Universität Rostock
Traditio et Innovatio

Professur für Geodäsie und Geoinformatik
Prof. Dr.-Ing. Ralf Bill/M.Sc. Markus
Berger/M.Eng. David Hennecke

Tutorial - Exercise:

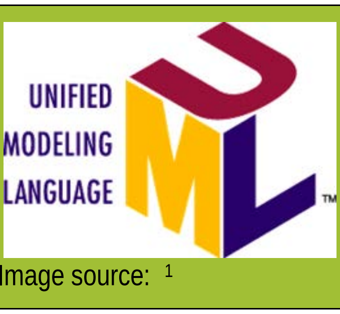
UML diagrams

OPEN GEO EDU

UML

UML can model both static and dynamic aspects of a system. Static aspects are described with class diagrams, package diagrams, component diagrams, distribution diagrams and composition structure diagrams, dynamic with state, activity and sequence diagrams.

The UML Class Diagram

In this course unit, you will practically apply the basics of the UML class diagram and the UML activity diagram you learned in the lecture tutorial.	 Image source: [1]

Introduction

To consolidate what we learned in the lecture, we will now look at how to draw ER or UML diagrams. We will focus on a simple example, a UML class diagram. We also use a free, beginner-friendly online tool called draw.io, which makes it easy to draw the desired structures, but has no advanced UML features. Instead of draw.io, programs such as Microsoft Visio can also be used. Scope and usability regarding UML are similar here.

To create formal, standards-compliant UML diagrams, you need to use special software such as ArgoUML or Enterprise Architect.

The following task builds on the lecture part. More precisely, on the class diagram designed there. As we remember, individual steps were always illuminated with sections of the overall example. The task is therefore obvious: Put all the individual parts together to form an overall model. After completing this task, the concepts of the class diagram should be solidified, as should the operation of the draw.io tool. With this you can easily create diagrams of all kinds.

