FP7 – 285229 – Collaborative Project

Knowledge-based energy management for public buildings through holistic information modelling and 3D visualization

Deliverable 2.2

Method development: building behavioural model creation, building specific ontology creation

Authors	Polina Häfner (KIT), Kris McGlinn (TCD), Anton Gerdelan (TCD), Hendro Wicaksono (KIT)
Reviewers	Preslava Dobreva (KIT)
Delivery due date	28.02.2013
Actual submission date	01.03.2013

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

Deliverable 2.2 “Method development – building behavioural model creation, building-specific ontology creation” represents a significant step forward towards meeting one of the main KnohoEM objectives – to manage the complex interlinking between the different elements within the project by mapping them to the central ontology. This ontology is derived from the generic ontology developed in deliverable 1.1 and which is undergoing iterative development cycles as new knowledge is gathered from the building objects. This generic ontology is then extended with building data to create and populate the building specific ontology, building on the building analysis which took part in deliverable 21.

As described in the DoW, deliverable 2.2 represents an overview of the work performed in the frame of task 2.2 and 2.3 of WP2 dealing with the specific ontology enhancement trough population of semantic extracted data from 2D building layouts and the occupants behaviour. Two separate tools have been conceptualised based on the requirements for correlation and knowledge extraction and have been implemented independently from the data gathering from the demo buildings. These modules are integrated directly into the core ontology. The first development and validation phase has been built upon data from the Forum Building in the Netherlands, the MediaTIC and Bluenet buildings in Spain.

Chapter 1 distinguishes the generic ontology (WP1) from the specific building ontology (WP2) and describes the role of the specific ontology in the KnoholEM approach. It presents an overview of the development process and the different connections to the building automations systems, layout plans and behaviour information so as to provide a coherent understanding of the requirements set up for the chosen methods.

Next, the technical chapter 2 gives a short description of existing technologies relevant to the KnoholEM research. Exchange formats, Computer Aided Design (CAD) construction drawing issues and technologies for building geometry visualisation and behaviour modelling are discussed. It examines methods for rapidly extracting data from the 2D CAD models for the purpose of visualisation. It explores the state of the art in activity modelling and issues related to visualisation of building geometry to support the methods presented in Chapter 4.

Chapter 3 presents the established concept to derive semi-automatically interpretation of the building’s geometric model (CAD layout plan) by using input coming from the generic ontology. The challenge hear was to set up a common matrix for extraction of data coming from many different formats, as well as to support extraction of as much semantic related data from the CAD

layouts as possible – thus including creation of automated connection between identified objects, pattern matching algorithms and to allow import of industry standards, especially IFC. The implementation development and first labour test of the OntoCAD tool are presented last.

Chapter 4 concentrates on the interpretation of the behaviour data and the population of the building specific ontology with user behaviour. It focuses on uses cases developed in deliverable 2.1 for occupants to model the activities they conduct and how they interact with the building. It documents initial experiences with building visualisation to support a flexible, extensible and accessible (to occupants and facility managers) web based tool to support activity modelling. Examples are given of the generic ontology and the instance data which is being generated by the tool. It also introduces a methodology for evaluating the level of usability of the developed tool, as its level of usability will impact its uptake by the aforementioned users.

The report finishes with a conclusion summarizing the achieved results and further tasks to be developed until the end of the project.

1 SPECIFIC BUILDING ONTOLOGY AND BEHAVIOUR MODEL CREATION – REVIEW

1.1 Building Specific Ontology (BSO) concept

The objective of the creation of building specific ontologies for all demonstration objects lies on the ability to enable interoperability and to harmonize different knowledge coming from different components (data extraction for energy modelling, mining, behaviour related events etc) in the KnoholEM architecture. One of the biggest challenges in task 2.2 and 2.3 is to correlate user-provided occupant activity with building structures, and further with the sensor information coming from the metering device as well. The BSO is therefore considered as **main interface between the knowledge providing components and the sensing devices**. Several methods for its enhancement are taken into account that corresponds to the structure of the established classes within the Generic Ontology (GO). The population of the ontology with `Building Element` instances is provided by the OntoCAD tool (task 2.3) ; `Behaviour` related instances are been provided by the SmartBuildVis too (task 2.2) ; instances related to the `BuildingControl` that send the information of the building's current states will be provided by the installed sensors in the frame of WP3. On next stage in the project, the BSO is expected to support the development of simulation models (WP3), the visualisation of the "user in the loop" and the energy anomalies prediction trough the SmartBuildVis tool and the data mining algorithms.

Due to the high inerrability required during the concept phase of the BSO, special attention has been given to the project objective that the solution must go a step ahead from the standards by providing reasoning capabilities, but still keep its alignment to the industry standards (especially IFC). Thus the IFC specifications have been taken into account by the definition of the ontology classes and additional efforts for the import of IFC files together with all different AutoCAD versions which have been made during the implementation of the 2D recognition tool. More details are provided in the following chapters. Figure 1 illustrates the role of the BSO and its relations with the knowledge applying components of the KnoholEM architecture.

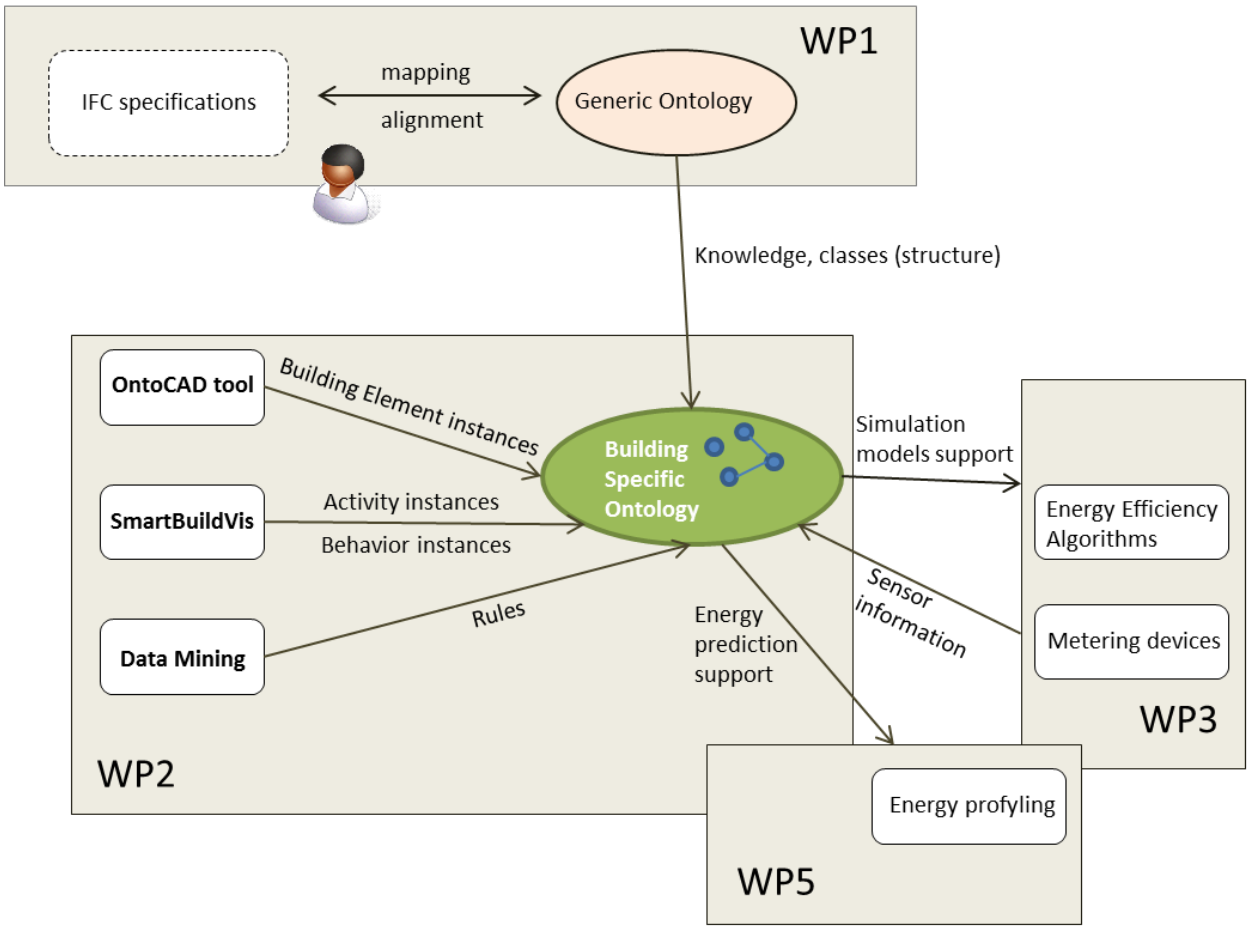


Figure 1: Role of the Specific Building Ontology

1.2 Overview of Ontology Development Methodology

In KnoHolEM project, the development of ontology can be divided at two main phases, i.e. development of generic ontology and development of building specific ontology.

Generic ontology represents the domain knowledge for building holistic energy management. It contains concepts/classes, terminologies, relation definitions, property definitions, and common rules. These components correspond to the T-Box knowledge. The generic ontology is aligned to industry standards, especially IFC, where the ontological classes are mapped explicitly with IFC entities. The generic ontology is created by domain experts, who in the case of KnoHolEM project, are experts from the Technical Group. The generic ontology is mainly developed in the WP1. The initial version of the generic ontology was presented in deliverable 1.1. This has been systematically extended through the analysis of the building demonstration objects presented in deliverable 2.1. Figure 2 illustrates a screen shot of classes contained in generic ontology.

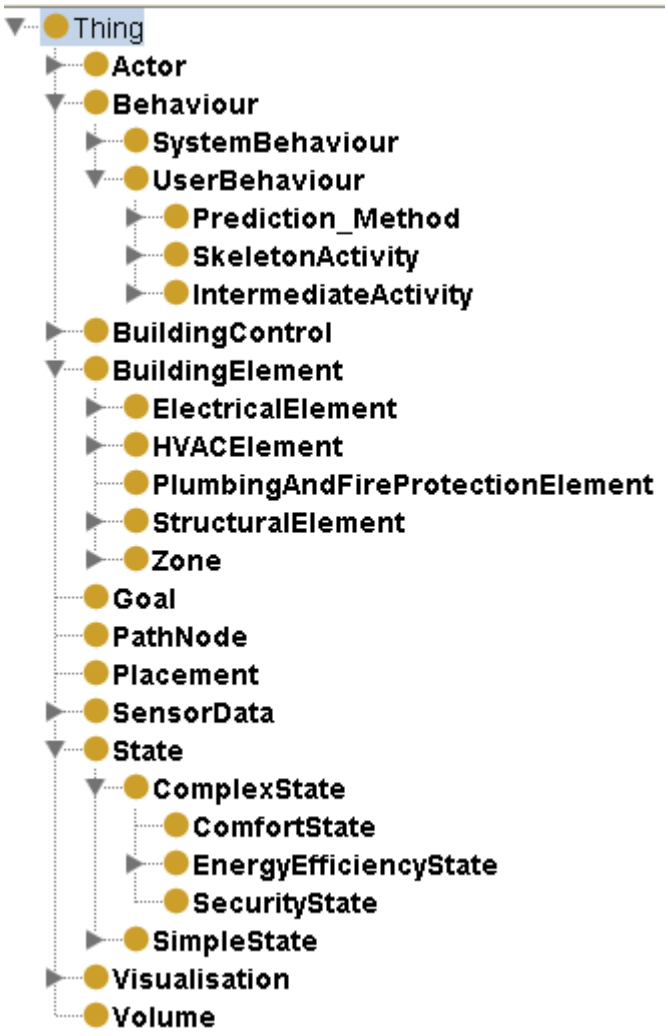


Figure 2: A screenshot of the classes in generic ontology

In order to make it applicable for supporting the energy management tasks, the generic ontology should be enriched and populated with building specific information resulting in the *building specific ontology*. The tasks to enrich and populate the ontology are mainly within the scope of WP2. Populating the ontology means creating the instances (A-Box components) and assigned them to the corresponding classes, setting the property values, and creating the relations between instances. However, this deliverable focuses on the enrichment and population of the ontology using building specific behavior model and geometrical data coming from 2D drawing and IFC files.

Figure 3 illustrates both phases in the development of the ontology within the scope of KnoHolEM project.

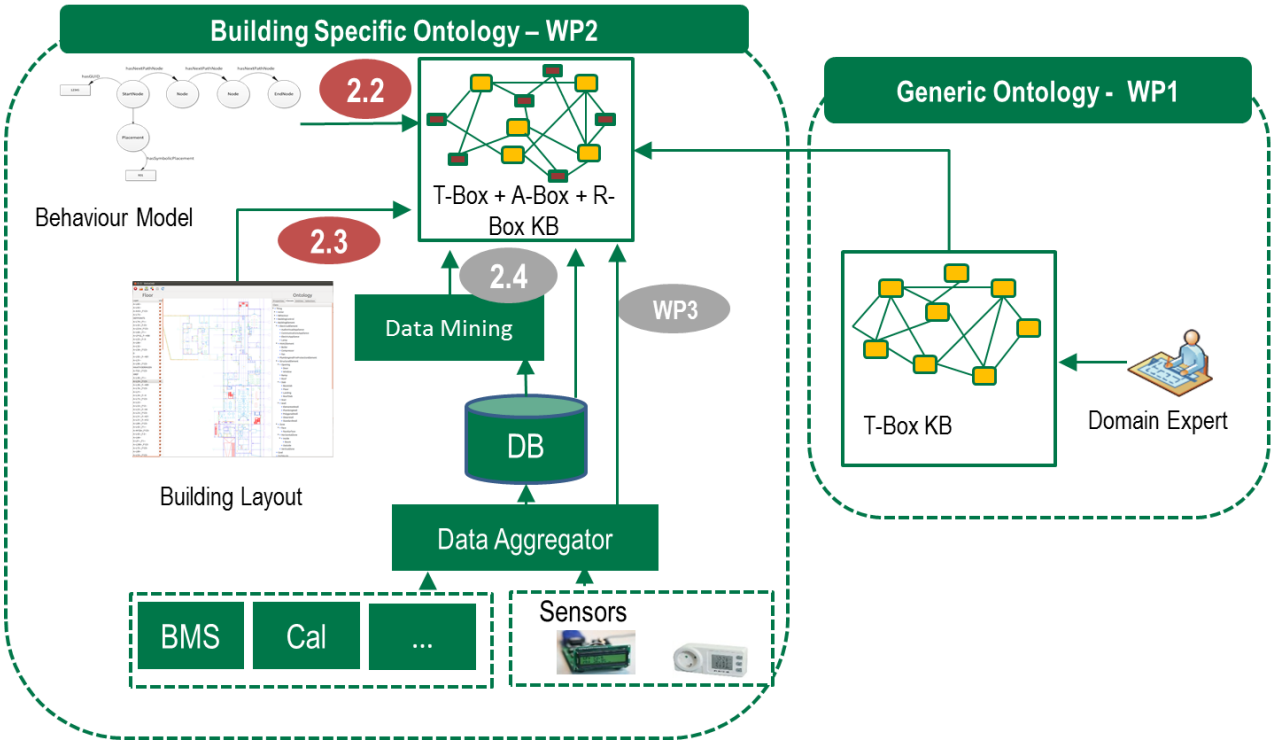


Figure 3: Methods to develop the ontology in KnoHolEM project

As shown in Figure 3, the building specific ontology is created using different methods, implemented in the KnoHolEM project. The instances representing building specific behavior are created using an activity modeling tool. The tool is based on an approach that identifies the different types of user in the building and the range of activities that can be associated with that user. These models are then used as input into the development of the tool that allows the building users input data on their activities, which in turn can be used to inform predictive simulations of energy consumption and also in the development of rules for building control. In other words, the tool populates the `Behaviour` class, its sub classes, and other related parts of the generic ontology.

