Teaching UML in Moodle environment

Zbigniew MROZEK[#1]

[#]*Malopolska Higher Vocational School in Cracow*

(Małopolska Wyższa Szkoła Zawodowa w Krakowie)

PL 31-532 Kraków, ul W. Pola 4

[1]zbigniew@mrozek.org.pl

***Abstract*— This paper describes author's experience in teaching object oriented programming with UML in a Moodle environment.**

I. Introduction

Author is teaching object oriented programming with UML from year 1999. Moodle and Blackboard were used by author to manage educational process. After one semester, Moodle was chosen as more suitable environment.

II. The teaching environment

The teaching environment may be called virtual learning environment (VLE), Learning Management System (LMS), Course Management System (CMS), Learning Content Management System (LCMS), Managed Learning Environment (MLE), Learning Support System (LSS) or Learning Platform (LP). The actual name depends on software package used to support the process and on services offered by that package.

There are many teaching environment software packages. The best known open source environments are: ATutor, Claroline, Dokeos eFront, Fle3, ILIAS, KEWL, LON-CAPA, Moodle, OLAT, and Sakai. The commercial environments are: Acrobat Connect, ANGEL Apex, Authorware, Blackboard, Brihaspati, Captivate, Content Point, D2L, Desire2Learn, eCollege, Elluminate, FirstClass, Knowledge Forum, Learn.com, Meridian, KSI, Saba, SAP Lectora, Tooling, TutorVista, WebCT, WebEx, Xmind and others. More information on each package may be found in internet.

The best known are **Blackboard** (commercial package, widely used in industry and in universities) and **Moodle** [12] (*Modular Object-Oriented Dynamic Learning Environment*), the largest open source competitor to Blackboard. Moodle and Blackboard were used by author to manage educational process. Both have all needed services, including: registration and creating groups of students, publishing documents in different formats, links to external resources, on-line and off-line assignments, statistics of the users activity and many others. Moodle and Blackboard are compared in [13]

After one semester of using Moodle and Blackboard for different groups of students, Moodle was chosen as more suitable and more effective environment. It is less complicated and more user friendly for the teacher. Moreover Moodle is a public domain software and this means that possible lack of founds will not stop using it.

III. UML language

UML (*unified modelling language*) is derived from three other products: **OMT** *(object modelling technique)* by James Rumbaugh, **Booch method** by Grady Booch and **OOSE** *(object oriented software engineering)* by Ivar Jacobson. From version 1.1, UML language is non-proprietary and open to all. It is maintained by the standards organisation: *Object Management Group* (*www.omg.org*)

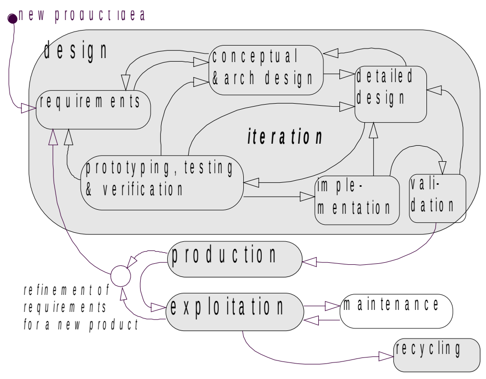


Figure 1: Design is a part of life cycle of a product

Bruegge and Dutoit in their excellent book [2] introduce UML as a base to object oriented software engineering. UML is independent of programming language chosen for coding the system and independent of physical nature of a hardware used. This approach (fig.1) is not limited to pure software engineering and may be successfully used in design of any system [3-8,10,11]. UML models may describe architecture and behaviour of the future system on high level of abstraction, including its software and hardware subsystems. Any complex system can be presented by a set of UML diagrams (models).

IV. Setting up Moodle for teaching UML

The course homepage is set up on local Moodle server. It contains standard blocks on the left and right side of the page and a central column containing several sections of the course content. Teacher is using simple menu (fig.2) to add resources and activities for each section of the course.

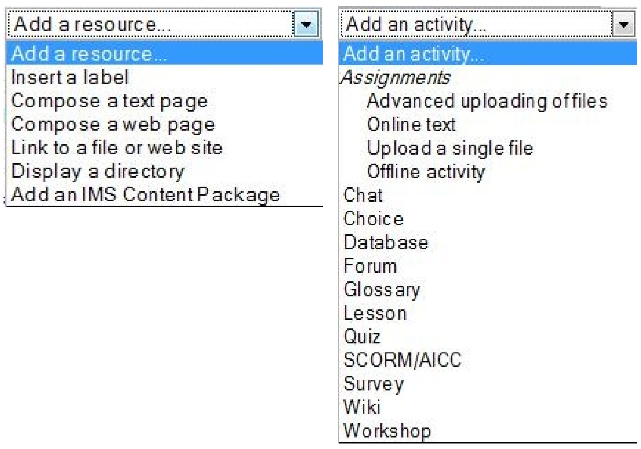


Figure 2: Adding resources (left) and activities (right) to the course section in Moodle. This menu is visible for teacher in edit mode only

Each section may be used during one or few weeks of teaching. Student will see only those sections, resources and activities, which are actually activated by a teacher as visible. Blocks and sections may be added, hidden, deleted, and moved. Teacher with editing rights will also have access to a course administration block.

