"The Babel Experiment": An Advanced Pantomime-based Training in OOA&OOD with UML

Vladimir L Pavlov, Intel[1]
30 Turgeneva str., Nizhniy Novgorod,
603950, Russia
VLPavlov@ieee.org

Anton Yatsenko, UNN
IT Lab, 23 Gagarin avenue,
Nizhniy Novgorod, 603950, Russia
Yatsenko@wl.unn.ru

ABSTRACT

In this paper we present the original method of intensive hands-on training in Object-oriented Analysis and Design (OOA/OOD) with the Unified Modeling Language (UML). The method has been successfully used by the authors for three years.

During the training, the students:

- go through the communication problems that are typical for large software development projects;
- obtain the successful experience of applying UML to overcome these problems.

The essence of the method is that a team of students is supposed to design a software system. They have several hours to complete the task. During this timeframe verbal and written communication is forbidden, and the UML is the only allowed language. This training is a kind of experiment for students – they are to discover whether UML is "a real language" that is suitable and beneficial for a project team.

The training was successfully delivered more than ten times in both academic and corporate environments and generated positive feedback from students and customers.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques – *Object-oriented design methods;* K.3.2 [**Computers and Education**]: Computer and Information Science Education – *Computer science education.*

General Terms

Documentation, Design, Experimentation, Human Factors, Languages.

Keywords

Unified Modeling Language, UML, IT-education, Object-oriented Analysis, Object-oriented Design, OOA, OOD, OOP, Software Development , Hands-on Training, Pantomime

1. INTRODUCTION

One of the major achievements in the industry at the present time is unification of various notations used in different object-oriented modeling methods. Today all major modeling languages have merged into one – the Unified Modeling Language [1], [8]. UML is de-facto and de jure industry standard for object-oriented analysis and design [8]. This situation also resulted in incorporating UML into standard IEEE/ACM curricula on computer science [3] and software engineering [4].

Although there are a lot of UML courses, many of them have the same drawbacks. Quite often courses are criticized for giving just the syntax of UML with some simple isolated examples of possible UML applications [5]. The authors' experience has shown that the main problem faced by the students, is the gap between knowledge of concepts and ability to effectively apply that knowledge [10]. The authors believe that the best way to improve in any area is practice. Practice makes perfect. As it was shown in Dirk Frosch-Wilke's research [5], after participation in a practical project, students realize the expediency of using UML (see Figures 1 and 2).

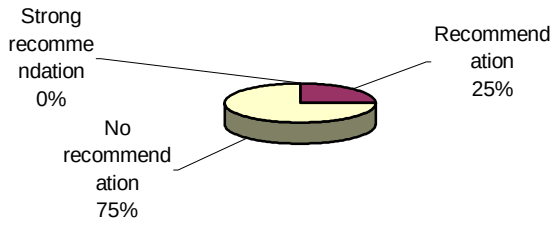

Figure 1 Recommendation of using the UML for software requirement analysis given by students before project work

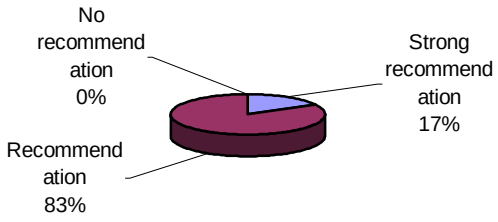

Figure 2 Recommendation of using the UML for software requirement analysis given by students after project work

This leads to another problem of teaching UML in universities: UML is used in large software development projects, and it's quite hard to incorporate such project into university curricula. On the other hand small projects (which could be

SIGCSE'05, February 23–27, 2005, St. Louis, Missouri, USA.

[1] This paper discusses the original teaching method that the author had developed prior to he joined Intel.

incorporated into educational process) do not allow students to taste the advantages of UML at full.

To address these challenges, in 2001 Vladimir L. Pavlov developed a training program called "The Babel Experiment". Since then it was successfully delivered a number of times in both software companies and universities. In June 2004 Vladimir L Pavlov and Anton Yatsenko joined their efforts to develop a TTT (Train The Trainer) course for the Babel Experiment.

2. THE TRAINING DESCRIPTION

2.1 Students' background

The method presented in this article was originally conceived as the final stage of traditional academic "OOD with UML" course. Later it was adapted for software development companies whose engineers needed the intensive advanced hands-on training in OOD/UML.

