# Tool Support for Collaborative Teaching and Learning of Object-Oriented Modeling

Klaus Marius Hansen
Department of Computer Science
University of Aarhus, Denmark
Aabogade 34, DK-8200 Aarhus N
marius@daimi.au.dk

Anne Vinter Ratzer
Ideogramic ApS
Mejlgade 45, DK-8000 Aarhus C
avratzer@ideogramic.com

## ABSTRACT
Modeling is central to doing and learning object-oriented development. We present a new tool, Ideogramic UML, for gesture-based collaborative modeling with the Unified Modeling Language (UML), which can be used to collaboratively teach and learn modeling. Furthermore, we discuss how we have effectively used Ideogramic UML to teach object-oriented modeling and the UML to groups of students using the UML for project assignments.

## Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computer and Information Science Education – *computer science education, information systems education.*

## General Terms
Documentation, Design, Experimentation, Human Factors.

## Keywords
Object-oriented modeling, collaborative learning, gestures, electronic whiteboards.

## 1. INTRODUCTION
Simplistically, object-oriented software development is a mapping of real-world phenomena and concepts of a problem domain to objects, classes, and their relationships in a solution domain [13]. The process of doing so is called *modeling*. Even though the reality of object-oriented development is much more complex, modeling enjoys a central position in object-orientation and is thus also important from a teaching perspective [13][14].

The Unified Modeling Language (UML; [16]) is the predominant industry standard for object-oriented modeling, and it is thus relevant to introduce it to students when teaching object-oriented software development. The UML has, however, a number of characteristics that inhibit learning:

- *Extent.* The UML has nine diagram types, more than 100 distinct meta-classes, and even more relationships between these. It is necessary to know many of these in order to use the UML fully.
- *Complexity.* The many elements of the UML have complex semantics and there is no concept of level of use related to the expertise of modelers. In this way, the UML has a steep learning curve and is primarily suitable for modeling experts.
- *Usability.* Even though the UML is large and complex, it has no support for many commonly occurring modeling practices such as freehand drawing and the inclusion of other diagramming styles [9]. The UML provides one formal view of what a model is and this view cannot be changed to accommodate a modeling situation.

In a use study designed to examine ways of dealing with these characteristics, we used a new tool, *Ideogramic UML,* to introduce the UML to a group of computer science undergraduates. Ideogramic UML is a gesture-based tool for collaborative modeling which is particularly effective on large-scale input and output devices such as electronic whiteboards.

The rest of this paper reports from the observations from this use study and discusses Ideogramic UML as a vehicle for teaching and learning object-oriented modeling.

## 2. BACKGROUND
Object-oriented techniques have been criticized as being too complex for teaching in certain situations [2][17]. Raner states that a notation such as the UML is excellent for experts, but bad for novices, since it contains a large number of new concepts, and novices need to learn both a new way of thinking, and a "highly non-intuitive graphic notation" [17]. Several possibilities exist for avoiding the complexity such as *hand waving about difficult details*, *providing a framework to cover difficult details*, or *avoid difficult details* [12]. With the "OVAL" notation, [17] provides an example of the second possibility in the context of graphical modeling notations.

In some cases, such as ours, object-orientation is introduced through object-oriented programming so that the basic concepts and principles of object-orientation are learned *before* graphical notations [14]. This enables teachers to introduce the UML directly in connection to a known object-oriented language by relating concepts of the language to concepts of the UML. As we investigate in this paper, a tool such as Ideogramic UML that does


*ITiCSE '02*, June 24-26, 2002, Aarhus, Denmark.

not in itself have a steep learning curve can further help to introduce students effectively to the UML.

Teaching situations provide a rich setting in which collaboration and communication co-exist in a number of forms. Different environments have tried to support these activities through "electronic classrooms" [1][2]. The "DEBBIE" groupware system [2], e.g., allows students to make personal annotations on a graphics tablet to an instructor's notes written on an electronic whiteboard. Our goal with using Ideogramic UML in a learning environment complements the goals of such systems: we envision the tool as a part of an *electronic learning environment* in which the tool can be used in many different situations. Particularly, Ideogramic UML may be an effective way to engage students in *active learning* [15].