UML course home page

The course home page has many sections of the course content. The actual sections for UML course are:

1. Introduction to UML language. Recommended books and articles. UML resources in internet
2. RUP (*rational unified* process) methodology and importance of early phase of design
3. Elicitation of requirements. Using RAD, the requirements analysis document. Preparing scenario and use case diagram
4. Building draft version of sequence diagram based on scenario.
5. Using Visual Paradigm or other CASE tool to build UML diagrams
6. Interactive verification and adoption of classes and sequence diagram to minimise risk of failure of the designed system
7. Final verification of the project with state space or activity diagrams
8. Uploading the final report

There are also several standard blocks as: "Latest News", "Blogs", "Upcoming Events", and "Recent Activity"

V. Teaching UML in the Moodle environment

There are different ways to enrol students to the course. I ask students to manually create their own accounts in Moodle. Then they receive an automatic email from Moodle administrator to confirm their account. Using different access key for each group, student will be assigned to proper group of the UML course.

During lab, student reads instruction from Moodle and works individually on computer. Results are uploaded to Moodle at the end of each lesson. Both student and teacher, they have access to uploaded documents and may see the grade assigned to results and comments from the teacher (fig.3, 4).

The due date and time may be set 5 minutes before end of the class, which guarantee that class will finish at time. More time should be given for homework.

