

Theoretische und empirische Untersuchungen der Benutzeroberfläche von zwei CAD-Systemen

Paul Tavalato, Karl-Heinz Weidmann, Wien

Zusammenfassung

Die Benutzerschnittstellen zweier komplexer Anwendungsprogramme, die für denselben Verwendungszweck entwickelt wurden, die allerdings stark unterschiedliche Benutzeroberflächen aufweisen, wurden sowohl mit Hilfe theoretischer Modelle als auch mittels eines empirischen Versuchs miteinander verglichen. Für die theoretische Untersuchung wurde ein aus der Literatur bekanntes formales Verfahren adaptiert und erweitert. Ziel des Projekts war einerseits der Vergleich zweier unterschiedlicher Benutzerschnittstellen und andererseits die Validierung theoretischer Modelle zur Feststellung der Benutzerfreundlichkeit einer Programmoberfläche durch empirische Untersuchungen.

1. Projektübersicht

Im Rahmen eines Forschungsprojekts wurden zwei funktional idente Planungsprogramme für den Architekturbereich (2-D-CAD-Programme mit Massenermittlung und Kostenberechnung) entwickelt. Die Benutzeroberflächen der Programme sind allerdings unterschiedlich gestaltet. Diese Konstellation eröffnete die Möglichkeit, die Benutzerschnittstellen sowohl mit Hilfe theoretischer Modelle als auch empirisch zu untersuchen und miteinander zu vergleichen.

Für die theoretischen Analysen wurde eine Erweiterung der Task Action Grammar (TAG) von Payne und Green (1986) entwickelt (ATAG – Advanced Task Action Grammar). Die Benutzeroberflächen beider Programme wurden mit Hilfe dieser Beschreibungsmethode modelliert und verglichen.

Bei den empirischen Untersuchungen wurde eine Gruppe von 30 Testpersonen mit den beiden Programmen konfrontiert; anschließend wurden von allen Testpersonen zwei eigens im Rahmen des Projekts entwickelte Fragebögen ausgefüllt. Die Auswertung dieser Fragebögen wurde dann den Ergebnissen der theoretischen Analyse gegenübergestellt.

Das Resultat der Untersuchungen dient zur Zeit als Grundlage einer Überarbeitung der Benutzerschnittstelle der Programme.

2. Die beiden Programme

Beide Programme sind funktional als Planungswerkzeug für den Wohnbau gedacht und richten sich in ihrer Problemlösungsstrategie nach der SAR-Planungsmethode (vgl. Boekholt et al., 1974 oder Chaslin et al., 1976). Diese unterscheidet zwischen der Planung der Primärstruktur eines Gebäudes (tragende Bauteile, Erschließungen, Ver- und Entsorgungsstränge) und der Planung der Sekundärstruktur (nicht-tragende Wände, Türen, Fenster, Installationen u.s.w.), um flexible und variable Wohnstrukturen zu ermöglichen. Unter Zuhilfenahme eines Maßrasters und einer Zonierung werden die Lage und Abmessungen von Bauteilen sowie deren Zuordnung untereinander geregelt.

Die Bauteile werden einem Bauteil-Katalog entnommen und in den jeweiligen Plan eingesetzt. Der Bauteil-Katalog enthält die grafische Repräsentation und andere Informationen (über Maße und Kosten); die Katalogelemente müssen mit Hilfe des Programmsystems manipuliert werden können.

Daraus ergibt sich eine Dreiteilung der Planungsaufgaben (Planung der Primärstruktur, Planung der Sekundärstruktur, Bearbeiten des Bauteil-Katalogs), die sich auch im Programmsystem widerspiegelt: Das System besteht aus drei einzelnen Programmen: Primärstruktur bearbeiten, Wohnungsplanung und Bauteil-Katalog bearbeiten. Die drei Programme sind einander sehr ähnlich; das umfangreichste ist das Programm zur Planung der Sekundärstruktur (Wohnungsplanung). Es erschien sinnvoll die Untersuchung auf diesen Bereich zu beschränken. (Die beiden anderen Bereiche stellen im wesentlichen Untermengen dieses Bereichs zur Planung der Sekundärstruktur dar.)