It is supposed that course attendees have some prior knowledge of OOD and UML. Obviously it is impossible to teach the practical aspects of UML if students are not familiar with the syntax of UML. Knowledge of the UML syntax does not guarantee the ability of using UML rationally but it is definitely necessary – if one wants to write meaningful sentences one has to know the letters.

From our point of view students' familiarity with some advanced areas such as design patterns, OCL and UML metamodel is an important landmark on their way to master object-oriented analysis and design [7]. So we usually asked participants of the training to learn these topics prior to attending the Babel Experiment.

Also we always wanted student to have some experience with the Charades game and acquaintance with the parable about "The Tower of Babel" – of course, most students meet these requirements without making any additional efforts.

2.2 The Pantomime Game

There is a well-known student's game called The Charades [2]. This game was one of the prototypes for author's method. This game is known under different names in different places: sometimes it is called "The Crocodile", sometimes – "The Charades" and some people simply name it "The Pantomime Game". In this article we will call it The Charades.

As well as its name, rules of this game vary from one place to another. Here are some exemplary rules.

There are two teams (each from 3 to 10 members). Members of the first team agree on a word and then tell this word to one person from the second team. The chosen person has to pantomime the proposed word to members of his team. At that he can not use speech or writing. Once the word is guessed correctly or the second team gives up, teams exchange their roles. Team, which has guessed the word, proposes new word and the other tries to guess, and so on. To make the game interesting it is preferable to choose really difficult words such as for example: melioration, placebo, expansion, encapsulation and the like.

Students who are going to take part in the Babel Experiment are supposed to practice this game.

2.3 The Tower of Babel

There is a well-known Bible's parable about The Tower of Babel:

"Now the whole earth had one language and few words. And as men migrated from the east, they found a plain in the land of Shinar and settled there. And they said to one another, "Come, let us make bricks, and burn them thoroughly." And they had brick for stone, and bitumen for mortar. Then they said, "Come, let us build ourselves a city, and a tower with its top in the heavens, and let us make a name for ourselves, lest we be scattered abroad upon the face of the whole Earth."

And the Lord came down to see the city and the tower, which the sons of men had built. And the Lord said, "Behold, they are one people, and they have all one language; and this is only the beginning of what they will do; and nothing that they propose to do will now be impossible for them. Come, let us go down, and there confuse their language, that they may not understand one another's speech."

So the Lord scattered them abroad from there over the face of the Earth, and they left off building the city. Therefore its name was called Babel, because there the Lord confused the language of all the Earth; and from there the Lord scattered them abroad over the face of the Earth." Genesis 11.

This legend shows that without having the common language human beings are incapable to achieve the common goal.

2.4 The Experiment description

A group of students who had prior learned OOD&UML is supposed to design some software system. At that verbal and written speech is not allowed and the only languages they may use are UML and pantomime.

Will they be able to design a system under such extreme conditions?

This experiment could be perceived as something quite uncommon for the regular teaching practice in universities. However, our experience shows that such approach is very effective and beneficial to achieve the following goals:

First, this experiment is a test, which can demonstrate to students whether UML is a "real" language or not.

Second, this experiment is aimed to raise students' interest in UML and to be a challenge for further improvement.

Third, experiment gives students an opportunity to obtain experience of team-working on a software design. It also gives them the opportunity to take part in collective research. Of course, the project environment is rather simulated. However, unusual conditions of this experiment allow students to get the teamwork experience similar to what they could learn from a "real" project that would last several weeks. While announcing the experiment to students we usually describe it as "joint eating of salt in the compressed time/space".

It is important to make sure that all experiment participants have committed to do their best during the experiment, that they have "if you pledge, don't hedge" mindset. One of the possible mechanisms to achieve this is to organize the presentation of the experiment results right after the experiment's completion and invite a significant number of spectators (guests). In fact, we did it for all of our Babel Experiments, and it has always been a highly motivating factor.

2.4.1 Approximate schedule of training preparation

The experiment was implemented according to the scheme shown in the Table 1.

Table 1

Days	Activity
1	The event is announced to students;
1-16	Students prepare for the contest (exam); Intensive Charades training
8-16	Spectators receive invitations
17	The contest (exam)
18	The Babel experiment

2.4.2 Participants selection

To simplify the communication within a team it is desirable that participants would be of close age – in this case they feel more uninhibited and behave more spontaneously. To ensure emotional balance between participants we've always tried to have both men and women in a team.

