-user point of view, the energy solver itself (logic and model processing) is seen as a “BLACK BOX” and simulations are configured by users by assigning values to parameters in the configuration files. To provide stochastic capabilities, the stochastically sampled datasets need to be converted to a set of deterministic input files as illustrated In Figure 6. If the sampling procedure generates samples with 100 elements, then 100 separate sets of configuration input files will need to be generated and simulated.

Depending on the specific input requirements for each energy solver, there may be different uncertainties related to the input parameters (e.g. relevant extreme and variability of weather conditions, different occupancy patterns, technical data and specifications). Consequently, there can be different number of uncertain parameters involved for different simulation tools and also for each simulation case.

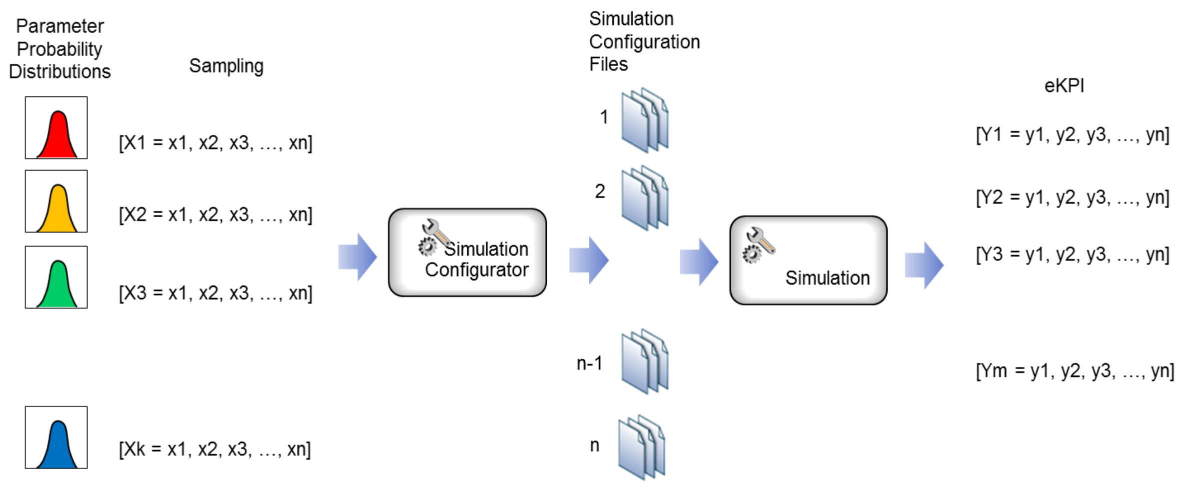


Figure 6. Stochastic simulation using deterministic simulation tools.

In a stochastic simulation process, input parameters (random variables) are expressed in terms of probability distributions D = [D_1, D_2, … , D_k] that characterize the uncertainty in input parameters X_1, X_2,..., X_k respectively. Distributions D are sampled accordingly to generate samples for variables X’ = [x'_1, x'_2, … ,x'_n] resulting in a matrix of [$x_n \bullet x_k$] elements (n = sample size and k = number of variables). The matrix is propagated through the simulation model n times to determine the statistics of the energy key performance indicators (eKPI) expressed by the mapping [X’I => Y(Xi)], i = 1,2 .. n.

In ISES, the eeTemplates are a key component in facilitating stochastic processing in the VEL simulation process. An eeTemplate is a collection of different data models data required to enhance BIM to eeBIM, that is, the relevant data mostly missing from design documents produced by CAD and BIM tools, but needed for configuring the simulation input files to the specification of the energy solvers. The eeTemplate is the ISES approach to abstract the configuration of the different simulation parameters and provide support for automation in this process. eeTemplates can deliver both deterministic data as well as stochastic data. This data can be presented in form of simple variables or as time series, variables that change over time, stored externally in distributed data sources or locally within the data repository.

Furthermore, an eeTemplate may contain embedded processing instructions that enable a special processor to dynamically generate stochastic data sets based on probabilistic methods as illustrated in Figure 7.

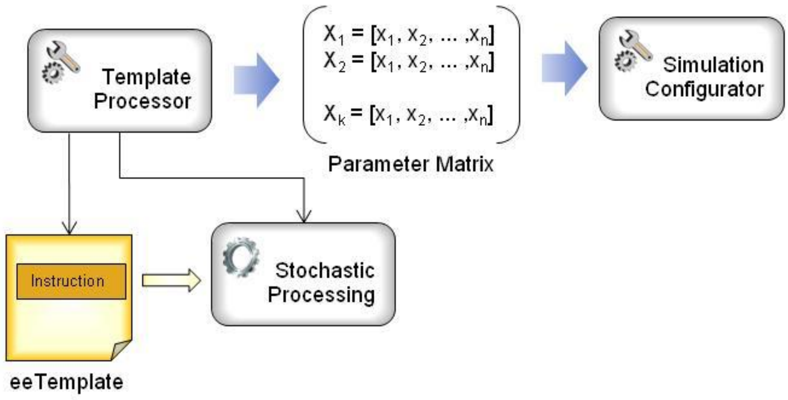


Figure 7. Stochastic processing using eeTemplates.

The overall stochastic approach is documented in ISES deliverable 2.1 (Zahedi et. al. 2012).

1.5 Framework Organisation

BIM concepts for linking external data

A formal semantics definition provides the necessary taxonomy for identification, organisation and navigation of the data framework content according to well defined building semantics. The IFC 2x3 schema defines concepts that can automatically associate external data to specific objects within the BIM/IFC instance model.

The following table summarizes the IFCx3 concepts and how they map to external data sources.

Table 1. IFC object matching to external data

Ifc Object	Description	Parameters	Data Source
IfcSite *IfcSpatialStructureElements*	The geometrical placement of the site, i.e. to the world coordinate system	Longitude Latitude Elevation	Weather data
IfcBuilding *Pset_BuildingUse* *Pset_BuildingUseAdjacent* *IfcSpatialStructureElements*	Representation of the spatial structure of the building	Building Use Type Building location on site and orientation	Templates by building use type repre-senting an aggregation of building floors, zone or space templates Weather data

Ifc Object	Description	Parameters	Data Source
ifcBuildingStorey	Represents the partial spatial structure of a building	Ground floor, intermediate floors, top floor	Template by building/floor type representing aggregation of zone or space templates
ifcZone *Pset_SpaceOccupancyRequirements* *Pset_SpaceThermalRequirements* *Pset_SpaceLightingRequirements*	A zone is an aggregation of spaces, partial spaces or other zones.	Zone activity type	Templates by zone Occupancy, equipment and lighting profiles
ifcSpace *Pset_SpaceOccupancyRequirements* *Pset_SpaceThermalRequirements* *Pset_SpaceLightingRequirements*	A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.	Space activity type	Templates by Space functional use or user activity Occupancy, equipment and lighting profiles
ifcBuildingElement ifcMaterial *IfcMaterialProperties sub-types* ifcDistributionElement	When used for physical constructs incorporated into the building	Element, Product and material type / material layer type Property identification	Product and material properties
IfcSpatialStructureElement IfcOccupant IfcRelAssignsToActor	The principal purpose is to determine the nature of occupancy of a property for a particular actor	Type of occupant in a building, story, zone or space	Occupant Profile
IfcSpaceProgram	Architectural program for spaces in the building detailing client requirements for the space before the building is designed		Space program templates (optional)

Which parameters in the IFC object model are populated is dependent on the design software export capabilities and the level of detail (LoD) in the BIM as prepared by the designers. Filtering the BIM model may be missing several of the necessary parameters to construct eeTemplate queries for inclusion in the eeBIM. It will therefore be necessary to include a GUI in the information retrieval process where end users can provide additional information not contained in the BIM.

More advanced and intelligent approach to be explored is to use the envisaged ontoBIM ontology in query processing in order to suggest parameter values or eeTemplates to the end user, or even to autonomously provide missing information on the basis of engineering rules.

Building, Zone and Space Classification

Energy loads and energy performance requirements are highly dependent on the functional use of the building and on individual spaces within the building. For energy simulation it may be necessary and beneficial to partition the building into different zones. For each one, the user designates zone types for different spaces of the building that share similar characteristics with regard to energy analyses. For most purposes, functional properties of zones determine by default the general occupant activity and density, related operating schedules for lighting, heating, cooling, ventilation and shading and energy loads stemming for activity based equipment and appliances.

There is no single consistent practise of how to partition a building into zones as different energy analyses systems may require different zone specifications. Many sources exist, on the other hand, for building and space classification in different level of detail found in energy calculation standards and formal building classification systems such as OmniClass™ (USA) and UniClass (UK).

In ISES, classifying external data sources and eeTemplates in accordance with building types and spaces will facilitate their logical organisation and help to establish association and mapping with different BIM objects that will support automated and transparent retrieval and inclusion in the eeBIM. For this purpose the ISES consortium has selected to use the open OmniClass™ construction classification system (www.omniclass.org).

OmniClass™ is a multi-faceted, ISO-12006-2 compliant classification system which is designed to classify all information related to design, construction and operation of the built environment over its entire life-cycle from conception to decommissioning and demolition.

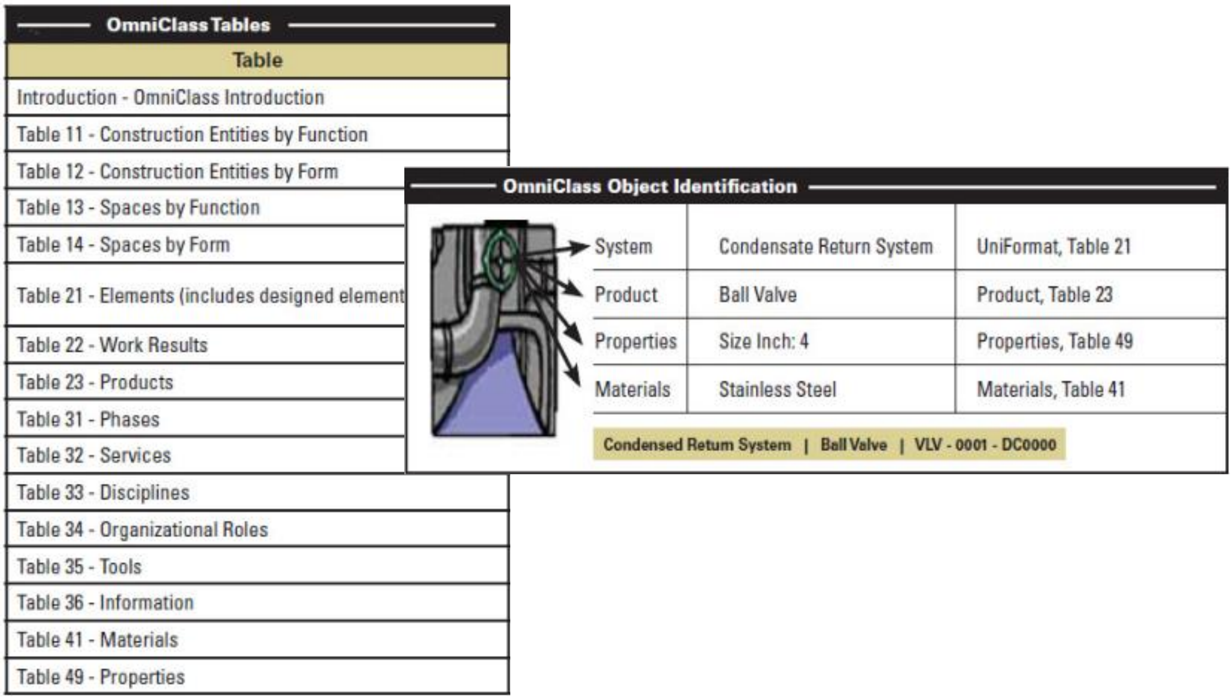


Figure 8. List of the OmniClass tables and example description in Table 23 "Products"

In ISES the classification done according to OmniClass™ Table 11 "Construction Entities by Function" and Table 13 "Spaces by Function" will be used, as detailed in section 2.1 further below. Later on, Table 23 "Products" will also be added in order to harmonise energy data and product component catalogue representations.