Das Programm MBP wurde im Architekturbüro Ottokar Uhl in Wien entwickelt und wird dort einerseits zur Wohnungsplanung mit Nutzermitbestimmung eingesetzt (das bedeutet, daß das Programm auch von den künftigen Bewohnern der geplanten Wohnungen selbst, also von Personen ohne jede EDV-Vorkenntnis, bedient wird) und andererseits wird es an der Technischen Universität Wien im Rahmen der EDV-Ausbildung der Architekturstudenten eingesetzt. Die Benutzeroberfläche des Programms kann – im Sinne von Shneiderman (1987) – als "modeless, direct manipulative" charakterisiert werden. Implementiert ist das Programm auf Rechnern der Type "Apple MacIntosh".

Das Programm PROSAR wurde an der Universität Karlsruhe am Institut für Gebäudelehre, Lehrstuhl für Bauplanung und Entwerfen entwickelt und wird dort im Rahmen der Architekturausbildung eingesetzt. Die Benutzeroberfläche dieses Programms ist ihrer Charakteristik nach der Gruppe der Menü-hierarchischen Systeme zuzuordnen. Implementiert ist das Programm auf Rechnern der Reihe "IBM/PC" mit EGA-Schirm und Maus.

3.Theoretische Untersuchungen

Im Rahmen der theoretischen Untersuchungen wurden die Benutzeroberflächen beider Programme mit Hilfe einer formalen Beschreibungsmethode modelliert. Als Beschreibungsmethode wurde eine Erweiterung der Task Action Grammar (TAG) von Payne und Green (1986) entwickelt, die Advanced Task Action Grammar (ATAG); dabei wurde in erster Linie auf Ideen der Command Language Grammar (CLG) von Moran (1981) und des Gulf-Modells von Norman und Draper (1986) zurückgegriffen. TAG bietet dabei eine Beschreibungsmöglichkeit des Bereichs zwischen Aufgabe (Task) und der hierfür auszuführenden (physikalischen) Aktion(en) (Action). Die formale Beschreibung gliedert sich in 3 Bereiche:

- > **Einfache Tasks**
- > **Task Features**
- > **Regelschemata**

- Die Liste der einfachen (simple) Tasks stellt eine Übersetzung der Features in eine formale Notation ähnlich den Fakten in der Programmiersprache Prolog dar. Die einfachen Tasks dürfen, konzeptuell gesehen, keine Kontrollstrukturen (wie z.B. Verzweigungen oder Iterationen) enthalten. Das heißt, sie stellen solche Aktionssequenzen des Benutzers dar, die ohne Problemlösungsaufwand durchgeführt werden können. Der Problemlösungsaufwand wird einem Planungssystem überlassen, das in TAG nicht näher untersucht wird. Nach Ansicht von Payne und Green (1986, S. 121) handelt es sich bei der Liste der einfachen Tasks um ein psychologisch relevantes Konstrukt. Begründet wird diese Haltung unter anderem mit dem Verweis auf die Arbeit von Moran (1981) über External-Internal Task-mapping. In diesem Zusammenhang wird die Notwendigkeit der Existenz psychologischer Abbildungen von externen auf interne Tasks unterstrichen und weiters auch der Erwerb der Semantik und der Syntax durch den Benutzer hervorgehoben. Beides findet in den Simplen Tasks seine Entsprechung.
- Die Liste der Task Features (strukturierte Gruppierungen einzelner Tasks) stellt konzeptuell Bausteine der Kommandos mit ihren möglichen Realisierungen zusammen. Das Konzept des Featuring stellt ein wichtiges Konstrukt für TAG dar, da es als psychologische Theoriebasis angenommen wird. Die Hypothese lautet: Mentale "Konzepte" werden durch Task-Strukturen realisiert.
- Die Regelschemata sind das Endprodukt von TAG. Ihre Notation hat eine gewisse Ähnlichkeit zur Syntax von Prolog. Die Variablen über die gesamte Regel werden vom Aufrufer her belegt. Es wird ein Pattern-Matching durchgeführt, das durch die Vorbelegung der Variablen der Regel mit Keywords, die in den Task Features aufgezählt sind, zum Ausdruck kommt. Terminale werden durch Anführungsstriche gekennzeichnet. Weiters gibt es sogenannte "known-items", die

das Gebrauchswissen (worldknowledge, common sense) ansprechen, um so eine angepaßte Anzahl von Regeln zu erhalten. Die Anzahl der Regeln ist deshalb so wichtig, da sie die Hauptmetrik für die Erlernbarkeit darstellt. Zusätzlich wird noch eine Metaregel eingeführt. Diese lautet:

Die Beschreibung durch die Regelschemata ist dahingehend zu optimieren, daß die Anzahl der verwendeten Regeln minimiert wird.

