Using Pantomime in Teaching OOA&OOD with UML

Vladimir L Pavlov, Intel*  
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

This paper presents the original pantomime-based training on Object Oriented Analysis/Objected Oriented Design (OOA/OOD) with Unified Modeling Language (UML). The essence of this training involves giving an assignment to create a high-level design for a software system to a team of students. They are required to use UML and are not allowed to use any other language (English, German, Russian, Ukrainian, etc), either spoken or written. Students are given 4-6 hours to complete the task. This assignment is always presented as an experiment – the students are to discover whether or not UML is “a real language” that can successfully serve as a communication tool within a team. The authors call this training “The Babel Experiment”.

This training allows students to work through typical problems of software development projects and to see how UML helps solve these problems.

This training was developed by Vladimir L. Pavlov in 2001 and it has been successfully conducted at universities and software companies over ten times. In universities, this training has been used as a kind of capstone project for traditional OOA/OOD/UML courses; however, for software companies this training was mostly delivered as a stand-alone course. In all cases this training course has received very positive feedback from students and customers.

1. Introduction

Currently the common view in the educational community is that OOA/OOD and UML [1], [16] should be incorporated into university curricula on Computer Science and Software Engineering – this fact is reflected in the ACM/IEEE Computing Curricula ([7] and [8]). However, the question about the most effective approach to teach this subject is still open [2-5], [10], [13-15], [18], [20-21].

While teaching OOA/OOD in universities, the authors have noticed that students tend to perceive it as something very abstract and theoretical--something that generally meets academic requirements rather than the practical needs of software developers. However, students change their opinion about UML once they try to apply UML in a real project. The authors’ subjective observation matches various studies that demonstrate the importance of practical application in teaching OOA/OOD and UML. For example, in Dirk Frosch-Wilke’s research [9] it was shown that after participation in a practical project students realized the expediency of using UML (see Table 1).

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

<table>
<thead>
<tr>
<th></th>
<th>No recommendation</th>
<th>Recommendation</th>
<th>Strong recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before project work</td>
<td>75%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>After project work</td>
<td>0%</td>
<td>83%</td>
<td>17%</td>
</tr>
</tbody>
</table>

* This paper discusses the original teaching method that the author had developed prior to joining Intel.
It is difficult, however, to incorporate large or even mid-size projects into university curricula [11], while small projects are not enough to allow students to experience all the advantages of UML. The authors experimented with different non-traditional techniques to discover a unique method to provide students with an intensive, yet not time-consuming practical teamwork experience of applying OOA/OOD and UML. As a result of this experimentation, several years ago Vladimir L. Pavlov began to use a pantomime-based approach to enhance his students’ practical experience. During the past three years, this approach has proven itself to be successful and applicable not only in the university curricula, but also in the industrial environment.

2. The training description

The training described in this article consists of the experiment (that takes one day) and the preparation phase (that takes up to several months). Students prefer the experiment phase because of the perceived challenge, as well as the enjoyment of participating in the experiment. However, from an educational point of view, both the preparation phase and the experiment are equally important. These components are discussed below in detail.

2.1. The Experiment

A group of students that has previously studied OOA/OOD and UML must design a software system. They have 4-6 hours to complete the task. Students may only use UML and pantomime and are prohibited from using verbal and written speech. Can students design a system under such constraints?

The experiment helps to achieve the following goals:

- to demonstrate to students whether UML is a “real” language or not;
- to provide students with the experience of designing software as a team;
- to model the communication problems that are typical in software projects;
- to demonstrate how UML helps overcome communication problems;
- to raise students’ interest in OOA/OOD and UML and to challenge them for further improvement.

Extreme conditions are applied to the experiment to destroy the standard stereotypes of behavior; this forces participants to find new ways to collaborate. The experiment stimulates creativity and pushes participants toward discovering innovative approaches of applying UML in their communication.

<table>
<thead>
<tr>
<th>Milestone/strike</th>
<th>Phase description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The event starts</td>
<td>Instruction of the participants (Duration: approximately 30 minutes)</td>
</tr>
<tr>
<td>2. Switch to speechless mode</td>
<td>Team opens an envelope with the task Team works on the task Light lunch Team moves to the presentation room (Duration: approximately 4-6 hours)</td>
</tr>
<tr>
<td>3. Presentation starts (speechless mode ends)</td>
<td>Presentation and discussion of results Postmortem (Duration: approximately 1-3 hours)</td>
</tr>
<tr>
<td>4. The event ends</td>
<td>The end</td>
</tr>
</tbody>
</table>
To expedite this experiment, the trainer strikes matches to mark the transition from one phase to the next. The sequence of the “Strike of four matches” milestones is shown in Table 2.

The experiment consists of two components: one component is unique and is created for each Experiment; the other component is the same in all Experiments that are conducted.

The unique component of the experiment is created in secret so that students become aware of it only when the experiment starts, right after the strike of the second match. The unique component is composed in a way that allows for varying solutions of the task. The trainers always try to design this component in a manner that supports some ambiguity, in order to test the students’ ability to understand each other and to uncover and clearly specify all assumptions.

