

# Potential application of UML for WiWiT

Takeru Shibayama

September 24, 2014

## Contents

<b>1</b>	<b>Business Modeling Methods</b>	<b>1</b>
<b>2</b>	<b>Basics of UML</b>	<b>2</b>
<b>3</b>	<b>Class Diagram</b>	<b>3</b>
3.1	Class . . . . .	3
3.2	Association . . . . .	4
3.3	Package . . . . .	4
3.4	Simplified diagram . . . . .	4
3.5	Standard diagram . . . . .	5
<b>4</b>	<b>UML modelling software</b>	<b>6</b>

## List of Figures

1	Basic UML Class . . . . .	3
2	Simple UML Class . . . . .	3
3	Basic example of UML association . . . . .	4
4	Simplified UML Diagram . . . . .	5
5	Standard UML Diagram . . . . .	6
6	Basic example of UML association . . . . .	7

## 1 Business Modeling Methods

A number of business modeling methods have been developed in the past. Among them, the mainstream ones are:

- Organigramm (organizational chart);
- Flowchart;
- DOA (Data Oriented Approach);
- IDEF (Integrated DEFinition methods) Family;
- BPMN (Business Process Modeling Notation);
- UML (Unified Modeling Language);
- CLD - SD (Causal Loop Diagram and System Dynamics).

Among these, flowchart, DOA, IDEF, BPMN and CLD-SD are used to model the business **process**, while organigramm is used to depict the business **structure**. UML is a set of 13 diagrams and this can handle both process and structure.

Organigramm is a classical method that was developed in the middle of the 19th century in the United States, and it is used widely today. UML is a new method developed in the 1990s originally for object-oriented software design, while since the beginning of the new millennium applications for business modeling have been sought by many and have been proved applicable. There is no official standardization for the organigramm, while the standard for the UML is managed by Object Management Group and the standard is published on <http://www.uml.org/>.

## 2 Basics of UML

The most useful diagrams, among 13 diagrams defined for the UML family, for organization modeling are:

- **Class diagram** to depict functional structures;
- **Activity diagram** to depict business procedures;
- **Use case diagram** to depict user interface.

When it comes to create a competence map, the most useful among these three is the **class** diagram. The class diagram is the most basic and most used diagram among the UML diagram family, and this is the one that can be used for depiction of competences in this research. Class diagram is used to depict the static relationship between various objects and thus this fulfills the requirements.

To describe process, among UML family, the most useful one is **Activity diagram**, which depicts flows of activities. This is essentially an extension of the flowchart with a set of specified marks that is useful for software design. This is fairly straightforward and no extra description is given. Especially useful is however a way to describe timing within the grammar of activity diagram.

## 3 Class Diagram

### 3.1 Class

The most basic element of the class diagram is *class*, each of which is shown by a box.

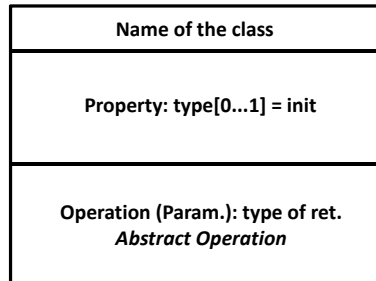


Figure 1: Basic UML Class Source: own graphics

The basic box consists of three compartments: the one on the top is for *name* of the class, the one in the middle is for *property*, and the one on the bottom is for *operations*, as shown in the figure 1. The property is to hold variables or parameters, while operation is to describe the actions that this class can carry out. In the language of object-oriented programming, operation is also called for method (e.g. Java).

A simplified class diagram can contain the UML class only with one compartment that shows the name of the class as shown in the figure 2.



Figure 2: Simple UML Class Source: own graphics

More sophisticated diagram can have the fourth compartment below the operation, called nested classifier. Here, this class can hold another class within it. This function is fairly advanced and not much used.

In the context of modeling organizational structure, a class can depict either an organizational unit or a competence. When the focus is on competence, it is more appropriate to model based on competence, and use the package to depict organizational unit (see below).

### 3.2 Association

Relationships between two classes are shown with lines. This line is called *association* in UML. The association can have some additional information written beside it:

- **Multiplicity:** showing the range of the numbers that each side of the association can take, or in other words, if the relationship is 1 to 1, 1 to 0, 1 to n, etc. Instead of "n", asterisk (\*) is used in UML.
- **Role:** the role of the class at the both ends can be shown at the end of the line.
- **Navigability:** a simple arrow attached to the end of the association indicates that the relationship shows the direction of the relationship. If there is no arrow on the both sides, this means either unknown navigability or bidirectional navigability.
- **Name** of the association can be shown next to the line.