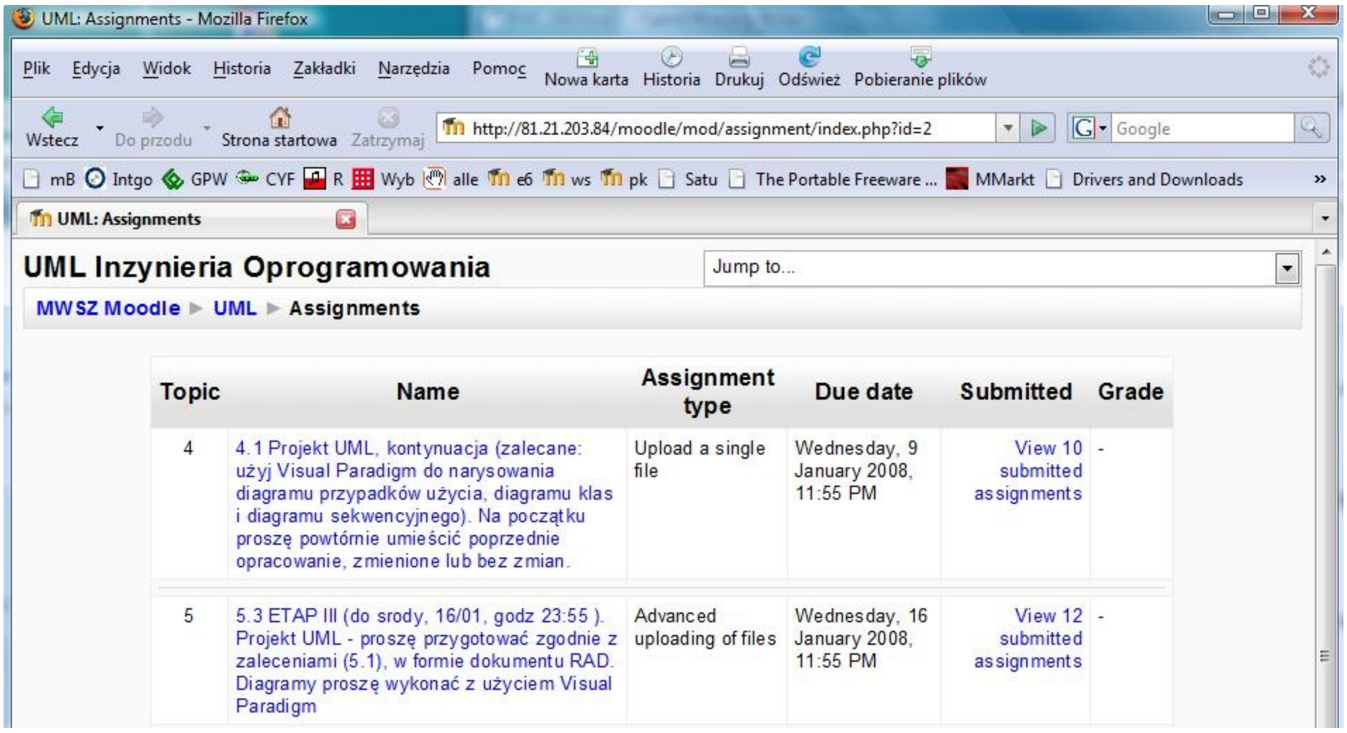
Objectives of the study program

The objective of the UML module is to provide the student with visualization and understanding of software development process (fig.1). Designing UML diagrams on computer screen is supported with **CASE** (*computer aided software engineering*) software tools. Best-known packages are Rhapsody [14] and Visual Paradigm for UML [16]. I do not use Rational Rose, as this software is not upgraded to UML 2.0. Some CASE tools offer simulation and animation of UML models. State diagrams are easily modelled, simulated and animated in MATLAB/ Stateflow package [9].

Before start of UML module, student should be granted free access to modern CASE software. He may install evaluation copies of different software and test them within limited period of time. Then he should use the software available permanently on university. Free and good quality software for educational purposes may be obtained by members of Visual Paradigm Academic Partner Program [16].

Bruegge and Dutoit (2004) point out, it is sufficient to have a deep knowledge of a small subset of UML to use it: "*you can model 80% of most problems by using about 20% of UML*". In author's opinion, *use case* diagrams, *scenario*, *class* diagram, *sequence* diagram and *state* diagram are the most important. There is no need to use all kind of diagrams during the design a real system.

Student should build and explore graphical UML models to verify his ideas, before he starts preparing code in chosen programming language. UML models form a good base for future detailed design and prototyping of the system. UML model is an important step to concurrent design and integration of subsystems of different nature, leading to solutions where control software, electronics and precision mechanics is integrated into one product or device.


UML: Assignments - Mozilla Firefox
Plik Edycja Widok Historia Zakładki Narzędzia Pomoc
Nowa karta Historia Drukuj Odśwież Pobieranie plików
Wstecz Do przodu Strona startowa Zatrzymaj
mB Intgo GPW CYF R Wyb alle e6 ws pk Satu The Portable Freeware ... MMarkt Drivers and Downloads
UML: Assignments
Google
Jump to...

http://81.21.203.84/moodle/mod/assignment/index.php?id=2

UML Inzynieria Oprogramowania

MWSZ Moodle ▶ UML ▶ Assignments

Topic	Name	Assignment type	Due date	Submitted	Grade
4	4.1 Projekt UML, kontynuacja (zalecane: użyj Visual Paradigm do narysowania diagramu przypadków użycia, diagramu klas i diagramu sekwencyjnego). Na początku proszę powtórnie umieścić poprzednie opracowanie, zmienione lub bez zmian.	Upload a single file	Wednesday, 9 January 2008, 11:55 PM	View 10 submitted assignments	-
5	5.3 ETAP III (do srody, 16/01, godz 23:55). Projekt UML - proszę przygotować zgodnie z zaleceniami (5.1), w formie dokumentu RAD. Diagramy proszę wykonać z użyciem Visual Paradigm	Advanced uploading of files	Wednesday, 16 January 2008, 11:55 PM	View 12 submitted assignments	-

Figure 3: Moodle: several useful information (e.g. name and description, the due date, link to submitted assignments, grade – if assignments is already scored) may be found in list of assignments

The design process

A typical design process may be presented as a sequence of:

- early design phase, which include requirements elicitation (inception) and conceptual design (elaboration). Off-line simulation may be used for testing and verification of the design,
- building (construction) phase, which includes detailed design, simulation and prototyping, as well as implementation and testing,
- deployment (transition) of the achieved solution.

Requirements elicitation (inception)

During elicitation of requirements, borders and external behaviour of the system are defined and criteria of consumer satisfaction are set. This may be influenced with development strategy of a company. Use cases are system boundaries identifying what the system should do. They capture subsystem functionality as seen from the point of view of end user or domain expert and help to understand how the system should work.

Conceptual design (elaboration)

During this design phase, feasibility study, estimation of needed resources and business plan for development (including implementation costs) are prepared. The objective are: to establish a sound architectural foundation, to develop the project plan and to eliminate highest risk elements of the project. Preliminary architecture is refined many times, as new UML diagrams are prepared and then iteratively analysed, modified and tested.

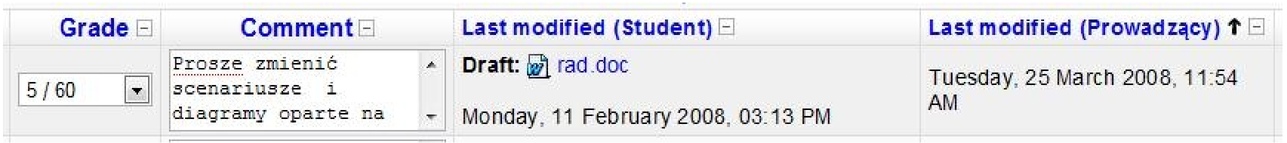
Student should decide on architecture of the system. At the beginning, student should analyse the *scenario* to identify candidates for *actors, objects* and *attributes* . Similar objects are generalised into class. This leads to preliminary version of the *class diagram*.