In cases where no detailed space planning is available, buildings may be considered as single zones. Similarly, when all the spaces are serviced by the same heating and cooling system and the set point temperatures for heating and cooling are similar between spaces, then it may be appropriate to use the assigned eeTemplates based on building type, e.g. school, office, hospital or hotel may be appropriate.

Partitioning the building into different zones not only adds accuracy and reliability to the simulation outcome, but it also enables higher granularity in the information on the energy performance of the building. However, there are limitations as to how detailed this partitioning to zones should be. For example:

a) The number of individual zones is limited in the simulation software, anywhere between 256 and 1200 depending on the specific simulation tool

b) Multi-zone calculation with thermal coupling between zones e.g. heat transfer by thermal transmission, ventilation or air infiltration between zones requires an order of magnitude more computational power and costs.

On the other hand, defining a small number of zones to simply minimize simulation costs may introduce considerable errors. Thus, using the eeTemplates for representing zone energy related information that consists of a large group of spaces may not necessarily be representative of all the individual spaces within the zone and energy dominant spaces may be neglected.

When partitioning a building into zones and deciding on zone types one should consider:

- Division by different occupancy or functional requirements based on space function and user activity e.g. single office spaces, hospital rooms, meeting rooms
- Division by heating, cooling and ventilation systems, thermal zones serviced by the same system or characterized by different energy loads, e.g. south vs north perimeter spaces of the building
- Division by internal heat sources, e.g. industrial kitchens, data centres and server rooms, freezers
- Division by zone construction, zone boundaries of different materials or building envelope partitioned by different construction materials, e.g. concrete, glass façade, prefabricated building elements
- Division as required in national codes or building regulations e.g. building energy certificates may stipulate certain requirements for zone types.

2. Data Framework

2.1 Overview and Remarks

This section describes the main concepts of the ISES data framework.

As outlined in the previous chapter, it is designed to facilitate the integration of energy related data that comes in variety of forms and from variety of data sources into the energy enhanced Building Information Model (eeBIM). The specification of the ISES data framework considers the following:

Data Sources: The data exists in various distributed sources, hosted by different technologies e.g. web based file repositories, data bases, web services

Data Models: The data is presented in specifications that have been developed using different data schemas

Data Formats: The data is delivered in various formats e.g. structured and un-structured data formats e.g. text based, XML, data tables

Furthermore, the data framework needs to support and provide consistency with defined exchange requirements and model transformations as defined in the HESMOS project (Bort et al. 2011; Liebich et al. 2012):

- BIM/IFC to energy enhanced BIM (eeBIM)
- eeBIM to Energy Simulation Model (eSIM)
- eSIM transformation to simulation input file formatting requirements.

The framework provides a common foundation and methods, by which, external data currently available in heterogeneous, varied data sources and formats can be modelled and described in a consistent and standardized manner for consolidation into the eeBIM. It further provides a logical structure for comprehensive representation of external networked data that is independent of particular simulation and calculation tools used in energy performance analyses. The framework is also open, flexible and scalable and implemented using current broadly available technology for straightforward adaptability.

From the framework point of view, data resources are both internal (local to the data repository) and external. For example eeTemplates are internal re-usable common resources created and maintained by ISES users and stored within the data repository. Networked data sources that exist out-side of the data repository and hosted by different technologies e.g. databases, web services and file repositories are considered to be external. Local data hosted in the data repository can be both project specific and common re-usable objects. Whilst the primary aim is to create re-useable data objects that can be used across projects, objects such as eeTemplates can be made specifically to suite a particular use case or a project as well.

The data framework (Figure 9) provides the overall high level view of heterogeneous data to support the consolidation of the eeBIM along with addition of meaningful semantics in form of meta-data, describing the various data entities, attributes, views and contexts.

Apparently, the framework is not intended to be a complete specification for every conceivable data type necessary for realistic energy simulation. However, it shall provide a stable implementation governed by the pilot requirements of the ISES eeBIM concept.

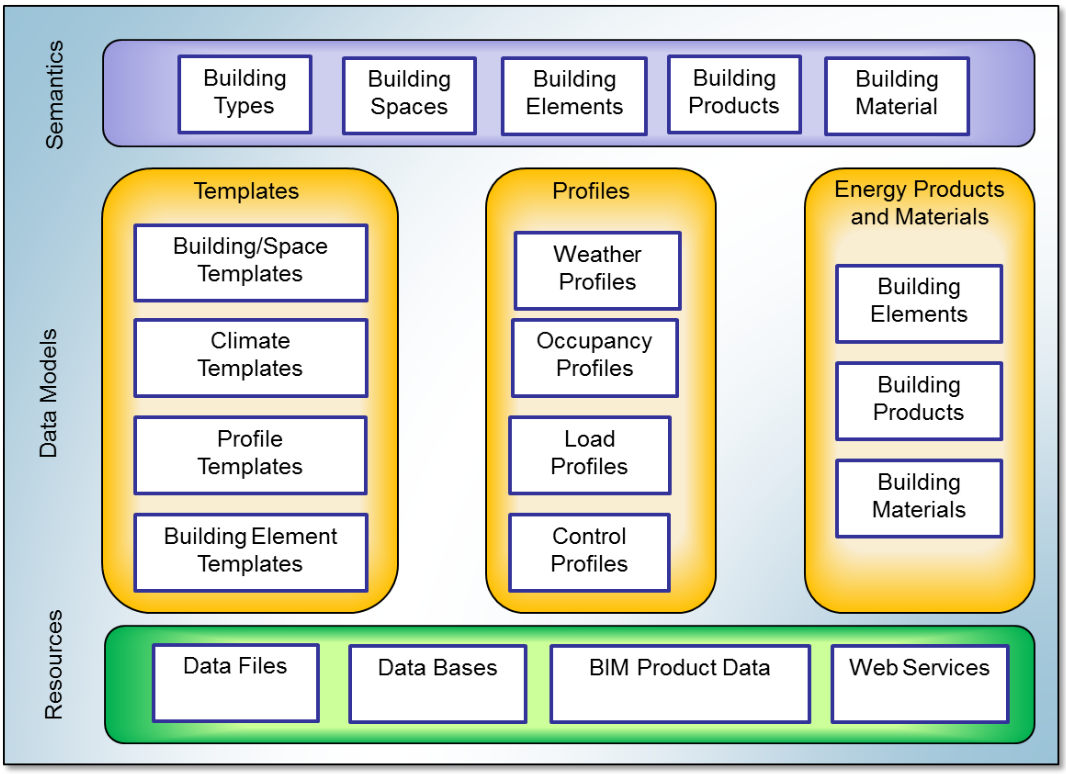


Figure 9. Conceptual representation of the data framework

The data framework is constructed in a three layer configuration, namely (1) semantic layer, (2) data model layer and (3) resource layer.

1. **Semantic layer**:

The semantic layer defines the common base for logical identification and organisation of data entities, attributes, views and contexts.

2. **Data model layer:**

The Data model layer forms the core of the data framework. It is divided into three main tiers. By documenting these tiers separately there will be a well defined interface to three types of essential data objects that can be used separately or together in consolidating eeBIMs.

The first tier "Templates" refers to the code that models on high level the external data arrangement into a collection of resources pertaining to a particular use specification.

The second tier "Profiles" refers to the constructs of energy related time dependent variables ordered in time series

The third tier "Energy products and materials" refers to the documentation of the details pertaining to the energy related physical constructs.

3. **Resource layer:**

The resource layer identifies and documents the actual distributed data sources, hosted by different technologies.

2.2 Framework Specification

2.2.1 Semantic Layer

The semantic layer defines the common syntax and context that enables a shared interpretation of information registered within the data framework and subsequent retrieval of this information for linking in the eeBIM (Figure 10). This layer further facilitates semantic enrichment that allows meaningful identification, organisation, and navigation of the content according to a well-defined building semantics. For this purpose, the base semantic terms and concepts have been derived from the IFC 2x3 schema and aligned with the OmniClass™ classification system to facilitate logical identification and organisation of templates, profiles, building elements, products and materials within the data framework. This will also facilitate the interlinking of the framework with the ISES platform ontology being developed in WP3.

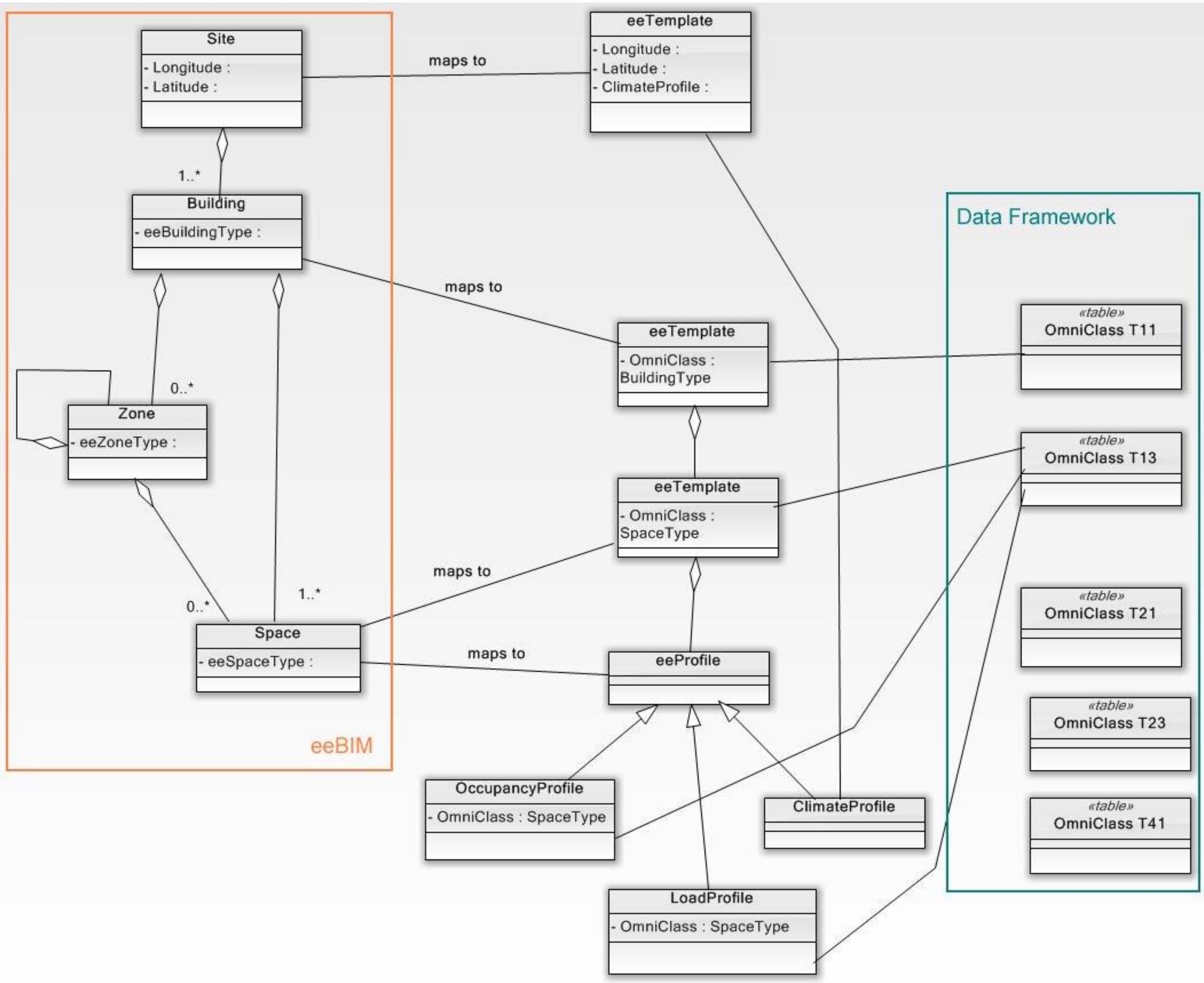


Figure 10. Overview of semantic mapping in the Data Framework

The basic semantic concepts further underline the hierarchical relationship of the content represented by the OmniClass™ classification tables in their analogy to the hierarchy found in the IFC model schemas.