The task 2.3 as depicted in Figure 3 populates the ontology from geometrical information coming from building layout drawings. The population begins with the transformation and extraction of geometric primitives from the existing geometric descriptions of the building. The extraction of semantic information from geometric data is achieved through semi-automated interpretation of the building's geometric model (the CAD layout plan) based on pattern-matching algorithms. The approach mainly populates the `BuildingElement` class and its sub classes. This approach will be presented in details in chapter 3.

There are further methods to enrich and populate the generic ontology with building specific information, such as enriching the ontology with rules and functions extracted from different data using data mining approach. The approach is documented in deliverable 2.3. Other method to extend and enhance the ontology is population of sensor and building automation classes from building management system data. This is performed, for example, with the help of tools developed in WP3. The ontology could also be extended using information from different sources such as calendar system, business process, etc. However these are not within the scope of KnoHolEM project.

At the end of ontology development, five building specific ontologies corresponding to the five demonstration objects will be derived from the generic ontology. A building specific ontology is used as knowledge base to allow better holistic energy analysis based on relationships between different resources and entities within a building. The algorithms that perform energy evaluation and optimization use ontology as the information/knowledge source. They will be mainly developed in the scope of WP3.

1.3 Methodology for Building Analysis and Requirements Gathering

The building analysis in DR2.1 set out to discover all information regarding the building which is of relevance to, and which can be used to help reduce, energy consumption. It began with examination of the existing building models which are required to develop the building specific ontology.

1.3.1 Building Models

The building models included 2D CAD blueprints of each building, as well as Google SketchUp [1] models. We found that the 3D models which were provided either covered only particular areas of the buildings or that they contained only the external walls and windows of the building. Also 3D Google SketchUp models provided only describe surfaces for visualisation and as such contain no information about the shapes that are represented, for example whether a surface is a wall, a window or the surface of a table. The 2D models provided did contain additional semantics, for example the materials of walls. This information is difficult to extract as it is provided in a “key” which associates certain shapes, for example a squiggly line, with semantics like “concrete”. The language alongside the key was also the local dialect, which contributed to the difficulties of extracting semantics from the models.

Much of the data regarding the building systems and devices were also contained in separate documentation. For example, information on sensors varied in their presentation from MS Excel

sheets, MS Word documents to PDF files with different levels of detail regarding their technical capabilities. These were often associated with separate CAD models which gave their 2D placement. The same situation occurred for other systems like the Heating, Ventilation and Air-conditioning (HVAC) and electrical installations. Details concerning the Mechanical and Electrical (M&E) services collated for the sampled buildings ranged from simplistic 2D plans colour coded for their service strategy type in parallel with additional explanatory notes to full, as built CAD drawings allowing each item of plant, it's size/rating and the area it served to be noted. The heterogeneous nature of the data sources and data representation contributes to the problem of extracting knowledge about the different building systems to support energy management solutions.

1.3.2 Site Visits, Interviews and Occupant Behaviour

Site visits were also conducted for each building object. These included interviews with each building facility manger (FM). These assessed how the FM interacts with the building and how they detect energy wastage as well as how they address these issues. Energy wastage can be due to behaviours as simple as leaving lights on to more complex behaviours like those due to the "proper" functioning of the automation [25]. For example, an air conditioning unit being turned on due to sensors detecting high CO2 levels, even as occupants begin to open windows and doors to improve air quality. This latter type of energy wastage can be difficult to detect and needs more careful consideration of the activities of building occupants in relation to the Building Automation Systems (BAS).

The FMs were therefore also questioned about how they record user behaviour and whether activity models for occupants exist, as inputs into the building specific ontology for user behaviour. Information regarding their experience with the building automation/control and management systems were also questioned. From these questionnaires and the analysis of the building models it was discovered that there was a distinct lack in activity models for the building occupants. Where these models exist they tended to only indicate the working hours of occupants. The existing energy saving strategies were also found to be entirely dependent on the FM who must, through his own analysis of the building behaviour and in adherence to building standards, configure the building systems. To address these issues, a framework for knowledge-based energy management has been developed.

2 BACKGROUND AND STATE OF THE ART

2.1 Building Information Modelling and Industry Foundation Classes

A building information model (BIM) describes an integrated data model for storing all information relevant to the BLC. This can include a 3D model of an architectural design, electrical installations, fire protection, occupancy, energy consumption, costs, CO2 emissions, etc. A BIM goes further than just providing consistent representation of objects; it also defines object parameters and relations to other objects. Collectively this data can be used for visualisation and simulation of the building throughout the BLC [10]. Within the architecture, engineering and construction (AEC) community the Industry Foundation Class (IFC) is an open, freely available, non-proprietary data model which can be used to exchange and share BIM data between applications without the necessity to support numerous data models [11]. Green Building XML (GbXML) is also described as a BIM which is prevalent in the energy simulation domains. Its focus however is on geometric data and often falls short in other areas, such as HVAC (Heating, Ventilation and Air-conditioning) information and descriptions of space boundaries [12].

Due to this shortcoming and as IFC is also the only BIM currently an ISO PAS standard [11] IFC is therefore a primary candidate for BIM. In practice though, IFC has yet to make a significant impact in the AEC communities [13]. A major issue is the complexity of developing full BIM [8] which contains not only detailed 3D models of the static architectural geometry for visualisation, but also other data, such as volumes of spaces for energy modelling, as well as descriptions of the different devices and controls systems required for performance simulation [1]. Modelling all this data in a consistent manner is a considerable challenge, and for existing buildings, the benefits of developing such models must be examined against the costs. Usable visualisation tools are therefore needed to support contributing towards BIM and which can then be evaluated to determine if they result in energy savings.

2.1.1 BIM Development

Development of BIM generally begins with an architectural plan, typically created in CAD software like AutoDesk [2] or ArchiCAD [3]. Sometimes a 3D architectural model is also developed, but currently 3D IFC models are not yet common across buildings. Energy modelling is greatly improved with the addition of 3D geometric volumes not only for visualisation and easier “zoning” (of similar thermal areas) but also allowing accurate sun-path and self-shading studies to be conducted to allow for solar gains. Due to the lack of 3D information available in building models, these are typically modelled and calculated separately to existing BIM [14]. For energy-related

visualisation, and any non-technical user interface, it certainly makes sense to attempt to visualise a 3D representation of the building. Day-to-day users of a building (for example, when plotting activity modelling) may also prefer to navigate the building as they would see it, rather than a floor-plan that may be less familiar.

Tools which support different building users, for example facility managers or office workers, contribute towards the BIM in a manner which can reduce energy consumption and thus building costs can therefore demonstrate the benefits of BIM. Tools like Google SketchUp have already set out to address this for developing simple surface models and more recently have begun work on extending to support BIM primitives like walls and windows. These models can also be converted into GbXML using tools like gModeller [16]. As yet, these models do not address the numerous systems involved in modern “smart buildings”, which also include entities like sensors [17] and models of the activities of building occupants. Tools are required which can enable different stakeholders contribute towards BIM.

2.2 Interfaces and Existing Tools for Extraction of Data from 2D CAD Drawings

Smart buildings need intelligent systems to address the issues of energy and resource efficiency or user comfort. Such systems need access to building and environment specific information to make decisions, thus the need for a knowledge database with semantic information. Getting such information is not trivial due to deficient interfaces and exchange formats. To accelerate and facilitate the process of information gathering, an automated approach is needed. We now review the existing main stream exchange formats for CAD layout data.

DWG is a binary file format that stores two and three dimensional design data and metadata. It is the native format for several CAD packages including AutoCAD, IntelliCAD (and its variants) and Caddie. In addition, DWG is supported non-natively by many other CAD applications. As the biggest and most influential creator of DWG files it is Autodesk [31] who designs, defines, and iterates the DWG format as the native format for their CAD applications. Autodesk sells a read/write library, called RealDWG, under selective licensing terms for use in non-competitive applications. An alternative is the also commercial Teigha runtime library from the Open Design Alliance [32]. Those libraries are interesting for industry products, but are not attractive for academia, where open exchange formats are preferred.

DXF stands for Drawing Exchange Format and is an Autodesk CAD data file format for enabling data interoperability between AutoCAD and other programs. Autodesk supports both ASCII and binary forms of DXF. As stated in the official Autodesk specification for DXF [33], all user-

specified information in a drawing file can be represented in DXF format. An approach that combines CAD information and ontology is presented in the work of Garcia [34]. This proposes using OWL as a CAD Data Exchange Format, giving the possibility for the addition of more descriptive information of products and processes in one self-content and self-descriptive file. He used the CAD data exchange format DXF to classify a two dimensional design as a set of classes and instance in order to populate geometry ontology. The classes of the ontology correspond to the lines and arcs in the drawing file and aims to facilitate the feature extraction using the ontology technology. However the main weakness of this methodology is that it overloads the ontology with irrelevant information of high granularity.

The open IFC standard is intended to enable interoperability between building information modelling software applications in the AEC/FM industry. Deliverable 1.1 gives an overview of the usage and existing extensions of the IFC standard. We investigated the potential of the IFC format because the IFC exchange format becomes more and more popular for its ease of interoperability between software platform and BIM concept, thus companies start trying to adopt it. The IFC standard promises semantic information going beyond the simple visual representation of the building [29] [35], but it is still rarely used in current construction projects. That makes methodologies using the IFC format as main input for semantic information in ontologies, at this stage, not attractive.

Wicaksono et al. [38] is another author that developed a method for semi-automated interpretation of semantics from 2D drawings. The AutoCAD export function was the first component of this framework. They used an XML-based transfer format that allows the development of a generic recognition engine that works with a neutral format and that can be extended by additionally exporting modules for various CAD applications. The function was developed using ObjectARX. The export function allows a variable degree of detail for the exported geometric primitives. Further they used configurable JavaScript based rules which define the relation between CAD-symbols and the semantics of the building components and energy consuming appliances and saved the result as ontology individuals. The drawback of this method is that the user must define diagrams and write recognition rules (for each drawing) for each type of objects to be mapped. It is difficult to check if the mapping rules give correct output for each object, therefore the ontology population errors will be difficult to be detected.

The challenge of extracting semantic information from CAD data is heavily dependent on the quality of the data exchange format. Important characteristics are the amount of information loss when exporting from CAD authoring tools and the compliance to open standards. For these

reasons we use DXF exchange data format. This is because it is easy to use, and is easy for authors to export and has minimal information loss. This approach will still be valid for standards with more semantic information than in simple CAD drawings. We gathered experience with the discussed exchange formats by implementing and parsing them into our solution. The results are described in more details in section 3.

2.2.1 User Behaviour and Activity Modelling

As discussed in deliverable 2.1 the use of HVAC systems, lighting and other devices (e.g. personal computers) collectively contribute to the energy consumption of a building and also have a strong correlation to the behaviour of occupants. Understanding how building occupants are using the physical building is therefore essential to the development of strategies to reduce energy consumption. These strategies can either be implemented by the FM in the configuration of energy consuming devices or the layout of the space and its occupants, for example, by placing individuals in locations that ensure comfort with the least demand on HVAC and lighting. Alternatively, building automation systems (BAS) can use data on building occupant behaviour to predict usage and therefore adjust HVAC, lighting and device power settings to best meet the needs of the occupants in the most energy efficient manner [19], [20], [21].

These methods require the development of activity models which describe the behaviour of the occupants. Activity models range from those to predict lighting energy performance [19], to user interactions with windows and its effect on thermal comfort and energy use [20], to larger sets of interactions including, in addition to aforementioned windows and lights, activation of heaters and fans [21] and additional activities like getting a drink or having a smoke [22]. These studies have mainly focussed on office spaces, due to the imperative to improve operational, safety and energy efficiency in office buildings in use. Office use cases also offer similarity between the activities of one office user to another (e.g. working at desk, going to meetings, etc.) and the predictability of work times for the majority of office users (e.g. a nine to five work day). These factors may make the task of modelling activities more fruitful and tractable than modelling activities for buildings which are less numerous and also have less predictable use (e.g. a university campus) and it has been claimed that building predictable usage is a pre-requisite for accurate predictions in simulation [23].

While this is the case for skeleton activities, as defined by Tabak [22], when modelling intermediate activities the type of building has less impact on the predictability of the activity. Skeleton activities are directly linked to the role of the person, like giving a lecture or cleaning the toilet while intermediate activities include actions like “going to the toilet” or “having a drink”.

Intermediate activities can be modelled using either probabilistic or S-curve methods [22]. More recently, Shen et al. [24] have built on Tabak's approach and developed a model based on BIM to conduct pre-occupancy evaluation of buildings. They provide 3D visual tools of how occupants are using the building to help the understanding of the designer on the impact of their design decisions. As their solution is for pre-occupancy, it is entirely dependent on existing statistical models of activities with no relation to the actual building in question. For buildings already in the operational stages, data from sensors and the occupants themselves can provide valuable input into the activity model. Their approach is also reliant on proprietary tools to develop the 3D models (e.g. 3Ds Max Design). This type of development in itself requires expertise. A method which can be rapidly deployed and is usable by all stakeholders (including occupants) in the development of activity models can provide valuable input also for the development of the building specific ontology and thus help reduce energy. Visualisation tools which support the building occupants in the process of modelling their activities, for example their path through the building, are therefore necessary. To make the solution acceissible, an understanding of the issues related to visualising aspects of the building, likes its geometry is required. The next section examines these issues.

2.2.1.1 Visualisation of Building Geometry with 3D Graphics Rendering

The KnoHolEM solution includes the "user-in the-loop" for Modern 3D graphics are processed on specialised hardware; the GPU, or graphics processing unit, which runs separately to the CPU. Modern GPUs have in the order of a hundred parallel processors. With careful design, and subdivision of rendering models, we can re-programme a GPU to process our rendering hundreds of times faster than with a CPU-based software rendering pipeline (one operation per processor in parallel, rather than in series). This allows us to construct considerably more complex visualisations or virtual worlds. Recent industry advances mean that GPUs can now be considered commodity hardware; most desktops and laptops either ship with a specialised graphics adapter, or have an integrated GPU chip on the main board. Smart phones, tablets, and even netbook computers are produced with GPUs as standard. Graphics software that re-programmes the GPU is written to a graphics rendering library interface, this is generally a variant of either the OpenGL library, or the Direct3D library.

In recent years, there have been attempts to wrap, or create abstractions of the interface software, in order to create a platform-independent graphics interface. Notable examples would be Java3D, JOGL, jMonkey Engine, and LWJGL, which wrap OpenGL in a Java interface; allowing the software to run on any device supporting the Java Virtual Machine [18]. A very recent innovation is the WebGL interface, which is designed to address this platform independence

issue, and is based on the OpenGL ES specification to support a wide range of devices. WebGL is implemented by the developers of web browsers, and will be installed in browsers transparently - making the platform or operating system the web browser, instead of the underlying system, or separate virtual machine. WebGL is designed to integrate with the HTML5 specification (available since January 2011), and the 3D viewing area is displayed inside the new <canvas> tag, inside a regular HTML page.

This is very attractive to 3D graphics developers for several reasons; the web browser is an ubiquitous interface available to almost every operating system, the programmer can make use of other web page elements as part of an interactive user interface, and we have a number of high-level communications tools at our disposal such as AJAX (Asynchronous JavaScript and XML). The 3D graphics software interface to WebGL is written in JavaScript, which allows the graphics software to use the DOM (Document Object Model) to manipulate the web page, or the visualisation to be manipulated by standard web form controls. Programming a user interface with user-input forms and text is a considerable development task in 3D programming, but with WebGL we can use web-page elements to do this with minimal effort. This combination of web tools opens 3D visualisation to a much wider range of programmers and web developers, who might not necessarily have had the low-level programming background traditionally required, providing a flexible and extensible solution to 3D visualisation.

An additional benefit to using a web-based visualisation interface is the profusion of ready-made distribution and user interface mechanisms available to high-level web languages. This can help us to de-couple our simulation model from the visualisation; essentially the end-user can be anywhere in the world, and use a web device to transparently query a centralised data-base, which tells the visualisation what to display. The web device relays this information to its GPU, which does all of the computationally-intensive graphics transformations, and returns the final 2D image to the web canvas. This separation of concerns is analogous to the model-view-controller design pattern - the "thin client" concept.