It is obvious that everything depends on students' understanding of UML, their imagination and the ability to apply their knowledge in extreme conditions. For the goals of experiment it is useful to organize a competitive selection of experiment participants from the student group. Usually a team of participants consists of 6-10 members who are selected from the group of 20-30 students. This competition between students in a group is made for the purpose of involving into the experiment the most prepared and interested students. It challenges students and encourages them to improve their skills.

Selection is organized as a verbal exam that includes questions both on the UML and on the modeling techniques. Instead of conventional multi-choice questions we prefer open questions. They give an opportunity to look upon the way of thinking of examinee. These questions usually lead to a deeper discussion; they are not supposed to have one simple answer. Most of them are exacted in further discussion by questions: "Why do you think so?", "Where, in which cases, it can be used?" and the similar. Here are some examples of such questions: "What is the difference between the *aggregation* and the *composition*?", "Draw the UML-diagrams, which will retell the fairytale about the Gingerbread Man", "Build the class diagram and statechart diagram for electric chandelier", "You are talking with alien, who understands only the UML language. Explain to alien what the traffic lights are, how it works and how it should be used".

As an alternative to viva voce examination we have always suggested students to earn industry recognized certifications on UML. For example, to be exempt from viva voce exam students can pass the IBM 486 Object-oriented Analysis and Design with UML exam [6] or one of the OMG Certified UML Professional exams [9]. However, students usually prefer not to utilize this option – probably, because of the additional cost associated with certification exams.

2.4.3 Roles

In the discussed training we usually had the following roles: participants, trainer, trainer assistant and spectators. In this section we will explain these roles in detail.

Participants are students who wanted to take part in the Babel Experiment and have successfully passed through the selection stage. Their functions in the experiment are:

- designing the system model without using speech;
- presenting the experiment results to spectators.

Trainer is the teacher and the experiment organizer. His role includes preparing the experiment task for participants and controlling conformation to the rules. He is not involved into designing a system, but looks after the process and later gives the feedback to participants.

Trainer assistant supports logistics, organizes lunch and connection with "the outside world". He meets and manages the spectators; he is also responsible for insuring that all Experiment participants strictly follow the rules.

And, as it was stated above, there are spectators (guests), who attend the presentation and whose feedback is important to make the final decision whether the experiment was successful or not (whether it "proves" UML to be a "real language").

2.4.4 Strike the four matches

The experiment consists of four phases, and the trainer strikes matches to mark transition from one phase to another. The sequence of the "Strike of four matches" milestones is shown in the Table 2.

Table 2

Milestone/strike	**Phase description**
The event starts	Instruction of the participants
Switch to speechless mode	Team opens an envelope with the task Team works on the task Light lunch Team moves to the presentation room
Presentation starts	Presentation and discussion of results
The event ends	The end

2.4.5 Approximate timing

Approximate schedule of the training is presented in the Table 3. We need a lot of guests on the presentation, so for the purpose of attracting the larger audience the Babel Experiment should be conducted in a non-working day. Usually authors organized it on Sunday because it is desirable to give the participants some time (Saturday) to rest after the working week as well as to get finally prepared for the Experiment.

Table 3

10:00-10:30 – the strike of the first match, the event starts
10:30-14:00 – the strike of the second match, switch to the speechless mode
14:00-15:00 – easy lunch, speech is not allowed
15:00-15:30 – the strike of the third match, switch to aloud mode, presentation starts
15:30-17:00 – presentation ends, the discussion with the spectators starts
17:00-18:00 – spectators leave the auditorium, postmortem with participants starts
18:00 – the strike of the forth match, the event ends

2.4.6 Experiment task

Each experiment task consists of two parts. One part is common for all Experiments. The other part is special for every Experiment.

The special Experiment task part is created in secret, so that students could learn it only when the experiment starts, right after the strike of the second match. Special task-part is composed in a

way that allows quite different solutions of the task. It should contain some ambiguity for the purpose to test students' ability to understand each other and to uncover and clearly specify all assumptions.

2.4.7 Special task part sample

Here is the example of real problem which was posed for the experiment participants in 2002.

"There is a company called BE (the Babel Electronics). The BE is going to launch home TV-systems of new generation. The feature of this product is that remote controls of new systems are something more than just traditional TV remotes.