Building (eeBuildingType)

Building spaces / zones

(Space boundaries)

Building elements

Building products

Building materials

The OmniClass™ classification system contains five classification tables that are of relevance for ISES and are implemented in the semantic layer of the data framework. These are:

- **Table 11 Construction entities by function.** The table identifies complete construction entities by its purpose or use. It is primarily defined by occupancy or user activity within the entity in 17 main classes of facilities. For example, single family residence, hotel, hospital, school, freezer storage facility, convention center.
- **Table 13 Spaces by function**. The table contains 964 hieratically organized space definitions, delineated by physical or abstract boundaries and characterized by their purpose or primary use such as occupancy of people or things or as medium of activity. For example, office space, kitchen, elevator shaft.
- **Table 21 Elements**. The table identifies major components, assemblies and constituent parts of construction entities without regard to materials or technical solution. for example exterior wall, roof, HVAC system.
- **Table 23 Products.** The table identifies the basic building blocks for construction e.g. single manufactured item, manufactured assemblies or manufactured operational stand-alone systems. For example door, window, curtain wall, wall covering.
- **Table 41 Materials.** The table identifies the basic substances, products and systems are made of without particular reference to physical form of the material but rather referring to a special material property. For example, metals, igneous rocks, cement, timber, glass, plastics.

Specific examples from Table 11 - Construction entities by function and Table 13 -Spaces by function are shown next.

Table 11 - Construction Entities by Function
11-11 Assembly Facilities
11-12 Learning facilities
11-13 Public Services Facilities
11-14 Cultural Facilities
11-15 Recreation Facilities
11-16 Residences
11-16 11 Single family residences
11-16 21 Multiple Family Residences
11-16 24 Hotels
11-17 Commercial Facilities
11-17 11 Offices
11-18 Production Facilities

Table 13 - Spaces by Function
13-55 Commerce Activity Spaces
13-55 11 Office Spaces
13-55 11 13 Dedicated Enclosed Workstation
An enclosed space used as one workstation occupied by a particular person on an on-going basis.
13-55 29 Commerce Activity Support Areas
13-55 29 21 Meeting Spaces
Space specifically designed for groups of people to interact on a occasional basis having appropriate seating and other amenities to support this activity.

In summary, OnmiClass^{TM} is a faceted classification system where each table provides a piece of the puzzle. By combining information from the various tables, it is possible to achieve very accurate and flexible identification of object. For example by combining Table 11 building types and Table 13 space types, it is possible to identify individual spaces within a particular building type, such as:

11-17 11 00 - 13-55 11 13 Private office in an office building

The Classifiable class is an abstract supertype of all classes which may be classified (organized into a hierarchical structure or partial order). A Classifiable object (Figure 11) may be classified by zero or more classification terms, by associating it with one or more classification terms.

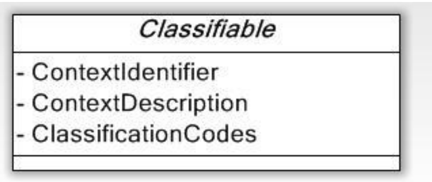


Figure 11. Classification in the Semantic Layer

Table 2. Attributes of the Classifiable Object

Property Name	Description
Context identifier	Identifies the type of object
Context description	Provides a description of the primary context that this object e.g. application area and restrictions of use
Assigns classification code buildings, spaces, building elements, building products, building materials based on the context identifier	assigned to several building types as defined by OmniClass^{TM} Table 11 assigned to several space types as defined by OmniClass^{TM} Table 13 assigned to several building element types as defined by OmniClass^{TM} Table 21 assigned to several building product types as defined by OmniClass^{TM} Table 23 assigned to several building material types as defined by OmniClass^{TM} Table 41

2.2.2 Data Model Layer

eeTemplates

eeTemplate is the principal method by which requests for external data will be communicated in the eeBIM framework. The template data model (Figure 12) provides great flexibility in defining the case, context and presentation of the multi-model data as needed in a project based eeBIM.

As already outlined, from the data framework point of view eeTemplates are documents stored as resources within the cloud-based data repository. In that respect the eeTemplate is a one-way, one-time data flow from the external data source to the eeBIM target. The data framework makes no assumptions about the template data elements other than those used to classify and otherwise describe the eeTemplate to be clearly identifiable within the framework. For this purpose meta-data is used that is consistent with the definitions give in semantic layer section.

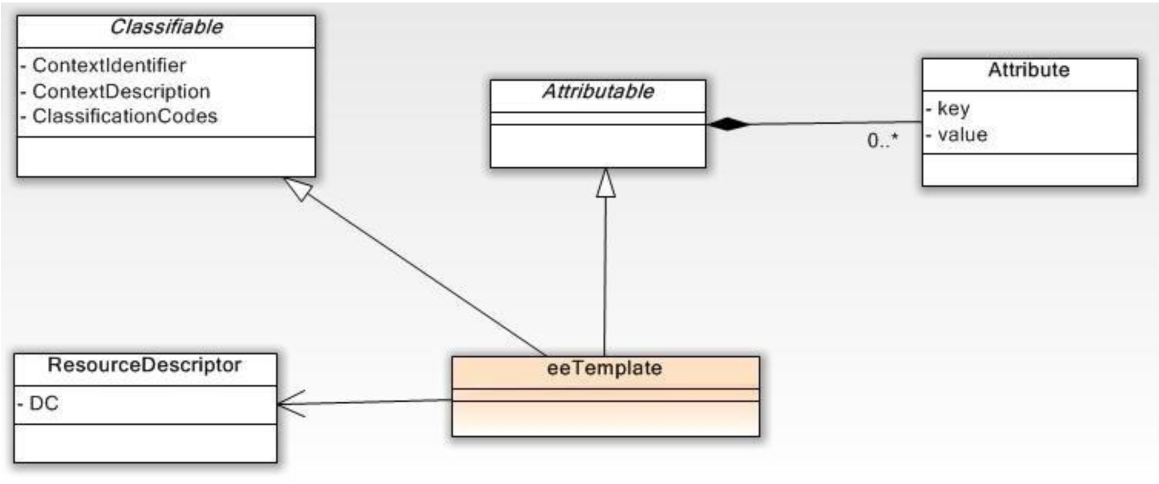


Figure 12. Overview of Template Data Model

The eeTemplates are described in detail in Chapter 3 of this document.

Profiles

The data framework identifies two classes of profiles: (1) deterministic profiles and (2) stochastic profiles. The profile classes share the same basic structure and the same characteristic or functional categories. Each profile category will vary in complexity and time steps, but are defined using the three basic time scales, consistent with most simulation software:

- Day Profile (schedule) defining a time series of 24 hours; the time step within each profile may vary depending on the level of detail (normally one hour or down to 10 min). Day profiles are normally distinguished between normal weekdays (Mon-Fri), weekends (Sat-Sun) and holidays
- Week Profile defines a typical seven day profile for a calendar week (Sunday-Saturday)
- Year Profile defines a typical full year profile as an ordered collection of weekly or daily profiles extending over the full calendar year and taking into account vacations and seasonal variations

Each profile may use a time series that is continuous with evenly spaced time intervals or it can be discrete where data is registered at discrete time intervals.

Profile categories

Weather profiles: A weather profile is a continuous time series with a fixed hourly time step. Several standard formats exist to catalogue weather data that have emerged over the years. Such weather profiles can be obtained for many European locations. The most common weather data formats used for energy calculation and simulation are listed below. Some simulation software vendors recommend using TMY2 or WYEC2 weather profiles. However, not all energy calculation and simulation software depend on the standard formats for weather data, but use their own proprietary formats. Such weather profiles can be constructed from available historic weather data or by filtering and transforming standard profiles, providing they contain all the necessary data.

Table 3. Weather data formats

Name	Description
Test Reference Year (TRY)	Early developed TRYs contain hourly values for an entire year including various weather parameters, e.g. dry bulb, wet bulb, and dew point temperatures, wind direction and speed, barometric pressure, relative humidity, cloud cover and type, and a place holder for solar radiation, but no measured solar data (e.g. calculated based on the cloud cover and cloud type information available for the TRY location). Initially, the method used to select the TRY data from an actual historic year of weather used a process whereby years in the period of record which had months with extremely high or low mean temperatures were progressively eliminated until only one year remained. This tended to result in a particularly mild year that, either by intention or default, excluded typical extreme conditions.
Typical Meteorological Year (TMY2)	Hourly values of solar radiation and meteorological elements for an entire year. Hourly data in this set consist of 12 months selected from long records to represent typical months to derive a TMY. The method used is similar to that used for the TRY but is based on individual months rather than entire years. The data include measured solar radiation (e.g. horizontal and direct normal or diffuse). The TMY months are selected based on a monthly composite weighting of the parameters (e.g. solar radiation, dry bulb temperature, wind velocity) as compared to the long term distribution of those values. Months that are closest to the long term distribution are selected. Each resulting TMY data file contains months from different years. Different procedures and methods are available for generating TMYs that are closer to the long-term average.
Weather Year for Energy Calculation (WYEC2)	Developed by ASHRAE, using the TMY format but including solar data (measured where available, otherwise calculated based on updated solar insolation models) and extended to include illumination data.
International Weather for Energy Calculations (IWEC)	Developed by ASHRAE, variations of typical weather data that are available for 227 locations outside the USA and Canada. The files are derived from up to 18 years of hourly weather data originally archived at the U. S. National Climatic Data Center. The weather data is supplemented by solar radiation estimated on an hourly basis from earth-sun geometry and hourly weather elements, particularly cloud amount information.