3 SEMI-AUTOMATED 2D LAYOUT RECOGNITION FOR BUILDING SPECIFIC ONTOLOGY POPULATION

3.1 Introduction

As stated in the DoW and project objectives, the building specific ontology needs to be populated with semantic information extracted from 2D CAD layouts, the geometric description of the building. Here is the place to highlight that the extraction of semantic information is based on input from the generic building ontology.

This bottom-up approach for identifying the building specific element classes, such as rooms, windows, etc. uses the 2D layout plan (AutoCAD drawings) contributed to all building demonstration objects. Geometrical information like positions, dimensions and topological relations between objects are extracted and linked with semantic information like identification keys, names etc. and the data is populated in the KnoholEM-ontology. Pattern recognition algorithms filter identical objects in the CAD drawing and map the geometrical properties, greatly accelerating the whole population process. This facilitates the flexible creation and manipulation of ontology individuals.

In our work we used the real world data from the Forum Building, BlueNet and Media TIC to validate the implementation and optimize the recognition process. The resulting tool “OntoCAD” combines user input with the pattern matching methods. The methodology tries to be as generic as possible, allowing various types of use-cases. The user will either be the facility manager or someone who knows the building or will interpret accurately the CAD drawing. Additionally some properties like room names or room numbers which are not explicitly depicted on the CAD layouts can be set using the tools graphical user interface.

The initial idea to use fully automated pattern matching was soon rejected as not practicable. Every drawing is different in its conventions; it might even not be consistent in itself because it is essentially user input. Thus we choose to focus on a semi-automatic approach, enabling the user to select a small set of primitives in the drawing and automatically segment and classify all identical others as logical objects like furniture. Geometrical properties contained in the drawing are the position and the bounding box of the objects. The class property is defined by the user. The implemented tool – OntoCAD allows a fast and flexible way to populate an ontology providing read and write capabilities for OWL with RDF model and a viewer for AutoCAD data. The interaction consists of a selection tool and a simple form to add single or groups of

individuals to the ontology. Another type of information are the zones and rooms which can easily be populated with the polygon selection tool.

The need for more semantic information has also been addressed in the recent versions of AutoCAD and the introduction of file formats like IFC and GbXML. The reason we cannot take advantage of those new features is that they will take effect in at least 10 years, because of their slow market uptake and low use [29]. Until then the present data for recent buildings consists of CAD drawings, and thus mostly primitives in 2D like points, lines, arcs, ellipses and circles with no semantic information whatsoever. This approach will also be practicable even when new standards emerge as import from the new formats like IFC is already allowed.

3.2 Requirements and Interfaces

The data collection from all demonstration objects has to be inspected in order to elaborate the methodology of geometrical data extraction. Most of the available data is in plain text or Spreadsheet form, 2D AutoCAD layouts and/or 3D geometries like SketchUp models.

3.2.1 Ontology requirements

When populating the KnoholEM generic ontology with CAD Layout data it essentially means to populate the “BuildingElement” class and its subclasses, enriched with geometric and object properties. The generic ontology must satisfy requirements regarding the definition of data and object properties. To map those properties to classes they have to specify the domains. This means that a property contains the information to which classes it belongs.
The building specific ontology, enriched with the layout data resulting from our methodology, is the basis for the data mining algorithms.

3.2.2 CAD construction drawings

There are many challenges when extracting semantic geometrical information directly from CAD drawings. CAD is one of the easiest and oldest technologies used in the industry [30]. At the same time CAD is the least effective technology when it comes to accomplishing building information modelling because it demands a great amount of effort. Recent research shows evidence that it does not ensure high quality, reliable, and coordinated information that the higher level of BIM produces [30]. There are many software products for CAD on the market used for drawing building layouts, these tools have versions and variants resulting in many differences and incompatibilities between them, different interfaces and import/export file formats.

One issue is the varying representation of the building elements in such a drawing. Often the quality of the information is fully dependant on the person who is inputting the data [30]. Fig. 3 shows some examples of different representation possibilities. It differs in the degree of information density and level of representation depth. Some authors create a legend to their drawings, which can help pattern matching (recognition) algorithms.

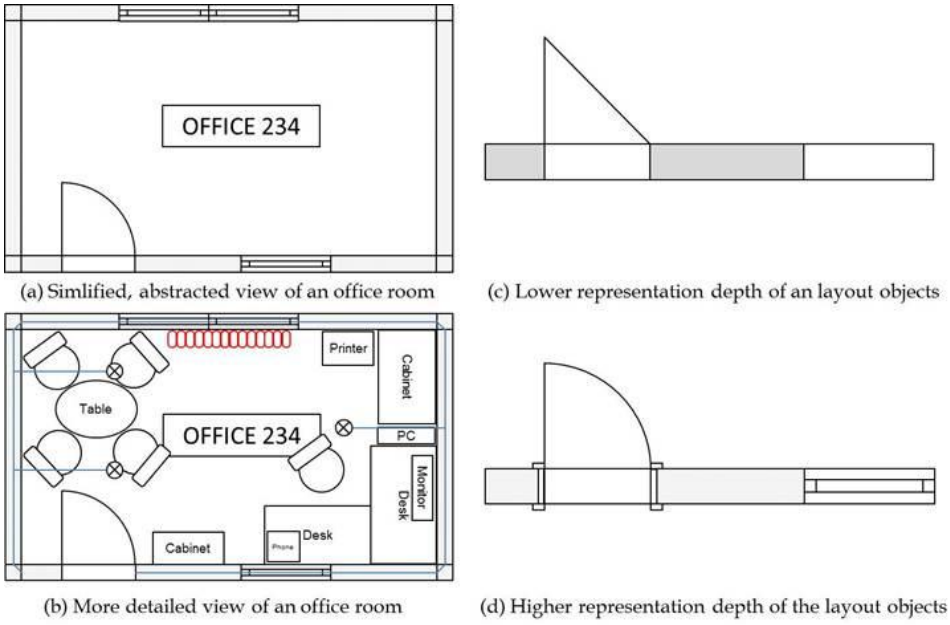


Figure 4: Possible representation of a CAD drawing

Another problem of the automated recognition of building elements from CAD drawings are the different languages used. Each author labels his drawing using a particular language, depending on the country. The need for translation increases the complexity of Optical Character Recognition (OCR) algorithms [39].

Fig. 4 presents examples of CAD drawings from one of the KnoholEM demo objects – the FORUM building in Eersel. Elements are grouped by colored layers. This layer information is only useful if there is a logical and consistent separation between them (like all walls or all furniture are in separate layer etc.), which is often not the case.
In addition, CAD drawings are often made for different perspectives of the building - for instance the front projection with all the floors of the building or the top view from a single floor. A recognition program has then to distinguish all perspectives. Other kinds of CAD plans are block schemes for building domains such as ventilation, heating, access controls, photovoltaic, electric circuit etc.

To extract the semantic information it is important to be able to use all layouts of the building (all floors and all perspectives) in order to avoid information loss. For example if we use only the CAD drawing for the ground floor, there will be no information for the height of the atrium.

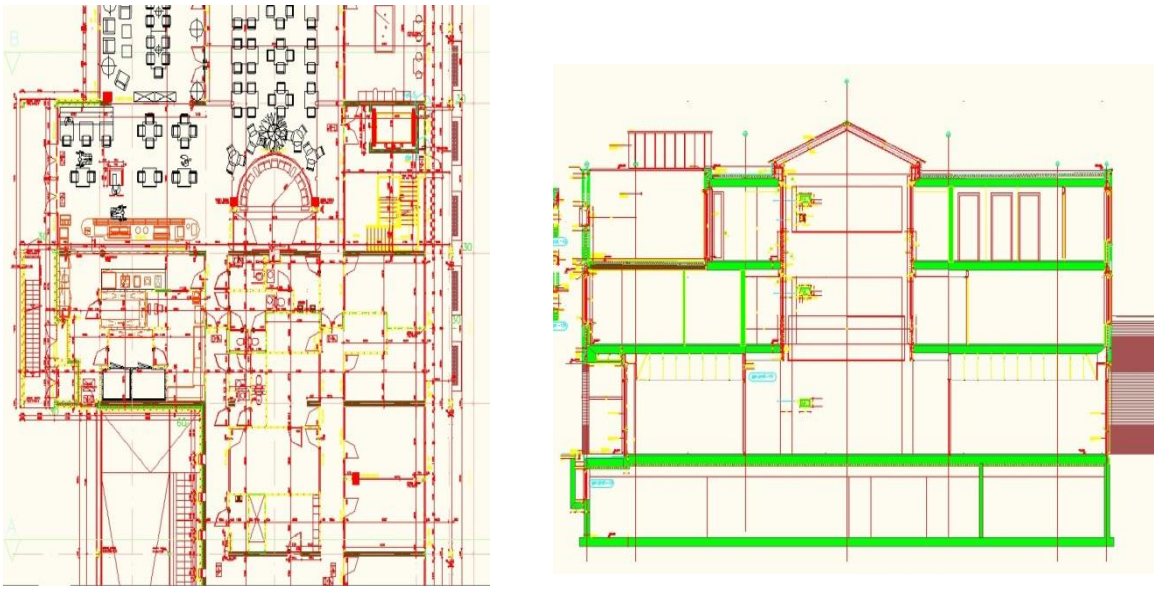


Figure 4: Ground floor and side perspective - CAD plans of the FORUM building in Eersel

Our experience with IFC files revealed flaws in the export from Autodesk software. We used different versions of AutoCAD to export IFC from the available DWG files. According to our expectations, the primitives were included in the IFC files. We wrote a parser and visualized the data and the most important issue was that primitives like circles, arcs and ellipses had been converted to lines when exporting from AutoCAD. This caused an exploding number of primitives which made all further pattern matching algorithms very slow. This kind of issues might get fixed in future versions of Autodesk software. Examples of OntoCAD with imported IFC file can be seen in Appendix B, Figure 22. The layouts of KnoholEM demonstration buildings have been drawn since 2005 and therefore there is no semantic information in the exported IFC construction plans. Thus we will focus on the DXF exchange format. This does not exclude the future use of the IFC standard.

To summarize, the discussed problems show the very high complexity of a fully automated approach. This makes the interactive evaluation and validation by the user of the semi-automated results far more interesting.

3.3 Methodology

Our methodology is an approach for the semi-automated extraction of semantic information from AutoCAD drawings using some user input at different stages. The drawings are exported from Autodesk software using exchange formats like IFC or DXF, where the DXF format is the one which we favour. We then import the primitives in our tool OntoCAD from the exported construction layout files. The primitives are extracted and clustered in layers like they were in AutoCAD. This vector based data representation is the basis for the viewer and the pattern matching algorithms. The methodology of the workflow is depicted in the BPMN diagram on Figure 5.

An important user input at the beginning is the mapping of the ontology specific data and object properties with the OntoCAD functions, for instance the computation of the object position. The functions are described in more detail in the implementation section. The implemented pattern matching and classification algorithms recognise building elements based on user defined templates. The user selects an object that can directly be populated in the ontology or he can choose to search for similar objects and then populate all of them at once. He has the possibility to directly validate the result and if necessary apply some corrections to the results. The results are continuously and automatically saved to the ontology file.

The advantage of a user centred method is that the user can provide meta information about the data like drawing type (top or side perspective), supported by an intuitive graphical user interface. The user can impact the performance of pattern matching algorithms by selecting and discarding the layers of the drawing which are not relevant. This helps the recognition algorithms to perform better, yield good results and work faster.

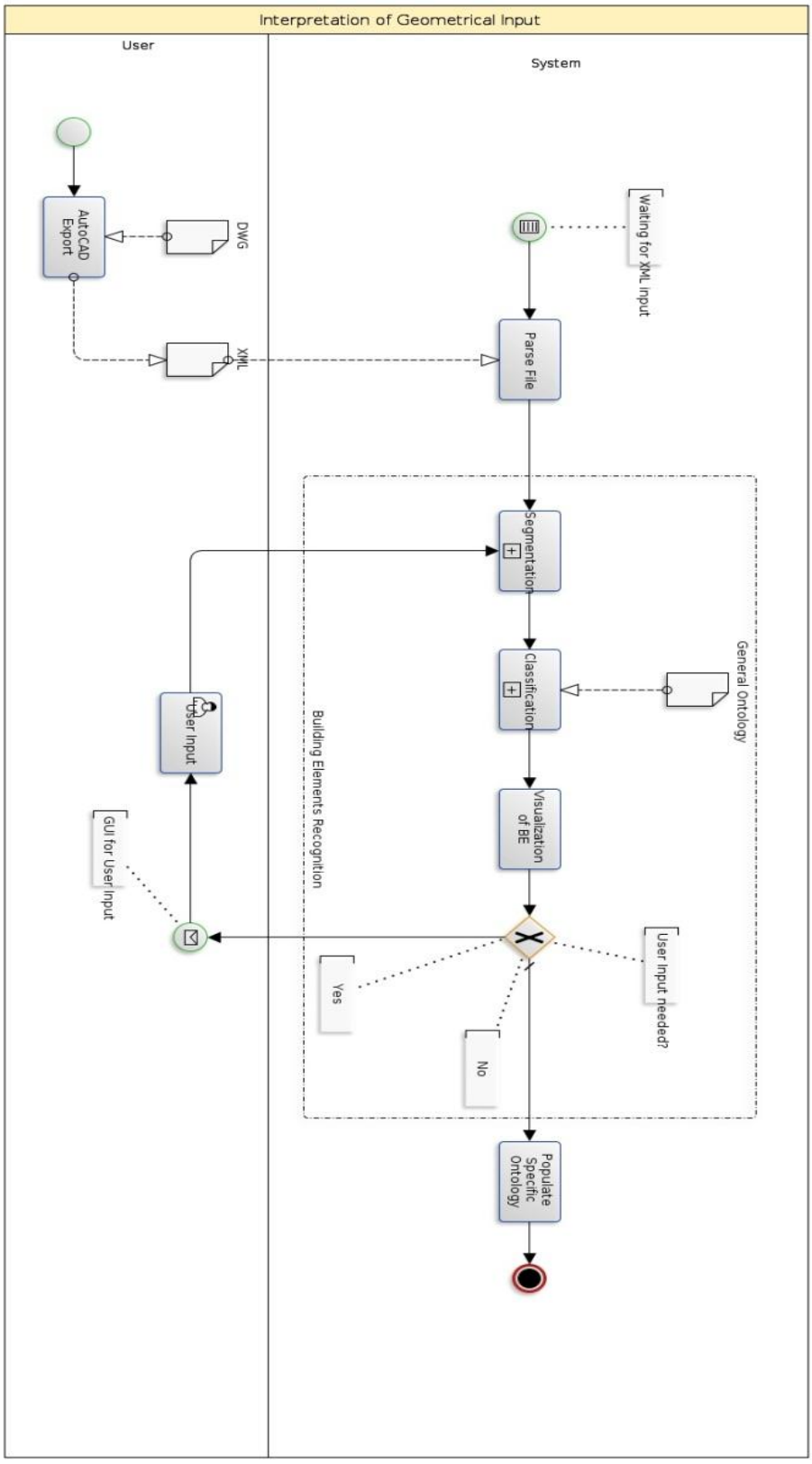


Figure 5: BPMN Diagram of the methodology for the CAD data segmentation and ontology population

3.4 Implementation of the OntoCAD Tool

OntoCAD relies only on Python and GTKs bindings PyGTK. This makes the tool highly portable. The implementation work has been done under the operating system Ubuntu 12.04. Later on the tool was ported to Windows 7. The software modules are as follows (see Figure 6):

- PyGTK Viewer for CAD data with a polygon selection tool
- CAD parser
- OWL parser
- Pattern matching based on primitives
- GUI Form to populate the ontology
- Module for automated population