Die Formalisierung einer Benutzerschnittstelle mittels TAG soll quantifizierte Aussagen über folgende Merkmale der Benutzerschnittstelle liefern:

- **Konsistenz**

Syntaktische Konsistenz:

Eine syntaktisch konsistente Benutzerschnittstelle weist eine einheitliche Struktur auf, die wie ein roter Faden wirkt. Das heißt, die Syntax der einzelnen Befehle sollte sowohl in sich (was die Parameter betrifft) als auch global strukturell konsistent sein: Gleiche mentale Konzepte sollen durch gleiche syntaktische Strukturen ausgedrückt werden. Außerdem soll es dem Benutzer möglich sein, Gruppierungen vorzunehmen.

Lexikalische Konsistenz:

Unter lexikalischer Konsistenz versteht man den Zusammenhang zwischen einem Wort und seiner Bedeutung einerseits und seiner Aktionsbelegung in einem Programm andererseits. Insbesondere kommt es dabei auf die Begriffsbildung an.

Semantisch-Syntaktische Konsistenz:

Darunter fällt der Effekt des Lern-Transfers im einschränkend reflexiven Sinn. Wenn zum Beispiel im Kommando zum Kopieren von A nach B die Parameter diese Reihenfolge haben, dann sollte auch ein Kommando zum Bewegen von A nach B in der gleichen Art und Weise aufgebaut sein.

Semantische Konsistenz:

Auch hier ist der Effekt des Lern-Transfers im einschränkend reflexiven Sinn wesentlich. Wenn es zum Beispiel ein Kommando zur Selektion gibt, so wird erwartet, daß auch eines zur Deselektion existiert.

- **Erlernbarkeit und Performance**

Für Aussagen hierzu werden Metriken über dem Regelschema verwendet.

- **Abkürzungen**

Sie sollen sowohl eine gewisse Regel- und Sinnhaftigkeit aufweisen als auch umgangssprachlich vorhandenen Abkürzungen nicht entgegenstehen.

Beim verhältnismäßig geringen Umfang einer Beschreibung kann also eine Reihe quantifizierter Aussagen über wesentliche Eigenschaften der Benutzerschnittstelle erreicht werden.

Allerdings ergaben sich bei der Anwendung von TAG auf eine komplexe Schnittstelle in der Praxis einige Schwierigkeiten, auch solche, die in der Arbeit von Payne und Green (1986) nicht erwähnt werden. Insbesondere sind jene Bereiche, wo Intentionen über die Tasks modelliert werden, wenig zufriedenstellend gelöst.

Daher mußte eine neue Lösung gefunden werden, die diese Schwierigkeiten löst: Es wurde TAG mit folgenden Erweiterungen versehen:

- Beschreibung der INTENTIONEN
- Beschreibung von KOMPLEXEN TASKS und AGGREGATIONEN
- Beschreibung der BENUTZEROBJEKTE und BENUTZERARBEITSDATEN
- Einführung von KONTEXTEN bei einfachen Tasks
- Unterscheidung zwischen Status- und Arbeitsvariablen bei der Notation der Regelschemata
- Verbesserung der NOTATION der einfachen Tasks und der Regelschemata (also der steuernden Struktur)

Es wird im Zusammenhang mit der Strukturierung der Tasks eine Schnittstelle zu den komplexen Tasks, also zu solchen, die eine Kontrollstruktur enthalten, hergestellt; dabei wird das Prinzip des Featuring nicht verletzt. Es wird nicht versucht, das kognitive Planungssystem zu beschreiben, sondern die Bedingungen, denen das Planungssystem zusätzlich Rechnung tragen muß. Vornehmlich haben diese Bedingungen den Charakter zeitlicher Einschränkungen: Wann hat was zu geschehen. Dadurch wird man beim Entwurf einer Benutzerschnittstelle gezwungen, sich über die Intention und die komplexen Tasks des Benutzers (nach Moran, 1981) klar zu werden.

Die eingeführten Benutzerobjekte gehören zur externen Taskwelt (nach Moran, 1984) und geben Auskunft über die Objekte, auf denen der Benutzer seine Operationen konzeptuell durchführt. Sie dienen vor allem zur Erleichterung der Überprüfung der semantischen Konsistenz und der Aufgabenangemessenheit.

