

Diss. ETH Nr. 10485

**Benutzerzentrierte Betriebssoftware
eines
Echtzeit-Bildverarbeitungs-Rechners
für die
interaktive Applikationsentwicklung**

Abhandlung
zur Erlangung des Titels eines
Doktors der Technischen Wissenschaften der

EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE
ZÜRICH

vorgelegt von

Dieter Benedikt Stokar von Neuforn

dipl. El. Ing. ETH
geboren am 3. August 1959
von Küsnacht ZH und Schaffhausen SH

Angenommen auf Antrag von
Prof. Dr. W. Guggenbühl, Referent
Prof. Dr. H. Krueger, Korreferent

Zürich, 1994

ZUSAMMENFASSUNG

Mensch, Computer und Applikation bilden das Spannungsfeld, in dem sich diese Arbeit bewegt. Die Hardware ohne Applikation (Software) ist ein toter Haufen Elektronik. Die Applikation fällt nicht vom Himmel, sondern muss entwickelt und aufgebaut werden. Hierzu braucht es den Ingenieur, den Menschen, der einen wichtigen Pol im Spannungsfeld darstellt.

Das Bindeglied zwischen diesen drei Eckpfeilern, Mensch, Computer und Applikation, ist die im Rahmen der vorliegenden Arbeit entwickelte *Betriebssoftware* SYMON. Mit einer geeigneten *Benutzerschnittstelle* ermöglicht sie einerseits dem Benutzer, die sehr komplexe Hardware des Echtzeit-Bildverarbeitungsrechners SYDAMA-2 interaktiv zu konfigurieren und zu programmieren. Auf der anderen Seite bildet sie eine *Programmschnittstelle* (API, application programming interface), mit der Applikationsprogramme (z.B. für die optische Qualitätskontrolle einer industriellen Fertigungsstrasse) die Hardware ebenfalls kontrollieren und nutzen können.

'Benutzerzentriert' weist darauf hin, dass die Gestaltung von SYMON dem Blickwinkel des Anwenders grosse Bedeutung einräumte. Dies ist für hardwarenahe Software eher unüblich, weshalb dem Thema *Software-Ergonomie* ein gesondertes Kapitel gewidmet ist. Es beinhaltet insbesondere ein Set von Begriffen, mit deren Hilfe die Aspekte der Usability von SYMON am Ende der Arbeit diskutiert werden.

SYMON ist intern als ein *Client-Server-System* aufgebaut. Die Server dienen als Controller für die Hardware-Knoten und bilden eine virtuelle Maschine zur Ausführung von *Code-blocks*. Code-blocks ihrerseits bilden die Grundeinheiten der Programmierung grösserer Applikationen, die mit Hilfe von *Mode-sequences* zu komplexen Abläufen zusammengefügt werden. Die Systemmodellierung durch die Software beinhaltet im übrigen eine Unterscheidung in *Interframe-* und *Frame-Actions*. Mit dieser massiven Einschränkung der Vielfalt von möglichen Programmkonstrukten konnte eine maximale Vereinfachung der Programmierung erreicht werden, wobei dennoch hinreichende Möglichkeiten für die Realisierung typischer Applikationen zur Verfügung stehen.

ABSTRACT

Man, computer and applications constitute the field within which this work is allocated. Hardware without an application (software) is not worth much. However an application has to be developed and because of that the engineer plays an important role in this scene.

The "glue" between the three corner stones man, computer and application is the operating software SYMON, which was developed in the course of the work presented here. With a convenient user interface it provides the application programmer the means to interactively configure and program the real-time image processing computer SYDAMA-2. On the other hand the operating software provides the applications programming interface (API) with which application programs (such as an optical quality control for a production line) may make use of SYDAMA-2's vast processing power.

As usability was considered to be an important aspect in the design of SYMON which is somewhat unusual for low-level software, a chapter was dedicated to software-ergonomy. In particular it contains a set of words which build a framework to discuss aspects of usability of software. In this way SYMON itself is discussed at the end.

The operating software SYMON is based on the client-server model. The servers act as controllers of SYDAMA-2's hardware nodes. They implement virtual machines which execute *code-blocks*. Code-blocks are the building blocks for programming larger applications. By means of *mode-sequences* they are combined to form complex sequences as are needed to run the machine SYDAMA-2. The programming model further distinguishes between *interframe-* and *frameactions*.

With these abstract programming constructs a good compromise could be found to allow for the realization of complex applications without sacrificing simplicity of programming despite of the complexity of the hardware and envisioned applications.