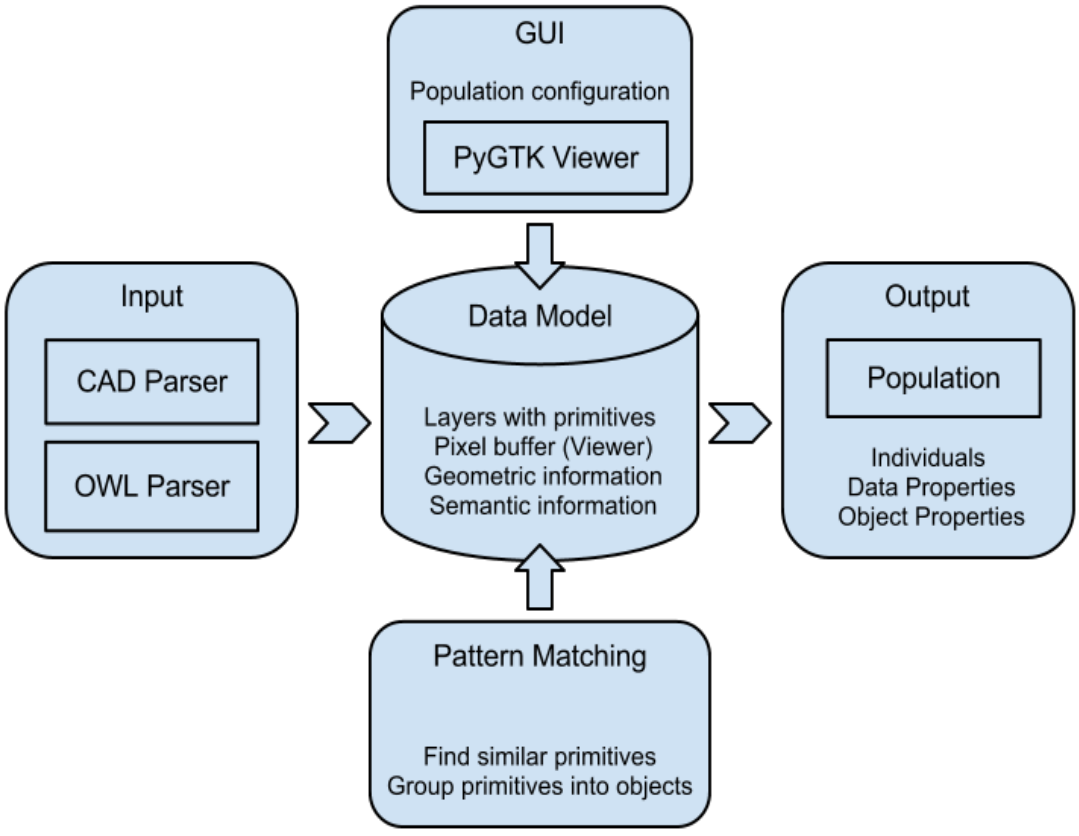


Figure 6: Software Architecture of OntoCAD

3.4.1 Input

3.4.1.1 CAD Parser

OntoCAD only uses the primitives of the CAD drawing to visualize it. Other information like metadata or complex data structures are discarded. Three file formats are supported for now, an own format based on XML, the IFC standard, and the DXF exchange format from Autodesk. From our experience with the export and import with those formats we focus now on the DXF import as this yields the best results with close to no information loss. Our DXF parser is partially based on the open implementation of Free-CAD. It is essentially a state machine that constructs the initial data hierarchy following the open DXF specifications. A second pass constructs the primitives. Then our implementation feed those to our viewer.

3.4.1.2 OWL Parser

In our work, we use OWL as the language to represent the ontologies. OntoCAD supports the import of OWL files with the RDF/XML format. The structure is essentially a very flat XML, with classes, data properties and individuals. We will not go into details of the RDF model, but to populate individuals with OntoCAD, it is important to know the available classes and the data properties for each class. All the parsed information are visualized in lists and trees.

3.4.1.3 Graphical User Interface (GUI) and User Input

The GUI consists of four main parts (see Figure 7), the Toolbar at the top, the layer dialog at the left, the viewer in the centre and at the right the GUI elements to manipulate the ontology. In the upper left corner are typical tools to open files, save, close and place for future tools. On the left side, all layers contained in the drawing file are listed. Each layer is visualized with different colours. Selecting a layer will highlight it in the viewer (see Figure 17, Appendix B). The checkbox can toggle the visibility of the layer.

The viewer in the centre allows panning and zooming into the data. We choose to set the navigation bindings on the scroll wheel and button of the mouse. The right mouse button appends a selection point to the polygon selection tool; a left double click closes the selection.

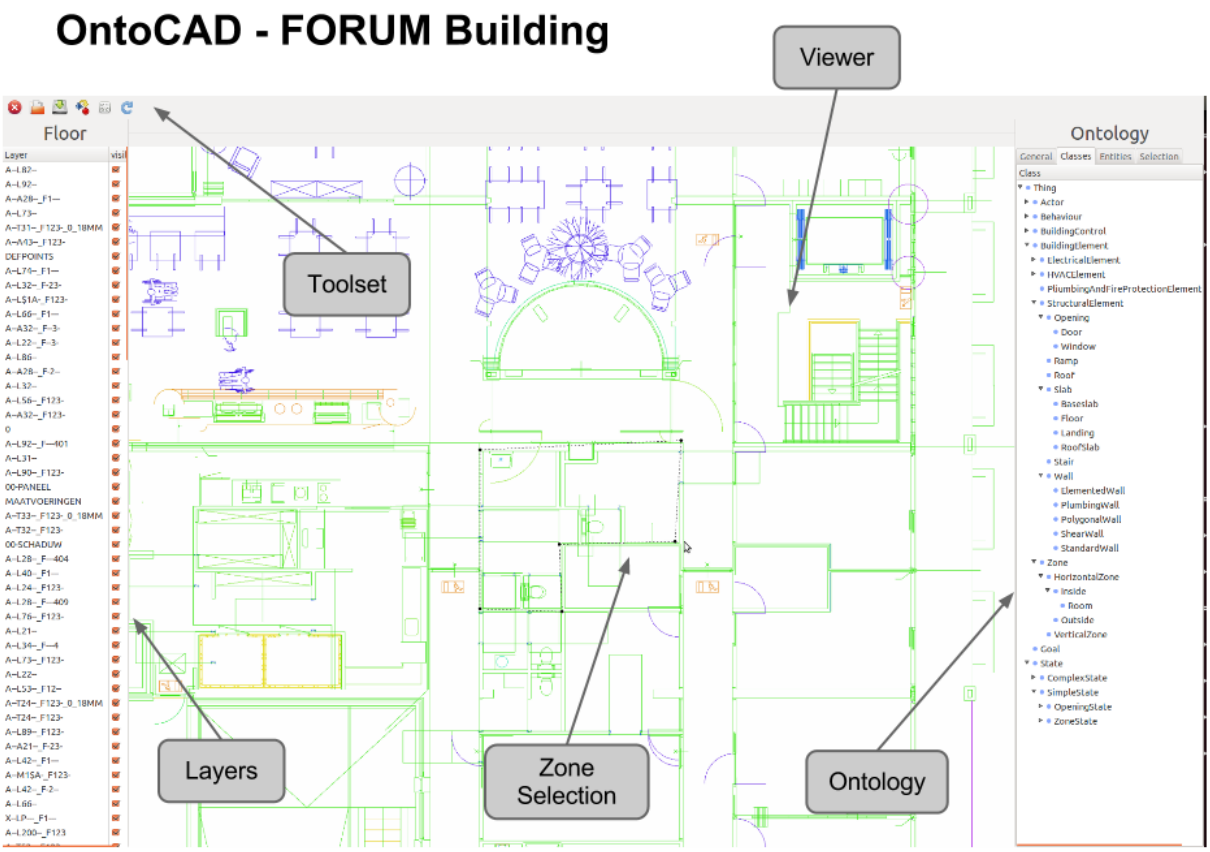


Figure 7: Graphical user interface of OntoCAD

The right side panel allows the user to manipulate the ontology. The first Tab “Properties” is a mapping of the data properties to a function of OntoCAD (see Figure 17, Appendix B). The reason that this is necessary is that the data properties are arbitrary strings with no information on how to compute them and thus cannot be automatically derived. OntoCAD functions are for instance geometrical data like position, area, length, etc. whereas a data property from the ontology that describes a position could be ‘hasPosition’, ‘hasArea’, ‘hasXCoordinate’, etc. Thus it is important that the user maps those before starting to populate individuals. This configuration is done only once.

Once the data properties have been configured one can start to populate the ontology. The “Entity” tab (Figure 19 and Figure 20) contains a dropdown menu with all available classes, a button to add individuals and a list of the data properties of the currently selected class. If a data property has been mapped to “User input”, then a text entry will appear next to it. The name of the individual is generally such a case.

When populating two cases are distinguished:

- Zones or rooms where only the polygon from the selection tool enters. There OntoCAD can provide the selection area, the polygon vertices, or the bounding box of the selection.
- Objects or other small sets of primitives that make up logical entities. In such a case it is often interesting to search for all duplicates in the drawing to populate all instances at once, for instance lamps or doors. In that case OntoCAD can provide the bounding boxes, the position, the width or the length.

That means we must first select for instance the room outline, then chose the “room” class form the drop down menu, then all data properties for the class “room” are visualized. For example additional information about the room type, room number and room label can be added by the user and after clicking on the button “Add”, an individual with a unique name and GUID is written in the ontology. Another tab “Ontology” visualizes the ontology class tree (see Figure 18). This helps the user to choose the right granulation for example if door is selected it can be populated as an “Opening” class or as “Door” class, using the tree hierarchy the user can see that class “Door” is under class “Opening”. The tab “Selection” shows the geometric data of the selection like the vertex list and area in real time.

The central viewer is the one of the more complex software modules. The whole methodology derives from the users experience and visual recognition of building elements. The viewer is pure PyGTK to avoid the loss of portability of the tool. There are open libraries available with limited functionality to draw and navigate primitives with PyGTK, but none was very convincing. Most development effort of the viewer went into the interface with the CAD data importer.

3.4.2 Data Model

The internal data model of OntoCAD consists of the CAD primitives, the ontology, the selection and pattern matching results. The five CAD primitives, points, lines, circles, ellipses and arcs, are organized by layers. The list of all layers is always visible for the user to quickly toggle the visibility of single layers. Pattern matching will only perform on visible layers. Further improvement to the data model could be an optimization of the structure of the CAD primitives. OntoCAD keeps a list of classes parsed from the ontology OWL file. This is important for the user to know the available classes of individuals but also to understand the structure of the OWL file. Also the list of individuals is present.

The polygon selection tool is a list of points, a polyline that is open until the selection is finalized, and then it closes. Every change to the selection computes the area of selection.

OntoCAD can compute certain geometric properties of the selected subset (see real-time calculation, “Selection” tab on Figure 21). Those so called OntoCAD functions are for instance the length and width of the group of primitive, the area, or the world position. When populating a new individual the data properties defined in the ontology for the chosen class need to be given. Only some of those may be calculated automatically using the OntoCAD functions, others have to be specified by the user like names and room numbers for instance. It is important to map the data properties to the OntoCAD functions before starting to populate. There is a simple form for this purpose described in this document with the GUI.

OntoCAD calculates the object properties after each populated building element and writes it automatically in the ontology. This is the semantic information about object affiliation, for instance a ‘door_x’ belongs to ‘room_x’ and the other way around. An algorithm for generating and saving a GUID for each individual is also implemented.

3.5 Module for Pattern Matching

3.5.1 Pixel Based approach

First we made some template matching based on image recognition used the OpenCV library [28]. We used threshold value to find not only exactly the same image but similar images. There are a lot of problems with this approach because of overlapping or intrusion of primitives and colours, bringing additional pixels which cannot be recognized as similar images. Some solution will be searching similar images for particular colour or layer. Another bigger problem is that finding rotated images cannot be efficiently implemented. It will be time consuming algorithm testing for all rotations. Summarized, pattern matching algorithms working on pixels in pictures or video streams have some disadvantages:

- Not rotation invariant
- Bad performance due to the 1D nature of most primitives, a lot of background color
- Very sensitive to resolution
- Very sensitive to overlaying primitives and other informations

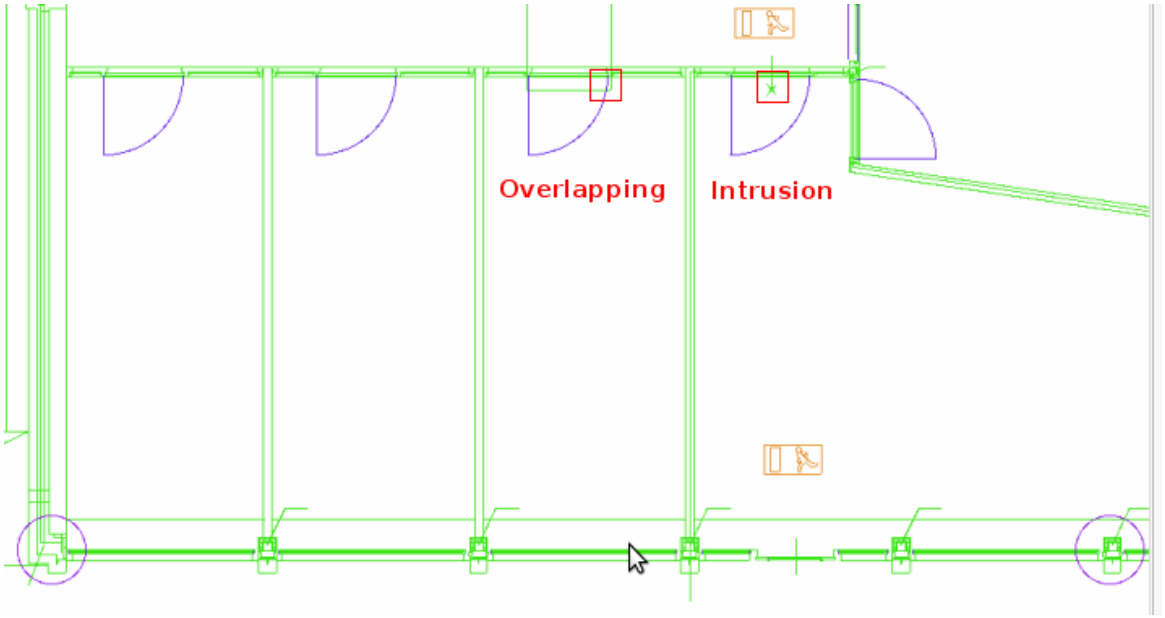


Figure 8: Disadvantages of pixel based pattern matching

3.5.2 Primitives Based approach

The aim is to get as much as possible semantic information about a building from its 2D layout and this as much automatically as possible. Our second approach consists of grouping primitives into objects based on a user defined selection of primitives. The user has the possibility to use a polygon selection tool to select all the primitives that belong to a logical object. The targeted objects are recurrent ones, this applies mostly to furniture like a table, chair, or any other object that has been inserted multiple times.

Our approach uses the primitives to find similar objects. We achieve this through the following steps:

- gather a subset with primitives similar to the selected ones
- compute the Set of relations between the selected primitives
- group the similar primitives into objects based on the above set of relations

Subset of similar primitives

The initial step is to take a subset of all the primitives using criteria that define two primitives as similar. We distinguish between rotation invariant and scale invariant criteria. For our purpose it is

interesting to have rotation invariance but to avoid scale invariance as this corresponds to identical objects which can be rotated. Our criteria for lines is the square length, for circles and arcs it is the square length of the radius and for ellipses the square length of the min and max radius.

Relations between primitives

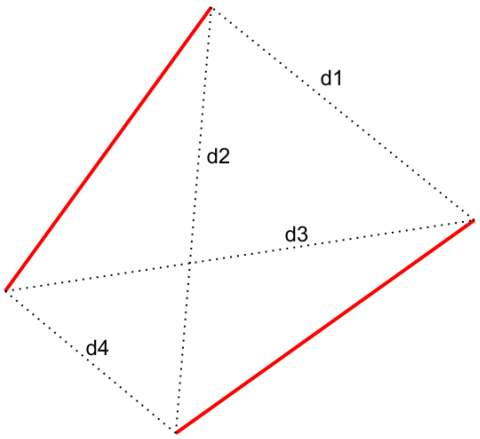


Figure 9: Similarity criterion

To compare patterns of primitives it is important to define how to describe a pattern with primitives. For this purpose we introduce relations between the different types of primitives. The first and most important relation is the segment to segment relation. As for the similarity criterion it is important to have rotation invariance but no scale invariance. We define the segment to segment relation as the set of 4 distances between the segment endings as illustrated on Figure 9.