Aufschluß über die mentale Belastung des Benutzers bzw. notwendige Feedback-Elemente des Systems geben die Benutzerarbeitsdaten. Sie dienen einer besseren Evaluierung des Systems.

Die Kontexte wurden hauptsächlich eingeführt, um auch einfache Tasks (im Sinne von Payne und Green, 1986) mit Iterationen (ebenfalls im Sinne von Payne und Green, 1986, S. 121) beschreiben zu können. Wobei dies nicht der einzige Vorteil ist. Die Anzahl der Kontexte kann auch als zusätzliche Metrik für die Komplexität der

Benutzeroberfläche verwendet werden. Auf diese Tatsache deutet auch die empirische Untersuchung hin. Die Kontexte teilen, formal gesehen, die Liste der einfachen Tasks (dictionary of simple tasks) in Kapitel (chapters) ein.

Im hier beschriebenen Projekt wurde die formale Analyse in zwei Teilen durchgeführt: Im ersten Teil wird die Beschreibungsmethode ATAG auf beide Programme (MBP und PROSAR) angewendet und die entsprechenden Metriken werden berechnet; im zweiten Teil werden die beiden Programme nach allgemeinen Richtlinien zur Gestaltung von Mensch-Maschine-Schnittstellen (wie sie etwa in DIN 66 234, 1986, in Haider et al., 1987, von Krüger und Müller-Limmroth, 1979, von Shneiderman, 1987, von Smith und Mosier, 1984 oder von Spinax, Troy und Ulich, 1983 definiert worden sind) beurteilt.

Die Ergebnisse der formalen Analyse zeigen, daß das Programm MBP konsistenter im Design ist, weniger Kontextwechsel benötigt und leichter erlernbar ist, als das Programm PROSAR. Auf Grund der Benutzerobjekte ergibt sich auch, daß MBP besseres Feedback liefert.

4. Empirische Untersuchungen

An dieser Untersuchung nahmen 30 Versuchspersonen teil; alle Versuchspersonen waren Architekturstudenten mit durchwegs sehr geringen Erfahrungen im Umgang mit dem Computer; alle waren absolute Anfänger in Bezug auf die beiden getesteten Programme.

Die Versuchspersonen absolvierten je eine Sitzung an jedem der beiden Programme. Nach einer Einführung in die Planungsmethode und die Grundprinzipien der Programmbedienung blieben sie sich mehr oder weniger selbst überlassen und sollten die Programme an Hand einer offenen Aufgabenstellung "erforschen". Am Ende jeder Sitzung wurden den Versuchspersonen je zwei Fragebogen vorgelegt.

Der erste Fragebogen diente zur Erfassung der Benutzerfreundlichkeit der beiden Programme. Beim zweiten Fragebogen handelte es sich um einen Test zur Erlernbarkeit der Programme. Die Fragebogen wurden speziell für diese Untersuchung konzipiert und vor ihrem Einsatz einem eingehenden Vortest unterworfen.

4.1 Der Benutzerfreundlichkeits-Fragebogen

Der Fragebogen zur Benutzerfreundlichkeit umfaßt 45 Items und besitzt im allgemeinen eine 5-teilige Ratingskala. Die Fragen gruppieren sich in folgende Subskalen: Physikalische Eigenschaften, Informationsangebot, Terminologie, Zeitverhalten, Aufgabenangemessenheit, Konsistenz, Hilfen, Fehlerbehandlung und subjektive Erlernbarkeit. Diese Subskalen stimmen im großen und ganzen mit den faktorenanalytisch ermittelten Subskalen: Operationen, Erlernbarkeit, Konsistenz, Informationsverarbeitung und Bildschirmgestaltung, graphische Darstellung, mentales Modell, Hilfen, Aufgabenangemessenheit, Bezeichner, überein.

Durch die hohe Korrelation innerhalb der Subskalen und zwischen den Subskalen ist es möglich, einen Gesamt-Score zu ermitteln und diesen zu vergleichen. Selbstverständlich wurden auch detailliertere Untersuchungen durchgeführt. Hier soll aber nur kurz das Hauptergebnis dargestellt werden. Zur statistischen Auswertung wurde unter anderem ein Wilcoxon (matched pairs) Test verwendet. Die Berechnungen wurden auf einer IBM 3083 mit dem Statistikpaket SPSSx durchgeführt (statistische Bezeichnungen beziehen sich im folgenden immer auf die SPSSx-Beschreibung von Schubö und Uehlinger, 1986).

