ex B

Diss. ETH No. 7425

XS-2:
THE USER INTERFACE
OF AN INTERACTIVE SYSTEM

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of
Doctor of Technical Sciences

presented by

JAN STELOVSKY
Dipl. Math. ETH, Master of Arts

born February 10, 1952
from Czechoslovakia

Accepted to the recommendation of

Prof. Dr. J. Nievergelt, examiner
Prof. Dr. E. Ulich, co-examiner

J. Nievergelt

Zürich 1984

Kurzfassung

Das eXperimentelle System **XS-2** ist ein integriertes, interaktives System mit dem Ziel, allen Klassen von Benutzern eine einheitliche Schnittstelle zur Verfügung zu stellen. Es richtet sich sowohl an den ungeübten, als auch den erfahrenen Benutzer, genauso wie an den Anwendungsprogrammierer und den Systemmanager. Der Benutzer kann sich auf ein explizites Systemmodell stützen. Um die Komplexität dieses Modells klein zu halten, wurden bei Analogien gleiche Konzepte verwendet.

Der Systemzustand ist in Daten, Befehle und Dialoggeschichte aufgeteilt. Diese Zustandskomponenten sind stets in separaten Bildschirmfenstern sichtbar und hierarchisch strukturiert. Wenige universelle Befehle genügen, um die Bewegungen und Sichtänderungen an diesen Strukturen auszuführen. Diese universellen Befehle sind immer aktiv. Die Sichtänderungen erlauben es dem Benutzer, den ganzen Systemzustand auf einfache Art zu erforschen, ohne ihn dabei zu verändern. So kann zum Beispiel jeder Befehl oder Parameter im ganzen System sichtbar gemacht und individuell erklärt werden.

Die XS-2 - Befehlssprache wird in Form von Befehlsbäumen dargestellt, die Befehle samt allen ihren Parametern enthalten: in XS-2 entfällt die Unterscheidung zwischen Befehlen und Parametern. Die Struktur aller Befehlsbäume unterliegt gleichen Regeln. Eine kleine Anzahl von Befehlstypen klassifiziert die Befehle. Die Semantik dieser Befehlstypen ist leicht verständlich; sie bestimmt die Bewegung auf den Befehlsbäumen. Auch die universellen Befehle bilden Befehlsbäume. Daher "sprechen alle Anwendungsprogramme sowie der XS-2 Kern die gleiche Sprache". Die ganze Syntax der Befehlssprache ist im Befehlsfenster sichtbar und kann zur Eingabe benutzt werden: die gezeigten Befehle können mit der Maus ausgewählt werden. Aber auch weit entfernte Befehle können einfach ausgewählt werden. Der Benutzer kann von einer beliebigen Stelle in XS-2 jeden Befehl ausführen, dessen Name ihm bekannt ist.

Der Benutzer von XS-2 kann die Befehle eines Anwendungprogrammes nach seinen Bedürfnissen massschneidern. Ein Befehl kann umbenannt, in einem Menü können die Befehle umgeordnet oder in Untermenüs aufgespalten werden. Mehrere alternative Befehlsstrukturen zu einem Befehl können in eine Anwendung eingefügt und getestet werden. Alle diese Änderungen können unmittelbar in XS-2 durchgeführt werden.

Die Anwendungsprogrammierer entwerfen die Befehle innerhalb von XS-2 mit Hilfe der standardmässig zur Verfügung gestellten Werkzeuge - wie zum Beispiel des syntax-gesteuerten Baumeditors. Ein Rahmenprogramm, das die Aktionen eines neuen Befehlsbaumes simuliert, kann innerhalb des Systems generiert und compiliert werden. Dadurch wird XS-2 zu einem "fast prototyping tool" für den Entwurf der Benutzerschnittstelle interaktiver Anwendungsprogramme. Neue Datentypen können in Form von hierarchisch strukturierten Grammatiken definiert werden. Der XS-2 Kern bietet systematisch entworfene Softwareschnittstellen für den Zugang zu den eigenen Betriebsmitteln.

Das XS-2 System versucht einen weiteren Schritt auf dem Weg zur Standardisierung des Mensch-Computer Dialogs auf eine konsistente und systematische Art zu tun.

Abstract

The eXperimental System **XS-2** is an integrated interactive system. Its goal is to provide uniform human-computer interfaces for several categories of users: from the casual to the experienced user as well as the application programmer and the system designer. The user is supplied with an explicit model of the system. The complexity of this model has been reduced by exploiting analogies and minimizing the number of concepts used.

The system state is partitioned into data, commands and dialog history. These state components are organized as trees and always visible in separate windows on the screen. A small set of universal commands performs motion and browsing on these structures. The universal commands are active at all times. The browsing facility allows the user to conveniently explore the entire state of the system without changing it. In particular, any of the commands - or parameters - in the entire set of applications can be visualized and individually explained.

The XS-2 command language is represented by command trees that contain commands with all their parameters: there is no distinction between commands and parameters. All command trees are built using a limited set of command types and obey a common set of rules. The semantics of the command types is easy to grasp; it determines the motion on command trees. Also the universal commands are embedded in command trees. Therefore, all the applications as well as the XS-2 kernel speak the same language. The entire syntax of the command language is visible in the command window and can be used as input: the commands shown there can be selected with the mouse. Also distant commands can be accessed conveniently. The user who marks a command or remembers its name can easily activate it from anywhere else in the system.

The user of XS-2 can tailor the commands of an application according to his needs. A command can be renamed, the commands in a menu can be regrouped or subdivided into a menu hierarchy. Several alternative command structures for the same command can be introduced and tested within one application. These modifications can be performed comfortably within the system.

Application programmers develop the command structures with the standard XS-2 utilities such as the syntax-directed Tree Editor. A skeleton program for the actions of a new command tree can be generated and compiled within the system. Thus, XS-2 is a fast prototyping tool for the design of the application's user interface. New data types can be defined using hierarchically structured grammars. The XS-2 kernel provides systematic software interfaces for the access to its own resources.

The XS-2 system represents an attempt to standardize of human-computer dialog in a consistent and systematic way.