UBUBI – WP1

Intelligent building eco-design software

Version: 0.3
Publish date: 2016-03-27

Software requirements specification (SRS)





Revisions history

Date	Version	Description	Author
2016-02-07	0.1	First version	Mathieu Dupuis
2016-02-10	0.2	Re-scope of document to fit to UBUBI v 1.0	Mathieu Dupuis
2016-03-24	0.3	Add the interaction between Spark and MySQL	Mathieu Dupuis

Table of content

1 Introduction 8

1.1 Foreword 8

1.2 Goal 8

1.3 Scope 8

1.4 Technology choice 8

2 Requirements 9

2.1 Functional requirements 9

2.2 Non-functional requirements 9

2.2.1 Usability 9

2.2.2 Performance 9

2.2.3 Scalability 9

2.2.4 Maintainability 9

2.2.5 Portability 10

2.2.6 Traceability 10

2.2.7 Reliability 10

3 Interfaces 10

3.1 User interfaces 10

3.2 System interfaces 10

3.2.1 Overview 10

3.2.2 Packaging 11

3.2.3 Class diagram 13

4 Input and Output File 13

4.1 IED File 13

5 Database 14

5.1 Type of data 14

5.2 Mysql 15

5.2.1 Front-end 15

5.2.2 OpenLCA 15
5.2.3 Back-end 19
5.3 Spark 21
6 Algorithms 21
6.1 Create the process tree 21
6.1.1 Process tree topology 22
6.1.2 Process linking 24
6.2 Resolve the process tree 25
6.3 Characterize the elementary flow 25
7 Standards used 25
7.1 Date’s format 25
8 References 26

List of figure

Figure 1: Context diagram 10
Figure 2: Package diagram 12
Figure 3: CSV BIM file format 14
Figure 4: OpenLCA entity relationship diagram 16
Figure 5: tbl_process_links 18
Figure 6: Result tables 20
Figure 7: tbl_result_process_link 21
Figure 8: Foreground process topology 22
Figure 9: Object's type and material process representation 23

List of table

Table 1: Definitions, acronyms and abbreviations list 7
Table 2: Parameters meanings by distribution type 17

Glossary

Term	Definition
Add-In	Term used by Revit for plugin
Assembly	Material set contained in a BIM object.
Background data	External database con
BIM	Building Information Modeling
BIM model	3D digital model composed of intelligent and parameterisable object.
BIM object	Intelligent and parameterisable representation of a physical object contained in a BIM model, like : a door, a wall, a window, etc.
Building's life cycle	Consecutive stage linked to a building, including final designs, resource extraction or harvesting, building product manufacturing, all related transportation, on-site construction, operations including maintenance and replacement, and reconstruction or demolition and disposal (ASTM, 2013).
Foreground data	All process and flow generated by UBUBI to describe a BIM model.
Impact indicator	Quantifiable representation of an impact category.
Midground data	Data created by UBUBI team to create link with the background data.
LCA	Life cycle assessment – « compilation and evaluation of the inputs, outputs and the potential environmental impacts of a product system throughout its life cycle » (ISO 14040).
LCI	Life cycle inventory – Compilation and quantification of inputs and outputs, for agiven product system throughout its life cycle (ISO, 2006a; 2006b)
LCIA methods	Set of characterization model (ex. TRACI, Impact World+)
Life Cycle	« Consecutive and interlinked stages of a product system, from raw material acquisition or generation of natural resources to final disposal » (ISO, 2006a; 2006b)
Life cycle inventory data base	LCI-DB – Database containing quantitative data on input and output of an unit process.
UBUBI	Uncertain But Usefull Building Information - Code name given to the LCA calculator.
Whole building LCA	« Life cycle assessment of the complete building enclosure, structural systems, interior walls, and interior finishes and trim of a building, which may include operating energy, but excludes furniture and attached cabinetry » (ASTM, 2013).

Table 1: Definitions, acronyms and abbreviations list

1 Introduction

1.1 Foreword

This document will be refined while the successive iteration of the project. Some part of this report target a future version of UBUBI and contain no information.

1.2 Goal

This document's goal is to identify all technical requirement of the software to get a global idea of the software to develop. This document is for the developer or someone that have technical knowledge on software engineering.

1.3 Scope

This document cover the realization of UBUBI, a LCA calculator prototype. This prototype doesn't include any login or account system, but support multiple projects. This prototype is use by a unique user and he can see all projects in the software.

1.4 Technology choice

The database used in this project are MySQL and Spark. Spark is used for is powerful calculation capacity. MySQL is used to interface with OpenLCA and to contain static or almost statics information like the menu of the web site. To interact with Spark, Spark offer an API for Python, Scala and Java. Developers choose that the project's backend will be in Scala.

For the frond end, this project will be based on Play framework. Play framework is really similar of .net mvc 4.5 with Razor extension and is based on Scala and Java. So it should be easy to learn and fully interoperable with the back-end.

2 Requirements

2.1 Functional requirements

Check vision document for this section.

2.2 Non-functional requirements

2.2.1 Usability

The usability is the main requirement in this project. UBUBI need to be simple for a non-expert LCA user.

2.2.2 Performance

UBUBI need to compute these result in real-time. In this document, real-time is define like this:
After a user action, the system have 2 seconds to return a result to the user.