WILCOXON MATCHED-PAIRS SIGNED-RANKS TEST

GESPC WITH GESMC (Summe über alle Ratingskalen; GESMC für die Scores des Programms MBP; GESPC für die Scores des Programms PROSAR)

MEAN RANK	CASES	
16.22	23	- RANKS (GESMC < GESPC)
13.14	7	+ RANKS (GESMC > GESPC)
	0	TIES (GESMC = GESPC)
	--	
	30	TOTAL
Z = -2.8899	2-TAILED P = 0.0039	

Ergebnis des Wilcoxon Tests für die Daten des Hauptversuchs

Aus den obigen Daten kann abgelesen werden, daß die Vergleiche der Wertungen der Versuchspersonen 23-mal eine geringere (=bessere) Wertung der Eigenschaften des Programms MBP gegenüber PROSAR ergeben hat und 7-mal die umgekehrte Variante gültig ist. Gleiche Wertungen (Ties) gibt es keine. Dieses Ergebnis wird auf einem Wahrscheinlichkeitsniveau von 0.039 angenommen (da das Signifikanzniveau zuvor auf 5% gesetzt wurde). MBP wird also signifikant besser beurteilt.

Zur Reliabilität des Fragebogens kann noch angeführt werden, daß Auswertungen zu Cronbach's Alpha (Kuder-Richardson Formel 20) für den gesamten Fragebogen Werte mit rund 0.8 annehmen.

4.2 Der Erlernbarkeits-Fragebogen

Der Fragebogen zur Erlernbarkeit ist ein multiple-choice Test mit 18 Fragen. Die Konzeption folgt den Richtlinien eines Power-Tests.

Durch eine Faktorenanalyse konnten folgende Subskalen ermittelt werden:

- **HÄUFIGE OPERATIONEN**
- **NICHT VORHANDENE OPERATIONEN**
- **DETAILWISSEN**
- **SELTENERE OPERATIONEN**

Auch hier konnte eine hohe Korrelation innerhalb der Subskalen und zwischen den Subskalen festgestellt werden. Daher kann ein Gesamt-Score ermittelt und dieser verglichen werden. Selbstverständlich wurden auch hier detailliertere Untersuchungen durchgeführt. Im folgenden wird kurz das Hauptergebnis dargestellt:

Zur statistische Auswertung wurde unter anderem wieder ein Wilcoxon (matched pairs) Test verwendet. Die Berechnungen wurden ebenfalls auf einer IBM 3083 mit dem Statistikpaket SPSSx durchgeführt.

WILCOXON MATCHED-PAIRS SIGNED-RANKS TEST

GESPC WITH GESMC (Summe über alle Ratingskalen; GESMC für die Scores des Programms MBP; GESPC für die Scores des Programms PROSAR)

MEAN RANK	CASES	
8.25	8	- RANKS (GESMC < GESPC)
15.24	17	+ RANKS (GESMC > GESPC)
	5	TIES (GESMC = GESPC)
	--	
	30	TOTAL
Z = -2.5965		2-TAILED P = 0.0094

Ergebnis des Wilcoxon Tests für die Daten des Hauptversuchs

Aus den obigen Daten kann abgelesen werden, daß die Vergleiche der Ergebnisse (Summe über alle richtigen Antworten als 1 gezählt; GESMC für die Scores des Programms MBP; GESPC für die Scores des Programms PROSAR) der Versuchspersonen 8-mal ein geringeres (=schlechteres) Ergebnis des Programms MBP gegenüber PROSAR ergeben hat und 17-mal die umgekehrte Variante gültig ist. Gleiche Wertungen (Ties) gibt es 5-mal. Dieses Ergebnis wird auf einem Wahrscheinlichkeitsniveau von 0.0094 angenommen da das Signifikanzniveau zuvor auf 5% gesetzt wurde. MBP wird also besser gelernt.

Zur Reliabilität dieses Fragebogens wurden wieder Auswertungen zu Cronbach's Alpha (Kuder-Richardson Formel 20) für den gesamten Fragebogen vorgenommen, die auch hier Werte von 0.8 annehmen.