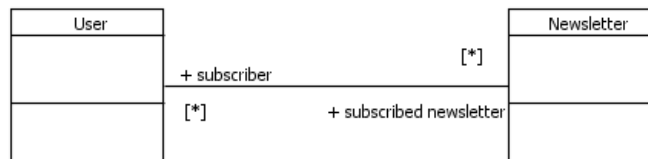


Figure 3: An example of association (Source: own graphic)

### 3.3 Package

UML's Class can be grouped using *package*. The role is self-explanatory as shown in the figures 4 and 5 and thus no additional explanation is made here.

In the context of organization modeling, the UML package is used to show the organizational or institutional unit as shown in the figures below.

### 3.4 Simplified diagram

In the standard UML class diagram, class has always three compartments, the one on top being the name, in the middle being the property, and the bottom one being operations, as explained above. Among them, the very basic element is only the name of the class: indeed it is possible to construct an UML diagram just with class names and associations. For the diagram constructors, this has an advantage in that they do not have to consider each class's property and operation; especially in the context of organization modeling, it is advantageous in that the diagram constructor can focus on the structure itself. For the readers, the simplified diagram is easier to understand compared to the standard diagram and it has better readability. Especially for those who are not used to the UML, this readability is a great advantage.

A simplified diagram is shown in the figure 4.

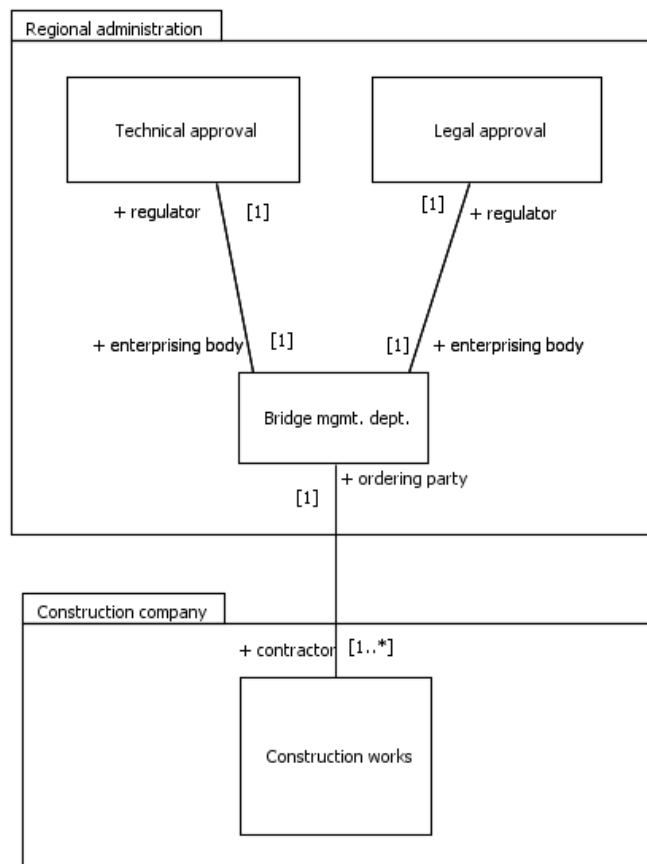


Figure 4: Simplified UML diagram without the property and operation compartments (Source: own graphic)

### 3.5 Standard diagram

The standard UML diagram contains, besides the name of the class, each class's properties and operations in the diagram. The diagram is thus more comprehensive while readability might be inferior to the simplified version of the diagram.

