

3815ICT-Software Engineering Workshop 9

Activity 1

Study the entry in Wikipedia for the “ARGO-UML tool” (<https://en.wikipedia.org/wiki/ArgoUML>). It is a short page, and maybe you can also investigate what is the Eclipse Public License.

We will be installing this software and using it in this workshop.

Response 1

Argo is an open source UML diagramming application which was developed by Jason E. Robbins. The software was built in Java and is available on any platform supporting Java SE.

Activity 2

There are several ways to install Argo-UML. We will recommend the stand alone jar file that does not require a connection over the Internet. The Argo-UML project is at argouml.tigris.org/

This page has a download button, but I do recommend to explore the “Other formats” at argouml-downloads.tigris.org/argouml-0.34/ Here, I recommend to download the latest binary distribution. Then, depending on where you unpack the binaries you start the package as follows.

```
java -jar argouml.jar
```

Response 2

Installation of ArgoUML was reasonably simple. I am running it in the ROS-kinetic environment.

Activity 3

It is recommended that you follow the complete Argo-UML tour; however, here we will just focus in a part of Building a Class Diagram.

The entire tour start at argouml.tigris.org/tours/ but the Building a Class Diagram part starts at argouml.tigris.org/tours/bdClassDiagram.html

Follow the 13 steps of this part of the tour. Not everything that is required is fully explained, but most of it is (you can also explore). For example, to make the BOX of the class Teacher light blue you need to use the presentation tab and not the properties tab. Also, depending on the version of Argo-UML and of course if you re using Linux, MacOS or Windows, the appearance of menus and other GUI elements may change. Figure 1 shows how the project looked for me at the end.

Response 3

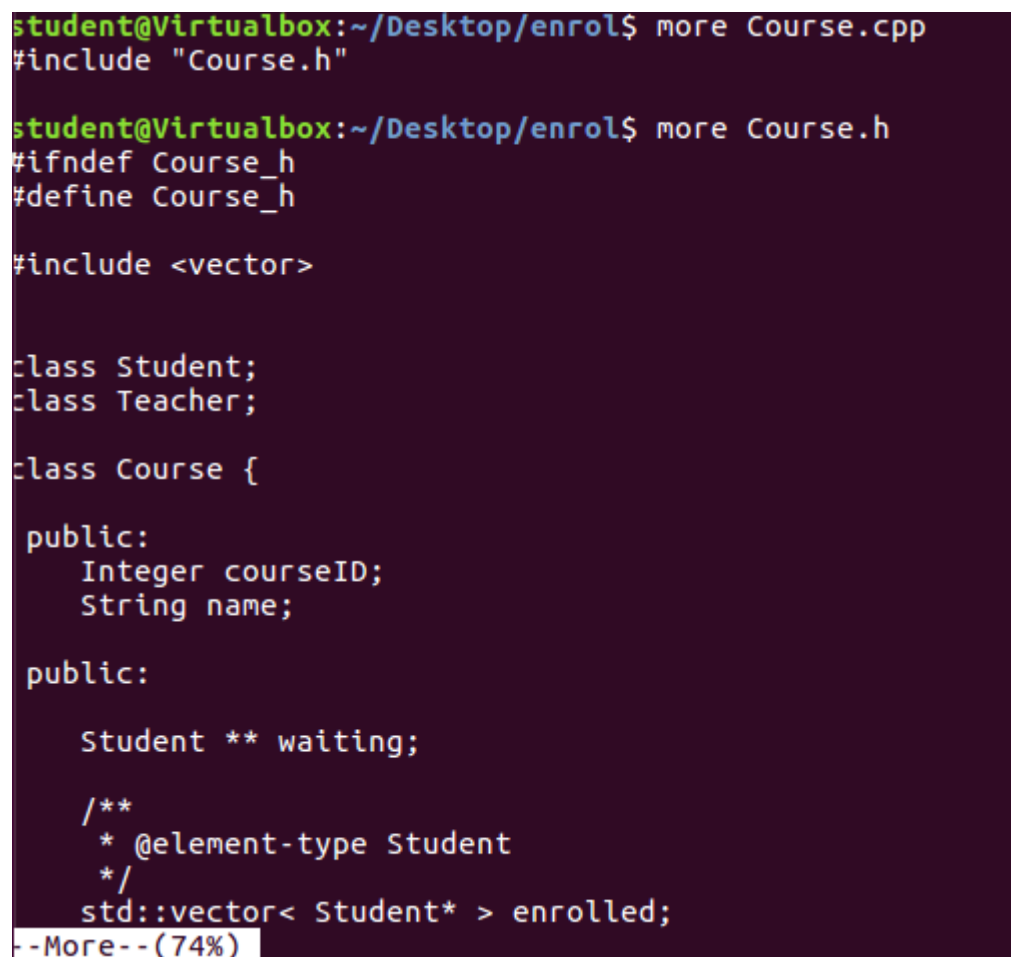
element. It is simpler to generate code from a static diagram, as stated, this simply creates a class structure which can then be expanded upon with manually written code. There is a set of basic rules which are followed, Eg, 'A class will become a class'. This will create a blueprint. Dynamic modelling conversion is more complex. There are different ways to create interactions with function calls and there are necessary implementations when dealing with asynchronous functions. Use of dynamic modelling conversion however, will generate a more complete piece of code in conjunction with static diagrams.