Es konnte somit gezeigt werden, daß beide Fragebogen valide, reliabel und objektiv (durch Konstruktion) sind. Die Auswertung der Fragebogen ergibt eine bessere Bewertung sowohl der Benutzerfreundlichkeit als auch der Erlernbarkeit des Programms MBP. Diese Ergebnisse decken sich erstaunlich gut mit denen der formalen Analyse, wie z.B. die Metrik über die Regelschemata in Bezug auf die Erlernbarkeit, die Benutzerobjekte und die Metrik über die Kontexte. Weiters konnten auch einige Richtlinien zur Gestaltung von Benutzerschnittstellen und besonders deren Anwendung verifiziert werden.

5. Zusammenfassung

Aus den Konstellationen des Projekts ergab sich die Möglichkeit einer ausgedehnten theoretischen und empirischen Untersuchung zweier unterschiedlicher, klar abgegrenzter Benutzeroberflächen. Das Besondere an dieser Untersuchung stellt die im Vergleich zu anderen Untersuchungen hohe Komplexität der untersuchten Programme dar. Dadurch wird die Relevanz der Ergebnisse im Hinblick auf Anwendungen in der Praxis stark erhöht.

Durch die hohe Übereinstimmung der Ergebnisse der formalen Analyse und der Ergebnisse der empirischen Untersuchung lassen sich positive Schlüsse auf die Adäquatheit der verwendeten Methoden ziehen. Dies wird durch die völlige Unabhängigkeit der beiden Untersuchungen voneinander möglich.

Schließlich konnten durch die Analysen einige Schwachstellen in den Benutzeroberflächen der untersuchten Programme festgestellt werden. Diese Schwachstellen werden zur Zeit bei der Erstellung einer neuen Version des Programmes MBP berücksichtigt. Die hier verwendete ATAG (Advanced Task Action Grammar) stellt also ein praktisch taugliches Mittel zur Bewertung und Verbesserung von Benutzeroberflächen dar.

6. Literatur

- J.T. Boekholt, A.P. Thijssen, P.J.M. Dinjens, N.J. Habraken (1974): Denken in varianten - het methodisch ontwerpen van dragers; Samson Uitgeverij.
- F. Chaslin et al. (1976): Dossier SAR; in: techniques & ARCHITECTURE, no. 311, Editions Regirex-France.
- DIN 66 234 (1986): Teil 8: Bildschirmarbeitsplätze – Grundsätze der Dialoggestaltung; Beuth Verlag, Berlin.
- Haider M., Kundi M., Kolm P., Moser U. (1987): Richtlinien zur Gestaltung von Bildschirmarbeit; Verlag ÖGB, Wien.
- Krüger H., Müller-Limmroth W. (1979): Arbeiten am Bildschirm – aber richtig! Informationsschrift des bayrischen Staatsministeriums für Arbeit und Sozialordnung, München.
- Moran T. P. (1981): The Command Language Grammar: A Representation for the User Interface of Interactive Computer Systems; in: International Journal of Man-Machine-Studies, Vol 15, pp. 3-50.
- Moran T. P. (1984): Getting into a System: External-Internal Task Mapping Analysis; in: Human Factors in Computing Systems, Proceedings CHI 83, North-Holland, pp. 45-49.
- Norman D. A., Draper S. W. (1986): User Centered System Design; Lawrence Erlbaum, pp. 31-63.
- Payne S. J., Green T. R. (1986): Task Action Grammar: A Model of Mental Representation of Task Languages; in: Human Computer Interaction Vol. 2, Lawrence Erlbaum, pp. 93-133.
- Schubö W., Uehlinger H.-M. (1986): SPSSx Version 2.2; Fischer Verlag, Stuttgart.
- Shneiderman B. (1987): Designing the User Interface: Strategies of Effective Human-Computer Interaction; Addison-Wesley.
- Smith S.L., Mosier J.N. (1984): Design Guidelines for User – System Interface Software; MITRE Corporation, Electronic Systems Division, AFSC, Hanscomb.
- Spinas P., Troy N., Ulich E. (1983): Leitfaden zur Einführung und Gestaltung von Bildschirm-Systemen; Verlag Industrielle Organisation, Zürich.

Paul Tavalato

Architekturbüro Ottokar Uhl

Maysedergasse 2
A-1010 Wien

Karl-Heinz Weidmann

Institut für Praktische Informatik 180B
Technische Universität Wien
Argentinierstraße 8
A-1040 Wien