2.2.3 Scalability

Project's performance are based on the scalability of the system. This project use Spark to distribute the effort on many computer. The goal for the scalability is to get at least 75% of the computation power of each computer added.

$$\frac{ProcessTimeAlone\ (*\ 1\ computer)}{ProcessTimeCloud * NbComputer} \leq .75$$

Where:
ProcessTimeAlone: Time to compute the result in nanosecond with one computer.
ProcessTimeCloud: Time to compute the result in nanosecond with multiple computer.
NbComputer: Number of computer used to computer the result.

Note: Maybe the number of computer isn't available or computer doesn't have the same computational power, in this case we can use any term of process power (Flop, Hz…) to replace the number of computer.

We expect to lose some power with Spark because every parallel computation need some intercommunication to sync all the work.

2.2.4 Maintainability

Each code file need to be commented in English and in Scaladoc format. The file's / class's comments need to include these information: author, project and description. Every method need at least have a summary and all parameter and return value describe.

2.2.5 Portability

The software is design for Google Chrome and won't be tested with other browser.

2.2.6 Traceability

This project is a prototype and doesn't have traceability requirement.

2.2.7 Reliability

This project is a prototype and doesn't have reliability requirement.

3 Interfaces

3.1 User interfaces

These presented user interfaces in this sub-sections are draft of the final interface and may change while the design and development stage.

To be added when frontend student begin

3.2 System interfaces

3.2.1 Overview

For this project, the team cannot develop all the tools needed to successfully do a LCA of a whole building. To save time they will use existing software for some task. All task related to create and modify a BIM model will be executed by Revit software. All data generated by Revit, will be accessed by an add-in. All task related to create, import and modify LCA background data will be execute by OpenLCA software. All data generated by OpenLCA will be accessed directly with a connection to the Mysql database.

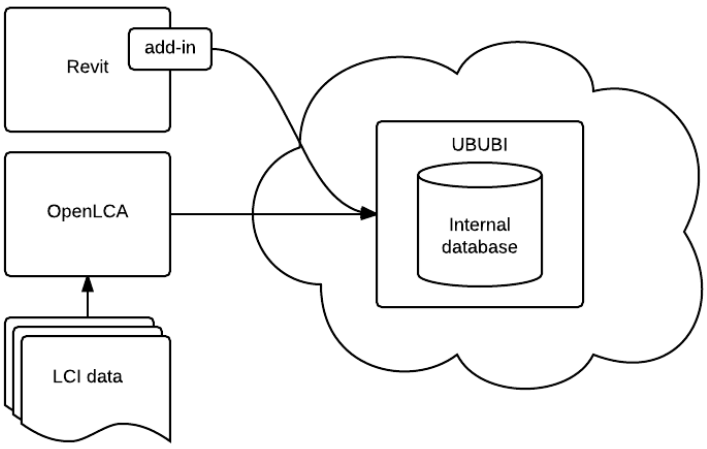


Figure 1: Context diagram

3.2.1.1 Revit & Add-in

Revit is an Autodesk's software to design BIM model. This software won't be modified in this project, but and add-in is designed to be integrated in it. Revit's task is to feed BIM data to the add-in. The Add-in use Revit like a BIM data provider and must use Revit API to work. The plug-in will convert BIM data in an intelligent environmental data (IED) format, describe in section 4.1.

3.2.1.2 LCI data & OpenLCA

LCI data can be in many format. The most common format are ILCD and eco-spold (1 and 2). This project can't handle the importation and the exportation of all these format. OpenLCA will be used as a data converter for this project. OpenLCA can communicate with a remote MySQL database. In this project, we will use this feature and reuse the generated data.

3.2.1.3 UBUBI

UBUBI is a cloud service for life cycle assessment. This cloud service offer some interface like:

- IED importation interface
- LCA report exportation interface
- LCA calculation interface

All these interface are invisible to the user when they use the web portal.

3.2.1.4 Internal database

The internal database support UBUBI software. This database is on Apache Spark technology and MySql. All the information shown on the web site is in Mysql database and all the computing is on Apache Spark.

3.2.2 Packaging

UBUBI is divided in many executable file that share some functionality. To avoid duplicated source code, all shared functionality is contained in an independent library. This strategy optimize the reusability of the code and facilitate automatize test.

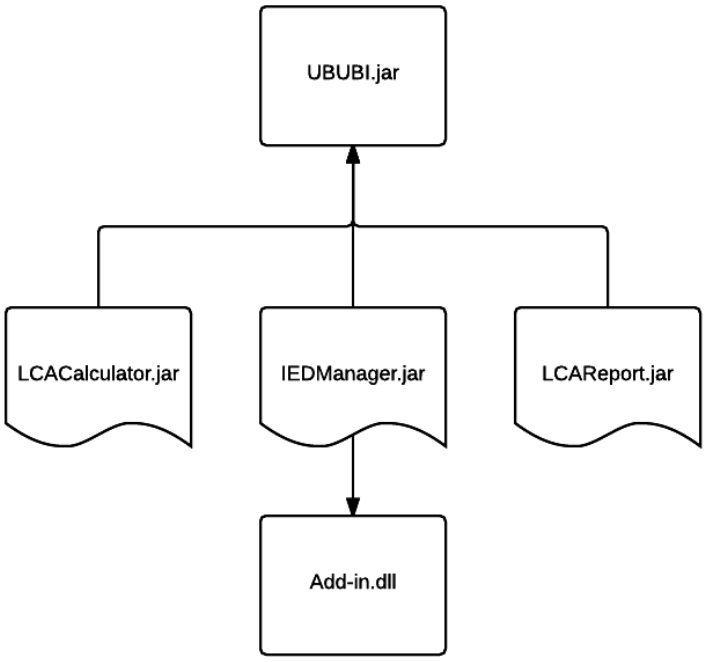


Figure 2: Package diagram

3.2.2.1 UBUBI.jar

Represent the web service once compiled. The web service is dependent of these library:

- LCACalculator.dll
- IEDManager.dll
- LCAReport.dll

All class used only by UBUBI.jar are in the namespace “UBUBI”. The web service is based on MVC pattern (Model, View and Controler) and are support by the Play framework.

3.2.2.2 Add-In.exe

Represent the add-in for Revit. All classes contained in this library is in the namespace “UBUBI.Add-In” except classes related to the generation of IED file.

3.2.2.3 LCACalculator.jar

This library contain all classes use to calculate and characterize impact of a product system. All classes contained in this library is in the namespace “UBUBI.LCACalculator”.

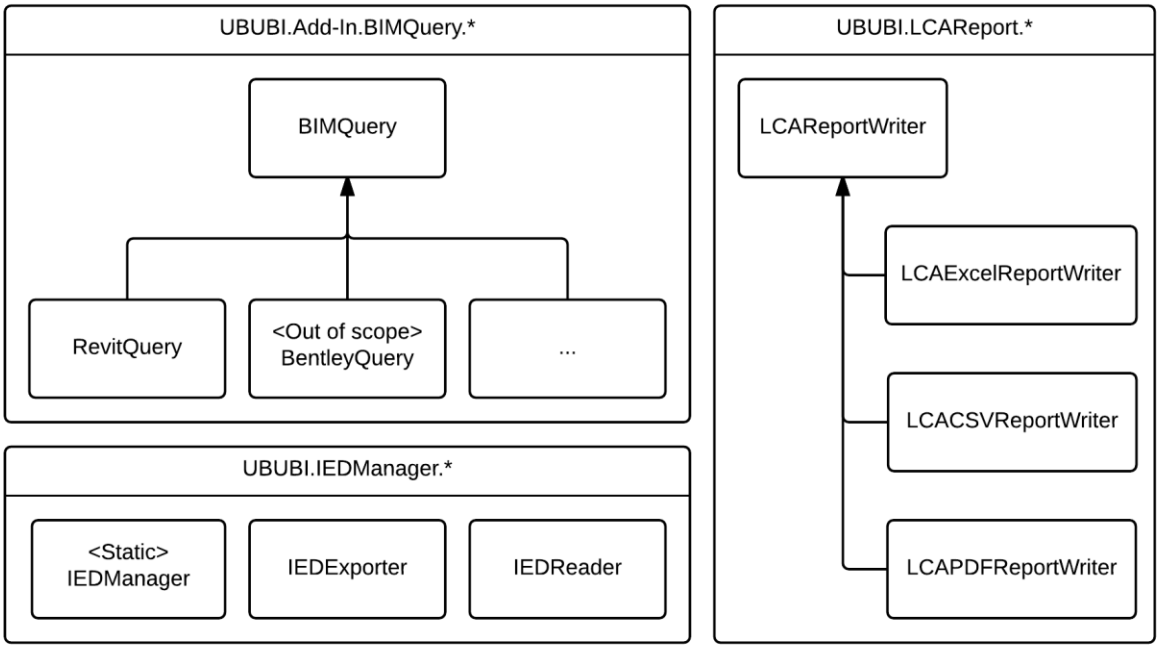
3.2.2.4 IEDManager.jar

This library contain all classes needed to write or read IED file. All these classes is under the namespace “UBUBI.IEDManage”.

3.2.2.5 LCAReport.jar

This library contain all classes needed to generate the report's content and report's file. All these classes is under the namespace "UBUBI.LCAReport".

3.2.3 Class diagram



4 Input and Output File

4.1 IED File

This section describe a new file format called Intelligent Ecological Data (IED). This new file type allow to transfer BIM data to life cycle assessment software. In further study, this format will allow to transfer differential update of the file and is one of many requirement that IFC file doesn't have. For now, the IED file is replace by three CSV file.

