S6-4

MATERIAL MATCHING PROCESS FOR ENERGY PERFORMANCE ANALYSIS

Jung-Ho Yu[1] **, Ka-Ram Kim**[2] **and Me-Yeon Jeon**[3]

[1] Assistance Professor, Department of Architecture Engineering, Kwangwoon University, Seoul, Korea
[2] Masters Course, Department of Architecture Engineering, Kwangwoon University, Seoul, Korea
[3] Masters Course, Department of Architecture Engineering, Kwangwoon University, Seoul, Korea
Correspond to karamiz@kw.ac.kr

ABSTRACT: In the current construction industry where various stakeholders take part, BIM Data exchange using standard format can provide a more efficient working environment for related staffs during the life-cycle of the building. Currently, the formats used to exchange the data from 3D-CAD application to structure energy analysis at the design stages are IFC, the international standard format provided by IAI, and gbXML, developed by Autodesk. However, because of insufficient data compatibility, the BIM data produced in the 3D-CAD application cannot be directly used in the energy analysis, thus there needs to be additional data entry. The reasons for this are as follows: First, an IFC file cannot contain all the data required for energy simulation. Second, architects sometimes write material names on the drawings that are not matching to those in the standard material library used in energy analysis tools. DOE-2.2 and Energy Plus are the most popular energy analysis engines. And both engines have their own material libraries. However, our investigation revealed that the two libraries are not compatible. First, the types and unit of properties were different. Second, material names used in the library and the codes of the materials were different. Furthermore, there is no material library in Korean language. Thus, by comparing the basic library of DOE-2, the most commonly used energy analysis engine worldwide, and EnergyPlus regarding construction materials; this study will analyze the material data required for energy analysis and propose a way to effectively enter these using semantic web's ontology. This study is meaningful as it enhances the objective credibility of the analysis result when analyzing the energy, and as a conceptual study on the usage of ontology in the construction industry.

Keywords: Building Information Model; Energy analysis; Material library; DOE-2; EnergyPlus; Ontology; Material matching process

1. INTRODUCTION

Along with the global awareness on green building, there is a growing interest in green building performance analysis. Energy analysis is necessary in analyzing green building performance and Building Information Modeling (BIM) technology is getting increasingly valuable in architecture. A number of energy simulation applications are being developed under BIM-based information management systems. DOE-2 and EnergyPlus (hereinafter ePlus) are the currently most-widely used energy simulation engines in the world. However, they were developed to suit the US and European standards. This, in order for an effective energy analysis in Korea, an input of BIM information suitable for the conditions of Korea is required.

Besides, notations are not standardized. During the design process, designers use different object names for 2D or 3D modeling. This is due to lack of standards for notations. Thus, additional work to re-input data is necessary, resulting in inefficiency.

Semantic database was proposed as one of the solutions to the above problems, and since then many attempts have been made to create ontology, which is one of the core technologies for implementing the semantic database. Semantic DB is an attempt to create a database (knowledge based) where each data elements are related to every other elements based on meaning. In the material data of a design documents, the type and unit of properties were different even from the same document frequently. For example, “Clay Brick” has the same concept with “Red Brick”. In addition, we cannot use an IFC file for energy analysis directly, because CAD software, such as ArchiCAD, Revit and etc, does not provide the properties for energy analysis. Therefore we have defined ontology for building material domain, and then defined the rules of objectified concepts corresponding to nonstandard terms that are frequently used in the design documents. Furthermore it is not necessary to add the properties, such as Thickness, Conductivity, Density, Specific Heat, and Resistance which are needed for energy analysis in additional work for energy analysis

Therefore, this study is to compare and analyze DOE-2 and ePlus to utilize, as input data for DOE-2 and ePlus, information on construction materials among BIM information created in Korea, and to suggest a process to match the building materials for energy performance analysis using an ontology based system. With this, a computer understands the used materials in a building for energy performance analysis automatically.

There are a number of input data for energy analysis such as climate, utilization schedule, and facility system, but this study limited its scope to information on construction materials. To use domestic drawing information for energy analysis, default libraries of DOE-2 and ePlus were compared and analyzed to figure out material properties, which are necessary for energy analysis. Additionally, OWL language was used using Protégé4.1 for ontology expression of material information and material information can be effectively found to improve efficiency in energy analysis process by comparing materials in domestic drawings and in ontology-based database.

2. PROCESS OF ENERGY PERFORMANCE ANALYSIS

2.1 BIM data format for energy analysis

2.1.1gbXML

The Green Building XML schema, referred to as “gbXML”, was developed to facilitate a common interoperability model integrating a myriad of design and development tools used in the building industry. gbXML is being integrated into a range of software CAD and engineering tools. This interoperability standard reduces the time to develop a building and also assures that when the building is put into operation it can meet the design intent.[1]

2.1.2 IFC

The Industry Foundation Classes (IFC) data model is intended to describe building and construction industry data. The IFC file format is developed by the IAI (International Alliance for Interoperability) to facilitate interoperability and is an open data exchange format that is used by model based applications to exchange data.[2] The IFC model consists of tangible components such as walls, doors, beams, furniture etc, as well as the more abstract concepts of space, geometry, materials, finishes, activities etc.[3]

2.2 DOE-2

The DOE-2 engine was developed by the Lawrence Berkeley National Laboratory and is one of the most widely used thermal simulation engines today. The engine was designed to study energy performance of the whole building during the design phase (Birdsall et al. 1990). Due to its long presence on the market several user interfaces have been developed for DOE-2.

