



Doctoral Thesis

Hide and idea tools for user-oriented application development

Author(s):

Greutmann, Thomas

Publication Date:

1992

Permanent Link:

<https://doi.org/10.3929/ethz-a-000635902> →

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 No. 9642

**HIDE AND IDEA:
TOOLS FOR USER-ORIENTED
APPLICATION DEVELOPMENT**

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of
Doctor of Technical Sciences

presented by
Thomas Greutmann
born April 3rd 1962
citizen of Beggingen SH

accepted on the recommendation of
Prof. Dr. Eberhard Ulich, examiner
Prof. Dr. Hans-Peter Frei, co-examiner

Zürich, 1992

S U M M A R Y

The main topic of this work is the development of applications with good usability. Since usability is related to (1) the application and (2) to the users, aspects of computer science and work psychology are equally important. "Designing an application with good usability" implies a joint optimization of technical and organizational aspects in order to improve the entire work system. "Usability" must not be reduced to a characteristic of the application or the user interface only, as it is often done in definitions of "usability". This implies that applications must be optimally adapted to task characteristics as well as to individual characteristics of the users. Since users learn and develop, this is a permanent optimization process which is not terminated after application development but continues during application use. Application development must be based on active user involvement, users and designers must cooperate closely in an iterative design process. Design tools and design methods have a well-defined role in this design process: they are communication means for supporting user-designer cooperation.

Design tools which support user-designer cooperation and the design of applications with good usability should fulfill the following requirements:

1. They must be flexible and extendible in order to allow an optimal adaptation to task-specific aspects.
2. Moreover, they must be portable with respect to user interface style in order to
3. support existing user interface standards.
4. They should easily allow to create realistic simulations of applications - a basis for user-designer discussions.
5. Features for evaluating designs as well as for
6. individual adaptations of applications must be also included in design tools.

The User Interface Management System (UIMS) HIDE and the User Interface Development System (UIDS) IDEA have been developed to fulfill these requirements. HIDE has an open-ended, extendible architecture: task-specific dialogue elements can be added to HIDE as building blocks. Each of the building blocks has a layered structure where user interface style dependent and independent parts of a dialogue element are separated. HIDE dialogue elements have an "abstract" application and editor interface: the abstract application interface makes applications portable with respect to user interface styles, the editor interface allows designers to create their own support tools for design. Due to this kind of flexible architecture, HIDE is very portable. To illustrate this, five HIDE versions with very different user interface styles could be implemented by replacing the user interface dependent parts. Although very different in appearance and behavior, these HIDE versions share up to 70% of the code. The interactive design tool IDEA can be used to edit user interface descriptions which are interpreted by the dialogue handler HIDE. IDEA includes a simulation

SUMMARY

facility by means of which user interfaces can be visualized dynamically, in order to support user-designer cooperation.

The development of the tool IDEA was done in a cooperative and iterative design process, too. In addition to cooperative design, a combined usability test strategy including ad-hoc, systematic and quasi-experimental usability tests was used. This combined test strategy helps to detect usability drawbacks in the development process.

Moreover, experimental investigations of the cooperative, tool-supported design process have been carried out. The results of these experiments - although carried out with a small number of subjects - allows to specify more precisely how designers and users should cooperate in the design process.

HIDE and IDEA have been designed to support usability and the users, so HIDE/IDEA differ from other UIMS and UIDS which usually are implementation aids "by designers and for designers". Differences can be found (1) for the dialogue handler architecture, e.g. the layered approach with editor and application interface which supports portability, (2) for the functionality of design tools, e.g. the simulation facility with fast switches between editing and simulation or possibilities for individual adaptations by the users and (3) especially for the integration of tools into a methodical framework, the cooperative design approach with active user involvement.

ZUSAMMENFASSUNG

Die Entwicklung von EDV-Anwendungen mit hoher Benutzungsfreundlichkeit ist das Hauptthema dieser Arbeit. Dazu müssen sowohl technische Aspekte (Informatik) als auch Benutzer-Aspekte (Psychologie, im speziellen Arbeitspsychologie) berücksichtigt werden. Es wird von einem umfassenden Verständnis der "Benutzungsfreundlichkeit" ausgegangen, welches von einer gemeinsamen Optimierung technischer und organisationaler Aspekte eines Arbeitssystems ausgeht, um das Funktionieren des Gesamtsystems zu verbessern. "Benutzungsfreundlichkeit" darf nicht zu einer Eigenschaft der EDV-Anwendung alleine reduziert werden. Dieses Verständnis von Benutzungsfreundlichkeit bedingt, dass EDV-Anwendungen sowohl an die Eigenschaften der Arbeitsaufgaben, welche mit dem System unterstützt werden sollen, als auch an individuelle Eigenschaften und Fähigkeiten der Benutzer optimal angepasst werden müssen. Da Benutzer nicht statisch sind, sondern bei der Ausführung ihrer Arbeitsaufgaben dazulernen und sich weiterentwickeln, ist die Entwicklung von benutzungsfreundlichen Anwendungen ein permanenter Optimierungsprozess, der nicht mit der Entwicklung der Anwendung abgeschlossen wird, sondern während der Nutzung weiterdauert. Aus diesen Gründen ist ein aktiver Benutzereinbezug mit enger Benutzer-Entwickler-Zusammenarbeit in einem iterativen Entwicklungsprozess unabdingbar. Die Rolle von Entwurfsmethoden und -werkzeugen ist dabei klar definiert: sie dienen als Kommunikationshilfsmittel, welche den Prozess unterstützen. Die Verwendung von Methoden und Werkzeugen garantiert nicht automatisch gute Lösungen im Hinblick auf Benutzungsfreundlichkeit, gute Lösungen sind vielmehr das Resultat einer adäquaten Zusammenarbeit zwischen Entwicklern und Benutzern.