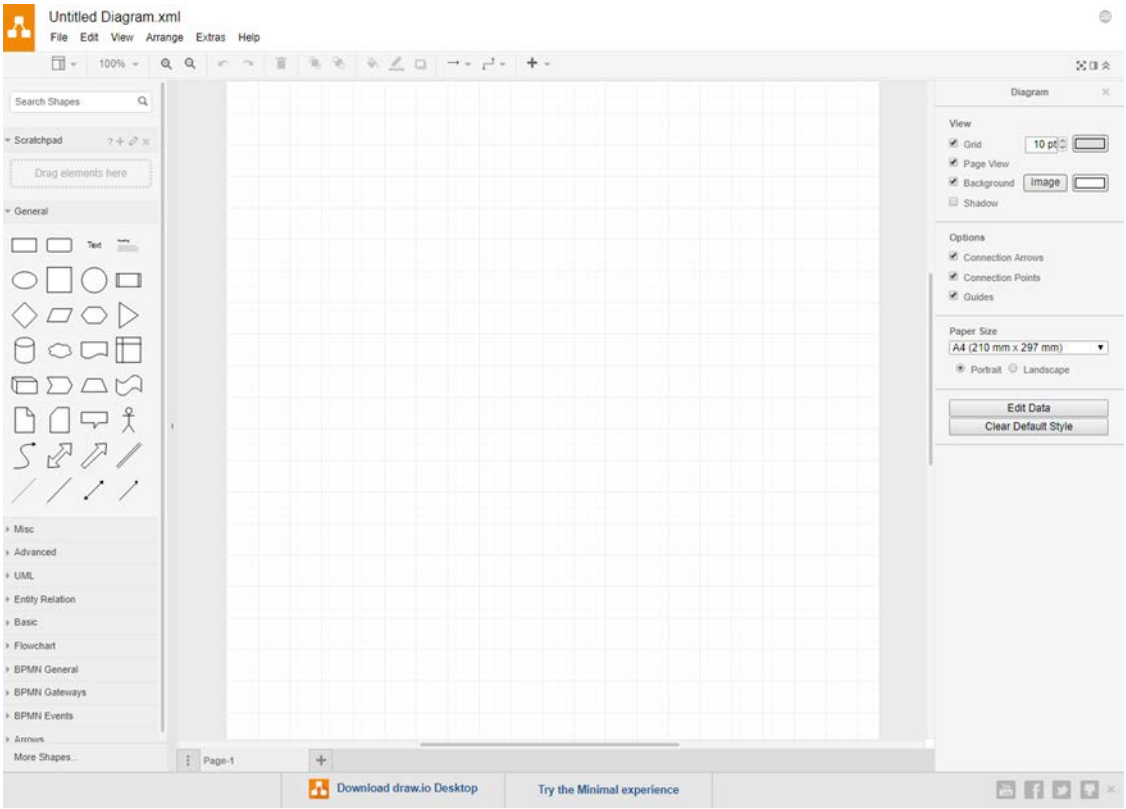
Sequence

Werkzeug draw.io

First we open draw.io. Then we select the local storage on the "Device", and the creation of a new diagram using the option "Create New Diagram". We stick to the "Blank Diagram" template. It should now welcome us to the interface shown in Figure 1.

[1] Picture source: https://de.wikipedia.org/wiki/Unified_Modeling_Language#/media/File:UML_logo.svg

The operating elements are distributed as known from many modern applications. At the top there is a menu bar, on the left there are the shapes for the diagrams, and on the right a context menu for the currently selected object, or the entire diagram. In the middle is the drawing field, which starts as an A4 page, but can be extended in any direction for as long as you like.



Untitled Diagram.xml

Diagram

View
☑ Grid 10 pt
☑ Page View
☑ Background
□ Shadow

Options
☑ Connection Arrows
☑ Connection Points
☑ Guides

Paper Size
A4 (210 mm x 297 mm)
● Portrait ○ Landscape

Fig. 1: User interface draw.io

Creating a Class Diagram

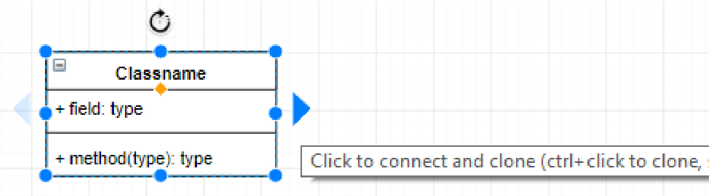
We open the "UML" tab in the diagram window with one click. Here we see the elements of all possible UML diagrams. It is helpful to go through all blocks once to avoid confusion later. It should also be noted that there are different notations, especially for line labels. We adhere to the established notation in the lecture part.

Let's start with the element marked on the right: The class. A rectangle with areas for titles, attributes and methods. It makes sense to first distribute all classes on the field so that you can arrange them before connecting.

We need the following classes from the lecture, with their attributes and methods:

- congregation
- settlement
- district
- plant
- tree
- moss

To create a class, we simply drag and drop the element onto the drawing field. We do this six times. Single blocks can be edited with a double click. New attributes and methods can be added by selecting an existing element and then clicking one of the arrows.



Classname
+ field: type
+ method(type): type

Click to connect and clone (ctrl+click to clone,

Fig. 2: Class diagram

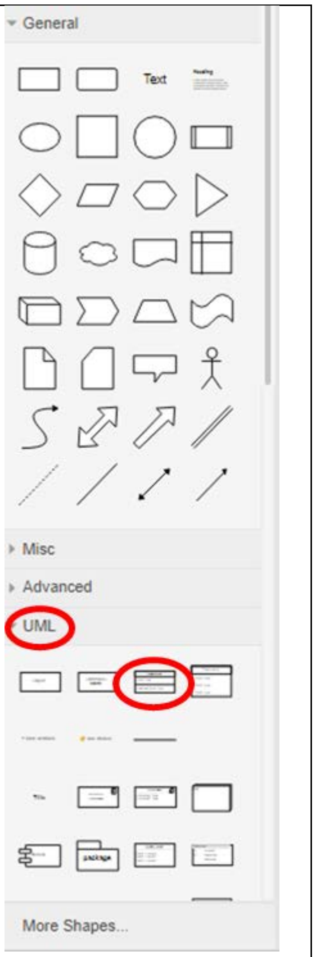


Fig. 3: draw.io UML

In this way we now enter all attributes and methods from the lecture. The result is as shown in the following figure.