Name	Description
Actual historic weather data (AHW)	Several sources for actual historic weather data are available for use with ISES. For example, 30 year data sets are available for Athens, Greece and Reykjavik, Iceland. Other sources such as the European Climate Assessment and Dataset project (http://eca.knmi.nl/), WeatherBank (www.weatherbank.com) and Meteonorm (www.meteootest.ch) have historic weather data available for a number of European locations.
Non reference years (NRY)	Standard weather profiles are assembled in a way that weather parameters are specific statistics calculated using a large number of years to represent a reference year and to account for different variability in weather data. Several profiles may be constructed that better reflect special weather events, extreme weather conditions and changing weather patterns now observed e.g. high or low temperatures observed over long periods, excessive wind speeds etc.

Occupancy Profiles

Occupancy profiles estimate the density of occupants within given space boundaries.

Energy load profiles

Energy load profiles determine the estimated energy load due to plug loads based on activity and occupancy levels. Typical profiles include equipment use profiles, machinery use profiles, service load profiles and appliances use profiles.

Control profiles

Control profiles estimate the energy load that stems from various devices that the user has control over. These normally pertain to regulating the user comfort levels and indoor air quality including lighting profiles, HVAC profiles, window operation and blind operation profiles.

Apart from weather profiles, where fixed time intervals are used for registered measured data, occupancy profiles, energy load profiles and control profiles are essentially discrete. This implies that, events occur at discrete intervals e.g. occupants arrive and leave at discrete times during the day, occupants interact with devices and systems and turn on/off lights at discrete times, open or close blinds based on discrete events. However, although profiles are discrete by default, they are presented using continuous time series with hourly time steps or less (down to 10 min) in simulation software.

Stochastic profiles

All profiles are inherently time-dependent, variable and uncertain. Using stochastic methods to produce profiles has many advantages over deterministic static profiles used in today's simulation software. Despite this fact, no main stream approaches have emerged that enable such features. Stochastic processing in energy simulation is still an uncharted area although several methods have been reported in the research literature. As illustrated in Figure 13, occupants have great influence on the energy loads that stem from their activity and behaviour in buildings. Research has shown that strong correlation exists between occupancy presence and the other profile categories. Occupancy presence is the ruling profile that regulates more or less the other dependent profiles. To obtain credible stochastic profiles, the stochastic occupancy profile needs to be first established. Several methods have been suggested for this including Bayesian Networks and Markov Chain methods.

The general approach would entail: (a) to establish a stochastic occupancy profile and then (b) to automatically generate the dependent profiles based on a correlation function.

In this context a stochastic profile is a set of time series generated by a stochastic process.

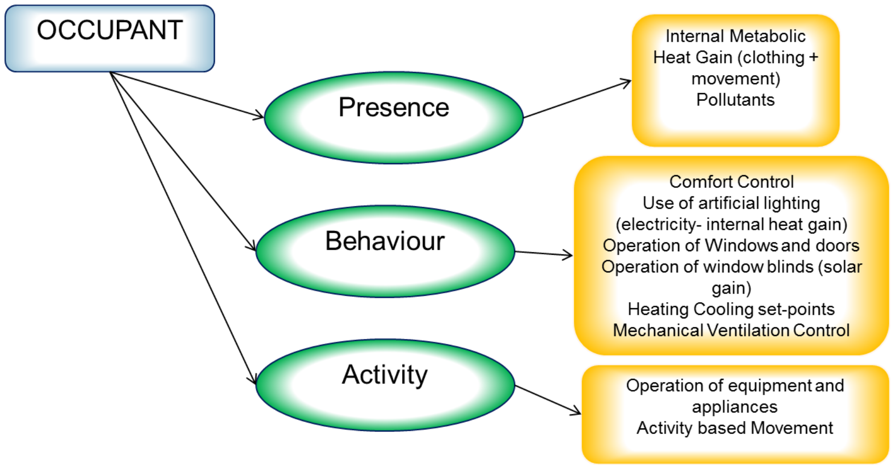


Figure 13. Occupants and their energetic relevance

The exact approach and detailed discussion on generating stochastic profiles will be included in the second version of this document, consolidating the finding from WPs 2-5.

The Data Model

From the data framework point of view, profiles are file-based resources mostly in text format. The file is usually structured in two parts. The first part contains meta-data describing the context and format of the profile, while the second part contains the actual data records.

The data framework provides a general data model to represent all types of profiles. However, there is a distinction made between weather profiles and other types of profiles. Figure 14 and figure 15 below show on high level the main classes in the profile data model and their interrelations. In these figures, only indicative properties are shown.

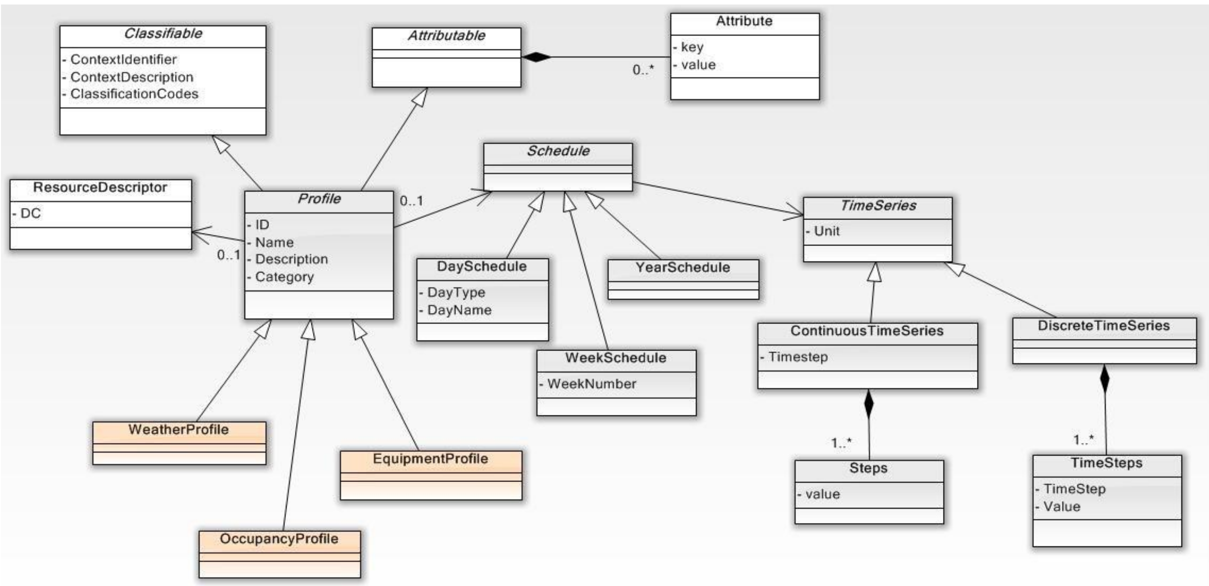


Figure 14. Overview of Profile Data Model

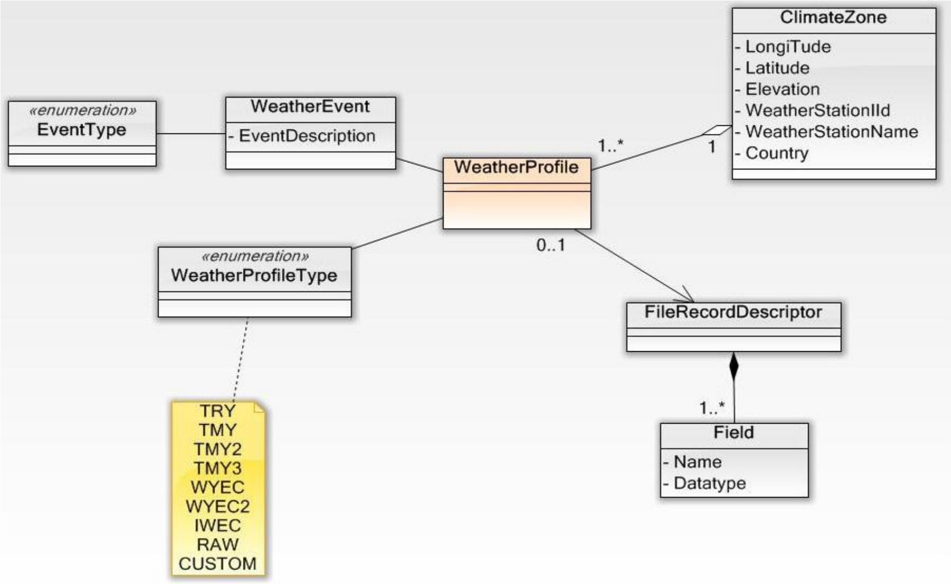


Figure 15. Overview of Weather Profile Data Model

To interface with historic and proprietary weather profiles that don't follow a recognized standard file format, a description of the record format will be necessary. The weather profile data model includes such a structure.

Energy products and properties

The ISES platform manages various types of model-based and energy related product and material properties that are essential in energy simulation, as described in the 5 main simulation scenarios presented in the process diagrams of ISES Deliverable 1.2 (Jung et al. 2012).

Host Models and Building Elements

Host models are virtual building models used in evaluating energy performance of building products during their development stages. The ISES data repository will support hosting a BIM based library of host models to be used for building product development. These can range from complete building models in which products can be inserted for performance evaluation as well as partial models such as Building story, Top floor of a building for roofing elements and roofing structures, or single zones for wall or window elements.

Building models of test products are similarly hosted within the data repository. ISES partner Trimo manufactures ready-made wall elements and facade systems, which are reprehensive of the type of building elements to be hosted in the data repository.

Building products and materials

For building energy simulation, detailed information on product and material energy related properties are needed which are normally not available in libraries as part of BIM and CAD design systems. In this regard, we distinguish between two types of objects, the so called design objects and manufactured objects. Design objects are generic virtual portrayals of product classes such as windows or wall elements, mainly with geometric and rendering properties e.g. colour and texture. Normally they include only representative properties, which are inadequate for energy calculations and simulations. Although most design software tools allow the addition of any number of properties to be attached to design objects. Currently there is a lack of standardized methods for property definitions including naming conversions, units, numerical precision that facilitates this in a transparent way. Accordingly, it is common practice by designers that only the most common product and material properties are included in the BIM.

Manufactured objects on the other hand, are the manufacturers' product and material descriptions, detailing *all* technical properties. However, most of these data are only available in proprietary un-structured format making them unsuitable for digital data exchange scenarios and as such they can only be manually processed for the required properties. Besides, many of the actual manufactured products and materials are not known until the construction process, i.e. in the "AS-BUILT" model.

In energy analyses, engineers and energy consultants will therefore need to depend on many different sources for product and material information. Whilst there are numbers of libraries and databases available for this type of information, they are still fragmented, encoded using diverse formats and in varying detail. Some examples are the International Glazing Database (http://windows.lbl.gov/materials/IGDB/) and the SRD 81 NIST Heat Transmission Properties of Insulating and Building Materials (http://srdata.nist.gov/insulation/).

Stochastic material properties

During the design stage not all values of product and material properties will be known to the designers. This is common, since the actual (manufactured) material to be used would not usually be decided upon at this stage. Instead, the potential material is represented by generic objects and properties.

ISES partner TUD-IBK has developed probability distributions for material properties from their historical library of measurement data on building materials by using data mining and clustering techniques. Applying this method they have compiled a collection of generic materials, each with its own property set. Properties are in this case, a probability distribution from which a sample can be computed using Monte Carlo methods. ISES has adopted this approach for performing any necessary uncertainty analyses of building materials in the simulation process (Zahedi et al. 2012).