# 3. IDEOGRAMIC UML: TOOL SUPPORT FOR COLLABORATIVE LEARNING

We have based Ideogramic UML on a series of user studies and subsequent evaluations of UML modeling situations. The tool is designed and implemented to support direct and fluid interaction, cooperative work, and an integration of formal and informal model elements [7][8].

## 3.1 User Interface

Ideogramic UML runs on a variety of input devices ranging from graphics tablets over ordinary desktop PCs to electronic whiteboards. The electronic whiteboard version provides a natural support for collaborative work (Figure 1).

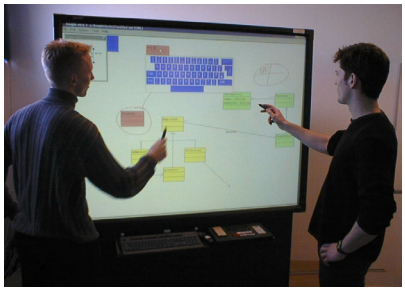

**Figure 1. Collaborative modeling with Ideogramic UML.**

The user interface (Figure 2) appears simple: it is a plain white surface, much like a traditional whiteboard, on which users draw gestures using dry pens.

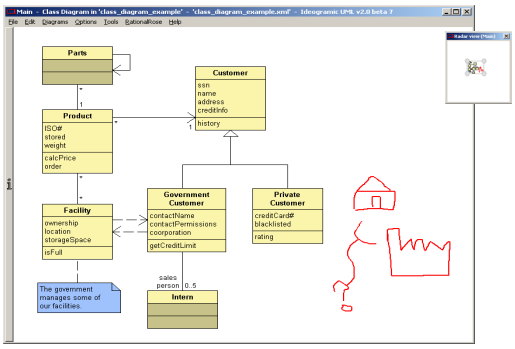


**Figure 2. The Ideogramic UML user interface.**

The use of traditional interface elements such as dialog boxes, buttons, and toolbars is kept to a minimum. When the user draws a gesture, Ideogramic UML recognizes the gesture incrementally and translates it into formal UML objects. Figure 3 shows how a box gesture (left) drawn in Ideogramic UML is transformed into a UML class (right).

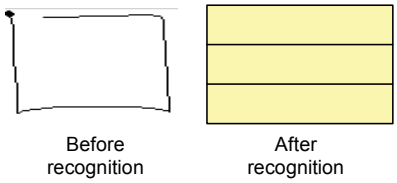


**Figure 3. Incremental recognition and transformation of gestures.**

The gestures largely resemble simple drawings on a whiteboard so that gestures are easier to learn and use. Apart from learnability, gestures allow input of a command and its parameters directly on the workspace, are fast to draw, and have a low cognitive overhead [4].

If the user presses and holds the pen, a context-sensitive, hierarchical pie menu (Figure 4; [10]) is opened. Pie menus provide a ready-at-hand way of invoking less frequently used commands.

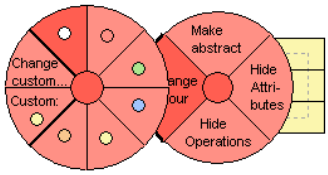


**Figure 4. Context-sensitive, hierarchical pie menus.**

Moreover, drawing a short gesture in the direction of a command on a pie menu invokes that command [11].

The upper right corner of Figure 2 show the *radar view*. The radar view shows a zoomed version of the workspace of Ideogramic UML, and it can be used for zooming and panning: dragging the view port in the radar results in a pan and dragging the handles of the view port in the radar results in a zoom.

## 3.2 Scenarios of Use

The following four general scenarios are examples of how Ideogramic UML can be used for educational purposes:

1. *Lecture use.* A lecturer can explain UML and object-oriented modeling techniques using Ideogramic UML in the classroom. As on an ordinary whiteboard there is a close connection between the lecturer's actions and the results of these actions.
2. *Peer instruction use.* Teaching assistants and students may use Ideogramic UML to present and develop solutions interactively. Teaching assistants may, e.g., present and review concepts from lectures and students may present and develop solutions to exercises.
3. *Peer group use.* Students may use Ideogramic UML when working in project groups, solving an exercise collaboratively. The use of Ideogramic UML can complement the use of ordinary whiteboards and blackboards and provide a more direct integration of modeling practices and computerized models.

4. *Individual use.* Ideogramic UML can be used on an ordinary PC as a modeling tool combined with all the scenarios above. Models may be transferred and used in the scenarios above.