2.2.1 Simulation Engine Structure

Figure 1 shows a process of DOE-2 engine. Basically, the program has one subprogram for translation of your input (BDL Processor), and three simulation subprograms (LOADS, HVAC, and ECON). The LOADS subprogram executes first for the entire simulation period, and its hourly data stored in a disk file for the HVAC simulation. Next the HVAC subprogram executes for the entire simulation period, with its hourly results stored in a disk file for the ECONOMICS calculations. Finally, the ECONOMICS subprogram executes. Each of the simulation subprograms also produces printed reports of the results of its calculations.

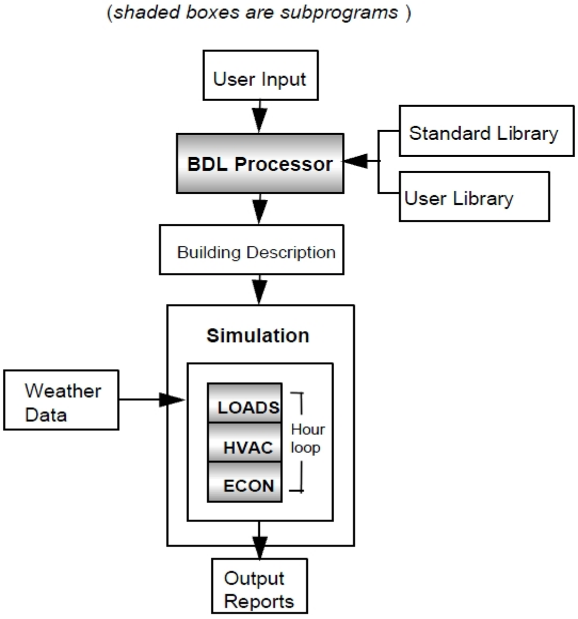


Fig. 1. Overall DOE-2 Structure (DOE-2.2 Manual Vol.1)

2.2.2 Process of DOE-2 program

1) BDL Processor

The Building Description Language (BDL) processor reads the flexibly formatted input data that you supply and translates it into a form recognized by a computer. It also calculates the response factors for the transient heat flow in walls and weighting factors for the thermal response of building spaces.

2) LOADS

The LOADS simulation subprogram calculates the sensible and latent components of the hourly heating or cooling load for each user-designated space in the building, assuming that each space is kept at a constant user- specified temperature. LOADS is responsive to weather and solar conditions, schedules of people, lighting and equipment, infiltration, heat transfer through walls, roofs, and windows and the effect of building shades on solar radiation.

3) HVAC

The HVAC simulation subprogram is divided into a secondary systems simulation (SYSTEMS) and a primary systems simulation (PLANT). The secondary HVAC systems simulation calculates the performance of air-side equipment (fans, coils, and ducts). It corrects the constant-temperature loads calculated by the LOADS subprogram by taking into account the outside air requirements, hours of equipment operation, equipment control strategies, and thermostat set points. The output of the secondary HVAC system simulation is air flow and coil loads. The primary HVAC system simulation calculates the behavior of boilers, chillers, cooling towers, storage tanks, etc., in satisfying the secondary systems heating and cooling coil loads. It takes into account the part-load characteristics of the primary equipment in order to calculate the fuel and electrical demands of the building.

4) ECON

The economic analysis subprogram calculates the cost of energy. It can also be used to compare the cost-benefits of different building designs or to calculate savings for retrofits to an existing building.

5) Weather Data

The weather data of a location consists of hourly values of outside dry-bulb temperature, wet-bulb temperature, atmospheric pressure, wind speed and direction, cloud cover, and (in some cases) solar radiation. Weather data suitable for usage in the program is produced by running the weather processor on raw weather files provided by the U.S. National Weather Service and other organizations. See “Weather Files” below.

6) Standard Library

The program comes with a library of input components. The elements in this library relevant to this document are wall materials, layered wall constructions, and windows. Other elements include HVAC performance curves, glass layers, between-glass gas fills, window blinds, pull-down shades, lamps, and luminaires.

7) User Library

The User Library contains building elements such as walls, windows, schedules, HVAC systems, etc., that you create.[4]

2.3 EnergyPlus

ePLUS is the official building simulation program of the United States Department of Energy, promoted through the Building and Technology Program of the Energy Efficiency and Renewable Energy Office. ePlus is a widespread and accepted tool in the global building energy analysis community [5]. ePlus models heating, cooling, lighting, ventilating, and other energy flows as well as water in buildings. The Building Energy Software Tools Directory [6], which is a directory that provides information on 375 building software tools for evaluation of energy efficiency, renewable energy, and sustainability in buildings, introduces ePlus as a tool for application on energy simulation, load calculation, building performance, simulation, energy performance, heat balance, and mass balance. ePlus is free and can be downloaded from the official website [7].