Data model

From the perspective of the data framework energy products and materials can be internal sources or external sources e.g. libraries or databases (Figure 16).

Internal sources are file based. Building models and building element models will be stored according to the IFC 2x3 schema in STEP physical format – PART 21. As there is no uniform description currently available for building products and materials, they will be presented along with their properties by different proprietary text based formats. As with other file based resources they can be described e.g. using the Dublin Core meta-data standard (dubincore.org) [*)].

External sources for product and material information follow the same principles as for other external resources.

[*)] This is yet to be decided in cooperation of WPs 3 and 4.

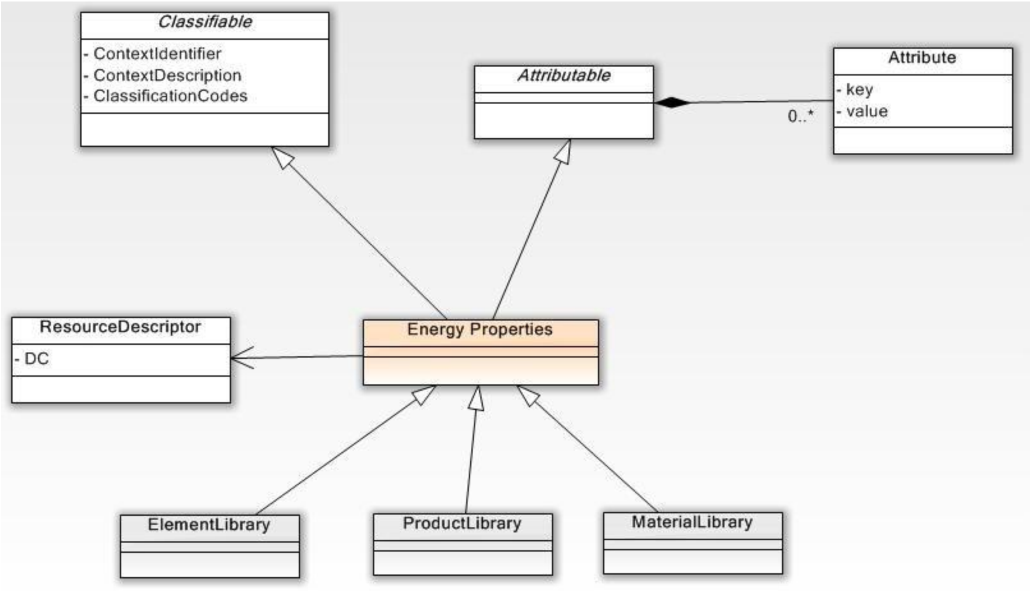


Figure 16. Overview of Energy Properties Data Model

2.2.3 Resource layer

The Resource layer presents a description of various resources both internal and external that provide the actual data to be linked with the eeBIM. These resources may be available in heterogeneous networked data sources, such as databases, web services, as files on a web site or in different data repository systems. To describe all these resource in comprehensive re-usable way requires a very complex framework. For example, a file that exists in a data repository or in an information management system is not accessible without the full knowledge of a service end point and the interface methods it exposes. Moreover, resources in the data framework exist only in a specific software context. It is therefore assumed that resources work within adaptors that understand the communication protocols and data format of the particular resource including the necessary authentication and authorisation. This may be expressed in a XML structure that only pertains to the retrieval of this particular resource. Resources are therefore, managed objects within eeTemplates in data framework.

Resource Descriptors are generic objects that provide a structure for parametric encoding which optionally can be passed as a parameter to an adaptor. Only descriptive information about the resource is registered with the Resource descriptor such as the owner of the data source, access privileges etc. Optionally, an interface definition or a web service may be included from which information about the resource can be learned e.g. WSDL or JSON file in the case of web services, a data specification etc. This may even be in a form of user manuals or guidance documentation. Sub-classes of Resource Descriptor identify the resource type and contain the detailed resource interface specification (e.g. end-point URI, user credentials, connection and parameter specifics) encoded in a XML structure. Resources together with adaptors are essentially what enables eeTemplates to be dynamic and re-usable. The eeTemplates use resources to dynamically retrieve or generate the required information.

Some resource descriptors and adaptors (providers) are provided as part of the data framework. An example of these data sources is shown below.

a) Data base of historic weather data, 30 year data sets for Athens, Greece and Reykjavik, Iceland

b) TUD-IBK web service for material data

c) TUD- IBK web service for weather data

d) Building elements by Trimo in IFC files

e) BIM Host models in IFC files.

Figure 17 shows an overview of the resource data model. Currently, the Dublin Core (DC) meta-data standard (dublincore.org) is used for the description of networked resources in this version of the data framework.

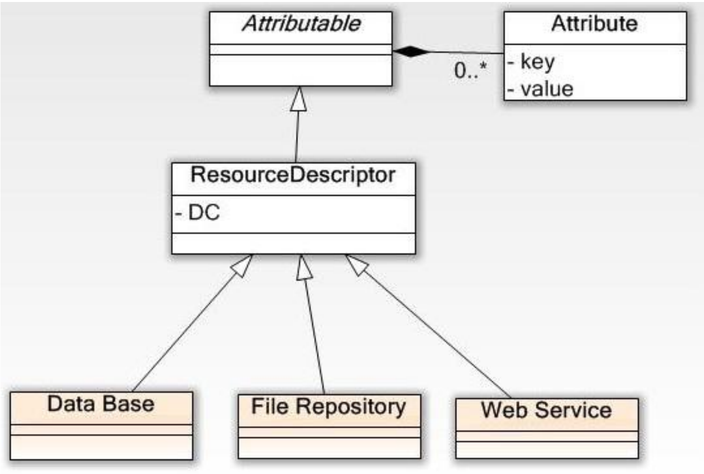


Figure 17. Overview of Resource Data Model

2.3 Notes on the Implementation

Currently there are certain restrictions in the developed data framework that need to be taken into account in its implementation.

1) The data framework is set up using a specific classification system, the OmniClass™. There are however many others national conventions for classifying this type of data and several standards provide taxonomies or dictionaries also suitable for this purpose. The argument for selecting OmniClass™ is based on its open access policy. It is comprehensive and includes facets to identify all object classes in ISES. There have also been talks to include OmniClass™ as part of the BuildingSMART IFD standard, although no confirmation is available.

   However, at present there is no provision in the data framework to include more than one classification system as an alternative solution, which may be of interest for later versions.

2) There are numerous meta-data standards currently available for describing digital networked resources that facilitate digital exchanges of information resources. For example,

   - Dublin Core - interoperable online metadata standard
   - RDF – the Resource Description Format of the W3C; it provides a general method for conceptual description or modelling of information that is implemented in web resources, using a variety of syntax formats

- METS - Metadata Encoding and Transmission Standard for management and exchange between repositories of digital objects
- Oasis CMIS - Content Management Interoperability Services; an interoperability specification for content management systems
- ISO19763 - Specifications developed for an interoperable web-service platform
- The WS Addressing and WS policy specifications of the W3C etc.

These standards provide means to describe and document networked resources in generic reusable ways in software-to-software settings. However, it is not the intention to implement these standards as part of the framework at this time. It is seen as more productive to describe the external resources in a more proprietary way that is more streamlined with ISES specific requirements. The Dublin Core meta-data standard is planned to be used and elements defined by other standards may be adopted when and if it is necessary.

3) The current framework specification does not have specific provisions or functionality to support proprietary versioning of external objects. Objects in the data framework are versioned through meta-data, yet underlying data sources may have their own versioning system. This is currently not in the scope of the data framework.

4) Finally, external resources can be volatile and unpredictable with regard to their availability. It is expected that the software components interfacing with external resources will provide robust error or exception handling in the event that broken links and connections are detected.

3. Stochastic Templates

3.1 Overview and Remarks

eeTemplate provide the principal method by which requests for external data are communicated in the eeBIM framework. They constitute a collection of different data model resources required for enrichment of a BIM to eeBIM. Accordingly, they include relevant data missing from design documents (e.g. CAD, BIM), but necessary for configuring the simulation input files to the precise specification of the corresponding software.

Configuration of simulation input files requires diverse parametric data. The eeTemplates are documents whose elements collectively define a simulation case and/or characteristic context. The template model enables abstraction of the many underlying data models that are binded to external data sources. Furthermore, the template model introduces some practical features, such as structuring hierarchical data templates, enabling successive granularity in defining the relevant data based on a custom logic processing.

In principle, the eeTemplates will deliver both deterministic data as well as stochastic data. This data can be presented in a form of simple variables or as time series, i.e. variables that change over time, so that they can be stored externally in distributed data sources or locally within the data repository.

The Data Framework addresses diverse types of data. However, the scope of the eeTemplates is primarily focused on enrichment of the eeBIM with these following data types

- eeBIM -> Weather profiles
- eeBIM - > Occupancy profiles and dependent load and control profiles
- eeBIM -> Building product and material properties

Three different template contents can be defined and used, which are elaborated next.

Constant content

The template consists of fixed or static values. The user defines a set of fixed values representing the template case. With this type, it will be necessary to define a large number of templates in order to represent all possible cases. Following is a hypothetical example that demonstrates the addition of a weather profile by reference, a IWEC data file for the location of Reykjavik into the template.

<Parameter name=”weatherprofile” type “IWEC”
source=”http://isesprofiles.org/ ISL_Reykjavik.040300_IWEC.epw”/>

Variable content

The template consists of user adjustable or configurable variables, where the user can select for example an item from a range of a predefined list of options or a value from a value range. With this type of template it is possible to represent a much larger number of cases with a single template as the user can configure and tailor the values to the correct context of a given case. This type of template is typical, for example, for preparing the configuration files used by simulation software.

Following is a hypothetical example that demonstrates a list of items that may be selected by a user for defining the different glazing materials used in a building.

```
<Parameter name=" Material" type "Glazing Types">
        <list>
                < listItem>Double Glazing 16 mm Argon 90%</listItem>
                < listItem>Double Glazing magnetron coated</listItem>
                < listItem>Triple Glazing 10mm Argon 90%</listItem>
        </list>
</Parameter>
```

Dynamic content

A template using dynamic content allows to automatically configure, using specialised software, the data objects contained in the template on the basis of the template context. Dynamic templates use embedded processing rules and instructions to build and manipulate the data according to the specific context of each case. The context is stipulated by user supplied conditions and the template data sets are configured accordingly. Furthermore dynamic templates can rapidly generate any kind of data against any data source. This template type needs a much smaller number of templates to represent a high number of cases. Following is a hypothetical example that demonstrates the addition of a stochastic occupancy profile into the template. The stochastic occupancy profile will be dynamically generated and then placed under the tag “<UserProfile>” in place of elements starting with $$.

```
<parameter name=" user profile" type="stochastic>
    <$$Evaluate>
        <$$TemplateTag name="UserProfile" />
        <$$Expression function="ISES.Adaptors.StochasticFunctions.getStochasticUserProfile() />
    </$$Evaluate>
</parameter>
```

In general, the data template provides a skeleton for collection of different data sets hosted in different data sources (Figure 18). Using the dynamic template model provides a flexible way to process data in different contexts and model schemas by establishing a data scope that inherits relevant information from internal or external data sources. These data scopes may execute codes that perform various functions such as data retrieval, calculations, stochastic or probabilistic processing or data formatting.