We will continue further only considering the segment to segment relation.

Grouping of primitives

The last step is to group the primitives into objects. The constraint is to match the selected set of primitives. Here we need to define what we consider being a match. First we need to have the same number of primitives, and second we need to find all relations that also exist in the reference set. A use case for finding similar chairs, even rotated, can be seen on Figure 20.

3.5.3 Output

After the user configures the data properties and selects the building elements, OntoCAD mapped them to classes. In the background there is a module that prepares the values for the individuals, data and object properties. Using the XML library and the RDF scheme, new ontology (.owl file) is generated. This can be opened and further edited with other ontology tools (for instance Protégé).

3.6 Conclusion

The created prototype has great potential as an extraction tool for geometric and semantic information from layout drawings. It accelerates the population process and can be used in further research activities. Improvements of the importer and viewer can be done, so that more exchange formats are supported. Using the full spectrum of manual and automated tests in means of software engineering will improve the graphical user interface and the usability of the prototype. Significant results on the automated recognition of related objects and properties between them have been also achieved.

4 BUILDING BEHAVIOUR AND BUILDING SPECIFIC ONTOLOGY POPULATION

Energy use during operation is strongly influenced by the operation and utilization of the different spaces [4] and the behaviours of occupants [5]. Therefore another key requirement of the KnoholEM solution is to provide tools for modelling how users interact with the building and for providing feedback on the energy consumption of users of the building to the facility manager. This section presents the user `Behaviour` class of the specific ontologies and a method for building occupants to model their activities and populate the building specific ontology through an accessible web based visualisation interface. We first give a brief recap of the use case which this section set outs to address and which was defined in deliverable 2.1.

4.1 Use Case

As discussed previously, IFC has yet to make a significant impact in the AEC communities [13]. This issue is routed in the manner in which different vendors implement the IFC model to their specific requirements, which has often resulted in data exchanges between tools resulting in imprecise or lost data [28]. To remedy this situation, the National Building Information Model Standard Committee (NBIMS) has developed IDM.

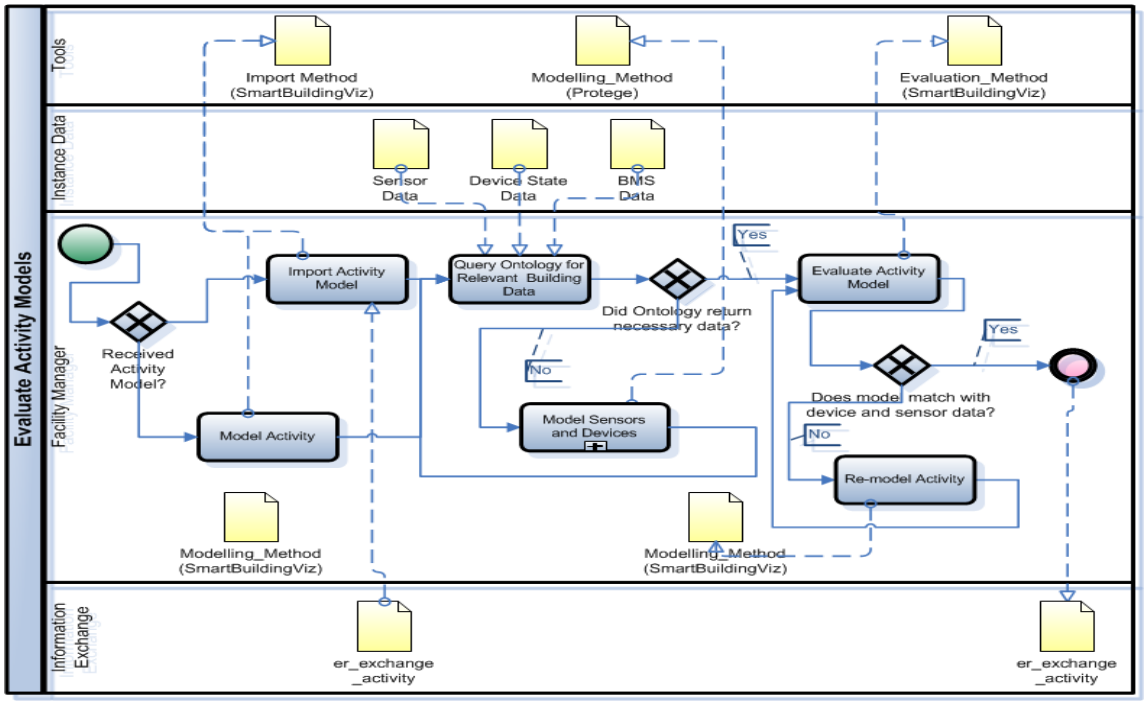


Figure 10: IDM Use Case - Evaluate Activity Model

The BPMN diagram which represents this use case is split into several swim lanes which represent the activities which take place within the particular use case (blue bubbles). Each of the activities can be themselves broken down into its own “use case” allowing complex tasks to be represented in ever greater detail. In Figure 10 there is one actor with several activities (the FM). The swim lanes are also used to indicate information exchanges. An activity can require information from outside that use case or result in some new data which can then be used in other use cases. In Figure 10 we see that an activity model information exchange can either be imported from an external use case, or exported. Finally, the swimlanes also indicate data which is required from the different building systems and also the different tools being used during the use case.

The use case begins with the FM receiving an activity model from a user from another use case (not presented here) which describes how the building occupant models their activities on a monthly basis for a period of one week. This process is repeated for six to twelve months to build up a model of their building use as an input into the energy simulation software and to enable energy efficient rule generation.

4.2 SmartBuildVis: Tool for Populating Ontology with Behaviour Modelling

To support better energy prediction, a method for user configured activity models has the potential to provide more accurate simulation models, as well as improving predictive automation. This requires a method for visualising the building geometry, here we present our findings with respet to visualising the building geometry.

4.2.1 Building Geometry Visualisation

As presented in the previous sections there are a number of tasks related to extracting relevant information from the different sources of data (e.g. 2D CAD drawings) to execute the KnoHolEM solution. This section examines issues related to building visualisation to support visualisation of building energy consumption and also the process of populating the building specific ontology with user activities. In the use cases identified in deliverable 2.1 which involve non-technical users’ such as building occupant, a 3D view of the building to aide their landmark recognition and spatial awareness of the building is required. As IFC is not yet common, a method is required to rapidly construct a 3D graphics model for visualisation. There are three basic options to meet this requirement:

1. Creating a 3D extrusion by hand, from an image of the floor-plan, in 3D modelling software

2. Parsing IFC files, assembling the 2D geometry, extruding this to height automatically, and sub-dividing this into units which can be rendered.

3. Exporting an intermediate format (such as XML) from CAD software, parsing wall elements into 2D points, and extruding these into triangular surfaces.

The first approach requires employment of a skilled technical artist, and has the advantage of being able to adjust the model to remove visually un-important building geometry, as well as correcting any changes to the building, or annotating visual landmarks into the model. The second approach may be desirable if IFC files exist for all of the buildings in the project domain, but requires additional import-export tools to be constructed. IFC files contain a nested transformation hierarchy that we have found takes an unacceptable length of time, from a user experience point of view, to parse and re-construct into the format of mesh required for web visualisation. It is feasible, however, to treat IFC as an intermediate format, and pre-compute these into a more readily interpreted web format such as JSON (JavaScript Object Notation).

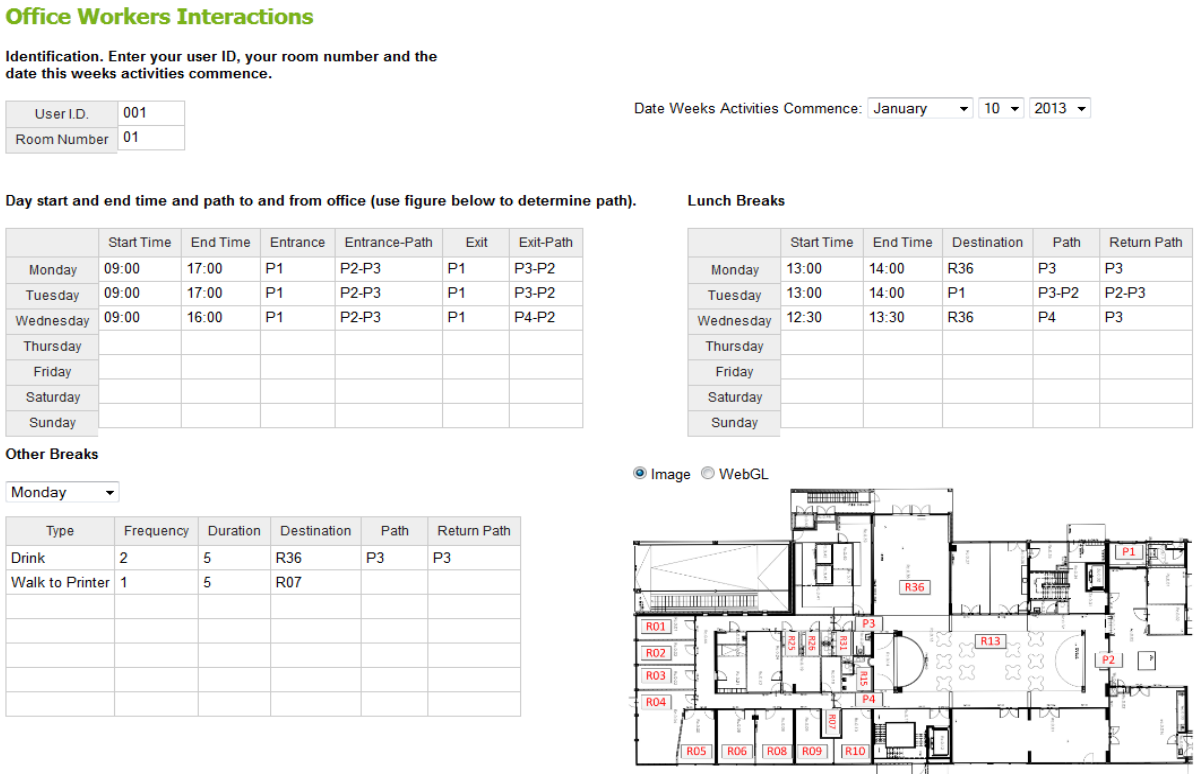
The third option, we have found to be more generally applicable to our test buildings, and are the least complex of the procedures. There must, however, be an existing 2D CAD model that can be exported into an additional XML format. We can quite quickly parse the XML with existing web tools such as AJAX (Asynchronous JavaScript and XML), and extrude the 2D points at wall ends into the 3D triangular surfaces required for rendering.

4.2.2 SmartBuildVis: Activity Modeller

This section describes the web-based interface for supporting occupant activity modelling (Figure 11). This is built upon activity models developed through analysis of the Forum building and preented in deliverable 2.1. The interface is implemented using jQuery, CSS, xHTML and WebGL. The tables make use of an open source library called “handsontables” and provide a rapid method for office workers to enter in activities on a weekly basis. Building occupants each have an activity zone assigned to them, which can be created through the web based interface (Figure 12). Occupants use the interface to populate the specific building ontology with start and end times of each working day, where and for how long they took their lunch break, as well as whether they had meetings and the location of those meetings.

Figure 13 shows an example of a skeleton activity in the ontology and the protégé tool [6]. The skeleton activity is an “OfficeWork” activity. This has a userID associated. Here it is described simply as “001” to ensure anonymity of the building occupant and to meet European ethical

requirements for conducting evaluations involving live participants [7] . Alternatively, a GUID can be generated using Base-64 character encoding mapping (see IFC2x3) or a URI may also be used.



Office Workers Interactions

Identification. Enter your user ID, your room number and the date this weeks activities commence.

User I.D.	001
Room Number	01

Date Weeks Activities Commence: January 10 2013

Day start and end time and path to and from office (use figure below to determine path).

	Start Time	End Time	Entrance	Entrance-Path	Exit	Exit-Path
Monday	09:00	17:00	P1	P2-P3	P1	P3-P2
Tuesday	09:00	17:00	P1	P2-P3	P1	P3-P2
Wednesday	09:00	16:00	P1	P2-P3	P1	P4-P2
Thursday						
Friday						
Saturday						
Sunday						

Lunch Breaks

	Start Time	End Time	Destination	Path	Return Path
Monday	13:00	14:00	R36	P3	P3
Tuesday	13:00	14:00	P1	P3-P2	P2-P3
Wednesday	12:30	13:30	R36	P4	P3
Thursday					
Friday					
Saturday					
Sunday					

Other Breaks

Monday

Type	Frequency	Duration	Destination	Path	Return Path
Drink	2	5	R36	P3	P3
Walk to Printer	1	5	R07		

Image WebGL

Figure 11: Web Interface to collect activity data for building occupancy energy simulation

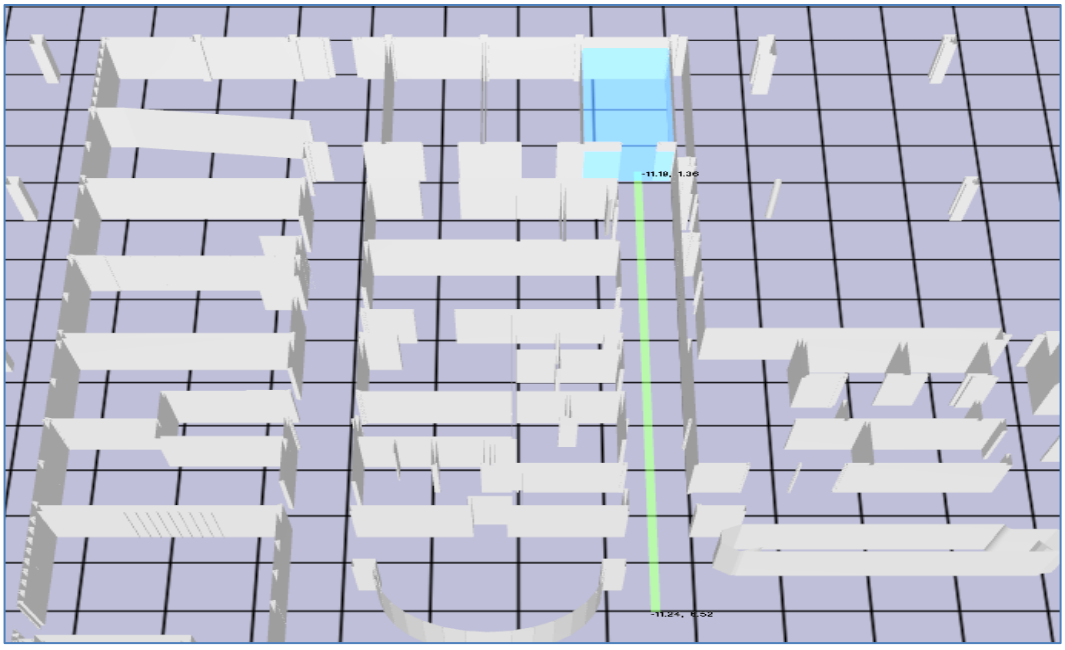


Figure 12: WebGL Visualisation for Modelling Activity Zones (blue) and Paths (green)

The activity also has a location associated with it, for example a room. Alternatively a zone may be associated, so that rooms can be sub divided into activity zones, for example in open layout offices with multiple desks. The unique id, the start date (hasStartDate) and its time and duration can be used to uniquely identify a skeleton activity (as it is assumed no two skeleton activities for a particular occupant overlap). The start time (hasStartTime) indicates its start time and with the duration (hasDuration) this can be used to calculate the end time of the activity. The user can also model intermediate activities, like how often they took breaks like getting coffee, or taking a walk and the duration of time they took to complete these activities.

