



# Effective Technology Transfer

[The Oberon Tribune, No 1/1, July 1995]

Prof. Niklaus Wirth, Institute of Computer Systems, ETH Zürich, Switzerland

The golden era of research for research's sake is over. The times when philosophers, inventors, and tinkerers could freely pursue their favourite ideas, problems, and gadgets while being supported by public funding belong to the past. *Relevance* is the catch word, relevance for the public who pays, relevance for the economy. No longer is it taken for granted that research is inherently a benefit; practical proofs must be delivered. And this is a healthy development.

Nevertheless, the path towards relevance-oriented research is not free of pitfalls. Applied research and development depends on previous basic research which becomes impossible if immediate results are demanded as a condition for further funding. Basic research naturally operates on a much longer time-scale than applied development. It requires larger investments into an uncertain future and carries higher risks. Such work has traditionally been the domain of universities, where the indispensable interaction between research and teaching is well established.

Universities doing basic, long-term research and industry pursuing applied research with short *time-to-market* therefore appear as ideal partners. For a proper functioning of a partnership it is essential that each partner is well aware of its specific role. If the interface, the contract becomes blurred, danger lurks. In particular, universities must not mutate gradually into business units, and professors should remain teachers rather than become business tycoons. Universities are ill-suited to do ultimate product engineering and ill-equipped for successful marketing, nowadays a prerequisite for every commercial endeavour in technology.

The modern term for university/industry cooperation is *technology transfer*. In its traditional form it works best in areas with established industries. In the modern field of high tech it typically manifests itself in the form of spin-off companies, founded by young scientists who transfer the new technology by transferring their heads and bodies.

*Oberon microsystems* is such a case in question. Its goal is to transform know-how acquired at ETH in the design of software systems into products meeting the needs of today's market. This is by no means an easy task, in particular because the goals of innovation and compatibility with existing products, tools, and habits are basically incompatible. The frequently encountered customer who seeks progress, but accepts it only when no changes are implied, cannot be satisfied - at least not in the long run. On the other hand, developers of new products should



muster some sympathy for the customers' resistance against too frequent, unnecessary or ill-motivated changes which appear as changes for their own (or marketing's) sake.

We believe that *Oberon microsystems* has a lot of substance to offer. It has found a sensible way to offer innovation without requiring total conversion. Oberon/F allows the programmer to make full use of a strongly typed, structured language, yet to generate modules that are fully integrated into the existing operating environment. It paves the way towards an industry based on software components, of interchangeable parts as it is customary in mature engineering disciplines. Oberon/F allows to build systems that are truly extensible, in which new modules introducing new functionality and new data types are merely added, with the full benefit of full compile-time interface checking, and without the requirement of recompilation and thereby the availability of source code.

The language Oberon, and with it the framework Oberon/F, is fortunately not a new product without connection with the past. On the contrary, it is, after Pascal and Modula-2, the youngest descendant in the Algol family of languages, standing for the ideas of structured and modular programming. To these cornerstones of programming methodology it adds object-orientation. Notably it does this without the need for a totally new set of terminology for commonly known concepts.

Oberon/F deserves to be investigated and to be given a try. Certainly, it is more convenient than conventional, a difference of character well-worth making.

---

The next article is [Component Software and Oberon: a Perspective on Oberon/F](#) by Bert Fitié.  
Go [back](#) to article selection.

---

Oberon microsystems, Inc.  
Technoparkstrasse 1  
8005 Zürich  
Switzerland

phone: ++41-1-445-1751  
fax: ++41-1-445-1752  
email: [oberon@oberon.ch](mailto:oberon@oberon.ch)

Back to our [Home Page](#).