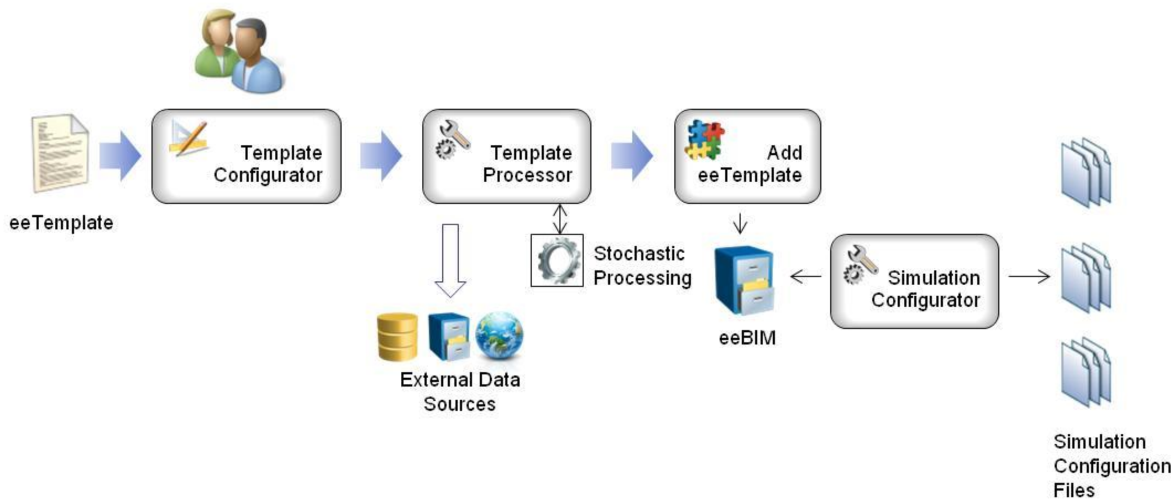


Figure 18. Integration of stochastic templates

Template Categories

The defined template categories are outline in the following table.

Table 4. Template Categories

Category Type	Description
Site	Geographic location, site conditions, weather profiles
Building	Collection of building envelope elements, building facilities, building equipment, building level activity profiles etc, which characterize the building by function
Space / zone	Collection of space boundary elements, energy systems, facility services, facility equipment, space level activity profiles etc, which characterize the space by function
Profiles (activity by space)	Collection of user based activity profiles confined to a given space e.g. workstation, hotel room, class room aggregating occupancy and dependent profiles e.g. ventilation/infiltration, lighting etc, which are characterized by the activity in a given space
Profiles (Schedules)	Individual profiles, occupancy profiles, load profiles or control profiles, which are classified by a given space type
Building element	Space boundary elements, i.e. building elements (wall, roof, door, window) constructed as assemblies or layered materials
Building service element	Facility service elements, e.g. HVAC or Lighting
Material	Collection of properties for a product or a material e.g. thermal conductivity, heat transfer, specific heat capacity, specific flow resistance etc.

Template Encoding

A data template can be encoded as a XML document whose elements collectively define how the data engine will process the template to generate the output XML.

A data template is encoded using W3C XML syntax. Templates can then easily be manipulated by many high performance tools implementing open source standards XML, XSLT, DOM, Xpath etc. Further-more, any text or XML editor can be used to generate data templates. The decision of using XML as the template modelling language further provides the benefit for extending and enhancing the template model transparently throughout the development process minimizing the development effort.

In addition, using XML as the template syntax, template engines or template processors ordinarily used for web based template systems can be adopted with minimum effort in order to be used as a pre-processor, filter, or template processor in the external data framework.

A data template does not feature built-in functionality to support the customization of data presentation such as reformatting or transforming any retrieved data. The encoding of the result set in templates is entirely dependent on the data source provider. For example, data providers can output its data in binary, text or XML format, structured in different ways such as lists or tables. However, it is desirable that the data is delivered encoded as a well-formed XML. Otherwise no assumption is made about the data format used by the provider. It is up to the target application to present the data in the templates in the right format appropriate for the end-user application.

In addition, templates are encoded using the Unicode global character set that enables multilingual support for localized deployment.

Template Processors

Several template engines are available both commercially and as open source or free ware, which meet the design criteria for the framework template processor (Wikipedia Template engine (web), http://en.wikipedia.org/wiki/Template_engine_(web)).

Candidate template processors that were considered and reviewed, include:

- *Thymeleaf*, Apache Licence, Java, full functionality (www.thymeleaf.org)
- *StringTemplate*, BSD licence, Java / C#, limited functionality (www.stringtemplate.org)
- *Casper*, MIT licence, Java, good functionality (code.google.com/p/casper)
- *Rythm*, Apache Licence, Java, good functionality (www.rythmengine.com)
- *Apache Jelly*, java, full functionality (commons.apache.org/jelly)

A decision was made to use the Apache Jelly script engine as the template processor to be adopted for the data framework template processor. It supports all of the defined criteria for dynamic handling of templates. Furthermore, Apache Jelly is fully stress tested and integrated with the MAVEN framework and has been used in a variety of applications (Clarity Software, CA Technologies Inc. etc.) where it has proved reliable and robust. Several features of the Jelly script engine are attractive for use with ISES eeTemplates. Some key features of Apache Jelly (commons.apache.org/jelly) are listed below:

– Apache Jelly is a Java and XML based scripting and processing engine that can be extended to support various custom actions. Jelly is totally extendable via custom actions (in a similar way to JSP custom tags) as well as cleanly integrating with scripting languages such as Jexl
– Jelly uses an XML Output which makes Jelly ideal for XML content generation. A single Jelly tag can produce, consume, filter or transform XML events, which leads to a powerful XML pipeline engine. Furthermore, Jelly has native XML support. It can parse XML and process it using XPath expressions (via the JSTL tags).
– Jelly supports dynamic tags and XML namespaces to allow different tag libraries to work together seamlessly in the same XML document.
– Jelly has full support for pluggable expression languages. The default expression language is a superset of the one used in JSP, JSTL and JSF which supports conditional expressions, navigating bean properties, and working with Maps, Collections, Lists, arrays etc. Jexl is the current implementation which adds some enhancements like method calls on beans etc. Jelly supports other expression and scripting languages like Velocity, beanshell, JavaScript, Jython, pnuts, etc. in separate tag libraries.
– Jelly has a powerful collaboration mechanism for passing information between tags/tasks. In Jelly variables can be any objects plus variable scopes that can be nested to allow nested scripts to work together neatly.

Stochastic templates

In the ISES data framework, the eeTemplates are not implicitly self-stochastic, but rather the data sets that they contain. For example, a material property may be presented as a stochastic sample generated from a probability distribution for that property using the Monte Carlo method. Similarly, an occupancy profile may be generated as a set of time series generated by a Markov Chain process.

Stochastic data sets in the eeTemplates are dynamically generated by probabilistic software, on demand, during the processing of a template by the template processor. Embedded processing instructions guide that execution of the stochastic processor that will be responsible for generating the stochastic data sets.

There is no provision for storing the result of processed templates and this should not be required. However, it may be beneficial to temporarily catch processed stochastic templates when several design options are evaluated to ensure a consistent testing environment and to minimize processing time.

Stochastic Processing

Stochastic processing can be handled by statistical software. MATLAB (MATrix LABoratory) is one of the most common tools used for engineering computing. It is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation (www.mathworks.se). A rich set of modules and tools based on MATLAB are available for diverse statistical, scientific and engineering data processing, statistical computing, stochastic processing and even simulation in modelling, primarily enabled by a library of over 12,000 MATLAB files and toolboxes donated by a strong MATLAB user community[*)]. However, due to high licence fees of MATLAB, ISES intends to look for and exploit open source projects for statistical computing to be used for the pilot development.

The open source **R environment** (www.r-project.org) bares many resemblances to MATLAB, although not a direct replacement (http://mathesaurus.sourceforge.net/octave-r.html).

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S programming language and environment that was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and it is highly extensible. The S language is often the vehicle of choice for research in statistical methodology.

Many features are built into the base R environment, and many others are supplied as packages. There are about 25 packages provided with R (called “standard” and “recommended” packages) and many more are available through the Comprehensive R Archive Network (CRAN) of internet sites around the world that store identical, up-to-date, versions of code and documentation for R (http://cran.r-project.org/). Most classical statistics, probability distributions, and much of the latest methodologies are available for use with R through R-developers community efforts.

3.2 Template Specification

The eeTemplate is defined as a XML document that consists of two basic element sections: (1) the Meta-Data section and (2) the Data Model section as shown in Figure 19.

The meta-data section contains the descriptive elements that identify the context of the template.

The Data Model section includes the content and processing part of the template. The Data Model section, may be further divided into two main sub-sections, namely the:

- Sub-template section: a hierarchical collection of sub-templates that are included as a whole within a given template
- Processing section: uses dedicated script syntax as its instruction base that is outlined next and will be detailed in the template user guide as part of ISES Deliverable D4.2.

[*)] See http://www.mathworks.com/matlabcentral/about/fx/

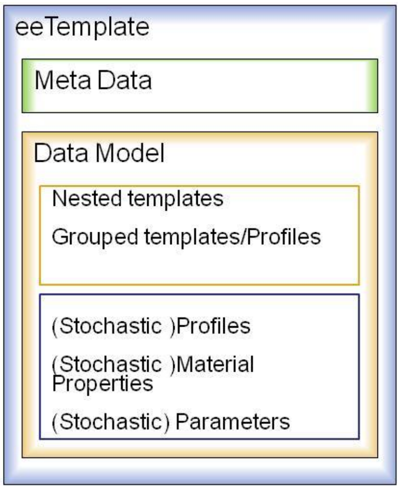


Figure 19. Structure of the eeTemplate

The <eeTemplate> element is the root element of the template. It has a set of related attributes expressed within the eeTemplate tag.

Table 5. eeTemplate Element Attributes

Attribute Name	Description
ID	The unique data template identification
name	The data template name
version	Version number for this data template
language	ISO language identification (EN, DE, IS, FI, GR etc.)

The template properties section contains the template metadata and other properties that define the specific template context, characteristics and general properties as well as controlling the template processor execution. The properties section starts with the <templateProperties> tag followed by sub-elements from Table 6.

Table 6. Property Section Elements

Element Name	Description
description	Short description of the content and use of this data template
templateType	Identification of this template as Stochastic or Deterministic
context Identifier	Functional type of the template by category (Table 4.)
contextDescription	Provides a description of the primary context that this template e.g. application area and restrictions of use
classificationCode	assigned to several building types as defined by OmniClass Table 11 assigned to several space types as defined by OmniClass Table 13 assigned to several building element types as defined by OmniClass Table 21 assigned to several building product types as defined by OmniClass Table 23 assigned to several building material types as defined by OmniClass Table 41

Element Name	Description
weightFactor	Weighting factors for generic templates to be weighed in order to account for varying complexity with the same type of templates: • Low: less than reference values • Medium / average: reference value • High: higher than reference value Examples include accounts for variability in Gross area, Occupancy density (occupants per m^2), Indoor air quality requirements, Energy load from activity, Energy load from lighting etc.

The Processing section contains the instruction set for processing the template content. The processing section contains two types of tags; a) Custom tags which are part of the "ises:" namespace and Jelly tags which are part of the Jelly Script Engine "jelly:" namespace. Table 7 and Table 8 give tag samples to be informative of custom tags and jelly tags respectively. The final specification together with user guidelines will be published as part of deliverable of task 4.2 in WP4.