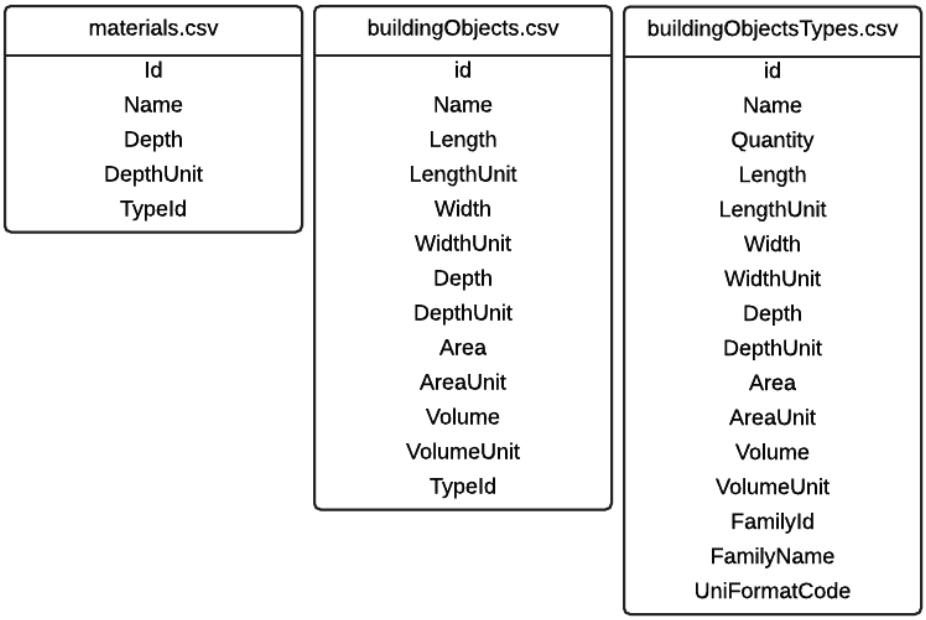


Figure 3: CSV BIM file format

All these files need to be process differently. The file “buildingObjects.csv” describe each object in the BIM model. The “TypeId” field refer to the field “Id” in the file “buildingObjectsTypes.csv”.

The files “buildingObjectsTypes.csv” describe each object’s type in the BIM model. Every measure field contain the sum of all object of this type. The field “Quantity” is the number of object with “TypeId” field that correspond to this type’s id.

The files “material.csv” describe all the layer contained in a type of object. These measure represent only the thickness of the material and need to be multiply with the type’s area to get the final volume.

5 Database

5.1 Type of data

UBUBI use tree kind of data:

- Foreground data
- Midground data
- Background data

Each kind have it own purpose. The most common known is the background data. Like every other LCA software the background represent all the process and flow provide by common LCI database (ex : EcoInvent, Gaby). This provide data commonly approved by LCA expert.

The foreground data is all processes and flows generated by UBUBI. These represent BIM model's elements (objects, object's types or Uniformat's classifications ranks).

The midground data is all the market process created by UBUBI team for a flow. These market process will regroup common process for a BIM model's elements. These process will facilitate the link between BIM object and LCA's elementary process.

5.2 Mysql

5.2.1 Front-end

To be determine with the UI student

5.2.2 OpenLCA

All these table are provided by the software OpenLCA. These table are the source of LCI data and LCIA data.