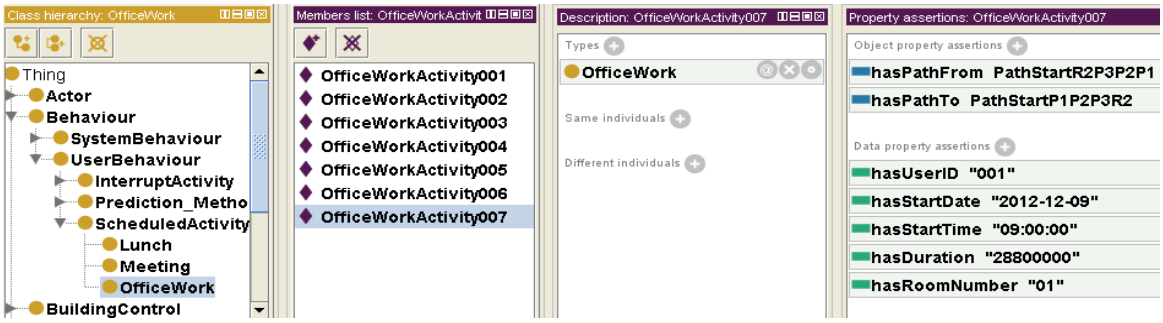


Figure 13: Skeleton Activity - Office Work

Figure 14: Intermediate Activity - Coffee Break

This method supports building up an activity model for each occupant of both skeleton and intermediate activities. Additional information can be supplied regarding any equipment interactions, for example, how long they had the light on in their office, how long the window was open, blinds were raised or lowered, whether the lighting was on or off and what settings the heating or air conditioning were set to. The interface also supports the user defining their path in the building. In Figure 11 this is done using a 2D image of the building which has been annotated with symbolic locations P1 – P4, each of which has motion sensors to detect presence. P2 – P4 have motion detectors on either side of the door, so it is also possible to determine direction, although when the areas become crowded the accuracy of this prediction decreases. The user

can choose which path they take to their office (R1- R10) from the entrance (P1). A path is modelled as a start node which has a number of subsequent nodes which end with a final end node. In the above example path is modelled as uni-directional (i.e. there is no relationship hasPreviousPath). This was a design decision made to support selecting any node along the path to become a start node. This can be extended to provide bi-directional paths where required.

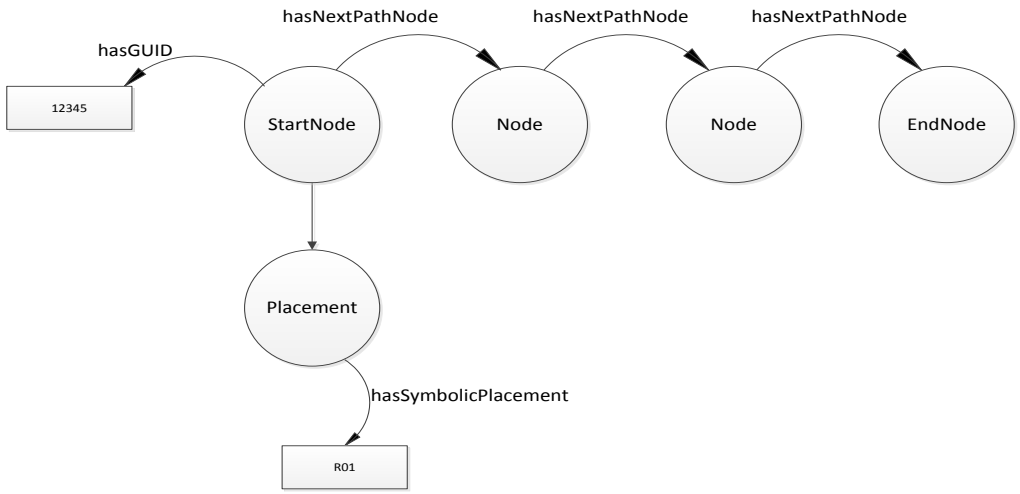


Figure 15: Describing Paths in Ontology

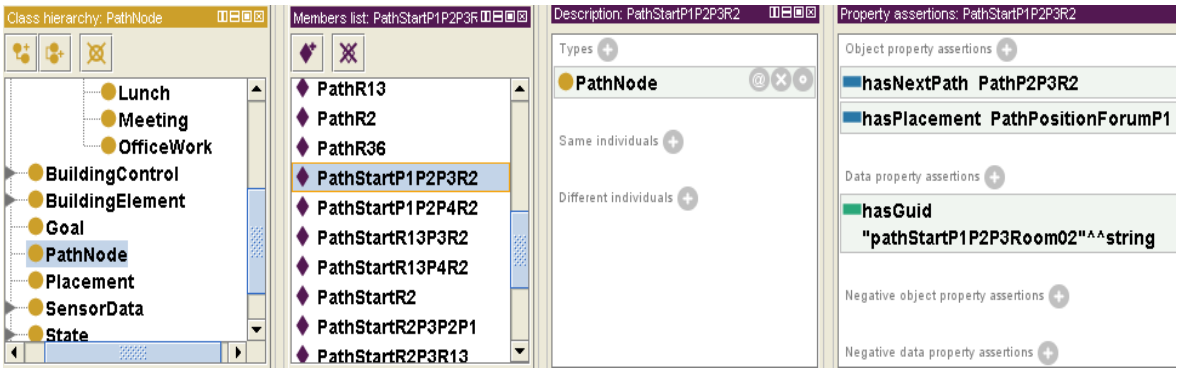


Figure: 16: Path Start Node in Ontology

In addition to the 2D image (Figure 11), the user can also select the WebGL interface (Figure 12). This is still under integration, but will support the user defining zones, for example a zone which covers their desk, and assigning an activity to it, e.g. office work. They will also be able to define their own path through the building through a point and click interface and thus meet the requirements for fault and energy wastage reporting (as reported in DR2.1). The ontology is currently stored in a Fuseki server and the interface communicates over HTTP using SPARQL queries.

4.2.3 Conclusions

This section presented a web-based tool to populate the building specific ontology with behaviour data from occupants. By taking a web-based approach the solution is made accessible to a wide range of users across different devices in any of the building demonstration objects. This makes it available to a large number of potential users, including building occupants and facility managers. Also, by ensuring that the underlying code supports all necessary visualisation while remaining optimised, further ensures that the solution will run smoothly on mobile and commodity devices with limited graphical rendering capabilities.

We also presented a selection of use cases which the solution addresses, demonstrating the flexibility of the approach to visualising, configuring and enhancing the BIM, in particular in the important area of occupant activity modelling and its impact on building energy consumption. As standards like IFC are as yet underdeveloped in respect to areas like energy, sensor and activity modelling, the solution also makes use of ontologies to help bridge the gap between IFC and further support flexible, extensible growth of the solution. By developing the ontological models using the information delivery manual (IDM), sharing and reviewing outside the project is supported, as part of a standardised process for extending BIM.

5 OVERALL CONCLUSION AND FUTURE WORK

This work should be seen in the context of the iterative development of the generic building ontology and the population of the specific building ontology. As such, the solutions proposed here will undergo continual development as new knowledge is garnered during the monitoring of the operational building objects. The KnoHolEM project has made significant progress towards developing the behavioural models and interfaces for the population of the building specific ontology to meet the requirements of the use cases defined in deliverable 2.1 and the DoW.

The development of the OntoCAD tool will continue in the context of searching for correlation with the visualisation process by providing automated semantic information on the connections between `BuildingElements` objects. Next important step is as well the application of the tool for the creation of the other two building specific ontologies (for the PICA and HHS buildings) and the improvement of the connection with other work packages.

SmartBuildVis will be extended to meet the various requirements of the identified stakeholders for populating the building specific ontology and the `Behaviour` objects. The inputs from the users on their behaviour will be analysed alongside data coming from the sensor deployments to decrease the uncertainty inherent in the process of determining user activities through automated means alone. This work will be done in conjunction with WP3.

The developed interfaces presented here will also undergo continuous usability testing as part of the iterative design and development of the behavioural models. These tests will determine the levels of usability of the implemented interfaces and determine whether they support the identified stakeholders (facility managers and building occupants) in achieving the tasks outlined previously. The methodology for conducting these tests is presented in Appendix B, and is originally presented by McGlinn et al [8]. Usability testing will begin with the first prototype SmartBuildVis solution (presented in Chapter 4) in the forum building in Eersel, and findings from these evaluations will feed into the next development cycle which will then be applied to all the building objects.

The methodologies presented in this deliverable are in the process of being deployed in three of the KnoholEM buildings - the Forum building, Bluenet and MediaTic. The developed methodology will continue undergoing an iterative design and development cycle, with its eventual deployment in the five building objects of the KnoholEM project. The data collected through its use will be used to inform the intelligent energy management solution, and the energy consumption of each

building will be evaluated against its current levels (as assessed through the comprehensive energy modelling and simulation being conducted). The goal of this solution is to contribute towards a 30 % reduction in the energy consumption of the buildings from existing levels.

6 REFERENCES

[1] "SketchUp BIM: Building Information Modeling in SketchUp." [Online]. Available: http://www.sketchupbim.com/. [Accessed: 21-Jan-2013].

[2] "Autodesk Revit Products." [Online]. Available: http://usa.autodesk.com/revit/. [Accessed: 21-Jan-2013].

[3] "ArchiCAD 16 - Overview." [Online]. Available: http://www.graphisoft.com/products/archicad/. [Accessed: 21-Jan-2013].

[4] P. Hoes, J. Hensen, M. Loomans, B. Devries, and D. Bourgeois, "User behavior in whole building simulation," *Energy and Buildings*, vol. 41, no. 3, pp. 295–302, Mar. 2009.

[5] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10*, p. 1, 2010.

[6] "The Protégé Ontology Editor and Knowledge Acquisition System." [Online]. Available: http://protege.stanford.edu/. [Accessed: 01-Mar-2013].

[7] "Find a document : Find a document." [Online]. Available: http://cordis.europa.eu/fp7/ethics_en.html. [Accessed: 01-Mar-2013].

[8] K. McGlinn, L. Hederman, and D. Lewis, "SimCon: A Context Simulator for Supporting Evaluation of Smart Building Applications when Faced with Uncertainty," *Pervasive and Mobile Computing*, vol. In Print, 2013.

[9] T. K. Landauer, *The trouble with computers: Usefulness, usability, and productivity*. MIT Press, 1995.

[10] T. Jokela, N. Iivari, J. Matero, and M. Karukka, "The Standard of User-Centered Design and the Standard Definition of Usability : Analyzing ISO 13407 against ISO," in *Proceeding CLIHC '03 Proceedings of the Latin American conference on Human-computer interaction Pages 53-60*, 2003, pp. 53–60.

[11] J. Nielsen and T. K. Landauer, "A mathematical model of the finding of usability problems," 1993, pp. 206–213.

[12] "ISO:9241-11: INTERNATIONAL Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability," vol. 1998. 1998.

[13] J. Scholtz, A. Wichansky, K. Butler, E. Morse, and S. Laskowski, "Quantifying Usability: The Industry Usability Reporting Project," *Proceedings of the Human Factors and Ergonomics Society Annual*, vol. 46, pp. 1930–1934, 2002.

[14] N. Bevan, "Common Industry Format Usability Tests," in *Proceedings of Usability Professionals Association Conference*, 1999, pp. 0–5.

[15] J. Sauro and E. Kindlund, "A Method to Standardize Usability Metrics Into a Single Score," pp. 401–409, 2005.

[16] J. Nielsen and T. Landauer, "A mathematical model of the finding of usability problems," *Proceedings of the INTERACT'93 and CHI'93 …*, pp. 206–213, 1993.

[17] J. Brooke, "SUS-A quick and dirty usability scale," *Usability evaluation in industry*, pp. 189–194, 1996.

[18] A. M. Lund, "Measuring Usability with the USE Questionnaire," *Usability Interface: The Usability SIG Newsletter of the Society for Technical Communications*, vol. 8, no. 2, p. 8, 2001.

[19] J. P. Chin, V. A. Diehl, and L. K. Norman, “Development of an instrument measuring user satisfaction of the human-computer interface,” in *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI ’88*, 1988, pp. 213–218.

[20] J. Kirakowski and M. Corbett, “SUMI: the Software Usability Measurement Inventory,” *British Journal of Educational Technology*, vol. 24, no. 3, pp. 210–212, Sep. 1993.

[21] T. S. Tullis and J. N. Stetson, “A comparison of questionnaires for assessing website usability,” in *Usability Professional Association Conference*, 2004, pp. 1–12.

[22] A. Bangor, P. Kortum, and J. Miller, “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale - International Journal of Usability Studies,” *Journal of Usability Studies*, vol. 4, no. 3, pp. 113–123, 2009.

[23] J. Nielsen, “Enhancing the explanatory power of usability heuristics,” in *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI ’94*, 1994, pp. 152–158.

[24] A. Woolrych and G. Cockton, “Why and when five test users aren’t enough,” *Proceedings of IHM-HCI 2001 conference*, 2001.

[28] OpenCV, http://opencv.org/, last review on the 22th of February 2013.

[29] Laakso, M., Kiviniemi, A.:The IFC Standard - A Review of History, Development, and Standardization. Journal of Information Technology in Construction, ISSN 1874-4753, May 2012.

[30] Vanlande, R., Nicolle, C., Cruz, C.: IFC and building lifecycle management, Automation in Construction, Volume 18, Issue 1, December 2008, Pages 70-78, ISSN 0926-5805.

[31] Autodesk, N.N: http://www.autodesk.de, last review on the 22th of February 2013.

[32] Open Design, N.N: http://www.opendesign.com/, last review on the 22th of February 2013.

[33] AutoCAD 2012 DFX Reference, Autodesc Inc., February 2011

[34] Ramos García L.: Ontological CAD Data Interoperability Framework. SEMAPRO 2010: The Fourth International Conference on Advances in Semantic Processing, 2010

[35] Venugopal, M., Eastman, C., Sacks, R., Teizer, J.: Improving the Robustness of Model Exchanges Using Product Modeling ‘Concepts’ for IFC Schema. International Workshop on Computing in Civil Engineering 2011, Miami, Florida, United States, June 19-22, 2011

[36] http://www.w3.org/TR/REC-xml/

[37] http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=773204

[38] Wicaksono H., Aleksandrov K., Rogalski S.: An Intelligent System for Improving Energy Efficiency in Building Using Ontology and Building Automation Systems. Book, "Automation", ISBN 978-953-51-0685-2, InTech, pp. 531- 548, 2012.