In all scenarios the strength of Ideogramic UML is its directness of use: the ability to express UML models in a familiar and powerful interactive way. The next section explores these aspects in the context of concrete use in learning situations akin to scenario 2 and 3 through a use study.

# 4. STUDY OF USE

## 4.1 Study Background

Our use study had as objective to examine how Ideogramic UML could be used to deal with the extent, complexity, and usability problems of the UML. Its context was a project assignment in a course on object-oriented programming for second-year computer science students at the University of Aarhus. During this four-week project, students in groups of two or three created an object-oriented application for archiving, browsing and replaying music files and CDs. Part of the assignment was to construct object-oriented models of the problem domain.

Four groups of 12 people total participated in the use study. All participants had attended a one-hour lecture in connection with the course, in which we presented a subset of the UML notation, and showed a short demonstration video of Ideogramic UML. This lecture covered the basic UML concepts needed for the project assignment requirements.

The table below shows how often the participants had used the UML and other techniques relevant to the use study:

**Table 1. Background information for test subjects.**

| | Programming | OO | UML | Gestures |
|---|---|---|---|---|
| **Never** | 0% | 0% | 58% | 42% |
| **A few times** | 25% | 17% | 33% | 41% |
| **Some** | 33% | 58% | 9% | 0% |
| **Frequent** | 42% | 25% | 0% | 17% |

The most commonly known type of gesture input device was a PDA such as the Palm™ handheld computer.

## 4.2 Execution

The study started with a summary of the lecture given in the course. We then had a brief trial round, where the participants became familiar with the less common interaction techniques: gestures and pie menus.

The groups then used Ideogramic UML to construct and discuss the models they needed for the project assignment. We did not specify any explicit tasks, but left it to the students to decide what to model and how. In a normal study group meeting, the groups would solve the assignment by constructing and discussing models around a whiteboard in groups, followed by implementation of the suggestions either in groups or individually. The setup of our use study thus came close to a typical situation at the beginning of an assignment (Figure 5).

During the study, two experimenters were present, one responsible for the execution of the study and observation, and one responsible for providing help to the participants. Both experimenters took notes, and we videotaped one of the groups. In our observations, we focused primarily on aspects such as *breakdowns* and *focus shifts* [5], both in use of the tool and in use of the UML. Breakdowns are situations in which a tool ceases to function for the work it is used for. Focus shifts are situations in which a tool detracts the focus of its users from the work it is used for.

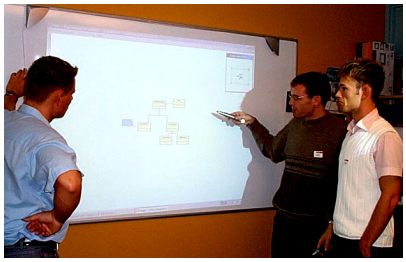

**Figure 5. Student use of Ideogramic UML.**

## 4.3 Help

During the test, the participants had access to two types of help.

- *Passive.* Two printouts showing the gestures of the tool and the basic elements of the UML syntax were available. The help menu in Ideogramic UML was also available.
- *Active.* The experimenter responsible for helping participants provided *learning scaffolding* [6][18] in the form of tool-specific help and modeling-specific help. The first category included, e.g., help for drawing a gesture or using the hardware. The second category included help, e.g., when the participants specifically asked for help on the UML, or when they used UML syntax inconsistently with their voiced intentions.

To get an impression of how novice users would react to the tool if they used it with no help, e.g. after downloading an evaluation copy from our website, one of the groups received very limited tool-specific help; we only explained to them how to use the pens on the whiteboard. They received the same amount of UML-specific help as the other groups.

The participants used the tool for approximately two hours. We then did an open-ended interview focusing on their qualitative impression of the tool.

The next section summarizes observations from the use study and comments from the interviews.

## 4.4 Results

Based in observations and analysis of our use study, we will now discuss how Ideogramic UML addresses some of the problems with teaching and learning object-oriented modeling using the UML.