Table 7. ISES custom processing elements

Element Name	Description
templates	This element starts a template include section in the template document. Any template can be included to aggregate separate templates as a whole, e.g. a space template may aggregate several profile templates. The templates tag uses sub-elements < template>
template	The template tag includes a template, which is identified by a resource descriptor e.g. <template resourceId="12345">
profiles	Starts a profile section. The profile section aggregates separate profiles given a template context. The profiles tag uses sub-elements <profile>
profile	The profile tag inserts a profile in the template. Sub-elements used are <stochasticTask> or <functionTask> to include a stochastic time series, <serviceTask>,<ioTask> or <functionTask> to include a deterministic time series
buildingElements	Starts a building element section. The building element section aggregates building elements to represent a component or constituents of a building e.g. wall, windows, doors. The tag uses sub-elements < buildingElement> and/or <construction> to identify the building elements
buildingElement	The tag inserts a building element in the template. Sub-elements are <ioTask> , <serviceTask> or <functionTask> to include a buildingElement identified by a resource descriptor. Building elements can be both ready-made building products or on-site constructed.
construction	The tag represents a construction as a multi-layered building component consisting of different material layers. Material tags represent each material layer in the construction.
materials	Starts a material section. The materials section aggregates different building materials using sub-elements <material>

Element Name	Description
material	Inserts a material in the template. The material tag identifies a material and the associated material properties. Sub-elements are <serviceTask>, <ioTask> or <functionTask> to include a material identified by a resource descriptor.
materialProperty	The tag inserts a material property in the template. Material properties are used in tag <material> which aggregates any number of energy related material properties e.g. thermal conductivity, specific heat capacity etc. Sub-elements of materialProperty are <stochasticTask> or <functionTask> to include a stochastic sample representing the property value or <serviceTask>, <ioTask> or <functionTask> to include a deterministic value.
stochasticTask	The tag stochasticTask handles probabilistic calculation of parameters. StochasticTask executes a program code identified by codeLibraryId. The code fragments can be a method of a class or R-code to be processed by the R environment.
serviceTask	The tag serviceTask handles invocation of web services for data retrieval. The serviceTask uses a resource descriptor to configure the service interface.
functionTask	The functionTask tag executes a programming code identified by the codeLiberaryId. This can be any arbitrary function to retrieve a resource or for probabilistic processing to provide custom solutions or extensions to the custom tag set.
ioTask	The ioTask tag handles retrieval of file based resources e.g. IFC files. The resource can be described both by an URI or a resource descriptor.
dataInput	The dataInput tag specifies input parameters to tasks. Each parameter is represented by a <parameter> tags
dataOutput	The dataOutput tag specifies how output from tasks should be handled. Sub-tags are <inline> or <toFile>. The inline tag instructs the output data to be inserted in the output template. Data formats can be, besides xml, text or binary, in which case the output is wrapped in a <![CDATA[...]]> tag. The toFile tag copies the output to a file. It is expected that the output is uploaded to the eeBIM and not to any arbitrary location. The sub-element is the <serviceTask> tag with the resource descriptor of the eeBIM upload service.
parameter	The parameter tag describes any input parameter. Attributes are name and type. Type can be any literal type, as specified in the XML Schema specification (www.w3.org/2001/XMLSchema) or a custom type.
selectOption	This tag groups one or more options to select from. Sub-elements can be any selectable element e.g. template, profile, material, material property or parameter.
selectRange	This tag enables the user to select a value from a range e.g. [-20,+20]

Table 8. Sample of Jelly processing instruction tags

Element Name	Description
switch	Executes the child <case>tag whose value equals the “on” attribute. Executes a child <default>tag when present and no <case> tag has yet matched.
case	A tag which conditionally evaluates its body if “my value” attribute equals “my ancestor” <switch> tag's "on" attribute. This tag must be contained within the body of some <switch>tag.
default	A tag which conditionally evaluates its body if none of its preceding sibling <case> tags have been evaluated. It must be contained within the body of a <switch> tag.
choose	A tag which conditionally evaluates its body based on some condition
when	A tag which conditionally evaluates its body based on some condition
otherwise	The otherwise block of a choose/when/otherwise group of tags
forEach	Iterates over a collection, iterator or an array of objects
while	A tag which performs iteration while the result of an expression is true
if	A tag which conditionally evaluates its body based on some condition
expr	A tag which evaluates an expression
file	A tag that pipes its body to a file denoted by the name attribute or to an in memory String which is then output to a variable denoted by the var variable.
import	Imports another script. By default, the imported script does not have access to the parent script's variable context. This behavior may be modified using the inherit attribute.
include	A tag which conditionally evaluates its body based on some condition
invoke	A tag which calls a method in an instantiated object
new	A tag which creates a new object of the given type
scope	A tag which creates a new child variable scope for its body; thus any variables defined within its body will no longer be in scope after this tag.
catch	A tag which catches exceptions thrown by its body; this allows conditional logic to be performed based on if exceptions thrown, or to do some kind of custom exception logging logic.
thread	A tag that spawns the contained script in a separate thread

3.3 Stochastic Template Processor (STP)

Figure 20 shows the suggested main components and modules of the Stochastic Template Processor (STP), external application, services and data resources.

Stochastic Template Processor (STP).

The only input to the STP is the eeTemplate XML file. Optionally, modifiers can be passed to the STP using the URL query syntax “eeTemplateFilename.xml? param1=value1& param2=value2& param3=value3”. The STP therefore provides only one interface method to invoke processing of the template. Similarly, the only output is the processed template. The STP can easily be integrated directly in automated workflows or other pipe-line processing.

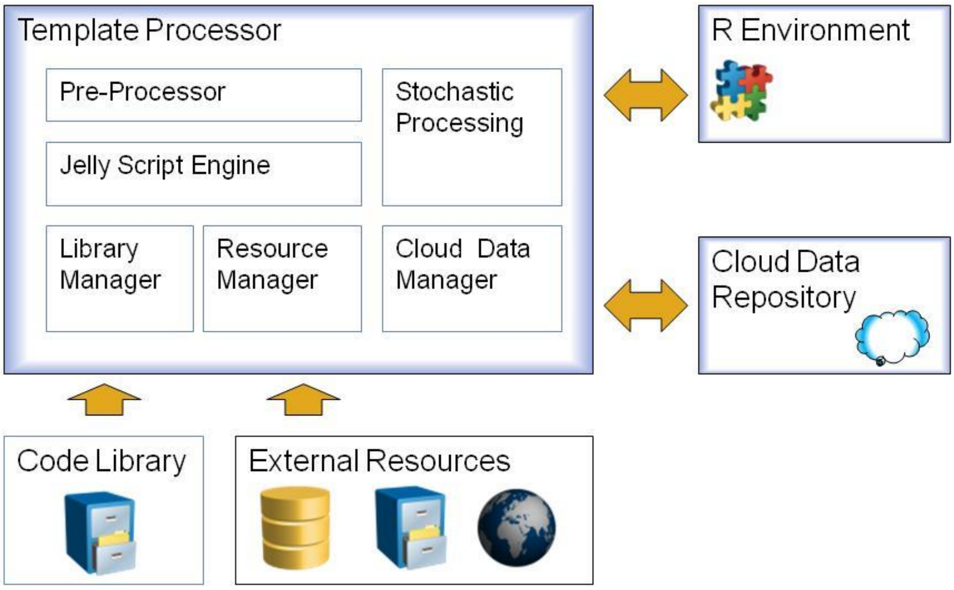


Figure 20. High level structure of the suggested Stochastic Template Processor

Pre-Processor

The pre-processor is responsible for parsing the template request and assigning parameters to Jelly variables. The pre-processor also handles any initialization and pre-class loading through the Library Manager when plug-in modules residing in the Code Library are being used as part of the template.

Jelly Script Engine

The Jelly Script Engine is the core of the Stochastic Template Processor. It is responsible for template parsing and processing and interaction with other modules.

Stochastic Processing and the R Environment

The R Environment is actually a single threaded stand-alone application, but includes features that enable it to run as a kind of a service, capable of accepting connections from other programs. Several language bindings are provided including one for the Java language, the Java R Interface (**JRI**). It allows R to run inside a Java application in a single thread. Basically, it loads the **R** dynamic library into Java and exposes a Java API to R functionality. It supports both simple calls to R functions and a full running REPL.

However, stochastic processing is not confined to the R Environment. Plug-in stochastic algorithms may also be integrated through the Code Library and used in custom tags or in Jelly script code.

Library Manager and the Code Library

The STP can include two types of code to be included in the STP, as follows:

- *Java code plug-ins*. The STP functionality can be easily extended using Jelly custom tag features together with java complied modules.
- *R code fragments*. Such fragments can be directly performed by the R Environment for probabilistic and stochastic processing.

The Code Library provides the needed flexibility in developing new or extending existing functionality of the STP. Through this feature the stochastic processing can be enhanced as further research develops.

Interfacing to the Code Library and basic class loading and error handling are thereby handled by a specialized Library Manager.

Following is a hypothetical example for extending the STP using custom tags and plug-in code.

```
public class ProfileTag extends TagSupport {
  private ResourceDescriptorID _resourceDescriptorID;
    private ResourceDescriptor _resourceDescriptor;
public void setResourceDescriptorID (ResourceDescriptorID id) {
    this._resourceDescriptorID = id;
    }
_resourceDescriptor = Repository.Resources.getResource(resourceDescriptorID);
  public void doTag(XMLOutput output) throws Exception {
                    Adaptors.profileAdaptor().run( context, output, _resourceDescriptor );
   }
}
```

In the Jelly script this custom tag can then be used to retrieve the profile described by the resource descriptor:

```
<ises: Profile ResourceDescriptorID="ABCDEF">
    Profile output …
</ises:Profile>
```

Following is an example using R Code:

Retrieve a sample of 100 elements form a normal distribution with mean 0 and standard deviation 1 into variable x

```
x <- rlnorm(n=100, meanlog=0, sdlog=1)
```

Retrieve a sample of 100 elements for the uniform distribution into variable x

```
x <- runif(n=100, min=0, max=1)
```

Generate a Latin Hypercube Sample of 100 elements in Y using (a) normal distribution, and (b) beta distribution.

```
X <- randomLHS(100, 2)
Y <- matrix(0, nrow=100, ncol=2)
Y[,1] <- qnorm(X[,1], mean=3, sd=0.1)
Y[,2] <- qbeta(X[,2], shape1=2, shape2=3)
```

Resource Manager and the External Resources

The Resource Manager provides access to resources both internal in the Data Repository and external that reside outside of the ISES framework. The Jelly tag libraries for example provide several ways to interface with external resources such as web services and databases. The STP relies on the data framework to supply information about resources to be accessed. For this purpose, the Resource Manager uses the resource descriptors contained in the cloud-based data repository to configure resource interfaces. It also handles thread management and exceptions. However, it only provides basic support in that respect as exposed by the resource web service. For handling complex interfaces that can be generally described, an *Adaptor* module may be used for handling all communication, interfacing and data formatting with the external resource. The Adaptor may then be tied to a custom tag for use in the eeTemplate. The *Adaptor pattern* is seen as appropriate for exposing service interfaces and handling the various tasks needed and thereby abstract the more complex underlying service proxies for more transparent template implementation.