Below is the common component for all experiments used in every Babel Experiment since 2001, some of which have included some non-essential variations.

“It is necessary to create and describe the high-level design of the software with UML. This software will be used in the above-stated systems. The use 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 the use of gestures or UML diagrams (according to the above-stated constraints) is allowed.

The above-stated constraints are applied to the period from the moment designing starts (strike of the second match at 10:30) to the moment of the results presentation (strike of the third match approximately at 15:00). The person(s), who will present the experiment results to guests is (are) chosen by pantomime.

If a participant breaks a rule he/she receives a warning. If a participant receives three warnings, he/she loses the right to participate further in the experiment. Both the trainer and the assistant may issue warnings.”

Although the experiment rules appear severe, the authors have never had the need to expel a participant from the experiment. Sometimes warnings were given, but not more than once to a single participant.

In the beginning the experiment seems to be very attractive for its participants. However, one to two hours into the experiment, some of the participants can start to feel helpless without the use of convenient communication tools, which can cause them to wish to leave the experiment. If even one person leaves the experiment, it can demoralize the other participants and negatively affect their efforts. It is important, therefore, to take some preventive measures to ensure that participants understand their responsibility for the final result so they will not retreat when problems arise. For this purpose the presentation of the experiment results occurs immediately after its completion. The presentation is announced in advance, which usually attracts many students from different faculties, employees of local high-tech companies and sometimes even reporters. Experts in OOA/OOD from local software development companies are invited as special guests to the presentation.

A discussion follows the presentation which is a main component of the experiment. The purpose of the discussion is to analyze the results of the experiment. During the discussion, guests provide their feedback on the presented model, including: Does it contain all of the important elements? Is it produced in a logical and consistent manner? They debate whether the model may serve as an input for further detailed modeling and
The discussion is concluded with the final agreed upon judgment as to whether the team has created a sound model. After the guests leave, the participants and the trainer conduct a postmortem. During this time, they discuss how they understood each other through the pantomime and UML, the model they developed, as well as the presentation they delivered. They also debate the feedback from spectators. This is a good time to thank each other; it is also an appropriate time to provide feedback to the trainer about the organization of the Experiment to help improve future trainings sessions. Finally, the trainer strikes the last match and the event is over.

2.2. The Preparation Phase

Experiment participants are expected to have a strong knowledge of OOA/OOD and UML. This includes familiarity with the UML syntax and semantics (together with OCL), analysis and design techniques, design patterns, and UML metamodel. To ensure that the participants meet these criteria, a verbal exam that includes questions both on the UML and on the modeling techniques is administered.

Instead of conventional multiple choice questions, open ended questions are preferred. These questions provide an opportunity to examine the potential participant’s way of thinking. These questions usually lead to a deeper discussion, and are not supposed to have one simple answer. Most of the questions are extended into further discussion through leading questions: “Why do you think so?”, “Where and in which cases, can it be used?” and so on. Here are some samples of the exam questions:

- Create a class diagram which explains the relation between a Citizen and a Policeman;
- Create UML-diagrams which will describe your plans for the next 3-5 years;
- You are talking with an alien, who understands only the UML language. Explain to him what a TV (an automobile, an air conditioner, an electro-guitar, etc.) is;
- Create UML-diagrams that describe your relations with your family (your neighbors, your colleagues, etc.);
- Create UML-diagrams which retell the fairytale about Cinderella (the Gingerbread man, etc.);
- Using UML, describe the political system in the United States (Russia, Israel, United Kingdom, France, etc.);

The exam is usually organized 1-2 days prior to the Experiment.

The authors want their students to not only know the UML syntax, but, more importantly, to be able to apply UML to solve practical analysis and design tasks. So, on the verbal exam special attention is given to students’ ability to model. In order to give students time for preparation, the Babel Experiment is usually announced several weeks (or even months) before the event. A number of previous demo-exams are conducted before the real (final) exam, so that students can understand their level of competency and discover gaps in their knowledge. As an alternative to the viva voice examination, students are encouraged to earn industry recognized certifications in UML [12] and [17]. However, students usually do not prefer to utilize this option – probably, due to the additional cost associated with certification exams.

The method presented in this article was originally conceived as the final stage of a traditional academic “OOD with UML” course. It is not intended that all attendees of the “OOD with UML” course should participate in the experiment – only the best of them. However, most of the students usually want to take part in this interesting and challenging event. The desirability of the event/experiment becomes an additional
motivator for students to improve their knowledge in OOA and OOD, to spend time on self-study and to learn more than the minimal requirements of university curricula.

Additionally, students are required to have some experience with the Charades game [6] and to become acquainted with the parable about “The Tower of Babel” [19]. (This parable shows that without having a common language, human beings are incapable of achieving a common goal.) Students generally meet these requirements without any additional efforts.

3. The Results

Although the Babel Experiment was initially designed to be used as an educational tool, the results have provided interesting possibilities, not only for professors/instructors, but also for analysts and designers.