Now, of course, we still have to arrange the classes and connect them with associations. There are two possibilities here. First, you can drag the desired arrows directly from the list of UML elements, and then drag and drop the ends to the attachment point of the classes.

Alternatively, you can click directly on the rectangles' connection points without selecting an arrow beforehand, and automatically drag an arrow with the mouse button pressed. The parameters of the arrow, such as ends and thickness, can then be set in the context menu on the right.

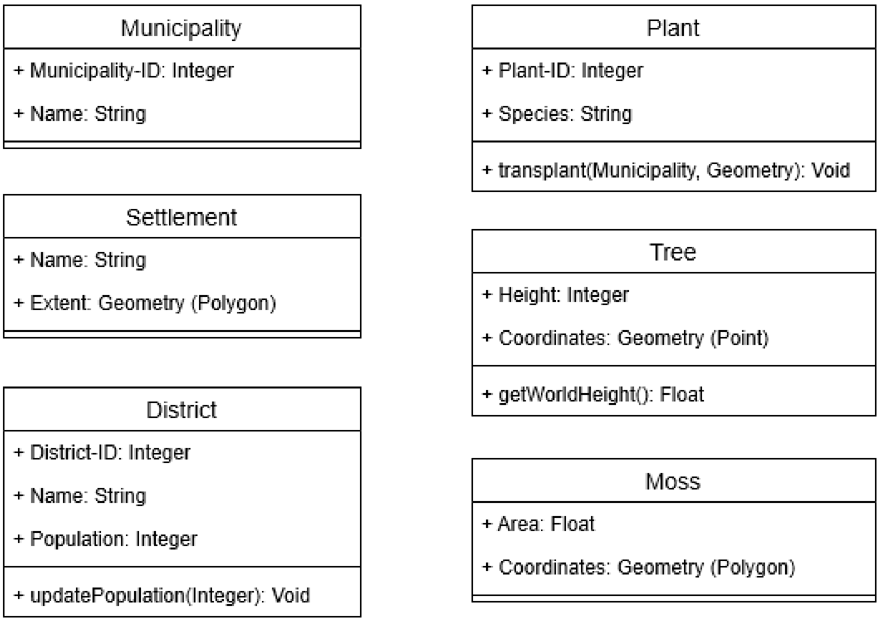

Municipality
+ Municipality-ID: Integer
+ Name: String
Plant
+ Plant-ID: Integer
+ Species: String
+ transplant(Municipality, Geometry): Void
Settlement
+ Name: String
+ Extent: Geometry (Polygon)
Tree
+ Height: Integer
+ Coordinates: Geometry (Point)
+ getWorldHeight(): Float
District
+ District-ID: Integer
+ Name: String
+ Population: Integer
+ updatePopulation(Integer): Void
Moss
+ Area: Float
+ Coordinates: Geometry (Polygon)