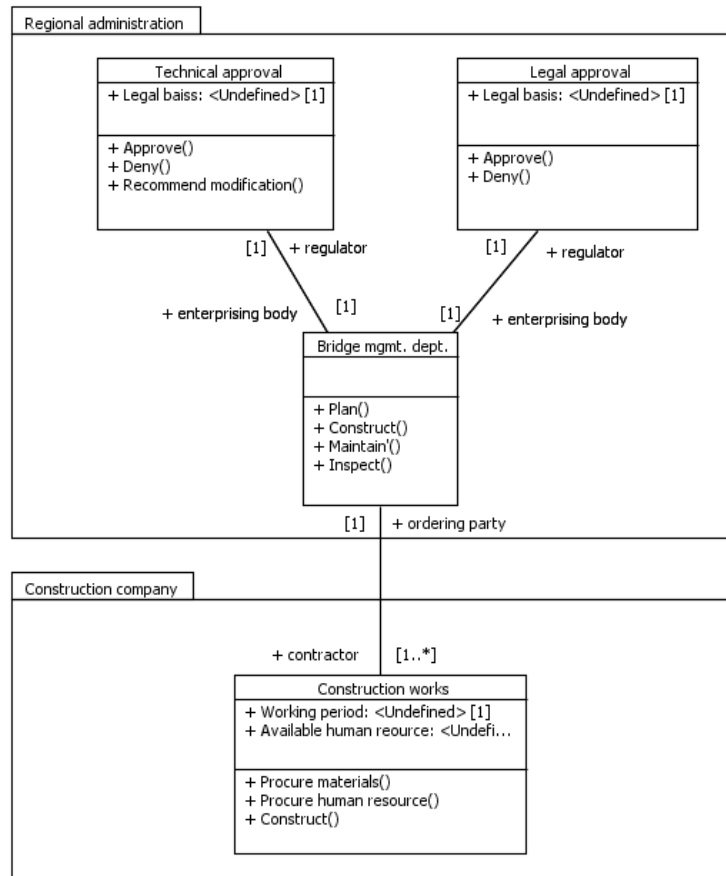


Figure 5: Standard UML diagram (Source: own graphic)

## 4 UML modelling software

There are a number of software packages that have been developed for UML modeling. A comprehensive list is on Wikipedia as List of Unified Modeling Language tools. There are a number of commercial ones such as *Prosa UML Modeller* or *BOUML*.

One of the most comprehensive one is *Papyrus* that is developed by French *Commissariat à l'énergie atomique et aux énergies alternatives* (English: Atomic Energy and Alternative Energies Commission). This is based on Eclipse and provided as Open Source, and provides

strong graphical modeling tool for UML. This works as a plug-in to Eclipse and thus works on the standard Windows environment.

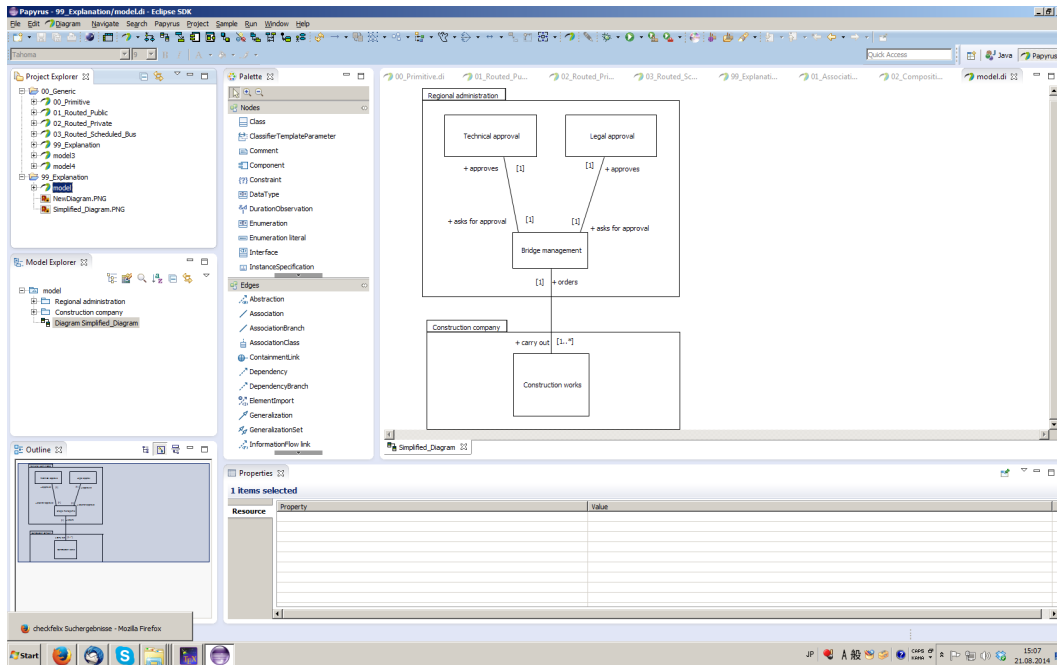


Figure 6: Screen of Papyrus on Eclipse (Source: own graphic)

Papyrus is a very comprehensive tool that calls for a good understanding of UML. One of the disadvantage is that the documentation of the software has not yet been fully provided and thus to find some function users have to search the documents available on-line.

The simplified UML diagram can be also created with standard office software that supports graphical drawings such as Microsoft PowerPoint or Visio. This is more advantageous in that the modellers do not have to learn the complicated user interface of Eclipse and Papyrus. Such standard office software's outputs are oriented on classical paper and thus the size of the diagram can be often limited by the maximum paper size (often DIN A3), while this may be overcome with user-defined paper size that is often supported in modern software.