The new remote control has no buttons; instead of them control has the sensitive screen on the whole control's surface. The control screen may display arbitrary images. The control screen can identify an exact location where a user pushes it by fingers or thick things.

Remote control is at the same time a mobile phone that complies with the GSM standard. This "smart" remote control is able to connect by Internet to the servers containing TV-schedules. Furthermore BE has made an arrangement with BT&T (Babylonian Telephone & Telegraph) company. According to it all WAP-connections of BT&T users with BE's server (where TV-schedules are allocated) will be free of charge during the 5 years from the moment of the new home TV-systems release.

The new TV-systems and their remote controls are supplied with BE-processors. They are capable to run complex software written in modern object-oriented programming languages, such as C++, Java, etc."

Note, that there are two possible usage scenarios for such systems: the user menu may be present either on the control screen or on the TV screen. None of these scenarios is explicitly mentioned to the training participants. This ambiguity was left intentionally in order to provoke students to make unspecified assumptions.

2.4.8 Common task part

Here is the common experiment task part which was included with some (not essential) variations in every Babel experiment since the year 2001.

"It's necessary to create and describe with UML the high-level design of the software. This software will be used in above-stated systems. At that using of English (or any other language) on the diagrams is allowed only for naming the diagram elements – use-cases, packages, stereotypes, classes, messages, attributes and so on. Notes, as the UML standard element, are banned.

In the process of designing only using of gestures or UML diagrams (according to above-stated constraints) is allowed.

Above-stated constraints are extended on the period from the moment of designing start (strike of the second match at 10:30) to the moment of results presentation (strike of the third match approximately at 15:00). So that person(s), who will present experiment results to guests, is (are) chosen by the pantomime.

If a participant breaks a rule he'll get a warning. If participant gets three warnings, he loses right of further participating in the experiment. Both the trainer and his assistant may issue warnings."

2.4.9 Presentation of the Experiment results

While the Experiment participants finish their assignment in the work-room, the spectators of the Experiment gather in the presentation-room. The spectators are met and hosted by the trainer assistant. When the presentation time comes, the participants move from the work-room to the presentation-room. The trainer strikes the third match and participants are allowed to speak. The presentation is made by person(s), who was (were) previously chosen by the pantomime.

2.4.10 Discussion

Discussion is one of the main experiment's stages. It follows the Experiment results presentation, which is made by the Experiment participants. The purpose of this stage is to make the decision about the Experiment results according to the spectators' feedback. During the discussion guests provide their opinions on the presented model (does it contain all important elements, is it produced in logical and consistent manner?). They debate whether it may serve as an input for further detailed modeling and consequent implementation. Discussion is concluded with the final consensual judgment on whether the team has managed to create a sound model.

2.4.11 Postmortem

After the spectators leave the auditorium participants and trainer may discuss the Experiment, talk over how they understood each other by the pantomime and UML. They debate the feedback from spectators, highs and lows of the model they have developed as well as the presentation they have delivered. It is a good time to thank each other; it is also an appropriate time to provide feedback to trainer about the Experiment organization – this helps improve future trainings of this kind. Finally the trainer strikes the last match and the event is over.

3. TRAINING RESULTS

The Babel Experiment was conducted more than ten times since it had been invented. It is important to notice that experiment had never failed. During every Babel Experiment students found the common language and generated the common ideas by UML communication, which led them to successful development of the proposed system model. The Experiment has always clearly showed to students that UML is a "complete" language.

It is interesting to mention that once we had two teams[2] working on the same task. One team was limited to using only the UML language and the pantomime in their communication. The other was allowed to use the speech in addition to the UML. The first team (which was not allowed to use speech) coped with a task more successfully than other team. Their diagrams were more detailed, more elaborated and elegant.

Discussing this fact we have come to the following explanation. When a task is discussed by team using the speech, everyone understands each other, so it seems to participants that there is no need to map "obvious" things in the UML diagrams. The problem is that in reality these "obvious" assumptions are quite different for different people. When the team uses only the UML and pantomime, in order to understand each other, participants have to map in the diagram everything very precisely

[2] They were participants of the educational project "Virtuoso" that was organized by University of Nizniy Novgorod (Russia) and supported by Intel, Microsoft, IBM, Borland and Kaspersky Lab.

and detailed, much more formal than in normal verbal discussion. This leads to making all assumptions explicitly captured in diagrams hence increases common understanding within a team and helps make final model more clear and coherent.