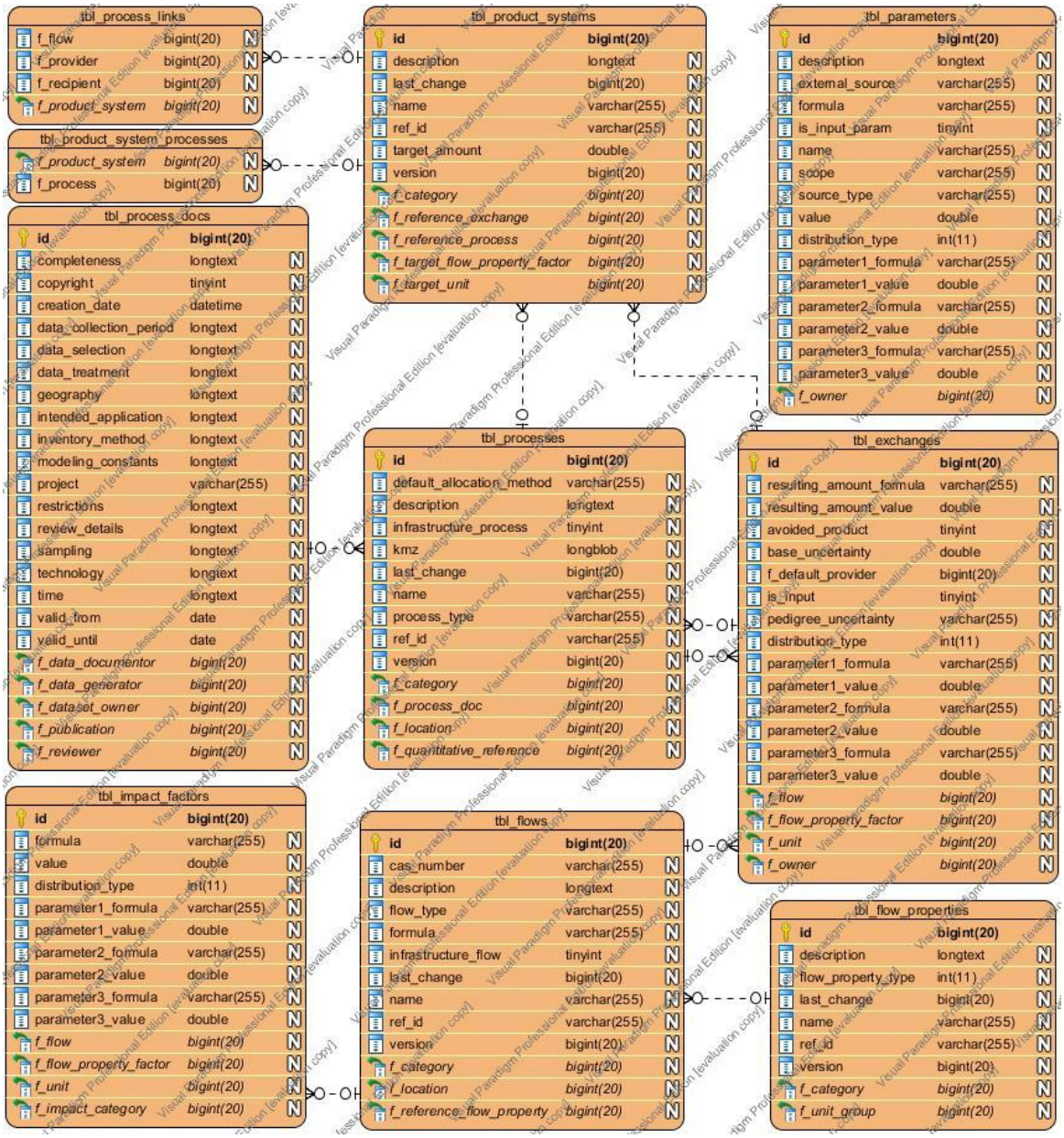


Figure 4: OpenLCA entity relationship diagram

5.2.2.1 tbl_categories

Categories is all the folder in the navigation panel of OpenLCA, except the first level.

f_parent_category

Description: Id of the parent category.

model_type

Value: [PROCESS | FLOW | FLOW_PROPERTY | UNIT_GROUP | IMPACT_METHOD].

Description: Type (first level label) of category in the navigation panel.

5.2.2.2 tbl_exchanges

Exchange is all the input and output flow of a process.

Is_input

Description: 0=> Output flow, 1 => Input flow

distribution_type

Value: [null|1|2|3]

Description: null→None, 1→Log normal, 2→ Normal, 3→Triangle, 4→Uniform

parameterX_value

Description:

Distribution type	Parameter1_value	Parameter2_value	Parameter3_value
Log normal	Mean	Standard Deviation	-
Normal	Mean	Standard Deviation	-
Triangle	Minimum	Mode	Maximum
Uniform	Minimum	Maximum	-

Table 2: Parameters meanings by distribution type

resulting_amount_formula

Description: Formula used to evaluate the parameter.

resulting_amount_value

Description: Value of the evaluated parameter.

5.2.2.3 tbl_flow_properties

Flow's unit.

Flow_property_type

0- User defined unit
1- Base unit

5.2.2.4 tbl_flows

Contain all flow: elementary and economic flow.

flow_type

Value: [ELEMENTARY_FLOW | PRODUCT_FLOW]

5.2.2.5 tbl_impact_categories

Contain all impact categories.

reference_unit

Description: Impact's unit express by a string.

5.2.2.6 tbl_impact_methods

Contain all the impact methods.

5.2.2.7 tbl_locations

Contain all location used in processes table.

5.2.2.8 tbl_parameters

All parmeters defined in processus or in openLCA.

f_owner

Process id or null for global parameter.

scope

Value: [PROCESS | GLOBAL]

Is_input

Description: 0=> Output flow, 1 => Input flow

5.2.2.9 tbl_process_docs

Contains the details on process contained in process table.

valid_from & valid_until

Date between the data is valid.

5.2.2.10 tbl_process_links

Contain all links between two process contained in a product system.

The root of the process tree is not contained in this table,

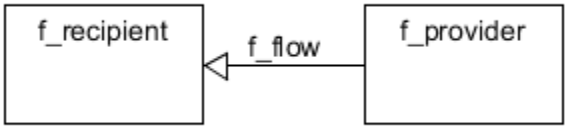


Figure 5: tbl_process_links

5.2.2.11 tbl_processes

Contain all process of the LCI database.

process_type

Value: [LCI_RESULT | UNIT_PROCESS]

5.2.2.12 tbl_product_system_processes

Contain all process used in a product system (root process too).

5.2.2.13tbl_product_systems

Contain all product system.

target_amount

Quantity for the functional unit.

f_reference_process

Root process.

5.2.2.14tbl_unit_groups

Unit groups represent a kind of measure. Ex: Length, mass, etc

f_reference_unit

Description: id of the reference unit of this group.

5.2.2.15tbl_units

Contain all the units used in flow.

name

Description: Unit symbol.

Conversion_factor

Description: Factor used to convert to the default unit in the unit group.

5.2.3 Back-end

In this section, all table needed to keep the LCA result. Keeping LCA result will allow the user compare easily two or many results. The system need to keep all information displayed to the user, so another user can change the input and output of some process and the result won't change.