2.3.1 Simulation Engine Structure

Zone loads calculated (by a heat balance engine) at a user-specified time step (15-minute default) are passed to the building systems simulation module at the same time step. The building systems simulation module, with a variable time step (down to one minute as needed), calculates heating and cooling system and plant and electrical system response. Feedback from the building systems simulation module on loads not met is reflected in the next step of the load calculations in adjusted space temperatures if necessary.

Accurate prediction of space temperatures is crucial to energy efficient system engineering—system size, plant size, occupant comfort and occupant health are dependent on space temperatures. Some of the more important prediction includes realistic system controls, moisture adsorption and desorption in building elements, radiant heating and cooling systems, and interzone air flow.

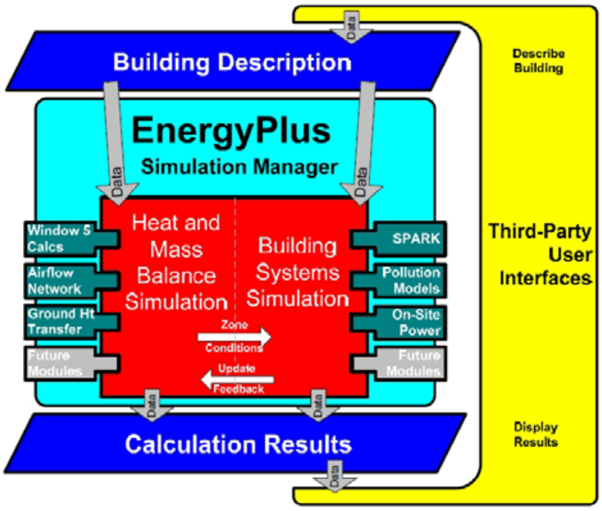


Fig. 2. Overall ePLUS Structure

Figure 2 shows overall ePLUS structure, with three basic components—simulation manager, heat and mass balance simulation module, and building systems simulation module. The simulation manager controls the entire simulation process. The heat and mass balance calculations are derived from IBLAST—a research version of BLAST with integrated HVAC systems and building loads simulation. The building systems simulation manager handles data exchange among the heat balance engine and various HVAC modules and loops, such as coils, boilers, chillers, pumps, fans, and other equipment/components, as well as their input and output data.

2.3.2 Process of ePLUS program

ePLUS input files are somewhat readable but can be cryptic and are not intended to be the main interface for typical end-users. ePlus uses a free format input format that contains a complete object-based description of the building and its systems. The basic syntax is: object, data, data,..,data, data; where ‘object’ is a pre-defined keyword denoting a building or system component, such as SURFACE, MATERIAL, LIGHTS, FURNACE, HEATING COIL, or BOILER. This keyword is followed by a list of data values and terminated with a semicolon. These data describe performance characteristics and intended use for that object in the simulation [8].

2.4 Comparison between DOE-2 and ePLUS

Fig. 3 shows data flow from 3D CAD program to energy analysis program when BIM is used in the energy analysis using DOE-2 and ePlus energy analysis programs.

To import to programs using DOE-2 or ePlus engine, DOE and ePlus respectively require *.inp format and *.idf format. However, 3D CAD does not provide exportation of data in *.inp or *.idf format. Thus, data have to be exported to gbXML and imported to

GreenBuilding Studio for conversion to *.inp or *.idf format and exportation.

Above method, however, has downside that even if BIM is currently being used, gbXML is not officially adopted. gbXML is a format developed by AutoDesk, Inc. and specialized in energy analysis, but not an interrnational standard, limiting its use. Add-on program needs to be separately installed to a CAD program to export if the CAD program is not developed by gbXML developer, and due to the characteristics of file format, it has limitation in expandability. Accordingly, this study analyzed information demand for energy analysis, but extracted material information using IFC format, an international standard format, for future studies.

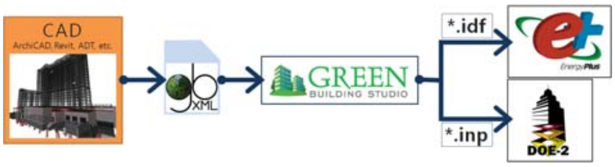


Fig. 3. Data flow by DOE-2& ePLUS

2.4.1 DOE-2 Library

DOE-2 default library provides a lot of libraries with information in its dblib.dat file such as climate information or schedule, facility system, information for economic comparative analysis as well as construction material, and material properties have values of thickness, conductivity, density, specific heat, or resistance in ft-lb unit.

```
AcousticTile 3/4in (AC03) = MATERIAL $Acoustic Tile, 3/4 Inch
    TYPE=PROPERTIES
    THICKNESS=0.0625          (ft)
    CONDUCTIVITY=0.0330       (btu / hr-ft2-b)
    DENSITY=18.0              (lb / ft3)
    SPECIFIC-HEAT=0.32 ..     (btu / lb-F)
```

