



Doctoral Thesis

## Einsatz und Programmierung des Computers als Werkzeug für den Unterricht

**Author(s):**

Ventura-Meyer, Andrea Marion

**Publication Date:**

1985

**Permanent Link:**

<https://doi.org/10.3929/ethz-a-000352511> →

**Rights / License:**

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH Nr. 7752

# **Einsatz und Programmierung des Computers als Werkzeug für den Unterricht**

## **ABHANDLUNG**

zur Erlangung des Titels eines

**DOKTORS DER TECHNISCHEN WISSENSCHAFTEN**

der

**EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE  
ZÜRICH**

vorgelegt von  
**ANDREA MARION VENTURA-MEYER**  
dipl. Math. ETH  
geboren am 5. November 1956  
von Baden (AG)

Angenommen auf Antrag von  
Prof. Dr. J. Nievergelt, Referent  
Prof. Dr. C.A. Zehnder, Korreferent

Zürich 1985

## Kurzfassung

Der Computer ist ein vielseitiges Werkzeug für den Unterricht, auch ausserhalb des Fachs Informatik. Er kann eingesetzt werden zur Aufbereitung von Schriftstücken mit Text und Graphik, als Rechenhilfe, zur Übermittlung und Verwaltung von Informationen, als interaktives Vorführgerät für dynamische Abläufe, als Experimentiergerät, als Programmierumgebung und als Lehrer. Besonders interessant ist der Einsatz des Computers als Vorführ- und Experimentiergerät, weil dabei im hohen Masse seine besonderen Fähigkeiten (schnell rechnen, auf grosse Datenmengen zugreifen und bewegte Bilder produzieren, wobei die Bewegung vom Benutzer gesteuert werden kann) ausgenützt werden.

Für einige der oben beschriebenen Unterrichtsfunktionen können Standardprogramme verwendet werden, zum Beispiel Textsysteme für die Aufbereitung von Schriftstücken. Speziell auf eine Unterrichtssituation zugeschnittene Programme hingegen müssen vom Lehrer selbst entwickelt werden.

In dieser Arbeit wird eine praktisch erprobte Entwurfsmethode für interaktive Unterrichtsprogramme vorgestellt. Ihre Anwendung ermöglicht es einem Lehrer mit Programmierkenntnissen, aber ohne vertieftes Wissen über die Gestaltung interaktiver Programme, brauchbare Unterrichtsprogramme selbst zu entwerfen und zu programmieren. Dabei müssen didaktische und informatische Überlegungen einbezogen werden. Die Didaktik liefert Richtlinien über das Vorgehen zur erfolgreichen Wissensvermittlung. Die Informatik liefert Richtlinien für eine benützerfreundliche Gestaltung der Mensch-Maschine-Schnittstelle und Hinweise für ein effizientes Vorgehen bei der Programmierung.

Die Befolgung der beschriebenen Entwurfsmethode führt zu Unterrichtsprogrammen, deren Inhalt in Seiten organisiert ist. Die Beschäftigung des Benützers mit dem Programm, das heisst, das Durchlaufen der Seiten, kann anschaulich als Bewegung in einem Netzwerk von Seiten aufgefasst werden. Das Netzwerk dient also als Spezifikation für die Struktur des Programms aus der Sicht des Benützers. Durch geschickt gewählte Regeln für den systematischen Aufbau von Netzwerken kann erreicht werden, dass eine gegebene Netzwerkstruktur besonders einfach in ein Programm abgebildet werden kann.

Neben der Entwurfsmethode werden in dieser Arbeit zwei Werkzeuge für Entwurf und effiziente Programmierung von Unterrichtsprogrammen vorgestellt, ein portabler Programmgenerator und eine komfortable Arbeitsumgebung für Autoren interaktiver Unterrichtsprogramme. Der Programmgenerator erzeugt aus einer formalen Beschreibung einer Netzwerkstruktur ein ausführbares Programm. Er kann in jede Programmierumgebung eingebaut werden. Die Arbeitsumgebung bietet verschiedene Operationen zur Manipulation von Netzwerken an. Der Autor kann ein neues Netzwerk erzeugen, beziehungsweise ein bestehendes Netzwerk verändern, und legt so die Gliederung seines zukünftigen Programmes fest (Netzwerk-Editor). Das Verhalten des Programmes, wie es später der Benutzer beim Durchlaufen der einzelnen inhaltlichen Teile erlebt, kann simuliert werden, um dem Autor einen Eindruck über das Benützerverhalten seines Programms zu vermitteln (Netzwerk-Interpreter). Die für den Menschen mühsame Abbildung der Netzwerkstruktur in ein Programm wird ganz von der Arbeitsumgebung übernommen (Netzwerk-Übersetzer).

## Abstract

The use of the computer as an educational tool is not restricted to the teaching and learning of Informatics. Computer applications in education include document preparation, data management, computation, communication, display of dynamic processes, simulation of experiments, programming and teaching. The use of the computer-driven screen for displaying and experimenting with complex phenomena is the least known of these applications, which would not be feasible without the computer's capabilities for fast computation, accessing large amounts of data and presenting animated images that respond to the user's commands. Standard software is available for some of the above applications, such as text processing. Educational software tailored to a specific situation, on the other hand, must usually be developed by the teacher himself.

This thesis presents a methodology for the development of interactive programs for education. Using this methodology, a teacher with some programming experience but without extensive knowledge of the principles of interactive program design can write his own educational software. This requires attention to both the educational and the technical aspects. The educational side provides teaching principles, the technical side provides guidelines for the design of the man-machine-interface and for effective software development.

These rules lead to programs that are organized in "pages". The path the user can follow through the program, i.e. the motion from one page to the next one, can be described by a network. The network specifies the program as seen from the user's point of view. An appropriate selection of rules for building networks simplifies the later translation of a network to a program.

In addition to this methodology, this thesis presents some tools for the design and programming of educational software. One is a portable program generator which takes as input a formal description of a network and produces as output an executable program. The other is a powerful workbench for writing educational programs. One component offers the possibility of editing networks, i.e. of defining the structure of the program as seen by the user on a very high level. Executing the program, i.e. moving through the network of pages can be simulated for the author, who can evaluate it and easily modify the network as necessary. The last component of the workbench is a network translator which produces executable programs from network definitions.