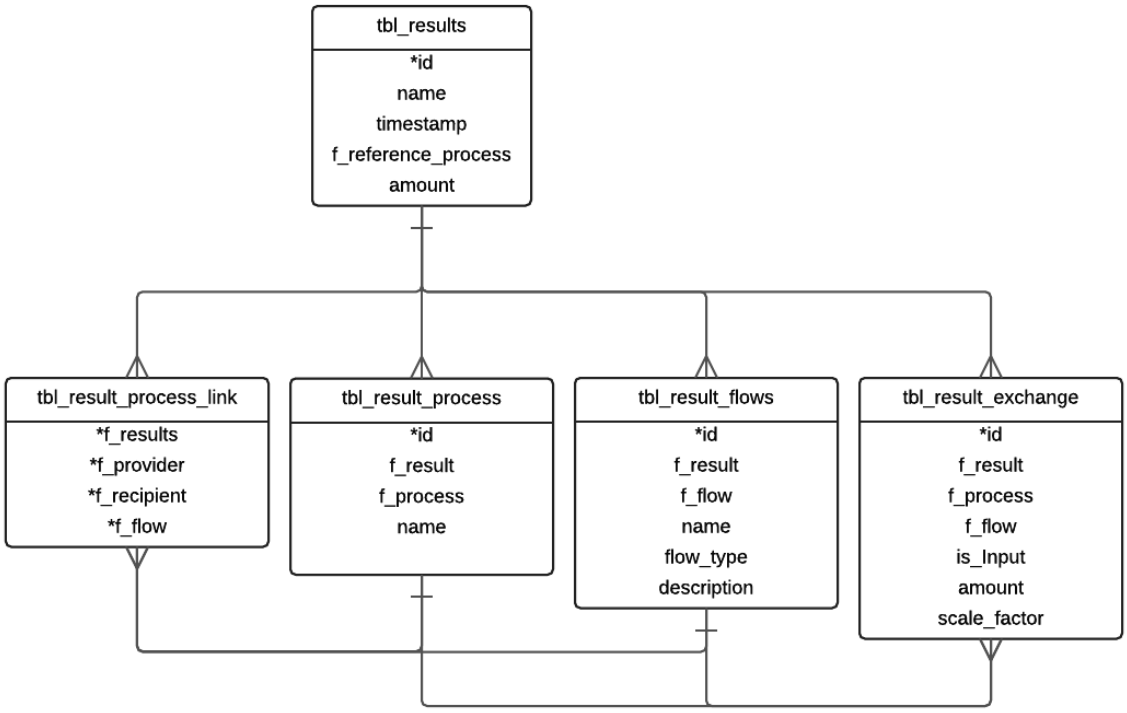


Figure 6: Result tables

5.2.3.1 tbl_results

Contain all the result header saved by the user.

amount

Quantity for the functional unit.

f_reference_process

Root of the process tree used for the result.

5.2.3.2 tbl_result_exchange

Exchange is all the input and output flow of a process from a result.

is_input

Description: 0=> Output flow, 1 => Input flow.

amount

Description: Value of the evaluated parameter.

scale_factor

Description: factor to apply to scale the input or the output to get the final amount needed.

5.2.3.3 tbl_result_flows

Contain all flow: elementary and economic flow in a result.

flow_type

Value: [ELEMENTARY_FLOW | PRODUCT_FLOW]

5.2.3.4 tbl_result_process

Contain all process used in a result.

5.2.3.5 tbl_result_process_link

Contain all the process contained in a result.

The root of the process tree is not contained in this table.

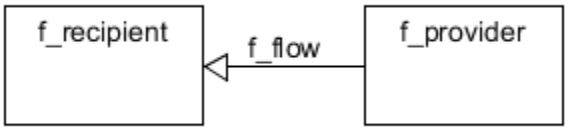


Figure 7: tbl_result_process_link

5.3 Spark

Spark is a big data framework with a huge scalability. His role is to distribute the computation describe in section 6 and to contain all the data needed. All the live data is contained in Spark and the system, can synchronize the data in MySQL, by the developer team actions, for validation purpose.

6 Algorithms

This section is used to explain algorithms used in UBUBI. To have a good algorithms, it need to fit with the data, we need to process. Let's describe the LCI database.

EcoInvent 3.1 cut-off database contain:

- 11301 process;
- Maximum product flow by process: 325;
- 19142 elementary flow;
- The biggest cycle is 7718 process (Kuczenski, 2015).

6.1 Create the process tree

6.1.1 Process tree topology

The process tree for a building have 2 parts: the foreground process and background process. UBUBI won't create background process. To create background process, an expert need to use OpenLCA and connect to the distant database. The foreground contain all the process that UBUBI create from the BIM model.