Fig. 4. Example of material in DOE-2 library (ft-lb unit)

2.4.2 ePLUS Library

Default library, ASHRAE_2005_ HOF_Material, has the information of code, material name and size information in Name, along with values of roughness, thickness, conductivity, density, and specific heat in SI unit.

```
Material,
    F16 Acoustic tile,        !- Name
    MediumSmooth,             !- Roughness
    0.0191,                   !- Thickness {m}
    0.06,                     !- Conductivity {W/m-K}
    368,                      !- Density {kg/m3}
    590;                      !- Specific Heat {J/kg-K}
```

Fig. 5. Example of material in ePlus library (SI unit)

2.5 Problems of Energy Analysis

part among default library provided by DOE-2 and ePlus was examined above. The comparison analysis results are as in the following table.

Table 1. Comparison of material data

	DOE-2	ePLUS
Thickness	○	○
Conductivity	○	○
Density	○	○
Specific Heat	○	○
Resistance	○	○
Roghness	×	○
Unit	ft-lb	SI

As the table shows, DOE-2 required material information includes all information except roughness, compared to that of ePlus, and thus two engines require similar data. In other words, both engines commonly require material data of thickness, conductivity, density, specific heat, and resistance for energy analysis. However, DOE-2 uses ft-lb unit for its properties whereas ePlus uses SI unit. Thus, to establish material data used in Korea for energy analysis, standards should necessarily be converted from SI unit data to ft-lb unit data. In addition, as described in section 3, one material may have different names. This is due to lack of standards for material names used for design documentation. Analyzers arbitrarily input material data for energy analysis, and depending on analyzer's skills, reliability of energy analysis results may vary. Meanwhile, schema of IFC file may include properties required for energy analysis, but such information cannot be expected from 3D CAD used modeling and the properties for energy analysis must be re-input. Therefore, this study is to suggest a system to manage material data using ontology as a way to solve such problems.

3. ONTOLOGY FOR BUILDING MATERIAL

The Building Material Ontology (BMO) has two parts: Building Material Description Ontology (BMDeO) and Building Material Domain Rules (BMDR). BMDeO relates to information on building materials description. It describes a concept of each material by several properties. In this paper, we use "Red Brick" as an example.

Based on concepts defined in BMDeO, we defined BMDR as rules the objectified concepts corresponding to subjective terms frequently used by an architectural designer. "Red Brick" is also used as an example in this part, too.

3.1 Building Material Description Ontology

As we know, BMDeO is the ontology for describing building material which is defined by user (i.e., architect or energy analyst). There are too many types of materials for concepts to be described. For this reason, this paper consists of an example (Red Brick) as a conceptual study. First for design BMDeO, we should know what the data is required for energy analysis. And we already defined the required data through section 3. However, the IFC file as a standard BIM data format contains not enough data for energy analysis.

Fig. 6 represents a part of IFC schema showing what entities relate to **IfcRoot**. **IfcConstructionMaterialResource** is a subclass of **IfcResource**, and **IfcSlab** is a subclass of **IfcProduct**. And **IfcProduct** and **IfcResource** are subclasses of **IfcObject**. In addition, there are many subclasses of

IfcbuildingElement, such as **IfcWall**, **IfcBeam** and so on. It means **IfcConstructionMaterialResource** class has relationship with **IfcBuildingElement**. Therefore we can extract where the material is used from IFC file.

By the way, material class has a label as a property. This label is made by users, and it contains a material name (if the material is made by CAD application). In addition, **IfcMaterialLayerSet** class has a thickness value as a property. Fig. 7 is shown the **IfcMaterial** and **IfcMaterialLayerSet** classes have several properties.

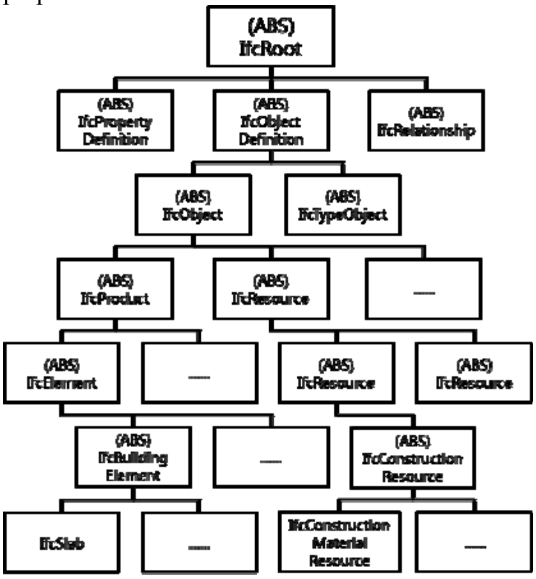


Fig. 6. A part of IFC schema

By the way, material class has a label as a property. This label is made by users, and it contains a material name (if the material is made by CAD application). In addition, **IfcMaterialLayerSet** class has a thickness value as a property. Fig. 7 is shown the **IfcMaterial** and **IfcMaterialLayerSet** classes have several properties.