[39] Mori, S.; Suen, C.Y.; Yamamoto, K.: "Historical review of OCR research and development," *Proceedings of the IEEE*, vol.80, no.7, pp.1029-1058, Jul 1992

APPENDIX A

OntoCAD Screenshots:

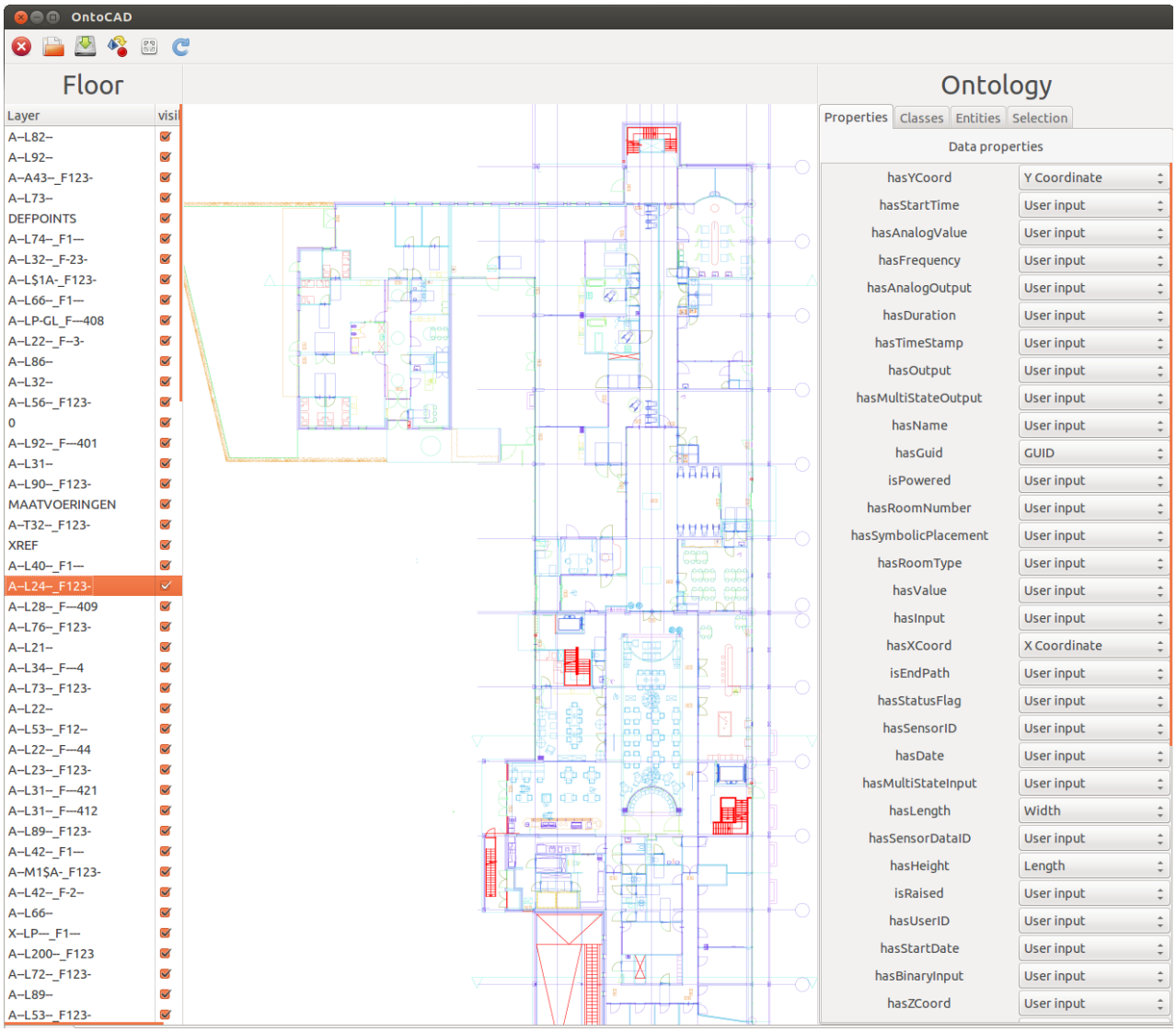


Figure 17: Forum Building CAD layout, OntoCAD layer selection and data properties configuration

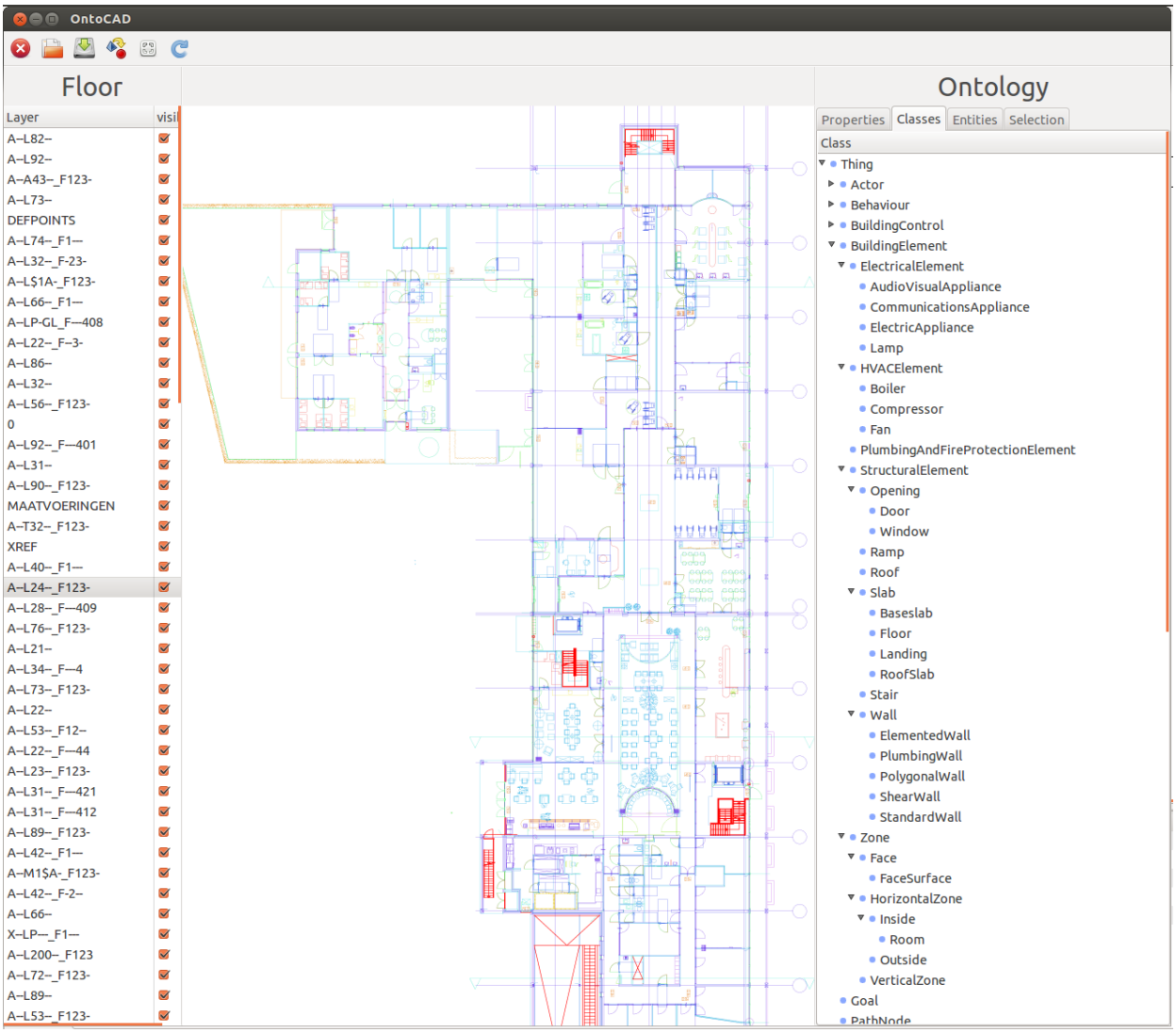


Figure 18: OntoCAD: Ontology tool, class tree (right side)