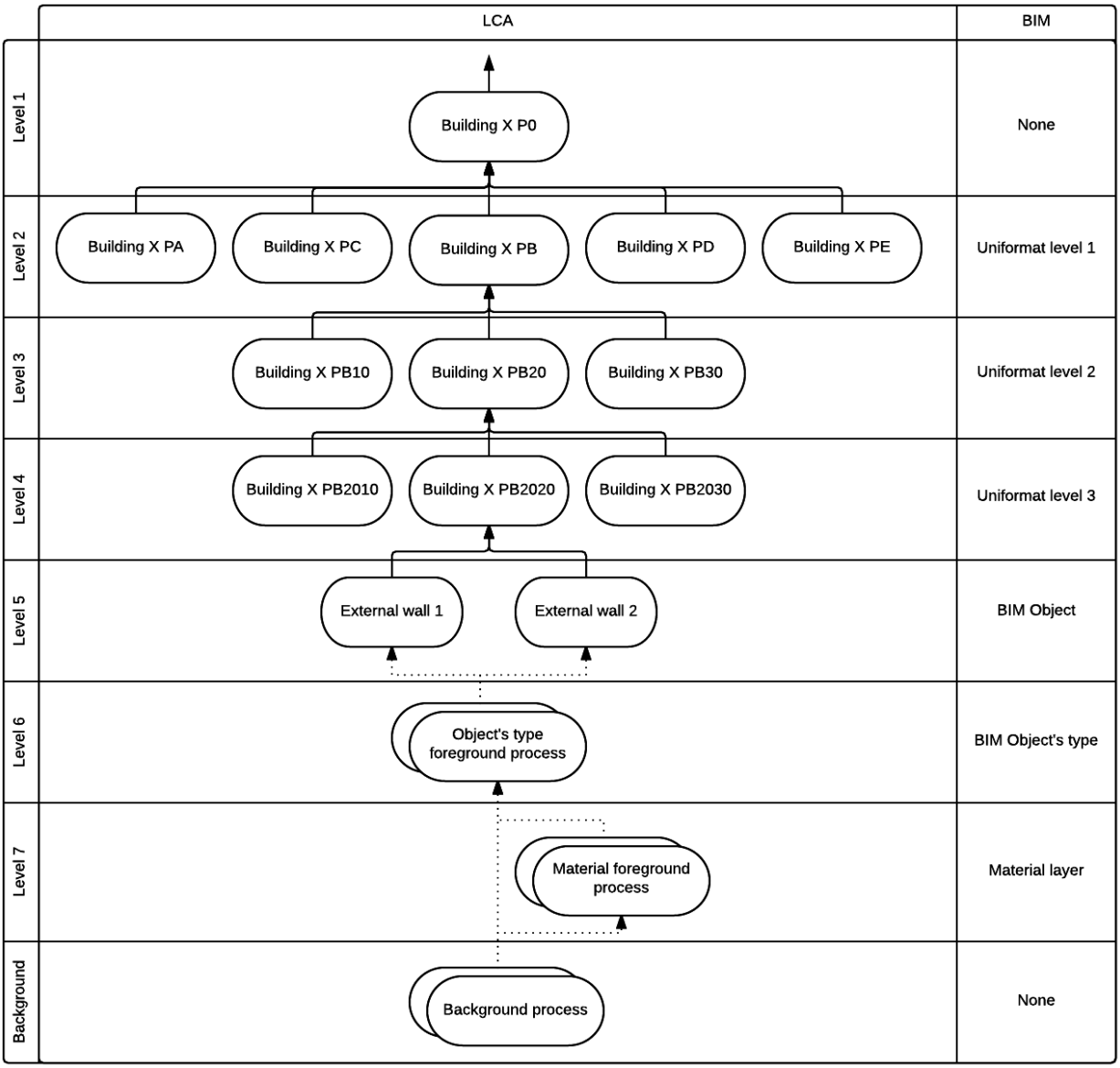


Figure 8: Foreground process topology

For the foreground process, the root is always the utilization of the building X. For the level 1,2,3 of the process tree correspond at the level 1,2,3 of Uniformat. The 4th level of the process tree is all the process to create the product represented by BIM object. In BIM model some object's type are compose of material layer. In the product system the object's type process can use material process instead using directly background process. Using material process instead background process allow to link less process in the foreground because one material is use in many object's type process. To support this feature a flag can

be swap to 1 or 0, to enable and disable the link with material process in the object's type process. For more detail check the Figure 9, all equation are specify.