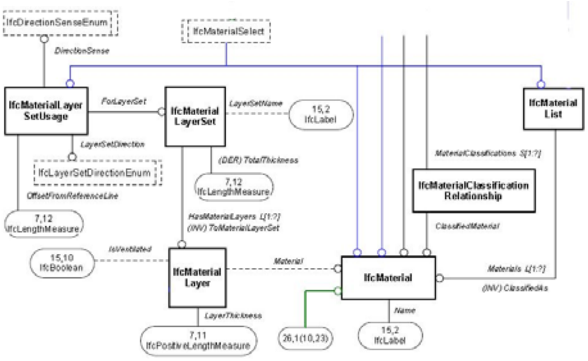
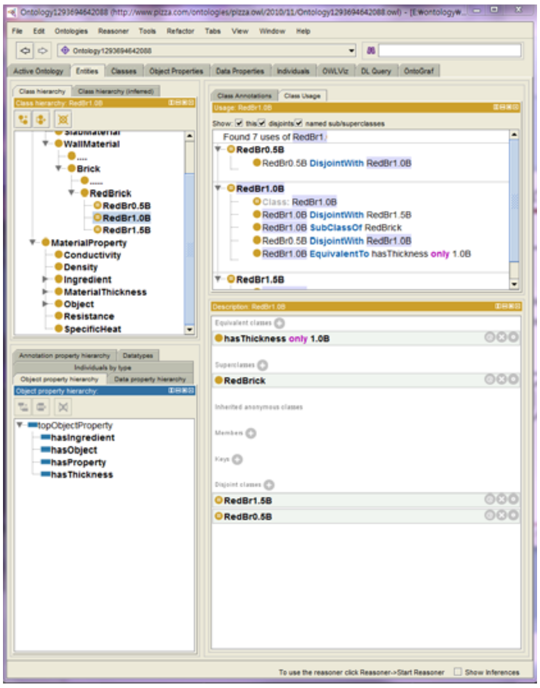


Fig. 7. A part of IfcMaterial and IfcMaterialLayerSet classes

As we show, the material data has three properties from IFC file: used location, name and thickness. It means we can extract those material data from IFC file.

Web Ontology Language (OWL) is used to build BMDeO, which is designed using Protégé v4.1, a Java-based ontology editor. Typically, OWL is used as the standard ontology language [9]. OWL, which was derived from DAML+OIL[10] and developed by W3C, has three different sublanguages; OWL Full, OWL DL and OWL Lite, each of which have different expressive powers. OWL Full is the most expressive, OWL Lite is the least expressive and OWL DL falls between the two, but it is significantly more expressive than OWL Lite. OWL DL is based on Description Logic [11], which is a decidable fragment of First Order Logic and permits automated reasoning. Therefore, OWL DL allows us to check automatically how classes defined in ontology are subsumed and whether they are consistent. For these reasons, we selected OWL DL from the three sublanguages of OWL for the construction of our ontology for building materials.

In OWL, a concept corresponds to a class, and a relationship corresponds to a property. A class defines a group of individuals with the same set of properties and axioms. A property is used to define a relationship between two classes, called ObjectProperty, or between a class and an individual, called DatatypeProperty. Classes can be organized in a hierarchal structure, using the property rdfs:subclassOf, where the prefix rdfs indicates that the namespace of the property subclassOf is RDFs (Resource Description Framework Schema) [12]. RDFs provides a mechanism for specifying the structure of RDF (Resource Description Framework) data model [13]. RDF was developed to facilitate the automated processing of web resources in a machine-understandable form and to provide interoperability between applications. However, owing to the limited expressive power of RDFs, the extended ontology language, OWL, was developed [14].


Ontology1293694642088 (http://www.pizza.com/ontologies/pizza.owl/2010/11/Ontology1293694642088.owl) - [E:\ontologyw...
File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help
Ontology1293694642088
Active Ontology Entities Classes Object Properties Data Properties Individuals OWLViz DL Query OntoGraf
Class hierarchy Class hierarchy (inferred)
Class hierarchy: RedBr1.0B
Class Annotations Class Usage
Usage: RedBr1.0B
Annotation property hierarchy Datatypes
Individuals by type
Object property hierarchy Data property hierarchy
Object property hierarchy:
Description: RedBr1.0B
Equivalent classes
Superclasses
Inherited anonymous classes
Members
Keys
Disjoint classes
To use the reasoner click Reasoner->Start Reasoner
Show Inferences

SlabMaterial
WallMaterial
....
Brick
......
RedBrick
RedBr0.5B
RedBr1.0B
RedBr1.5B
MaterialProperty
Conductivity
Density
Ingredient
MaterialThickness
Object
Resistance
SpecificHeat

Show: ☑ this ☑ disjoints ☑ named sub/superclasses

Found 7 uses of RedBr1

RedBr0.5B
RedBr0.5B DisjointWith RedBr1.0B
RedBr1.0B
Class: RedBr1.0B
RedBr1.0B DisjointWith RedBr1.5B
RedBr1.0B SubClassOf RedBrick
RedBr0.5B DisjointWith RedBr1.0B
RedBr1.0B EquivalentTo hasThickness only 1.0B
RedBr1.5B

topObjectProperty
hasIngredient
hasObject
hasProperty
hasThickness

Equivalent classes: hasThickness only 1.0B
Superclasses: RedBrick
Disjoint classes: RedBr1.5B, RedBr0.5B

Fig. 8. Screen picture of Protégé v4.1.