### *4.4.1 Extent*

Using a gesture-based tool for teaching UML helps the teacher limit the amount of functionality presented to novice users. Through an instructor or a manual, the users can learn a small set of gestures necessary for their tasks and become familiar with these before moving on. Unlike a tool based on toolbars and pull-down menus, the users do not constantly have to face and ignore other, less relevant options for interaction. To ensure that the novice users can remember the gestures for their subset of commands, the gestures should resemble the objects they create

as closely as possible. This is generally the case in Ideogramic UML, as shown for the class gesture in Figure 3.

In the use study, we observed that the participants had no problems remembering the most basic gestures, and when one member of a group had problems with a more complex gesture, the others could in most cases remind him how it should be drawn. In the interviews, the groups supported this observation, saying that they found the tool easy to use and easy to start using.

### *4.4.2 Complexity*

Ideogramic UML reduces some of the complexity of UML, as described above, by presenting fewer possibilities to novice users at a given time. The context-sensitive pie menus, showing the commands possible for a specific UML object, helps novice users understand the structure of UML. The same is true for UML-specific feedback such as automatic layout of object hierarchies, which shows the user how UML objects are related and linked to each other.

Some tools similar to Ideogramic UML have additional software engineering capabilities such as the ability to keep models and code synchronized. This can help users transfer their models to code, but several of our test users commented on this, saying that they preferred Ideogramic UML because it did not have this functionality. The reason was that they felt safer using a tool that did not change their code automatically, and feeling safer, they also felt more encouraged to explore different variations of models.

### *4.4.3 Usability*

The freehand mode supports drawings that are not in the UML syntax, but that in some way supplements the UML model. In the use study we saw examples such as drawings of a user interface, sketches of a runtime object structure, and small code drafts.

This is a benefit for both expert and novice users. All users can at any point switch to freehand mode if the UML syntax is too restricted for a specific purpose. Novice users can switch to freehand mode if they are unsure of the exact UML syntax, and discuss their model just as they would on a standard white board. All groups in our study said that they found the freehand mode very important for UML modeling.

Another way Ideogramic UML increases usability of UML modeling, especially for novice users, is by being less restrictive than the formal UML specification. Ideogramic UML allows various types of incomplete elements such as classes without names, elements with the same name, and disconnected associations. This enables learning by allowing users to start modeling while they still have a very basic understanding of UML syntax.

Figure 6 shows the model made by one of the groups in our study. The group created it in roughly 1½ hour, having no prior experience with UML modeling, and only a very basic introduction to the Ideogramic UML tool.

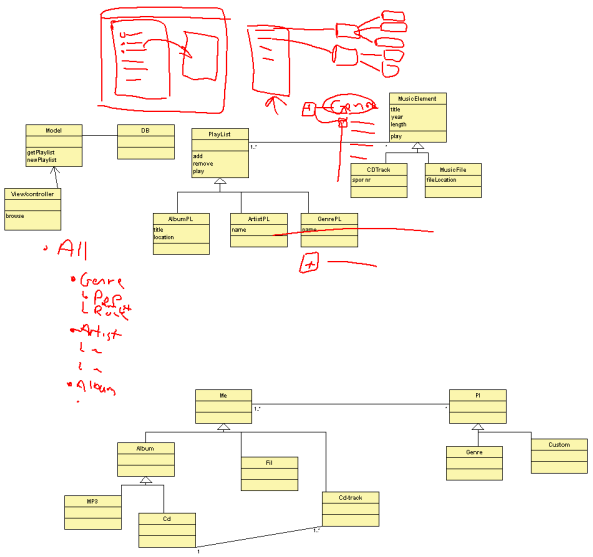


**Figure 6. Model made by students using Ideogramic UML**

During this time, the group sketched and discussed aspects of the user interface, examples of runtime object models, as well as several suggestions for code architecture.

## 5. SUMMARY AND FUTURE WORK

As the discussions above show, we have successfully applied Ideogramic UML to teaching and learning UML and object-oriented modeling

- based on an object-oriented language,
- through collaborative peer work, and
- through active learning scaffolding.

Ideogramic UML helps in this through a direct and fluid interaction that supports initial use and a gradual revelation of more advanced UML use through gestures. The user interface supports a collaborative style of use effective for teaching and learning.

Some of the major future directions for Ideogramic UML that our use study pointed to with respect to teaching and learning are

- *Support for understanding object-oriented design processes.* Active, adaptive templates of, e.g., design patterns will provide a first means of doing this.
- *Support for iterative learning of the UML.* The initial means for this will be various form of context-sensitive help on UML design preferably coupled to the support suggested above.
- *Support for distance learning.* We are currently extending Ideogramic UML to support distributed collaboration. One of the goals of this extension will be to support learning situations involving geographically distributed sites.