As the retrieval and configuration of external resources during the template processing can be quite lengthy, it can be beneficial to hard-code and optimize the retrieval of reliable frequently used resources by providing a custom tag and special adaptor for those resources. An example would be the TUD_IBK weather profile web service

<ises:TUD_IKB_WeatherProfile>

Cloud Data Manager and the Cloud Data Repository

The Cloud Data Manager is intended as a thin wrapper for the interface to the Cloud Data Repository.

4. Implementation of the framework and its integration in the ISES Platform

The overall architecture of the ISES Virtual Energy Lab (VEL) is detailed in the ISES Deliverable D2.2 (Baumgärtel et al. 2012). The VEL CORE provides the basic support for the data framework and stochastic templates. It further provides support for user software e.g. nD Navigator to search, access and select matching templates and further configure and realize stochastic templates to be integrated as part of the eeBIM. This will include the following software components:

1. Data management component as plug-in for the nD Navigator
2. Interface module to the cloud-based data repository providing support for intelligent search and retrieval of meta-data about templates, profiles and product and material information
3. Interface module for retrieving data stored externally or in the cloud based data repository
4. Template Processor providing support for programmatic and stochastic processing of templates
5. Stochastic Processor enabling probabilistic calculations for stochastic data.

The VEL Core is responsible for most of the processing work and interaction with the end-user GUI e.g. management and formulation of end user data requests, multi-model management of eeBIM and executing the Stochastic Template Processor (STP) for dynamic template realization and interface to the cloud-based Data Framework. This functionality is mostly realized using web services compliant with the ISES SOA approach. The process of obtaining all the required eeBIM data may take several round-trips as different templates need to be realized for the eeBIM.

As the cloud-based data repository provides full-bodied meta-data for data objects, configuring the eeBIM by the end-user, that is searching and selecting data objects for consolidation in the eeBIM, is minimized with regard to service load and data traffic to/from the cloud data repository. The actual data is not retrieved until the eeTemplate is processed by the Stochastic Template Processor or a user directly requests its upload. Furthermore extending the capabilities of the data framework and stochastic processing and introducing additional functionality such as data filtering, transformation between formats etc. can be accomplished without any modification to the overarching ISES platform.

Figure 21 illustrates the conceptual approach to the integration of the data framework and stochastic templates.

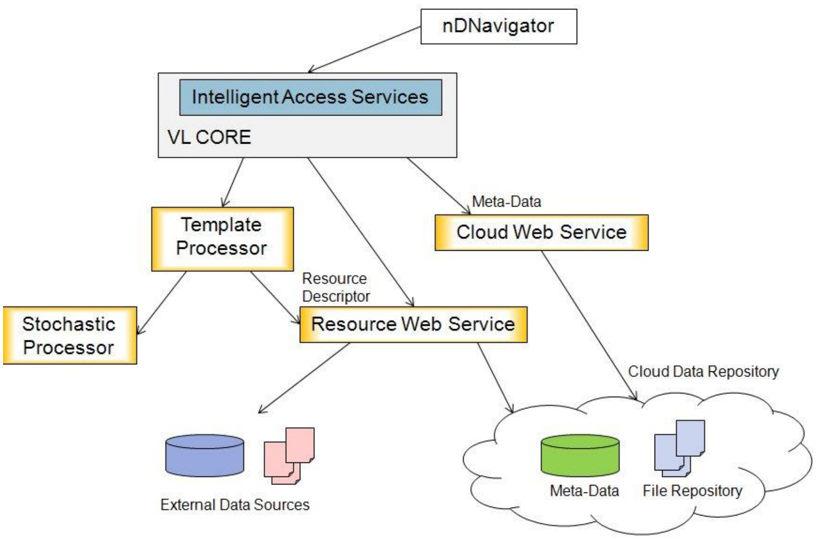


Figure 21. Integration of the data framework and stochastic templates

5. Conclusions

The preceding chapters of this report presented an overview of the work performed under task T4.1 “Framework and stochastic templates for product life-cycle”, under WP 4 “Energy Profile and Consumption Patterns for Built Facilities and Their Components”. This task has two main goals:

a) Specification of a service framework for the search and retrieval of energy related data in distributed resources

b) Specification of characteristic stochastic profiles for climate data, activity and usage patterns and energy related product information.

Based on the findings in WP1 reported in ISES Deliverables D1.1 (Kavcic et al. 2012) and D1.2 (Jung et al. 2012) an approach was developed that provides a generic and innovative solution to the framework design and specification of stochastic templates. Under this concept, the overall process is simplified by transposing it to a pure data framework containing descriptive meta-data of the data services and various data types necessary for enrichment and consolidation in the eeBIM. In addition, the stochastic template became a dynamic entity capable of integrating external services, data resources and stochastic computational algorithms based on a particular simulation context.

The main benefits resulting from the proposed approach are as follows:

1) User selection of data for eeBIM is based on light-weight meta-data originating from one source, thus minimizing network traffic => **lean platform**

2) Instantiation of the actual data is deferred until it is actually needed => **just-in-time processing**.

In addition, this approach can be more transparently implemented in the overall architecture of the ISES Virtual Energy Lab (VEL). Given that the solution is generic and not confined to any specific energy calculation or simulation software, it is exploitable beyond ISES thus providing broader adaptation and implementation potential.

The work achieved so far will provide a sound basis and technical foundation for implementation of the data framework and stochastic templates and their integration in the overall VEL platform. The specifications are, however, not as detailed as set out due to early scheduling of this deliverable in the overall project plan. Accordingly, an expanded and revised edition of this deliverable is planned to be published that will provide more details, once the stochastic algorithms to be implemented are defined in detail. Further enhancements are also anticipated to surface during the implementation phase and the development of the Stochastic Template User Guide.

References

Baumgärtel K., Katranuschkov P., Hollmann A., Laine T., Protopsaltis B., Dolenc M., Klinc R., Gudnason G. & Balaras C. (2012): *ISES Deliverable D2.2: Architecture and components of the Virtual Lab Platform*, © ISES Consortium, Brussels.

Bort B., Geißler M.-C., van Woudenberg W., Guruz R., Kaiser J., Hensel B., Zellner R., Laine T., Jonas F., Liebich T., Katranuschkov P. (2011): *HESMOS Deliverable D1.1: Gap Analysis, Use Case Scenarios and Requirements Specification*, © HESMOS Consortium, Brussels.

Bourdeau M., Laresgoiti I. (2011): Energy and behavioural modelling for managing energy resources in residential buildings, *Proc. 2nd Workshop on EE-Building Data Modelling, CIB W78-W102 Conference*, Sophia Antipolis, France.

Bourdeau. M et. al. (2011): *FIEMSER Deliverable D5: FIEMSER Data Model*, © FIEMSER Consortium.

Davis. J.A. III, Nutter D.W., (2010): Occupancy diversity factors for common university building types, *J. of Energy and Buildings* 42 (2012), 1543-1551.

Drury B. C. (1997): *Does It Matter Which Weather Data You Use in Energy Simulations?* U.S. Department of Energy, Building Energy Simulation User News, Spring 1997 – Vol 18, No. 1.

Eames M., Kershaw T., Coley D. (2012): A comparison of future weather created from morphed observed weather and created by a weather generator, *Building and Environment*, 56 (2012), 252-264.

Grunewald J., Kaiser J., Guruz R., Zellner R.: (2011): *HESMOS Deliverable D5.1: Specification of interfaces to material and climate databases*, © HESMOS Consortium, Brussels.

Hand J.W., Crawley D.B., Donn M., Lawri L.K. (2008): Improving non-geometric data available to simulation programs, *Building and Environment* 43 (2008), 674–685.

Hopfe C. (2009): *Uncertainty and sensitivity analysis in building performance simulation for decision support and design optimization*, Ph.D. Thesis, TU Eindhoven, The Netherlands, 229 p.

ISO 12006-2 (2001) *Building construction -- Organization of information about construction works -- Part 2: Framework for classification of information*, International Organisation for Standardisation, Geneva.

ISO 12006-3 (2007): *Building Construction -- Organization of Information About Construction Work -- Part 3: Framework for Object-Oriented Information*, International Organisation for Standardisation, Geneva.

Jung R., Mansperger T., Guruz R., Forns-Samso F., Gudnason G., Kaiser J., Katranuschkov P., Kavcic M., Laine T., Zahedi A., Zingel U., Zschippang S. (2012): *ISES Deliverable D1.2: Use Case Scenarios and Requirements Specification*, © ISES Consortium, Brussels.

Kavcic M., Balaras C., Gudnason G., Guruz R., Kaiser J., Katranuschkov P., Laine T., Mansperger T., Pappou T., Černe B. & Zahedi A. (2012): *ISES Deliverable D1.1: State of the Art and Gap Analysis for Energy Performance Assessment*, © ISES Consortium, Brussels.

Liebich T., Stuhlmacher K., Katranuschkov P., Guruz R., Nisbet, N., Kaiser J., Hensel B., Zellner R., Laine T., Geißler M.-C. (2011): *HESMOS Deliverable D2.1: BIM Enhancement Specification*, © HESMOS Consortium, Brussels.

Mahdavi A. (2009) Cogitative buildings: concepts, technologies, implementations, *ITcon Vol. 14, Special Issue Building Information Modeling Applications, Challenges and Future Directions*, pg. 692-704, http://www.itcon.org/2009/45

Page J., Robinson D., Scartezzini J-L. (2007): Stochastic Simulation of Occupant Presence and Behaviour in Buildings, *Proc. Building Simulation Conference 2007*.

Widen J., Nilsson A.M., Wackelgard E. (2009): A combined Markov-chain and bottom-up approach to modelling of domestic lighting demand, *J. of Energy and Buildings* 41 (2009), 1001–1012.

W3C XML Schema Specification (2004): XML Schema Part 0: Primer (http://www.w3.org/TR/xmlschema-0), XML Schema Part 1: Structures (http://www.w3.org/TR/xmlschema-1), XML Schema Part 2: Datatypes (http://www.w3.org/TR/xmlschema-2), Second Edition, W3C Recommendation 28/10/2004.

Zahedi Khameneh A., Scherer R. J., Gudnasson G. (2012): *ISES Deliverable D2.1, Overall Stochastic Approach for the Virtual Energy Lab Platform*, © ISES Consortium, Brussels.

Appendix I: Acronyms

AEC	Architecture, Engineering and Construction
API	Application Programming Interface
BIM	Building Information Modelling
CFD	Computational fluid dynamics
eeBIM	Energy enhanced Building Information Modelling
FM	Facilities Management
GUI	Graphical User Interface
HESMOS	EU Project No 260088 "ICT Platform for Holistic Energy Efficiency Simulation and Lifecycle Management Of Public Use FacilitieS"
HVAC	Heating, Ventilation and Air Conditioning
ICT	Information and Communication Technology
ID	(unique) Identification
IFC	Industry Foundation Classes
IFD	International Framework for Dictionaries
LoD	Level of Detail
NIST	The National Institute of Standards and Technology of the USA
TMY	Typical Meteorological Years
TRY	Test Reference Year
URL	Uniform Resource Locator
VEL	Virtual Energy Laboratory
WYEC	Weather Year for Energy Calculations
W3C	The World Wide Web Consortium
XML	Extensible Mark-up Language
XSD	Extensible Mark-up Language Schema Definition