

miniGRADE – a tool for conceptual modeling by class diagrams

Audris Kalnins, Janis Barzdins, Karlis Podnieks

Institute of Mathematics and Computer Science (IMCS), Raina bulv. 29, LV-1459, Riga, Latvia
e-mail: {audris, jbarzdin, podnieks}@cclu.lv

A demonstration of the tool **miniGRADE** developed by Infologistik GmbH and IMCS is proposed. miniGRADE is a tool specially oriented towards **conceptual modeling** by UML class diagrams.

There may be two goals of applying UML class diagrams:

- UML based software system design. Currently most UML tools on the market support this goal, with the main emphasis on code generation, reverse engineering etc.
- conceptual modeling of complicated systems as defined by J.Rumbaugh in his OMT approach [1] or by J.Bubenko et al [2].

Though conceptual modeling of a system may be considered as the first phase of the system design, it has some very specific requirements to the tool due to the fact that the results of conceptual modeling must be evaluated by a group of humans:

- easy building and modification of highly readable class diagrams
- support of extra large class diagrams
- easy-to-use support of graphical stereotypes in diagrams.

miniGRADE has solutions for all these requirements, in addition, it supports class diagram based presentations – “talking class diagrams”.

The specific features of miniGRADE are now described in more detail.

1. Maintaining the readability of class diagrams

miniGRADE supports building of highly readable diagrams by its powerful controlled **automatic diagram layout** mechanism. This autolayout mechanism permits the user to insert a new class symbol where it is most desired. The existing symbols and lines are automatically moved, to give the necessary space and avoid any overlapping of the new symbol by existing symbols or lines. The movement is “delicate”, it doesn’t destroy the existing relative placement of symbols. Thus the main graphical aspect of readability – appropriate positioning and clustering of class symbols is supported. Association lines are automatically positioned so that unnecessary line crossings are avoided, thus good traceability of lines is obtained. Any diagram element – a class or line

segment (rectangular or oblique) can be freely moved, to improve the layout manually.

The automatic diagram layout in miniGRADE is based on smart algorithms which have won the first award in the recent Graph Drawing contest '99 [3].

2. Supporting large class diagrams

Conceptual models of complicated systems tend to be large because human understanding frequently requires to see the whole “big picture”. miniGRADE supports maintenance of extra large class diagrams, firstly, by its autolayout mechanism, which works efficiently also for diagrams with hundreds of classes. In addition, a special diagram zooming feature is provided, similar to that typically available in camcorders.

But for extra large diagrams even this may be insufficient. Therefore miniGRADE supports **views** for class diagrams. The user can maintain a large diagram via views corresponding to subsystems, then any updates will be automatically transferred to other relevant views and to the main diagram. The automatic updates retain the specific layouts for each of the views. It should be noted that the standard UML package mechanism, which is also completely supported by miniGRADE, does not enable a view-like principle – it is impossible to see an integral diagram of a package.

3. Supporting graphical stereotypes

Introduction of stereotypes in UML has really been an ingenious idea, and stereotypes have two very special applications in conceptual modeling:

- to give a class symbol a “look and feel” typical in a problem area
- to enable a class diagram to substitute for any other traditional modeling type because at a “first glance” to a system it is extremely difficult to separate the structural modeling from a simple dynamics modeling via activities, states etc.

These stereotypes should be completely graphical, i.e. include icons, symbol shapes, background colours and other elements of the symbol style. In miniGRADE a stereotype definition for a class or association can determine any element of its style. Introduction of new stereotypes is extremely easy, because they are kept in

a special stereotype table global for a model. Therefore a good practice in conceptual modeling can be applied - first the stereotypes for a problem area are introduced in the stereotype table and then applied to model elements.

4. Support of class diagram based presentations (“talking class diagrams”)

Modeling experience shows that class diagrams are the best means for comprehending a complicated system and therefore should be used as far as possible to cover all facets of a system. However, even they in essence are limited to a set of simple assertions over the model elements. In order to extend these limits, miniGRADE proposes the use of **class diagram based presentations**, which add two principally new possibilities:

- the order in which a class diagram should be viewed best (a "guided tour" through an object model)
- spoken comments which can involve any number of class diagram elements in any context.

Thus a new dimension – **sound** is added to the two-dimensional presentation of a system by means of class diagrams.

miniGRADE contains both authoring tools for creating such presentations and a very simple presentation playback mechanism. During the playback a special cursor dynamically points to class diagram elements mentioned in a sentence of the spoken comment. Thus the presentation can “imitate” the author of the class diagram giving a lecture on it and using a pointer during his narration. The presentation in miniGRADE is a regular part of a model documentation.

5. Other features

miniGRADE, certainly, supports all these thousand-and-one small features so essential for modeling (navigation to any other occurrence of a concept in the model, glossary service, export to HTML, export of diagrams to Word documents etc), and the main emphasis here is on the usability of these features. A few more to be mentioned are the possibilities for including informal pictures as parts of class diagrams and a support for team development of large models. There is also a bridge to the world of design – class diagrams and packages can be exported from miniGRADE to Rational Rose.

Technical requirements for miniGRADE: Windows 95, 98 or NT 4.0, Pentium class processor, 32 MB of RAM, 20 MB on hard disk, sound card (for presentation feature).

miniGRADE is available for commercial use from October 8, 1999 (see the URL <http://www.gradetools.com/>).

References:

- [1] J.Rumbaugh, M.Blaho, W.Premarlani, F.Eddy, W.Lorensen, Object-oriented Modeling and Design, Prentice-Hall, 1991
- [2] J.Bubenko, P.Johannesson, M.Boman, Conceptual Modelling, Prentice-Hall, 1997
- [3] F.Brandenburg et al, Graph-Drawing Contest Report. Symposium on Graph Drawing, GD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings. Lecture Notes in Computer Science, Springer, 1999