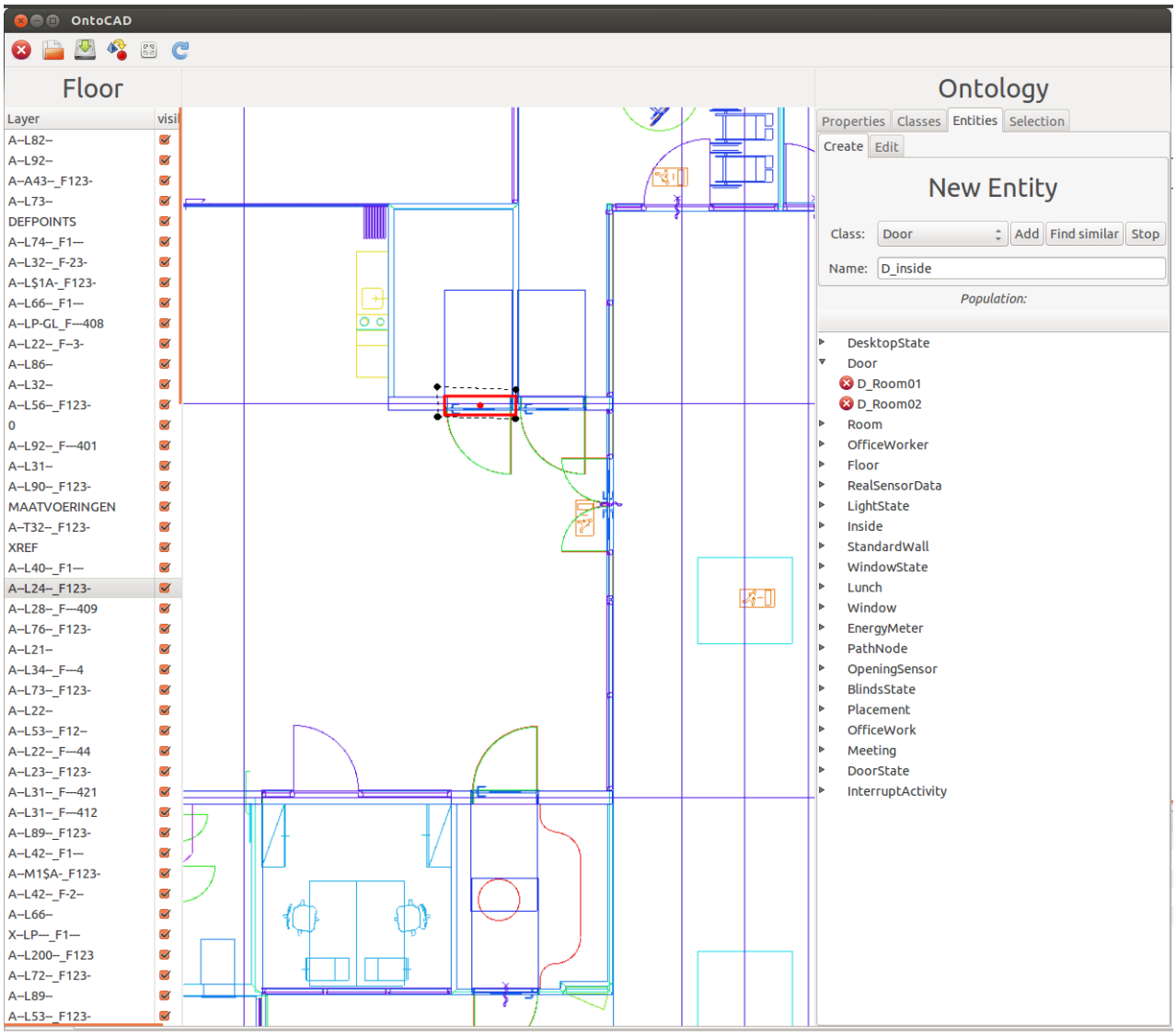


Figure 19: Slection of building element (door) and its population

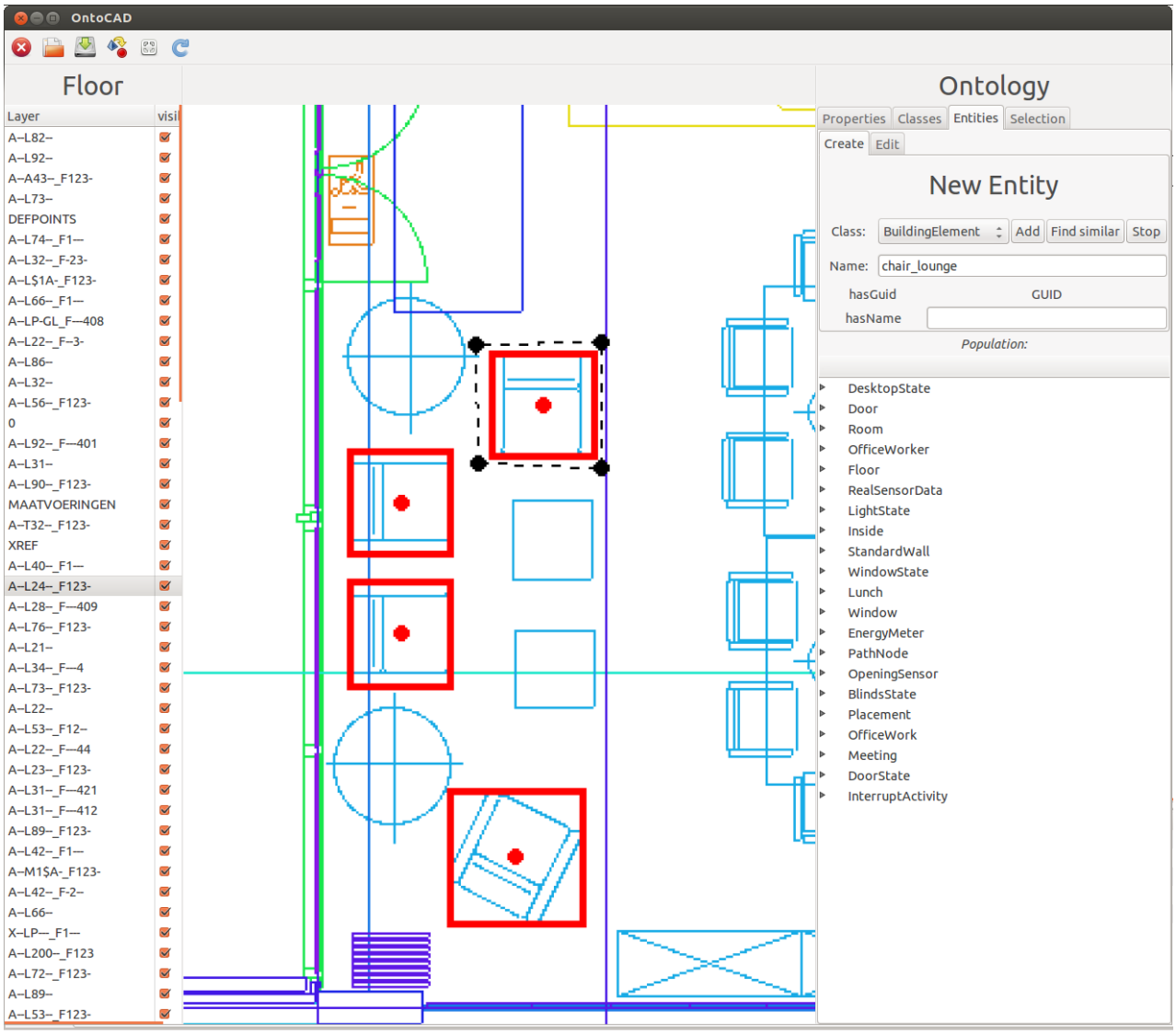


Figure 20: Pattern matching of selection

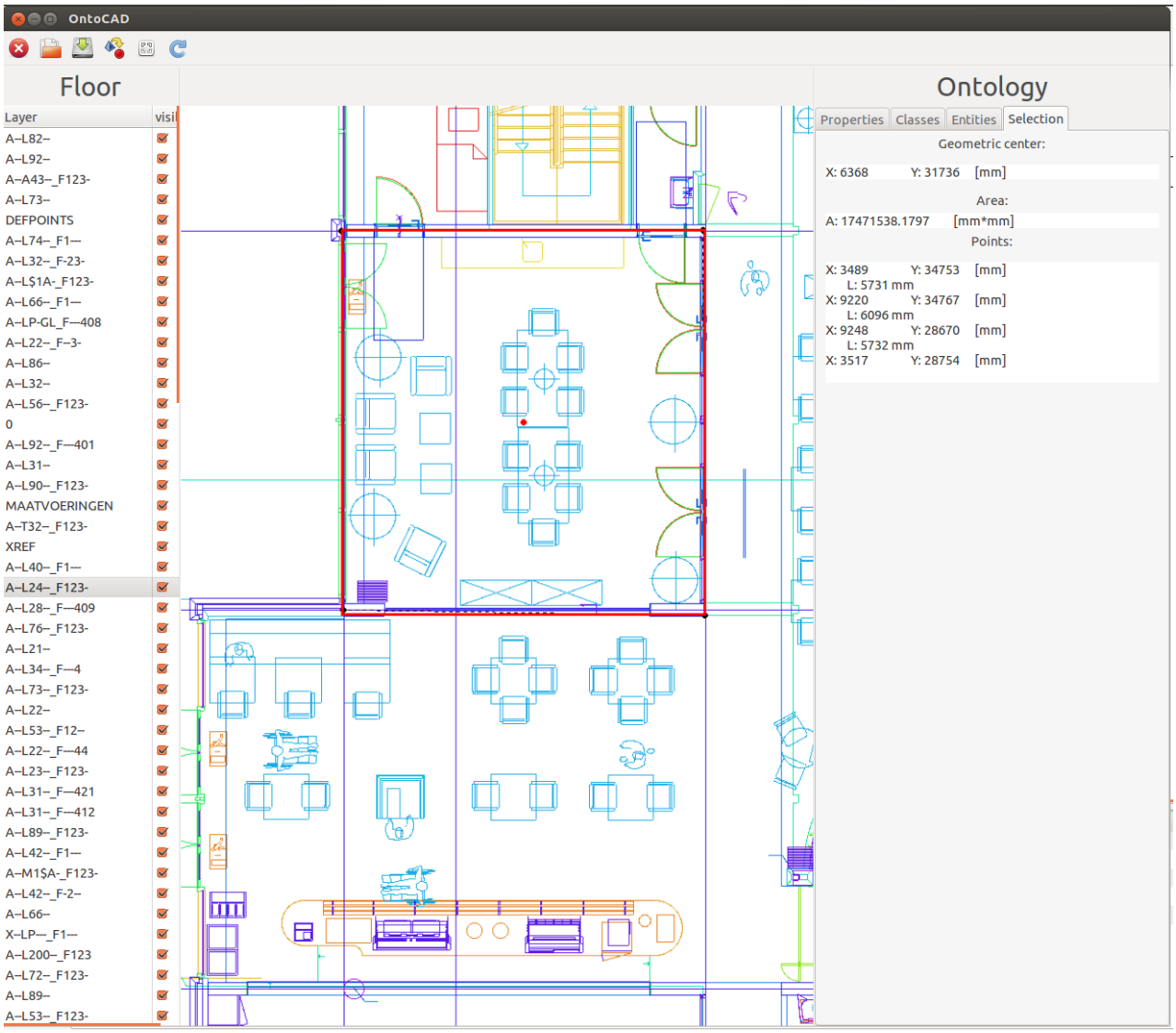


Figure 21: Room selection and geometric properties

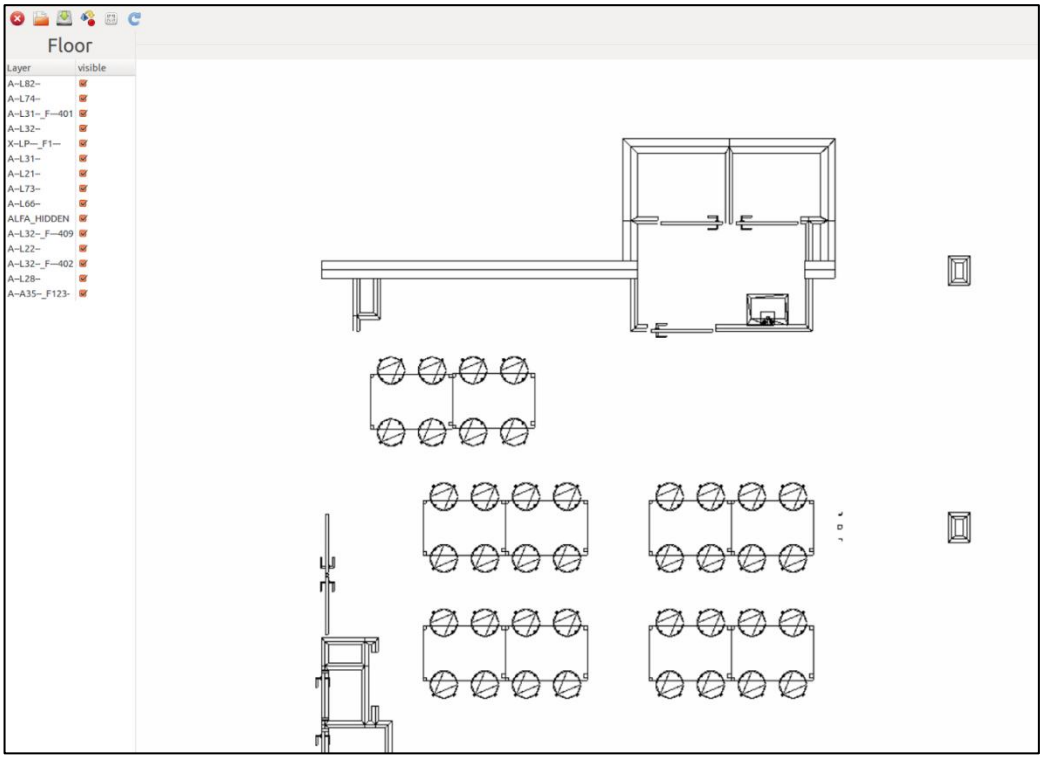


Figure 22: Representation of IFC file into OntoCAD

APPENDIX B

7 METHODOLOGY FOR USABILITY TESTING

This section presents the methodology for conducting usability testing of the models and interfaces presented in the previous section.

7.1 Introduction

For any newly developed object, be it a model, a tool, or anything a human interacts with, the usability of the object is an indicator of whether it has met its design requirements. The term usability can often be interchanged with usefulness [9]. Useful features enable users to “do things” and usable features those that make the “doing” easy. Nielsen defined usability as a sub category of usefulness and has given what may be the best known measure of usability [10]. These are: easy to learn, efficient to use, easy to remember, few errors and subjectively pleasing [11]. In 1998 the International Organization for Standardization (ISO) released a multi-part standard for aspects of human computer interaction. In this they defined the usability of a product as “the extent to which the product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use” [12]. Usability studies, or testing, is therefore the evaluation of a product, or model, by testing it with real users [11]. The ISO definition has now becoming the de facto standard in usability studies and as such will be used in this thesis [10]. Now that the term usability has been defined, a methodology for conducting usability testing is presented.

7.2 Usability Evaluation Structure

Building on the usability definition identified in ISO 9241-11, ISO also adopted the Common Industry Format (CIF) as a standard for usability reports (ISO 2006) with industry backing from companies such as Microsoft, Hewlett-Packard, Boeing, US West and Kodak [13]. CIF does not describe how a usability test should be conducted, but it does make explicit the requirement that an evaluation, of the product/model, include measurements of its effectiveness and efficiency, as well as a measure of the user’s satisfaction. CIF therefore provides a framework for conducting summative evaluations of usability. CIF makes a distinction between “summative” and “formative” evaluations. Formative evaluations are conducted during the development of a product; they are done to mould or improve the product and can be conducted without the need for a test administrator and participant to be co-present. Outputs of formative evaluations may include

participant comments (attitude's, sources of confusion, and reasons for actions) and other usability problems and suggested fixes determined through observation.

In contrast, summative evaluations are carried out at the end of the development stage. They set out to measure or validate the usability of the product. They look at comparing usable metrics and generating data to support claims about usability. Outputs of summative evaluations may include statistical measures of usability, for example success rate, average time to complete tasks, number of errors and/or number assists. It should be noted that summative and formative evaluation should not be seen as either opposing or even conflicting approaches, but rather complementary, being best used together to develop a model.

7.3 Methodology and Metrics

Formative and summative evaluations each look to capture a different set of metrics. Formative evaluations take place early in the development of a product and tend to rely on qualitative data, whereas a summative evaluation takes place towards the end of the development stage and looks at generating quantitative data to support claims about usability. The usability evaluations in KnoholEM shall take this approach by applying formative evaluations in the early and middle stages of development culminating in final summative evaluations. CIF is employed for structuring the usability evaluations. To comply with the CIF standard a usability report must include the following information [14]:

- A description of the product/model.
- The goals of the test.
- Context of Use
    - The test participants and background.
    - The tasks the participants were asked to perform.
- The method or process by which the test was conducted.
- The experimental design of the test.
- The usability measures and data collected.

- Numerical results and analysis.

7.3.1 Goals of Evaluations

For each evaluation, whether formative or summative, the goals are to be set out at the beginning. This includes the research question which will be addressed by the evaluation, as well a null and alternative hypothesis. These are used to determine whether the experiment has met that objective, i.e. to answer a particular research question. The formative evaluations may have a null and alternative hypothesis which is less specific than a summative evaluation, as they rely less on qualitative results.

7.3.2 Metrics

Formative and summative evaluations generally set out to measure different metrics. Sauro and Kindlund have created a quantitative model of usability based upon the ISO 9241 standard, which has resulted in four metrics. These are time to complete tasks, number of errors, whether a task is completed and the average satisfaction of users Figure 23 [15]. These are similar to those defined by Nielsen [16], which are: Success rate (whether users can complete a task), task time (time a task requires), error rate (number of errors per task), and user satisfaction. Measures of effectiveness have a strong correlation to the types of tasks. The web based solution requires that evaluations are designed so that the participants can complete the tasks with no help from an instructor. Efficiency is measured by how long it takes to complete a task and to achieve this times must be recorded when interactions with the interface are taking place.

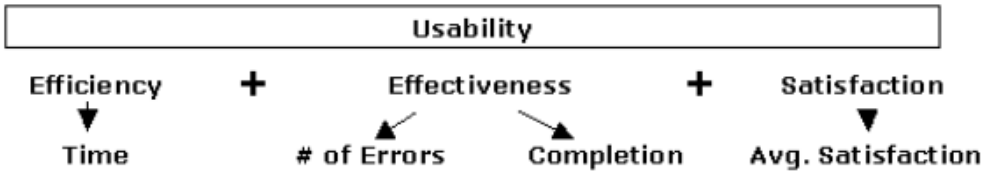


Figure 23 The ISO definition of Usability Converted into Quantifiable Metrics

User satisfaction is assessed through the use of questionnaires. A number of questionnaires are available to assess usability, for example the System Usability Scale (SUS) [17], the Ease of use (USE) questionnaire [18], the Computer System Usability Questionnaire (CSUQ), the Questionnaire for User Interaction Satisfaction (QUIS) [19] and the Software Usability

Measurement Inventory (SUMI) [20]. Some of these, for example USE and CSUQ, suffer from bias in the form of positive phrasing, i.e. all the questions are positive, for example “It helps me be more effective” “It is easy to use”. Tullis and Stetson have evaluated SUS, CSUQ and QUIS and concluded that even though SUS is one of the more simple approaches, it resulted in the most reliable results across sample sizes [21]. They also note that when evaluating only one design, the most important information from the usability questionnaires is related to its diagnostic value with respect to improving the design. The SUS is used in this context for the formative evaluations.

SUS is a simple ten item scale giving a global view of subjective assessment of usability [17]. It is a Likert scale in which the participant must choose between five or seven points ranging from agreement to disagreement on a particular statement. The statements in SUS are chosen to identify extreme expressions or attitudes. SUS also provides a point structure to assign to the answers of a particular test which rates overall satisfaction between zero and a hundred. Bangor and Kortum have provided a rating for these points Figure 24 [22]. They warn though that the rating of OK can be misleading and should not be considered satisfactory. They suggest that products with a score in the seventies should be deemed acceptable, and those below seventy still have usability issues which are cause for concern.

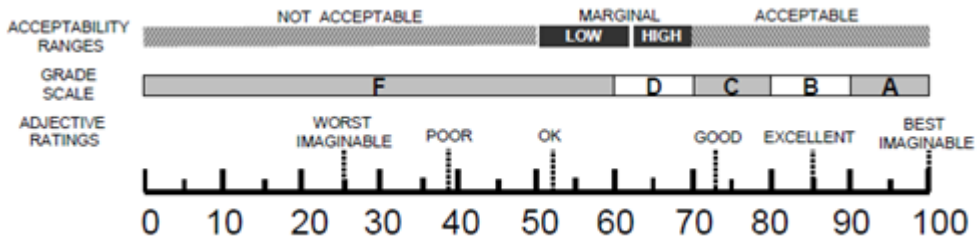


Figure 24 A comparison of the adjective ratings, acceptability scores, and school grading scales, in relation to the average SUS score (Bangor, Kortum et al. 2009)

Finally, while summative evaluations rely on quantitative metrics, these metrics are not exclusive to summative evaluations and may be incorporated into formative evaluation. For instance, task times can be a valuable diagnostic measure during formative-type tests. Now that the metrics to be taken have been explored, the next section will look at the selection of participants.

7.4 Number of Participants

Another question regarding usability is what number of participants is required for an effective evaluation of an implemented model. Nielsen and Landauer have created a mathematical model to determine the number of users needed for a usability evaluation [16]. Their focus was on user testing and heuristic evaluation of interfaces. Heuristic evaluation requires experts in a field to sit and judge a model using established usability principles, for example those given by Nielsen [23]. Usability testing requires real users being observed in real time using the model. The mathematical model they use is a simple Poisson distribution. It is based on the assumption that the probability of any participant discovering a problem is independent of the outcome of a previous test. For user testing the discovery of a problem relies on, a: the participant experiencing the problem, and b: the test administrator realizing an issue has occurred. Unlike the participant, the test administrator's own understanding of the issues is increasing with each evaluation, i.e. they may begin to recognise a particular issue only after observing it multiple times. This potential issue can be ameliorated by carefully considering these possibilities when developing the experiment. Nonetheless, Nielsen and Landauer assert that it would seem likely that the experimenter's findings of issues can also be approximated by a Poisson distribution.

During evaluation, failures which result in the complete cessation of the product will be quick to identify as the participant will no longer be able to interact with it. Other less significant problems will be harder to discover. They assert therefore that the number of usability problems found in a usability test with *i* participants are:

Equation 1: $Found(i) = N(1-(1-L)^n)$

Based on this model, Nielsen and Landauer have argued that a user evaluation with only five users will be enough to identify between 50 and 85% of the problems [16]. This is based on the assumption that the probability of discovering each problem L is 0.31, and they acknowledge that this is not always the case. Woolrych and Cockton have pointed out that to account for this, the value L should itself be based on a probability density function (PDF) that recognises variability in the probability of discovering a problem, as in situations where L is very low [24]. The number of users will consequently need to be raised to discover the same number of errors as situations where L is a higher value. Woolrych and Cockton also point out that in heuristic evaluations experts differ in their ability to discover errors. In the worst case scenario where the experts

selected for an experiment are all “bad” problem finders, the numbers of participants needed to discover errors consequentially increases.

The individual differences between non-expert users, their interaction with the tool under test, and the complexity of the tasks are also key variables which are neglected by Nielsen and Landaeur. Again, to account for these, Woolrych and Cockton recommend that L should be replaced with a PDF parameterised by values that represent beliefs about the differences of the likely impact of the users, the task and the tool under test. Nielsen and Landaeur state that tests of five users should not be done in isolation but rather as part of an iterative evaluation cycle. Their reason for choosing five is related to cost/benefit issues. While the exact number of users to find all potential problems may vary according to the users, the tasks, and the system under test, they claim it is still likely a better strategy, when dealing with constrained time and resources, to evaluate initial evaluations of a design less thoroughly. This will be reflected in the number of users in earlier evaluations. This is the approach taken when conducting usability evaluations in the next sections.

In conclusion, this section identified a method for producing reports and metrics on the usability of a model which can be applied to the evaluation of the developed models and interfaces. It identified a set of metrics which can be used to evaluate the usability of the system quantitatively. Finally, it identified the use of iterative design and evaluation cycles with small groups of users as an approach to discovering early errors in a model as part of an iterative design cycle, culminating in a larger summative evaluation.

Questionnaires: SUS Questions

1. I think that I would like to use this product frequently.
2. I found the product unnecessarily complex.
3. I thought the product was easy to use.
4. I think that I would need the support of a technical person to be able to use this product.
5. I found the various functions in this product were well integrated.
6. I thought there was too much inconsistency in this product.

7. I would imagine that most people would learn to use this product very quickly.
8. I found the product very awkward to use.
9. I felt very confident using the product.
10. I needed to learn a lot of things before I could get going with this product.