Fig. 8 is a screen picture of Protégé v4.1. In this ontology, the root class is *Thing***,** which is the default name of root class in Protégé. The subclasses of Thing are *Materials* class and *Properties* class. *Materials* class represents every building material in this ontology.

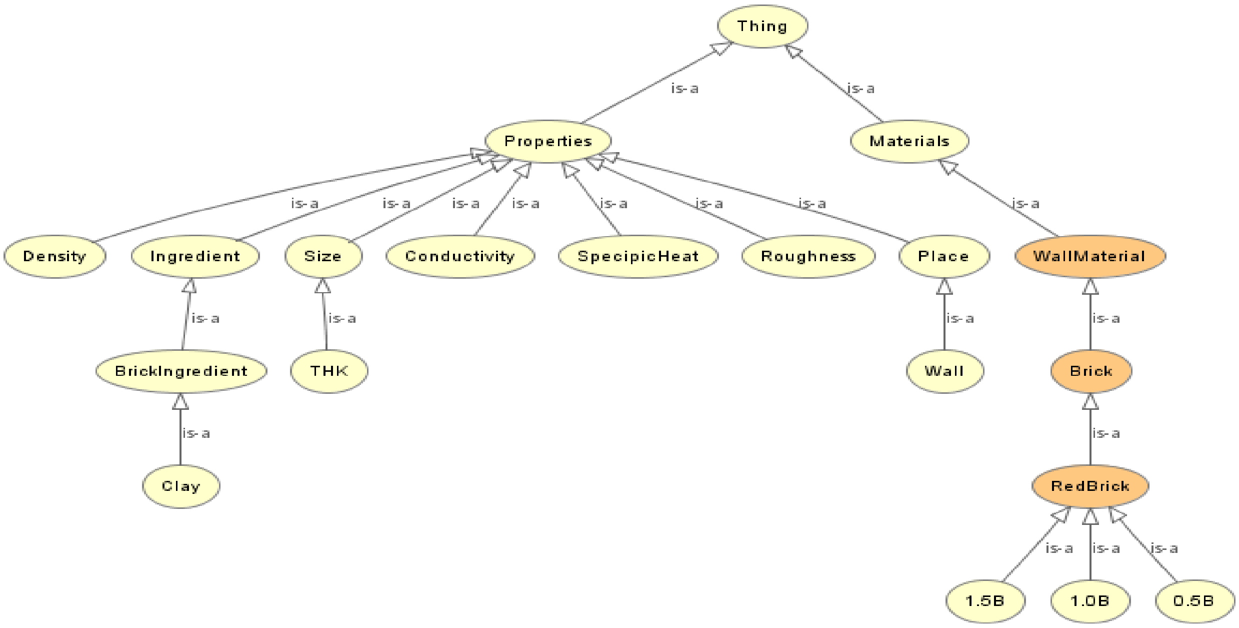


Fig. 9. Hierarchy of the inferred model

However, in this paper, we have only one subclass for an example as a conceptual study. *WallMaterial* class has *Brick* class which is a superclass of *RedBrick*. In this paper, *RedBrick* class has just three type of thickness: *0.5B, 1.0B, 1.5B*. Another subclass of *Thing* is *Properties* class which represents characters and attributes of materials.

Properties class has not only energy analysis properties: *SpecipicHeat, Roughness, Density, Conductivity, Thickness*, but also *Ingredient* and *Place* (used location of building elements). Fig. 9 is an export within OWLViz, which is a plug-in tool of Protégé v4.1. We can see a hierarchy of the inferred model of our ontology. The colored circles are defined by BMDR (Building Material Description Rule), we will discuss it on next section.

3.2 Building Material Domain Rule

When a subjective term like Red Brick is used in the design document by acronym or synonym, it may have different name for each user. It might solve this problem using a thesaurus, but it has limits when there is too much data in the database. Therefore, this paper proposes definitions of materials by metadata from IFC files. For example, Red Brick can be defined as a kind of Brick made by Clay in this ontology.

That a kind of WallMaterial which made by BrickIngredient. To the definition, rules consist of equivalent class on this ontology.

"≡" in the brown circle means equivalent class, it means the class reffered as a necessary and sufficient criteria. Domain rules can be defined in Semantic Web Rule Language (SWRL) [15, 16] which enables Horn clause-like rules to be added to semantic knowledge base in the form of a set of OWL axioms. Each rule in SWRL takes the form of **Antecedent→Consequent**, where antecedent is a conjunction of atoms, and variables are letters prefixed by a question mark. In the "Red Brick" case is :

0.5B Red Brick (?a) ∧ Size (?a, ?b)
∧ Ingredient (?a, ?c)
∧ swrlb : Equal (?b, 90)
∧ swrlb : Equal
(?c, BrickIngredient)

This rule reads that if a size of instance is 90mm and an ingredient is a brick ingredient, the instance is a 0.5B Red Brick.