One of the great training results is stimulation of students' efforts for learning OOA/OOD and UML. In order to pass the UML exam and to win a contest students have to study a lot of materials on the subject. It is necessary to notice that prior to the Experiment we always organize not only the UML exam, but a number of exam rehearsals where everybody has a chance to practice in modeling and get detailed feedback from experts. So, Experiment should be apprehended not as thing-in-itself, but as the culmination of a long-period preparation. The process of intensive students' preparation for the Experiment is overwhelmingly important.

Students always provide very positive responses after they participate in this training. Here are some typical students' comments:

- *"this experience of practical UML-using forced me to change my attitude with regard to UML – now I do understand how it could help me in real-life projects";*
- *"participating in the Experiment helped me to realize that UML is not a kind of abstract "Glass Bead Game", but rather a very practical tool";*
- *"thanks to the Experiment I was able to understand that usage of different UML diagrams (class diagram, sequence diagram, use-case diagram, etc.) really makes sense when they all are used together, representing different points of view for the modeled system".*

Several times we have conducted The Babel Experiment as a kind of capstone project that culminated "traditional" university course in OOD with UML. Students who attended this modified course have shown higher motivation and they have finally received better grades on the final exam in comparison to other students.

So our experience of implementing The Babel Experiments shows that the speechless approach to practice UML is an effective and productive method that helps students fully experience the might of UML.

4. CONCLUSION

In this paper we have presented a pantomime-based method of hands-on training in OOA/OOD and UML. This method has proven itself to be an effective approach to simulate teamwork on complex software projects. It gives students an opportunity to feel the expressive power of UML that significantly simplifies communication within a team.

The training is delivered to students in the form of an experiment with the declared goal to check "Whether UML is a real language". The results of such experiments were always positive – in all cases training participants managed to develop quite elaborated design models without using speech. Once we had two teams working concurrently on the same design task – one team was allowed to use speech while the other was not; the result presented by the speechless team was more elegant and thoroughly developed.

The experimental nature of this method inspires students for deep studying OOA/OOD and UML; the practical nature of this method helps them to recognize OOA/OOD with UML as a practical tool rather then a theoretical method. As a result, after attending the experiment students report that they are eager to apply UML in their real-life projects.

The method was successfully implemented in both academic and corporate environments.

5. ACKNOWLEDGEMENTS

We would like to thank all people who reviewed different versions of this article (either in form of article or presentation) and whose comments helped us to polish our paper:

- Alex Zverintsev (Nokia, Poland);
- Andrey Terekhov (Microsoft, Russia);
- Anna Akousheva (Intel, Russia);
- Dmitry Malenko (Dnepropetrovsk National University, Ukraine);
- Michael Wrinn (Intel, US);
- Nikita Boyko (Dnepropetrovsk National University, Ukraine);
- Nikolay Mitushin (Intel, Russia);
- Stanislav Busygin (University of Florida, US);
- Yury Bushenko (University of Westminster, UK).

6. REFERENCES

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

[2] Charades: http://www.cs.umd.edu/~nau/misc/charades.html

[3] *Computer Curricula 2001: Computer Science* (December 15, 2001) http://www.computer.org/education/CC2001/cc2001.pdf

[4] *Computer Curriculum: Software Engineering* http://sites.computer.org/ccse/volume/FinalReport-5-21-04.pdf

[5] Dirk Frosch-Wilke. Using UML in software requirements analysis – Experiences from practical student project work. *Informing Science + IT Education Conference Proceedings,* Informing Science Institute, Santa Rosa, CA, June 2003, 175-183.

[6] IBM 486 Object-oriented analysis and design with UML. http://www.ibm.com/certify/tests/sam486.shtml

[7] Lew Della, David Clark. Teaching Object-oriented development with Emphasis on Pattern Application. *Proceedings of the Australasian conference on Computing education,* ACM Press, New York, NY, 2000, 56 – 63.

[8] OMG , Unified Modeling Language Specification http://www.uml.org

[9] OMG, UML Certification Program. http://www.omg.org/uml-certification/exam_info.htm

[10] Philip J. Burton, Russel E. Brunn. Using UML to Facilitate the Teaching of Object-oriented Systems Analysis and Design. *Journal of Computing Sciences in Colleges,* vol. 19, issue 3, January 2004, 278-290.