3.1. The Experiment Results

The Babel Experiment has been conducted more than ten times since it was created, and it has always been successful. During every Babel Experiment, students were able to work together and to communicate by UML. It led them to successful development of the proposed system model.

The developed models usually include 5-15 (sometimes more) diagrams that cover at average 5-20 use-cases with several actors, up to 30-40 classes, etc. Sometimes students decide to create a lot of sequence diagrams; sometimes they pay significant attention to collaboration diagrams; in any case, the whole spectrum of UML diagrams (from use-case diagrams to deployment diagrams) is always covered.

The presentation guests, who represented both academy and industry, almost always provided very positive comments regarding the quality, consistency and sufficiency of the presented models. The Experiment has always clearly shown students that UML is a “real” language.

The authors were surprised to discover that during the experiment students sometimes used UML not only to solve their task, but also as a tool for general-purpose communication. Figure 1-a shows the message that was received by a trainer when the lunch was delayed during one of the experiments.

In Figure 1-b, a sample UML diagram that was created by Experiment participants is presented. It was created when they organized their cooperation and planned their work on the task – a good example to show to students while telling them about SPEM.

The main goal of the first Babel experiments was to “test” whether UML is a “real” language or not. Because the experiment has always been successful, the intent of the experiment has changed over time. It does not make sense anymore to propose the event to students as an experiment that “tests” UML, rather it becomes an experiment that “tests” students: whether they know UML well enough to use it as an all-sufficient communication tool.

Of course, the experiment evolves and changes its form over time. For example, in August 2004, two teams were working on the same task†. One team was limited to using only the UML language and the pantomime in their communication. The other team was allowed to use speech in addition to the UML. The first team (which was not allowed to use speech) coped with the task more successfully than the other team. Their diagrams were more detailed, more elaborated and elegant.

† They were participants of the educational project “Virtuoso” that was organized by the University of Nizniy Novgorod (Russia) and supported by Intel, Microsoft, IBM, Borland and Kaspersky Lab.
Having discussed this fact, the following explanation was determined. When a task is discussed by a team using speech, everyone understands each other, so participants see 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 everything in the diagram very exactly and detailed, much more formal than in normal talkative discussion. This leads to making all assumptions explicitly captured in diagrams, hence common understanding within a team increases and it helps to make the final model more clear and coherent.

![Diagram](image1.png)  ![Diagram](image2.png)

Figure 1 Examples of the UML usage during the Experiment.

3.2. The Training Results

One of the great results of this training is the 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. The Experiment should be apprehended not as thing-in-itself, but as the culmination of a long-period of preparation. The process of students’ intensive preparation for the Experiment is overwhelmingly important.

Students always provide very positive and enthusiastic responses after attending this training. The main theme of the students’ comments is that participation in the Experiment allows them to understand the expediency and utility of using UML in real software development projects.

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

During numerous postmortem discussions the authors also recognized that this approach helps to model work in complex multicultural environments. In today’s global market it is typical for large international companies to have people from China, France, Israel, Russia, Ukraine, United States, etc. (for most of them English is a second language) working together on the same task. So this training also helps students prepare to work on international teams.

4. Conclusion

This paper presents the original, intensive hands-on training in OOA/OOD and UML called “The Babel Experiment”. This training is based on the usage of pantomime. This approach was designed by the first author in 2001 and has proved itself to be an effective way to:

• simulate the typical communication problems that engineers face while working on complex software development projects;
• provide students with an opportunity to utilize UML to successfully overcome these problems.

The training is promulgated to students as an experiment with the goal to verify: “Is UML a real language?” Since its invention, the training has been conducted over ten times and it has never failed – students have always developed sound and elaborate models. In one instance, the experiment was carried out simultaneously for two teams with the same design task; one team used only the UML and pantomime while the other was allowed to use speech in addition to UML. The first (speechless) team resulted with a more mature and detailed model.

Regardless of the results of the experiment, the pantomime-based approach in teaching OOA/OOD and UML is an effective and powerful tool that helps raise students’ interest to the subject and to provide them with a vivid experience of utilizing UML in practical teamwork.

The training, “The Babel Experiment,” has been successfully delivered in both universities and software development companies.

5. Acknowledgments

We would like to thank all people who contributed to the development of The Babel Experiment and/or reviewed different versions of this article (either in form of article or presentation):

• Alex Zverintsev (Nokia, Poland);
• Andrey Terekhov (Microsoft, Russia);
• Bekah Sondregger (SCC Soft Computer, US);
• Dmitriy Malenko (Dnepropetrovsk National University, Ukraine);
• Irina Mozgovaya (Dnepropetrovsk National University, Ukraine);
• Michael Wrinn (Intel, US);
• Nikita Boyko (Dnepropetrovsk National University, Ukraine);
• Pavel Kudryashov (Volgograd State University, Russia);
• Stanislav Busygin (University of Florida, US);
• Yury Bushenko (University of Westminster, UK).
6. References