Like this, the Rule Interchange Format (RIF) working group started with developing a set of formats for interchange of rules in rule-based system on the semantic web. RIF has a great deal of extensions to support features such as frames and objects, URIs, and XML schema data types. Although RIF is a part of semantic web architecture having compatibility with XML, RDF, OWL, and SPARQL, it is designed to enable interoperability among rule languages in general, thus not limited to the web.[14]

4. SEMANTIC MATERIAL MATCHING PROCESS ARCHITECTURE

We suggest the matching process for implementing a semantic material match system, as shown Fig. 10. At first, BIM information is applied to create a building description by Building Description Module. This Module is made an INP file to DOE-2 engine. At the same time, the material data is sent to material matching module to add the required data from energy analysis. It consists of four layers and two storages: (1) semantic resolution conflict layer, (2) control layer, (3) reasoning layer, (4) query processing layer and (a) thesaurus storage, (b) RDF storage.

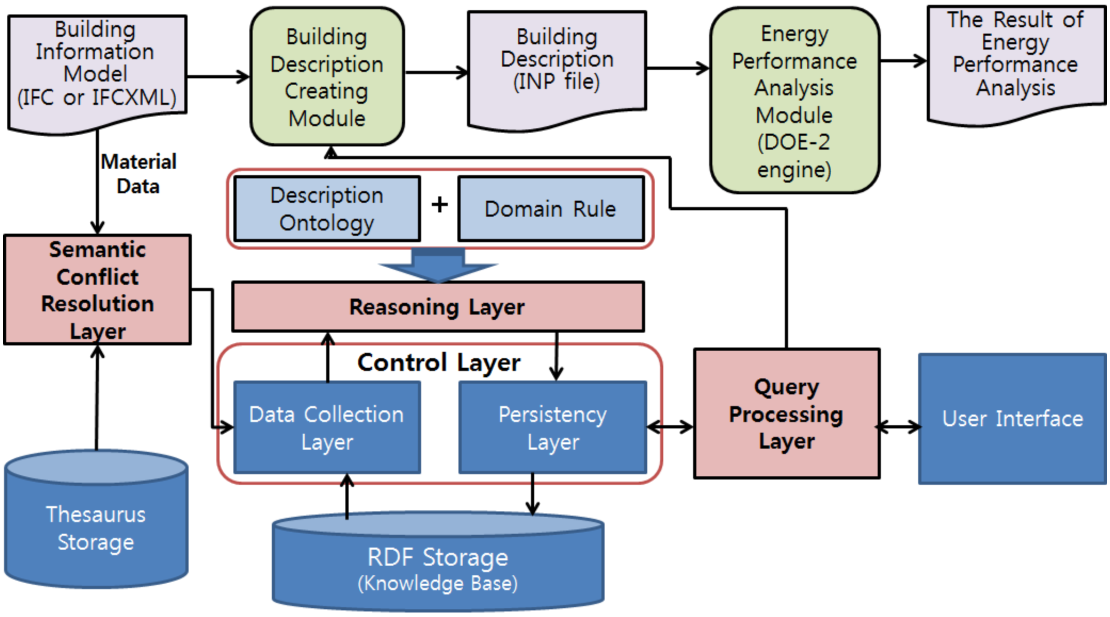

Building Information Model (IFC or IFCXML)
Building Description Creating Module
Building Description (INP file)
Energy Performance Analysis Module (DOE-2 engine)
The Result of Energy Performance Analysis
Material Data
Description Ontology
+
Domain Rule
Semantic Conflict Resolution Layer
Reasoning Layer
Control Layer
Data Collection Layer
Persistency Layer
Query Processing Layer
User Interface
Thesaurus Storage
RDF Storage (Knowledge Base)

Fig. 10. Semantic material matching process

4.1 Semantic Resolution Conflict Layer

The data should be considered to use it in this system. When user creates a material data within design document, there is no duty to use standard material name. It means there are many different names for one material in just one building material. So it is necessary to standardize the material name using (a)Thesaurus storage by semantic resolution conflict layer. Thesaurus storage is organized by standard terms and usage terms from materials. If defined by any synonym or homonym in this layer, the name of material is standardized by matching with the Thesaurus storage. For example, if the user used a material which is named “Red Br”, this name is changed by “Red Brick” with this layer. By the way, when a new material appears the thesaurus storage has to be upgraded.

4.2 Control Layer

The data of the system is consist of two types, entered data and extracted data. Entered data is created by user on the web site of the system. It might be information of building materials, and it does not include the properties which are needed for energy analysis. Besides, extracted data is come from a BIM data. The extraction is not our scope, so we need to formulate a hypothesis about it. Both of data types are sent to the reasoning layer.

The other part of control layer is the persistency layer. It undertakes import / export RDF data in the system. It stores inferred knowledge into (b)RDF storage. RDF storage is organized by standard material data, includes energy performance properties. The inferred knowledge is generated by reasoning layer

Control layer is implemented using Jena system which is a Java-based Semantic Web framework. It provides a programing environment for Semantic Web technologies such as RDF, OWL and SPARQL.