Fig. 4: UML class diagrams for 6 object classes

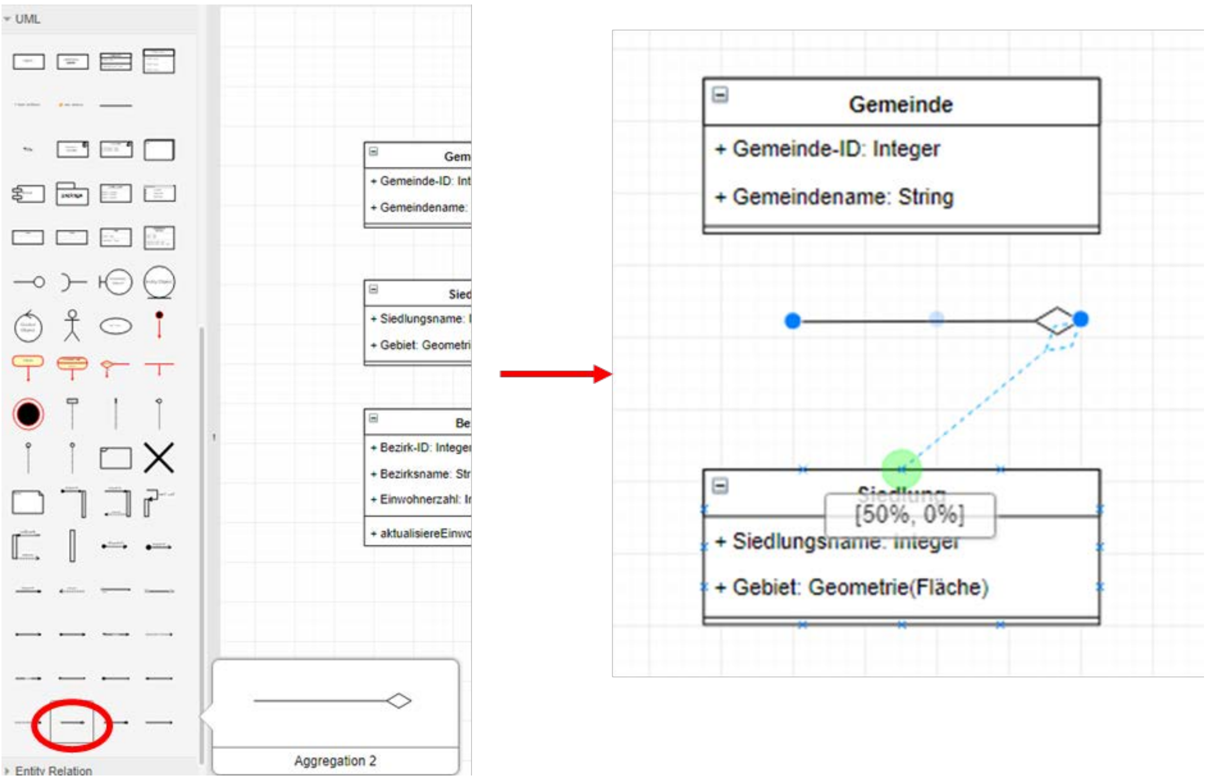


Fig. 5: Relationships between the object classes

Whatever the arrows are now placed, cardinalities are still missing. Text can be attached to arrows by double-clicking on the desired position. In our case, this means at both ends. With the appearing yellow dot finer shifts can be made, so that the arrow is not covered by the text:

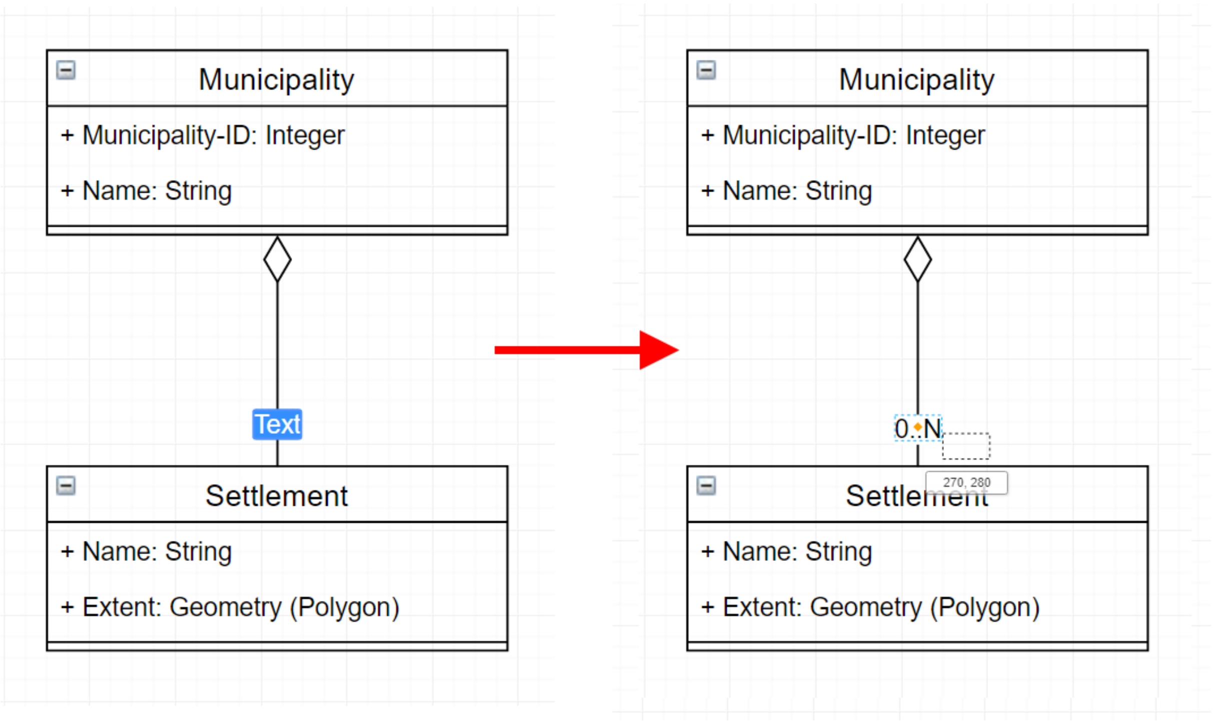


Fig. 6: Cardinalities

With these means it is now possible for us to create the whole diagram. The final result could look like this:

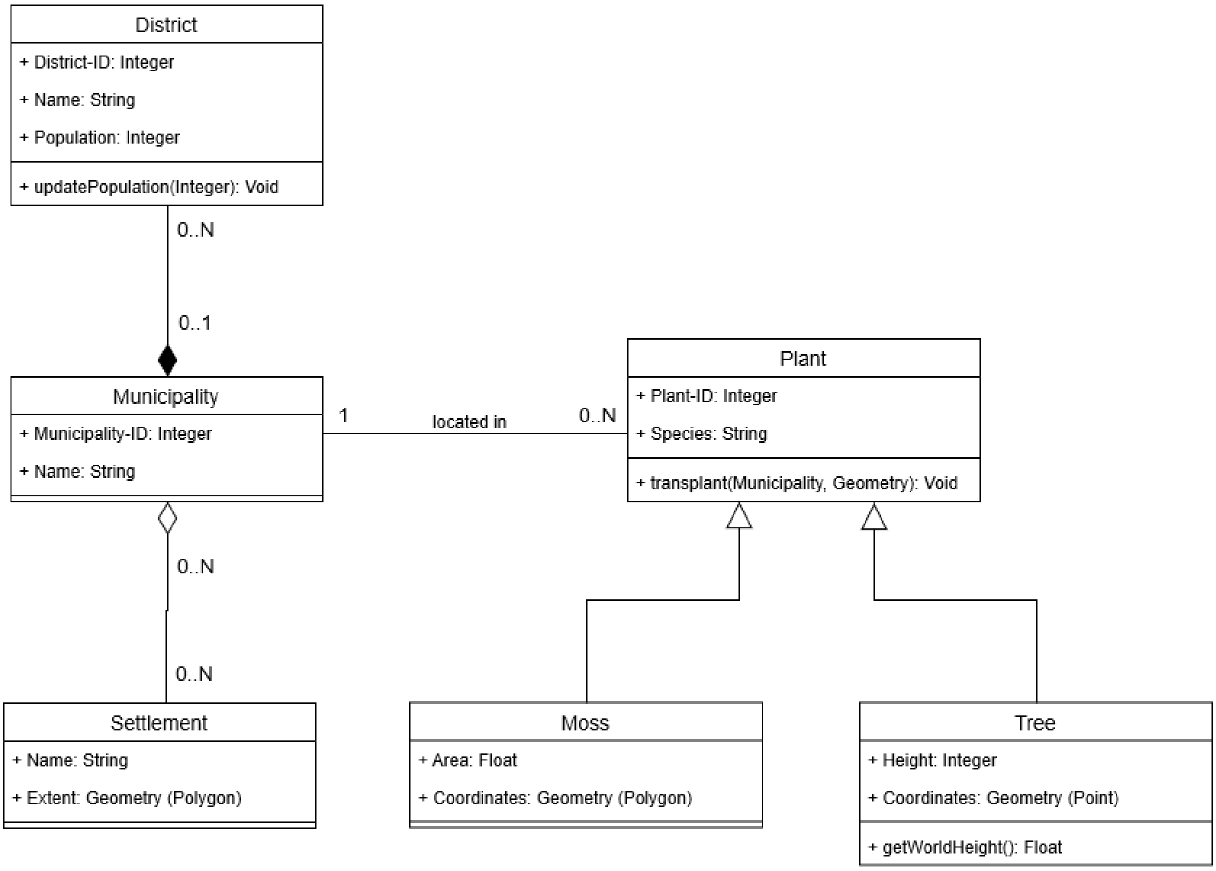


Fig. 7: Cardinalities

Since this diagram is only a collage of different examples, it is not designed for a real application. However, it does illustrate well the different aspects of modeling with class diagrams. For a practical implementation of a tree cadastre it is important to think about the role of geometry (closely related to knowledge about spatial databases, see OpenGeoEdu course) and possibly about programming problems like interfaces and abstract methods. These, however, exceed the scope of this course, and should be looked up in appropriate technical literature, or learned in more specialized courses.