Werkzeuge, welche diese Art von Entwicklungsprozess unterstützen, sollten folgende Eigenschaften aufweisen:

1. Sie müssen flexibel und erweiterbar sein, um eine optimale Anpassung an die Eigenschaften der jeweiligen Arbeitsaufgabe zu gewährleisten.
2. Sie müssen portabel im Hinblick auf die Art der Benutzungsoberfläche sein, um
3. eine Unterstützung von existierenden Standards zu ermöglichen.
4. Mit Hilfe der Werkzeuge sollen rasch realistische Simulationen von EDV-Anwendungen erstellt werden können, da diese eine geeignete Grundlage für Diskussionen zwischen Benutzern und Entwicklern sind.
5. Die Werkzeuge sollten Komponenten zur Evaluation von Entwürfen und
6. zur Anpassung an individuelle Benutzerbedürfnisse beinhalten.

Ausgehend von diesen Anforderungen wurde das User Interface Management System (UIMS) HIDE und das User Interface Development System (UIDS) IDEA entwickelt. HIDE besitzt eine offene und erweiterbare Architektur: aufgabenspezifische Dialogelemente können als Bausteine jederzeit hinzugefügt werden. Jeder solcher Baustein ist schichtenförmig aufgebaut, wobei die Teile,

ZUSAMMENFASSUNG

welche abhängig bzw. unabhängig von der Art der Benutzerschnittstelle sind, klar getrennt wurden. Auf die HIDE-Dialogelemente wird über "abstrakte" Anwendungs- und Editorschnittstellen zugegriffen. Die abstrakte Anwendungs-schnittstelle macht die Anwendungen unabhängig von der Art der Benutzerschnittstelle und daher portabel, die Editorschnittstelle ermöglicht es den Entwicklern, eigene Unterstützungswerkzeuge zu erstellen. Durch diese flexible Architektur ist HIDE sehr portabel, es konnten so fünf Versionen mit sehr unterschiedlichen Arten von Benutzerschnittstellen implementiert werden. Obwohl sehr unterschiedlich in Aussehen und Dialogabläufen haben diese Versionen bis zu 70% des Codes gemeinsam. Benutzerschnittstellenbeschreibungen können mit dem interaktiven Entwurfswerkzeug IDEA erstellt werden und werden dann von HIDE interpretiert. IDEA beinhaltet eine Simulationskomponente, mit welcher das dynamische Verhalten von Benutzerschnittstellen visualisiert werden kann, was die Benutzer-Entwickler-Kooperation unterstützt.

IDEA wurde in einem kooperativen und iterativen Entwurfsprozess entwickelt. Zusätzlich zum kooperativen Entwurf wurde eine kombinierte Benutzerteststrategie mit Ad-Hoc-Tests, systematischen Tests und quasi-experimentellen Tests verwendet. Diese kombinierte Teststrategie hilft, Mängel aus Benutzersicht zu entdecken und zu verbessern.

Daneben wurden experimentelle Untersuchungen zum kooperativen, werkzeu-gunterstützten Entwurfsprozess durchgeführt. Obwohl nur eine kleine Anzahl Versuchspartner an den Experimenten teilnahm, konnten gezielte Hinweise zur Kooperation von Entwicklern und Benutzern im Entwurfsprozess gewonnen werden.

HIDE und IDEA wurden ausdrücklich als benutzerorientierte Werkzeuge konzipiert, daher unterscheiden sie sich von anderen UIMS und UIDS, welche in der Regel als Implementationswerkzeuge "von Entwicklern und für Entwickler" ausgelegt sind. Unterschiede manifestieren sich (1) bei der internen Architektur des UIMS, z.B. der Schichtenaufbau mit Anwendungs- und Editorschnittstelle, (2) bei der Funktionalität der Entwurfswerkzeuge, z.B. die Simulationskomponente sowie die Möglichkeiten zur individuellen Veränderung der Benutzerschnittstelle und (3) besonders bei der Integration der Werkzeuge in einen methodischen Rahmen, den kooperativen Ansatz mit aktiver Benutzerbeteiligung.