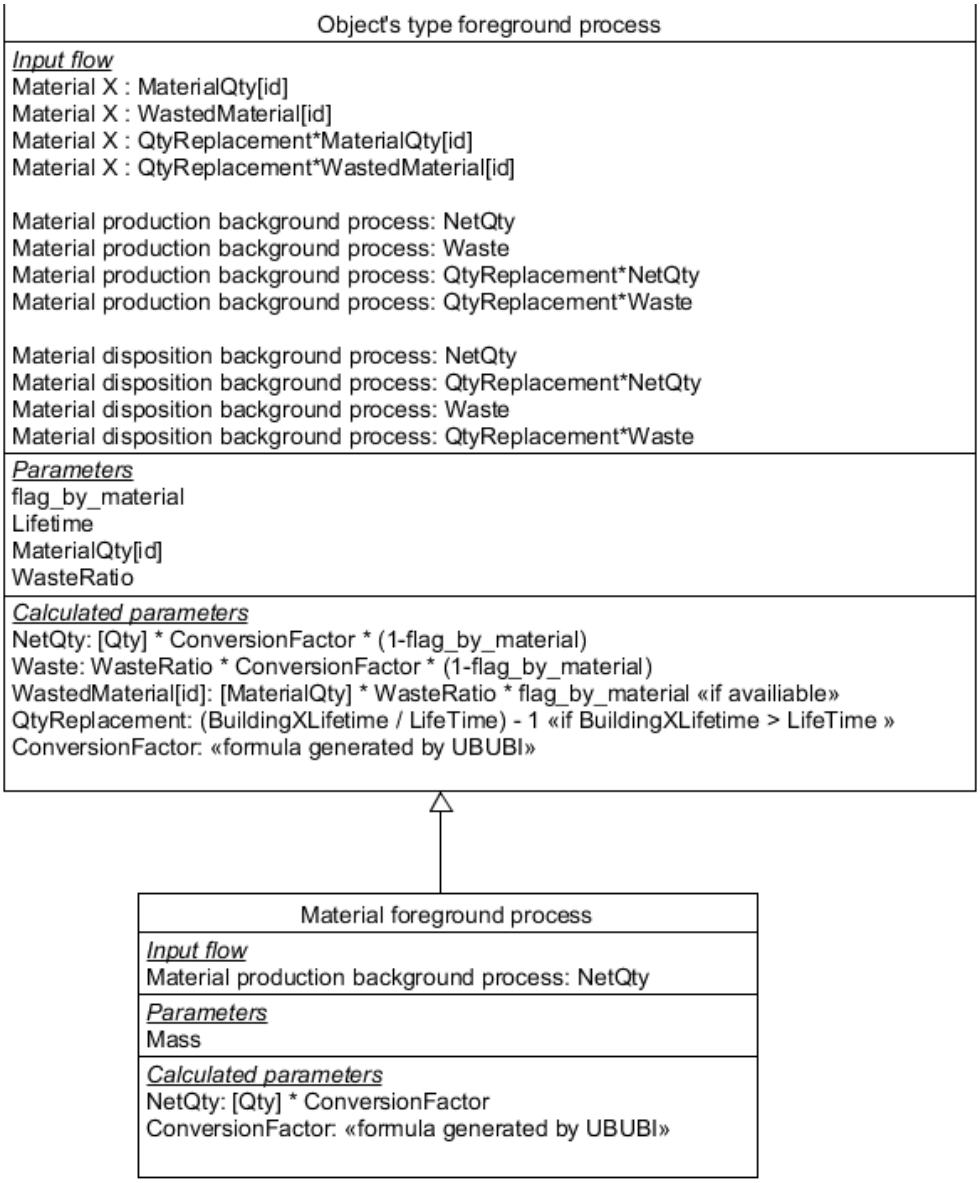


Figure 9: Object's type and material process representation

This typology is ideal for the creation and modification of the process three, because all the life cycle impact of one object's type is bound to one process. If it change, the user or the automation in UBUBI have only one place to change it.

This typology is maybe not the best way to explore the result. All stage of the life cycle have a different stakeholder. For example, the maintenance and replacement stage are more important for the owner than

the end of life of the building. For these purpose, a system of tag will be used to reorganize the data. The method to process all the tag is to be determined.

6.1.2 Process linking

For now, we will use the same method as OpenLCA. This method is contain in the class “ProductIndex” in “build” method. OpenLCA create the product system’s process tree by wide first and the detail for this method is describe below.

1. Add the root process for the product system in the list of process to handle
2. Initialize the list of handled process
3. While the list to process is not empty
4. Initialize the list of process to handle next
5. For each recipient in list of process to handle
6. Add recipient in the list of handled process
7. Get the list of all input product flow for the recipient
8. For each input product flow in the list
9. Find a process provider for this flow
10. Create a link between the recipient, the flow and the provider
11. If the provider is not in the list of handled process and not in the list of process to handle next then
12. Add the provider in the list of process to handle next
13. End if
14. End for each
15. End for each
16. Assign list of process to handle next in the list of process to handle
17. End while

To choose between different processes that produce the same flow, OpenLCA follow these criteria:

1. The default provider;
2. The desire process type (System process / Unit process);
3. The first found.

To save all the product system in the database, OpenLCA follow these step:

1. Delete every line of “tbl_product_system_processes” and “tbl_process_links” that refer to this product system
2. Write every flows in the table “tbl_process_links”.
3. Write every process in the table “tbl_product_system_processes”.

Warning: OpenLCA use native SQL for this.

Another method is used by OpenLCA to create a product system. This method introduce a cutoff and is contained in the class “ProductIndexCutoffBuilder”. This method won’t be analyzed or used for now.

6.2 Resolve the process tree

To resolve the linear system the fastest way is possible we need to characterize our data:

- The first try to resolve the linear system will be by using matrix representation.
- Two matrix is used, the technological matrix and the elementary flow matrix.
- The technological matrix is always square and reversible.
- This matrix is not triangular and contain decimal number (positive and negative).
- This matrix could be consider as a sparse matrix because the link between processes are few.

Matrix inversion is really cost full but for the first version it will be the solution used. The data collected will be used like a baseline to compare with.

6.3 Characterize the elementary flow

For this version of the application, we will use the same formula used by OpenLCA:

$$G = E * A^{-1} * B * f$$

Where:

G: Characterized impacts.

E: Characterization matrix.

A: Technology matrix.

B: Elementary flow matrix.

f: Scaling vector.

7 Standards used

7.1 Date’s format

All string that contain date use this format “AAAAMMDD” when is for storage or process purpose and “AAAA-MM-DD” when is for display purpose.

*This standard doesn’t affect numerical date like “DateTime” data type.

8 References

Autodesk. 2015. « Building Design Software | Revit Family | Autodesk ». < www.autodesk.com/products/autodesk-revit-family/overview >. Consulté le 2015-10-16.

ASTM E2921-13, Standard Practice for Minimum Criteria for Comparing Whole Building Life Cycle Assessments for Use with Building Codes and Rating Systems, ASTM International, West Conshohocken, PA, 2013.

Institute for Environment and Sustainability. 2012. « International Reference Life Cycle Data System (ILCD) Data Network - Management of UUID and version number of data sets ». Publications Office of the European Union, p. 25.

ISO. 2006a. *Management environnemental : analyse du cycle de vie : exigences et lignes directrices*. Genève: ISO, vi, 49 p. p.

ISO. 2006b. *Management environnemental : analyse du cycle de vie : principes et cadre*. Genève: ISO, vi, 23 p. p.

Kuczenski, Brandon. 2015. « Partial ordering of life cycle inventory databases ». *The International Journal of Life Cycle Assessment,* vol. 20, nº 12, p. 1673-1683.