4.3 Reasoning Layer

In this layer, we can create inferred knowledge by reasoning with semantic contents and model. Reasoning is carried out by a reasoner which checks logical consistency of the ontology and rules and generates inferred knowledge. Ontology is represented in OWL DL and domain rules are represented in SWRL. Both languages are based on different logic. OWL DL is based on description logic and SWRL is based on horn-clause logic. Therefore the reasoned must be able to support both languages. The reasoner makes inferred knowledge and model. And it could be sent to control layer again to store it.

4.4 Query processing Layer

For semantic query processing, we can use SPARQL which is developed by W3C RDF Data Access Working Group. SPARQL retrieves information stored in inferred knowledge which contains a large set of RDF triple structures. To support user interface for SPARQL in the web environment, Joseki, which is an implementation of an experimental Web API for the Jena RDF framework, is needed. When SPARQL query is submitted to the query engine via Joseki, it retrieves relevant information from the inferred knowledge.

5. CONCLUSIONS

This rule reads that if a size of instance is 90mm and the ingredient is brickingredient, the instance is 0.5B Red Brick.

In this paper, we have studied standard file format for energy analysis. There are two kinds of format: IFC and gbXML. gbXML is very powerful format for energy analysis within specific applications which are developed by specific private company. So IFC standard format should be used for energy analysis. DOE-2 and ePlus are the most useful energy performance analysis engines in the world. And we have described each engine's structure, input data and utilization scenario. Nowadays, however, there are three problems to energy analysis by these scenarios: (1) the difference of naming a material by each user, (2) IFC contains not enough data for energy analysis, (3) a computer cannot understand the concept of material. Finally, we have suggested the conceptual ontology based building material matching system on the web. This system has four layers: Semantic Resolution Conflict Layer, Control Layer, Reasoning Layer and Query processing Layer, and each layer has specific function.

Current system, however, has some limitations which are summarized as follows:

- The ontology domain is only just one material which is Red Brick, and we defined only Red Brick by SWRL.
- In the system, those layers are not real world contents
- It is only theoretical story, so it could be more technical problems.

These limitations should be removed by the next study in the near future.

ACKNOWLEDGEMENT

This work (00043726) was supported by Business for International Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2010.

REFERENCES

[1] About gbXML, Green building XML Schema, *wiki article*, accessed 29 November 2010, http://en.wikipedia.org/wiki/Green_Building_XML

[2] Khemlani, L, "Building the Future", *Exhibit Hall Highlights from Autodesk University 2006'AECbytes*, viewed on 11 September 2007.

[3] Sumedha Kumar, "Interoperability between Building Information Models(BIM) and Energy Analysis", *Masters Abstracts International*, Vol. 46, no. 6, 2008.

[4] DOE-2, http://www.doe2.com/

[5] M. Stadler et al., "On-site Generation Simulation with EnergyPlus for Commercial Buildings", *Conference: 2006 ACEEE Summer Study on Energy Efficiency inBuildings., Asilomar, California*, August 13-18, 2006.

[6] U.S. Department of Energy, Energy Efficiency and Renewable Energy Office, Building Technology Program, Building energy software tools directory. Available from: http://apps1.eere.energy.gov/buildings/tools_directory/.

[7] U.S. Department of Energy, Energy Efficiency and Renewable Energy Office, Building Technology Program, EnergyPlus. Available from: http://apps1.eere.energy.gov/buildings/energyplus/.

[8] Drury B. Crawley et al., "EnergyPlus: An Update", *Proc. SimBuild 2004, Building Sustainability and Performance Through Simulation*, August 4-6, 2004, Boulder, CO, USA

[9] McGuinness, D. et al., "OWL web ontology language overview", *W3C Recommendation*, 2004, At http://www.w3.org/TR/owl-features/.

[10] Hendler, J., & McGuinness, D., "The DARPA agent markup languae", *IEEE Intelligent Systems*, Vol. 15(6), pp. 67-73, 2000.

[11] Baader, F. et al., *The description logic handbook: Theory, implementation and application*, Cambridge: Cambridge University Press, 2002.

[12] Brickley, D. et al., "RDF vocabulary description language 1.0: RDF Schema", *W3C Recommendation*, 2004, At http://www.W3.org/TR/rdf-schema/.

[13] Beckett, D. et al., "RDF/XML syntax specification(revised)", *W3C Recommendation*, 2004, At http://www.w3.orf/TR/rdf-syntax-grammer/.

[14] Gunwoo Kim et. al., "Semantic business process space for intelligent management of sales order business processes", *Information Systems Frontiers* DOI: 10.1007/s10796-010-9229-1, 2010.

[15] Horrocks et. al., "OWL rules: a proposal and prototype implementation", *Journal of Web Semantics*, Vol. 3(1), pp. 23-40, 2005.

[16] Horrocks et. al., "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", *W3C Member Submission*, 21 May 2004, At http://www.w3.org/Submission/SWRL/.