Activity 5

Use the Menu Tab for "Generation" and select to generate code for all the classes in the project. When the pop-menu shows up select to generate code for all the classes in C++ and also in java (note that there is an option to also add SQL).

Note that for each class, in C++ we have two files, a .h file that defines the class structure and a .cpp file that defines the class behavior. With an editor examine the file Course.h. It should look like Figure 2.

Response 5



```
student@Virtualbox:~/Desktop/enrol$ more Course.cpp
#include "Course.h"

student@Virtualbox:~/Desktop/enrol$ more Course.h
#ifndef Course_h
#define Course_h

#include <vector>

class Student;
class Teacher;

class Course {

public:
    Integer courseID;
    String name;

public:

    Student ** waiting;

    /**
     * @element-type Student
     */
    std::vector< Student* > enrolled;
--More-- (74%)
```

Running Course.h Gave this output.

Activity 6

Consider the code generated above in Figure 2. Does the UML diagram in Figure 1 display all the dependencies? How about the file Student.h, this file has `#include "Grade.h"`?

Given an object of the class Student, how would we find the course(s) the student is enrolled in, that is what would the C++ code need to do, ie, what is the algorithm and data structures used? Given an object of the class Course, how would we find the students enrolled in the course? Given an object of the class Course, how would we find the Teacher teaching the course?

Are there any instruments in the code generated to ensure that given a Student s , when we find the set C of courses s is enrolled in, and then for each course $c \in C$, we find the students enrolled in c , we always find s among them?

How would we find given a teacher t , the courses that are delivered by t ? What part of the UML diagram implies this?

Can you answer the same questions with the java generated code?

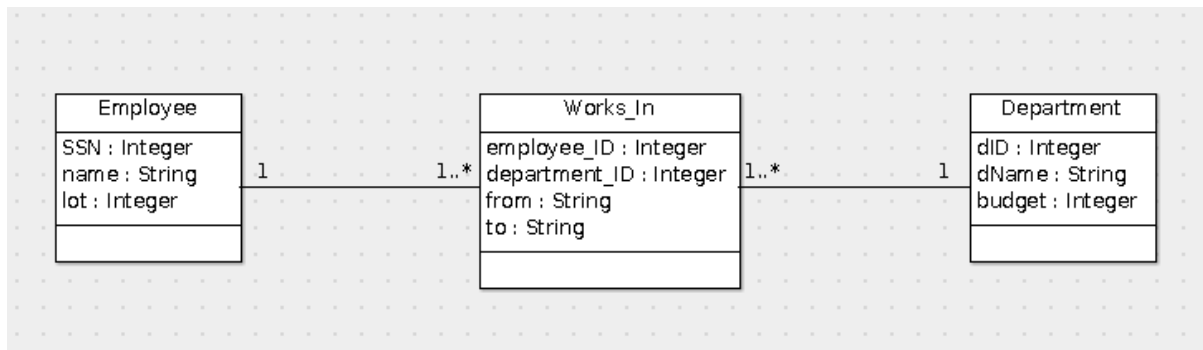
Response 6

The information in Figure one has the required information necessary in terms of dependencies. Observing the class student, there has been a vector generated which corresponds to the courses that the student is enrolled in. The same applies to the courses class which has a vector of students. At this point there appears to be no method of either class to populate this vector. This also applies to the Teacher which teaches each course, There is an element generated which corresponds to the 0 – 1 relationship with the Teacher class. The Classes will require a method which enrolls students to courses and teachers to courses based on their cardinality. There are no real instruments provided by the code which would suggest that we can find any of the information we might require of this structure. I believe that a more complete program would require a dynamic diagram along with a static one. The dynamic diagram would model the interactions between each Class and a code generator could extrapolate on this to determine correct methods to define the relationships between classes.

Activity 7

Consider the Entity-Relationship Diagram in Figure 3. Use Argo-UML to design the corresponding UML class diagram. What are the challenges? Does Argo-UML generate SQL code for the corresponding model? What are the difficulties?

Response 7



I put together the following diagram to illustrate the contents of the ERD. It is difficult to generate a model which correctly depicts the data structure for conversion to SQL as there is no way to add constraints such as Foreign Key and Primary Key (as far as I am aware).

I received the following error message upon attempted generation.

Model not valid
Primary key missing for Employee
Primary key missing for Works_In
Association name found more than once
Primary key missing for Department
Foreign key attribute missing for association

These capabilities are inherent to the construction of an SQL database and therein lies the difficulties of lifting an SQL structure from a model such as this.