The UML activity diagram

The activity diagram graphically illustrates the networking of elementary actions and their connections with control and data flows and is therefore well suited to illustrate the workflow in a GIS evaluation process.

Activity is a model for a behavior that is described as a set of elementary actions between which control and data flows exist. Actions are the elementary building blocks for modeling a behavior. An activity diagram depicts the logical flow of a system as a series of actions, connections, and possibly choices and conditions and is therefore comparable to a classical flow diagram. It is particularly suitable for modeling business processes and procedures. The aim is to identify the logical steps in a use case. The individual work steps of the overall process are represented as activities. Who carries out the activity can also be represented. Possible sequences of activities are marked with arrows.

An activity diagram begins at a start state, which is transformed into a final state by activities (both represented by circle symbols). An activity is displayed as a rounded rectangle, has a name, and acts on a class. The control flow determines the sequence of the activities. Branches based on conditions are possible here, activities can be synchronized again.

Create Activity Diagram

Starting from a start node (black circle symbol), the process is described by the sequence of activities (rounded rectangles). Decisions/branches are described by diamonds. The input and output products are named at the edge of the activity diagram.

The following figure describes the processing sequence in the GIS exercise "Sustainable Development Goals", in which the two input data sets SDG indicator table and country boundaries are processed and then linked in ArcMap and visually presented in a result map. The evaluation in the form of diagrams, texts and maps is then combined in a scientific document and submitted for evaluation.

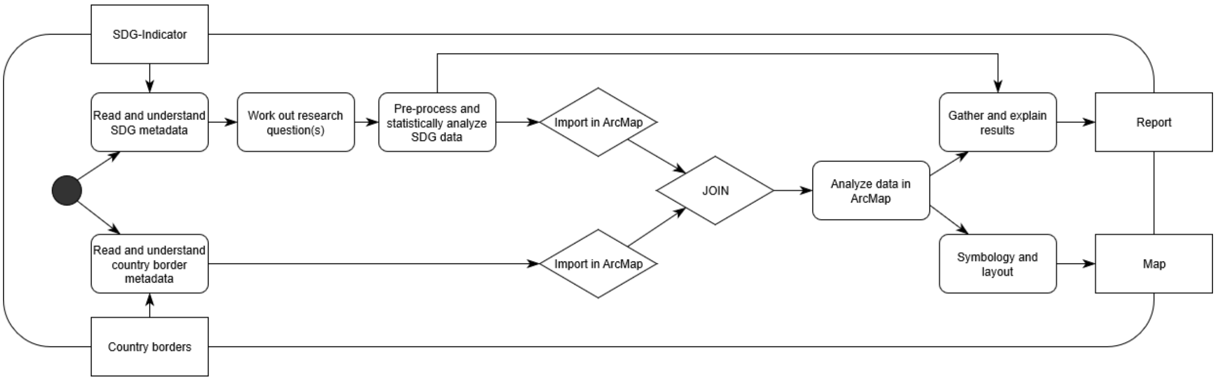

SDG-Indicator
Read and understand SDG metadata
Work out research question(s)
Pre-process and statistically analyze SDG data
Import in ArcMap
JOIN
Analyze data in ArcMap
Gather and explain results
Report
Read and understand country border metadata
Import in ArcMap
Symbology and layout
Map
Country borders

Fig. 8: Activity diagram for the Sustainable Development Goals exercise.

Literature

Bill, R. (2016): Grundlagen der Geo-Informationssysteme. 6th edition. Wichmann publishing house. Offenbach-Berlin. 867 pages. Chapter 4.2.

Booch, G., Rumbaugh, J., Jacobson, I. (1999): The UML User Manual. Addison-Wesley. 592 pages.