Obviously, further studies of the practical, situated use of Ideogramic UML is also on the top of the agenda. Hopefully, a new flexible working environment that is being created for the students at our department will provide means for flexible use of Ideogramic UML personally or in group settings as discussed above. These further studies should also include the study of use

of Ideogramic UML for lecture use and for personal use with a particular focus on learning.

Finally, an evaluation version of Ideogramic UML can be downloaded from `http://www.ideogramic.com` where a video demonstrating it in action is also available.

## 6. ACKNOWLEDGEMENTS

The Centre for Object Technology, partially funded by the Danish National Centre for IT Research, has funded the work reported in this paper. We thank the students for participating in the use study and Michael E. Caspersen for comments on a draft of this paper.

## 7. REFERENCES

[1] Abowd, G.D., Atkeson, C., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N., and Tani, M. (1996). Teaching and learning as multimedia authoring: the Classroom 2000 project. In the *Proceedings of the ACM Multimedia'96 Conference*.

[2] Astrachan, O. (2001). OO overkill: when simple is better than not. In *Proceedings of the SIGCSE Symposium 2001*.

[3] Berque, D., Johnson, D.K., Jovanovic, L. (2001) Teaching Theory of Computation Using Pen-Based Computers and an Electronic Whiteboard. In *Proceedings of ITiCSE'2001*.

[4] Buxton, W. (1986). There's more to interaction than meets the eye: Some issues in manual input. In Norman, D.A. and Draper, S.W. (eds.): *User Centered Systems Design: New Perspectives on Human-Computer Interaction,* Lawrence Erlbaum Associates.

[5] Bødker, S. (1996). Applying activity theory to video analysis: How to make sense of video data in HCI. In Nardi (ed.): *Context and Consciousness*, MIT press, Cambridge 1996.

[6] Chalk, P. (2001). Scaffolding learning in virtual environments. In *Proceedings of ITiCSE'2001.*

[7] Damm, C.H., Hansen, K.M., and Thomsen, M. (2000). Tool support for cooperative object-oriented design: gesture based modeling on an electronic whiteboard. In *Proceedings of Computer Human Interaction* (CHI'2000).

[8] Damm, C.H., Hansen, K.M., Thomsen, M., and Tyrsted, M. (2000). Creative object-oriented modeling: support for intuition, flexibility, and collaboration in CASE tools. In *Proceedings of ECOOP'2000*.

[9] Damm, C.H., Hansen, K.M., Thomsen, M., and Tyrsted, M. (2000). Supporting several levels of restriction in the UML. In *Proceedings of UML'2000*.

[10] Hopkins, I., Callahan, J., and Weiser, M. (1988). *Pies: implementation, evaluation and application of circular menus.* University of Maryland Computer Science Department Technical Report.

[11] Kurtenbach, G. (1993). *The Design and Evaluation of Marking Menus.* Unpublished Ph.D. Thesis, University of Toronto.

[12] Lewis, J. (2000) Myths about object-orientation and its pedagogy. In *Proceedings of the SIGCSE Symposium 2000.*

[13] Madsen, O.L., Møller-Pedersen, B., and Nygaard, K. (1993). *Object-Oriented Programming in the BETA Programming Language.* Addison Wesley.

[14] Madsen, O.L., Røn, H., Thorup, K.K., and Torgersen, M. (1998). A conceptual approach to teaching object-oriented programming to C programmers. In *Proceedings of the Educators' Symposium at OOPSLA'98.*

[15] McConnell, J.J. (1996). Active learning and its use in Computer Science. In *Proceedings of ITiCSE'1996.*

[16] Object Management Group (2000). *OMG Unified Modeling Language Specification. Version 1.3.* Document formal/00-03-01.

[17] Raner, M. (2001). Teaching object-orientation with the Object Visualization and Annotation Language (OVAL). In *Proceedings of ITiCSE'2001.*

[18] Wood, D, Bruner, J. S. and Ross, G. (1976). The role of tutoring in problem solving. In *Journal of Psychology and Psychiatry,* Vol. 17(2).