The most crucial part of design process is testing and verification of requirements and system architecture Student should understand that an important reason for building UML diagrams is to mitigate possible failure of the project. During this process, student verifies specification of requirements and *scenarios* against omissions and inconsistencies .

It is not a good idea to design complete diagrams sequentially. Instead, an iterative and concurrent approach is advised. Objects and classes are used to build other diagrams. Changes in object hierarchy, in naming, operations and attributes are inevitable, when other diagrams are under *design*. This is especially true when *sequence* or *state diagram* is prepared.

Verifying responsibility of subsystems

Statechart diagram is used to describe and verify behaviour of the system or its part (subsystem, use case, object). Comparing with sequence diagram or single scenario, the state diagram shows all states the system or its part (class, use case) may go through during its lifecycle.



Grade	Comment	Last modified (Student)	Last modified (Prowadzący) ↑
5 / 60	Prosze zmienić scenariusze i diagramy oparte na	Draft: rad.doc Monday, 11 February 2008, 03:13 PM	Tuesday, 25 March 2008, 11:54 AM

Figure 4: Moodle: grade obtained, teacher's comment, name and date of submitted assignment; scoring date and time

Each state represents a named condition during the life of an object. Object stays in actual state until it is fired by some event or when given condition (that must be fulfilled before the transition) will occur. Lines with arrow show possible transitions (change of state). A black ball shows a starting state. The end state (if exists) is denoted as black ball in a circle.

Detailed design, simulation, prototyping implementation and deployment is not included in this course.

VI. Conclusion

Students adopt very well to Moodle. They use internet access to see resources and to upload results of their work. On the other hand, teacher has a lot of extra work preparing resources and activities.

Preparing courses in Moodle environment is never ending job. Research in computer science area is so fast, that resources and proposed student activities need to be upgraded before start of each academic year. His work is finally rewarded at the end of semester, when final grade is easily assigned to each student, based on statistics from Moodle.

VII. References

[1] Booch, G. Rumbaugh, J. Jacobson I The Unified Modeling Language User Guide, Addison Wesley, 1999

[2] Bruegge B, Dutoit A, Object-Oriented Software Engineering, Prentice-Hall, 1999, 2004.

[3] Mrozek Z, UML as integration tool for design of the mechatronic system, in Second Workshop on Robot Motion and Control, pp 189-194, Bukowy Dworek, Poland, ed Kozlowski K, Galicki M, Tchoń K, Oct 18-20, 2001.

[4] Mrozek Z, Metodyka wykorzystania UML w projektowaniu mechatronicznym, pp.25-28, Pomiary Automatyka Kontrola 1, 2002.

[5] Mrozek Z., Mrozek B., Osei Adjei, Teaching object oriented software engineering with UML, *13th EAEEIE Conference "Innovations for Education in Electrical and Information Engineering",* April 8-10, York 2002

[6] Mrozek Z, Tao Wang, Minrui Fei. UML supported design of mechatronic system, *Proc of Asian Simulation Conference/5-th Int. Conference on System Simulation and Scientific Computing*, Shanghai, China, Nov. 3-6,. *2002*

[7] Mrozek Z. Computer aided design of mechatronic systems, *International Journal of Applied Mathematics and Computer Science*, vol **13** No 2, pp 255-267, 2003

[8] Mrozek Z, Importance of early design phase in mechatronic design. *In: Proceedings of 10th IEEE International Conference on Methods and Models* in *Automation and Robotics* (Domek S., Kaszynski R (Ed)), vol **1** pp 17-28, Miedzyzdroje, Poland, 2004

[9] Mrozek B, Mrozek Z. *MATLAB i Simulink, poradnik użytkownika* Helion, Gliwice, 2004

[10] Mrozek Z *An effective graphical approach to define objectives and structure of a control system* , 16-th Word Congress in Prague, 2005

[11] Mrozek Z, Bridging the gap between computer science and technology, Proc. of ACE´06, 7th IFAC Symposium on Advances in Control Education, 21– 23 June 2006

[12] Moodle, http://docs.moodle.org/en/Teacher_documentation, 2008

[13] Munoz K.D, Duzer J.V, Blackboard vs. Moodle http://www.humboldt.edu/~jdv1/moodle/all.htm (2005)

[14] Rhapsody, 2008, http://modeling.telelogic.com/products/, 2008

[15] UML Resource Page, (Object Management Group), 2008 http://www.uml.org/

[16] Visual Paradigm Academic Partner Program. http://www.visual-paradigm.com/partner/academic, 2008