Oberon News

January 1994
Institute for Computersystems
Swiss Federal Institute of Technology
Zürich, Switzerland

Oberon Day '93
Oberon for End–Users

A brand new overlapping windowing model and document architecture for Oberon were some of the highlights presented at the first Oberon Day held in Zürich, Switzerland. The Oberon Day, organized by the Oberon Users Group of Switzerland, was attended by approximately 250 Oberon supporters.

After an introduction by Prof. Niklaus Wirth, one of the fathers of the original Oberon, the day proceeded with an introduction to Oberon System 3 by Prof. Gutknecht, followed by a sequence of impressive demonstrations in the afternoon. The main thrust of the demonstrations was directed towards Oberon for end–users.

Prof. Gutknecht presented the extensions his research group had made to the Oberon system for users to build graphical user interfaces from graphical elements called *gadgets*. In the demonstrations, the audience was presented many different gadgets, ranging from simple elements like buttons, checkboxes and list boxes, to document editors and gadgets that that were specially designed for research purposes. Gadgets may be combined with each other, or may have actions associated with them, to form graphical user interfaces.

The use and composing of gadgets happen at the same time, giving the system a highly dynamic nature. It is possible to borrow prefabricated gadget constellations from existing applications and embed them in new applications. Persistent collections of gadgets are called *documents*. Documents can be organized in an overlapping way on persistent work surfaces called *desktops*.

The first demonstration gave examples of how user interface composition is done. The other demonstrations introduced some of the applications built using the gadgets system. The first demonstration presented the *Leda* page layout system. Leda is an "intelligent" document editor that knows about good layout and typography. The system provides defaults for different document types, and can change these defaults to adjust to a new document style. For example, a single column document can be converted painlessly into a three column text, with Leda automatically adjusting the text styles with smaller fonts, resulting in a document with a typographical optimal appearance. Gadgets can be incorporated and edited in place in a Leda document.

In a demonstration entitled *The Electronic Book*, the audience was shown how the Oberon system can manage diverse sets of data. Formula gadgets in the *Script* text editor can be edited and evaluated with the symbolic mathematics package *Maple*. The formulas can be plotted and the results inserted into other documents. A *Teletext gadget* shows live teletext pages in the way as we know it from television, but allows the user to browse without waiting through an online and continually updated page database.

The stock exchange prices can be extracted and visualized using a up–to–date stock database. A further impressive feat is the Geographical Information System of Switzerland. Maps are generated in real time from a topographical grid of $250m^2$ resolution. A map zooming facility and a database of cities allows the user to explore Switzerland on a 80486 PC. A panorama tool allows the user to generate a panorama viewpoint anywhere in Switzerland, all in shades of green and simulated snow. To complete the picture, the actual weather conditions can be shown on an image of Switzerland.

After the demonstrations industry and other non–ETH institutions had their chance to present their work on Oberon. An Oberon compiler for Windows applications was presented by the Johannes Kepler University in Linz, Austria, and the company *Oberon Micro Systems* from Basel, Switzerland, described their efforts to turn Oberon into a commercial product. At the end of the day each attendee could take home the newest Oberon System 3 version for DOS– PCs, which by the way, still fits on a single high–density diskette!

What is Oberon System 3?

Oberon System 3 is the latest evolution of the Oberon System. Designed and implemented by Prof. Gutknecht and his group, it makes Oberon a true object–oriented operating system. The *Gadgets user interface* system, based on the new system features, shows the new possibilities of Oberon and provides the system with an up–to–date graphical user interface.

Graphical user interfaces can be interactively composed and utilised through visual objects called *gadgets*. Gadgets cover the whole spectrum of elements from buttons and checkboxes to more complicated ones such as illustration and text editors. They are usable and editable in place wherever they are located. For example, gadgets can flow inside text streams or can be embedded in a page layout system. Gadgets can be placed inside each other, allowing more complicated hierarchical structures to be built. Such structures are called functional units and can be reused.

Composing a user interface consists of building own functional units or by borrowing prebuilt functional units from other applications. Using a tool called the *Inspector*, the behaviour of gadgets can be modified, for example, by specifying the Oberon commands to be executed when they are manipulated. In many cases this is enough to add a graphical user interface to an existing non–graphical application. Should this not suffice, the system can be extended by making sub–classes of existing gadgets or by building new gadgets.

Oberon System 3 Version 1.4

The latest version of Oberon System 3 incorporates several new features and corrections. In addition to the many corrections, several cleanups in the module structure have been made. A major change is the introduction of documents and the desktop.

Documents are collections of gadgets that can be stored as self–contained units on disk. Being self–contained, they know how to open, store, and print themselves. A document gadget acts as the agent for the objects it contains, and in addition to the features listed above, can provide information about the preferred document size on the display, the menu–bar contents and how it should look in iconized form. This change has drastic consequenses for the user of the Gadget system. The opening, storing and printing of documents have been centralized in a few commands. No applications programs are explicitly started by the user. Opening a document is as simple as clicking onto its icon. In addition, the creation of new document types has been centralized so that you can simply indicate on a list of document types what is needed.

The Desktop organizes documents in an overlapping fashion on a large storable work surface. Althought partial overlappings are now possible, the user still has the possibility of working with documents in the well–known Oberon viewer system. In addition to documents, the user can lay any gadget on the desktop to create his own personal working environment. The desktop can be saved allowing the creation of different environments for different tasks.

Name scoping of gadgets is a new feature that simplifies the construction of functional units. Like in a programming language, the scope for object names is determined by where the objects are located. The structural hierarchy of gadgets is used to allow names to be visible in a restricted scope. For example, panel gadgets form a scope for the objects they contain. Previously, object names had to be unique in each user interface, often leading to conceptual problems. Now panels and other objects build scopes and can be freely copied without influencing each other (i.e. they are truly independent from each other).

The abstract gadgets have also been extended. It is now possible to program your own abstract gadgets which act as models for existing gadgets. For example, you can add new abstract record types. These gadgets can act as models for existing gadgets; buttons, checkboxes and text-fields etc. This is made possible by a message passing mechanism between model and view, that indicate in which field of the composed abstract gadget the view is currently interested.

The new **TextGadgets** provide the system with text editor objects that can be used for either full text editing or the creation of small text "post–it" notes. The *List* gadgets have been improved with user–selectable scroll-bars and new document icons have been introduced.

DOS–Oberon Device Driver Kit

Oberon System 3 for DOS runs as a native 32–bit DOS application using our own extender. Display drivers for VGA, ET4000 and S3 display cards are included. Printer drivers include Postscript, HP500, HP500C, HP550 and HP Laserjet. Due to the limited selection of drivers we provide a special drivers kit diskette containing the source code for the above drivers. This allows developers to add support for their own display cards and printers. We expect that device driver developers will return the drivers they create to ETH so that we can distribute them further.

This is a service ETH provides but cannot fully support (regarding device driver programming techniques) due to the lack of man–power and higher–priority research tasks. If you are a programmer and are interested in developing and testing an own device driver, please let us know and we can provide you with the needed material.

Note that Oberon for Windows works together with the installed drivers of Windows and does not require our device drivers.

The Leda Document Editor

Leda is an advanced, fully integrated and easy–to–use document editor for Oberon System 3. Leda allows the simple and efficient creation of documents of different types and complexity. No distinction is made between short letters, a long report or complicated page–layouts. Leda provides enough functionality to solve sophisticated document layouts. This document has been created using Leda. It distinguishes itself from other modern document editors in two ways.

The Leda–System respects basic typographical rules of documents. The typography influences sizes and attributes that can be inserted into a document. Not only are fonts, but also the formatting of the text and the layout of the pages influenced by typographical rules. The consequent use of the typographical rules simplifies document editing. A good–looking document can be created in a short time with little effort by using defaults provided by the system. Additional functionality allows the expert further control over the attributes of a document.

Additionally Leda provides a seamless transition between simple text formatting and page–layout. In the base functionality Leda supports block adjusted formatting of text. Multiple columns and a marginal form is also available. A more flexible layout can be obtained by placing diagrams and pictures in a layout raster on the page.

Gadgets can be integrated into Leda documents, either flowing inside the text stream or as freely place–able boxes. In addition, Leda supports the direct in–place editing of mathematical formulas.

Leda is the result of long history of document editors at the Institute for Computersystems. The system will be sold by the ETH and commercial companies. The Leda–System will become available in the first quarter of 1994 and is expected to cost about Sfr 250. We will keep you informed of the availability of the system.

Literature & Other Information Sources

All literature about System 3 is available electronically with each release. The books about Oberon can be regarded as an important information source for the concepts of Oberon System 3:

N. Wirth and M. Reiser: Programming in Oberon. Steps beyond Pascal and Modula. Addison Wesley, 1992, ISBN 0–201–56543–9.
Tutorial for the Oberon programming language and concise language reference.

M. Reiser: The Oberon System. User Guide and Programmer's Manual. Addison Wesley, 1991, ISBN 0–201–54422–9.
User manual for the programming environment and reference for the standard module library.

N. Wirth and J. Gutknecht: Project Oberon. The Design of an Operating System and Compiler. Addison Wesley, 1992, ISBN 0–201–54428–8.
Program listings with explanation for the whole system, including the compiler for NS32000.

H. Mössenböck: Object–Oriented Programming in Oberon–2. Springer, 1993, ISBN 3–540–56411–X.
Principles and applications of object–oriented programming with examples in the language Oberon–2.

You may participate in discussions about Oberon on the Internet news group *comp.lang.oberon*. The developers of the Oberon system monitor the discussions and sometimes give advice and make version announcements. The developers can be contacted directly by internet e–mail address *oberon@inf.ethz.ch*. If you do not have e–mail access you can send comments and questions to the address at the end of this news letter. We appreciate constructive feedback of all kind and would like to know what you think about Oberon.

To further the Oberon language and System in education and industry, the Swiss Oberon Users Group has recently been founded as sub–group of the Schweizer Informatiker Gesellschaft. They can be reached by e–mail at oberon–user@inf.ethz.ch or the address *Fachgruppe Oberon, Bergstr. 5, CH–8044 Zürich, Switzerland*

How to get Oberon

All the different Oberon versions are available free of change from our Internet FTP server *neptune.inf.ethz.ch* in the Oberon directory. Classic Oberon is available for Amiga, DEC-Station, MS–DOS, Microsoft Windows and Windows NT, HP 700, Mac II, IBM RS6000, SPARC and Silicon Graphics machines. Oberon System 3 Version 1.4 is currently only available for MS–DOS computers.

If you do not have access to Internet, you can order on diskette from the address below. We charge a fee of Sfr 50.00 to cover our costs. We accept payment via *Eurocard/Mastercard* or *VISA*. To order by credit card, specify your credit card number, expiration date, and your name exactly as it appears on the card.

When you order the *MS–DOS Oberon System 3 version*, you will receive on a high–density diskette the Oberon base System with compiler, the Gadgets User Interface system, an Intel assembler, a discrete event simulation toolkit, and the complete online documentation. Several example programs illustrate programming your own gadgets.

Our Address

Institut für Computersysteme
ETH Zentrum
CH–8092 Zürich
Switzerland

Telephone +41 (1) 632 73 11
Facsimile +41 (1) 261 53 89