

Prototyping mit "Hyper-Tools"

Der Einsatz von Intermedia und Supercard als Prototypinginstrumente
für ein komplexes Bank-Informationssystem

Josef Bösze und Doris Aschacher, Zürich

Partizipative SW-Entwicklung bedingt unter anderem rasches Prototyping, damit der Dialog mit dem Benutzer möglichst rasch anhand von konkreten Fragestellungen an einem konkreten System in Gang kommt. Dies ist umso wichtiger, je innovativer das zu entwickelnde System sein soll. In diesem Erfahrungsbericht wird aufgezeigt, wie die Produkte Intermedia und Supercard während verschiedener Phasen im Prototyping Zyklus optimal eingesetzt werden können. Es wird gezeigt, dass die Verwendung von mehreren verschiedenen Prototyping Werkzeugen und die damit verbundenen Werkzeugwechsel und Redesigns durchaus sinnvoll und nützlich sein können.

1. Kurzbeschreibung des Projektes KAP-92

Im Rahmen eines Forschungsprojektes wird zur Zeit ein Informationssystem entwickelt, das alle Mitarbeiter - Sekretärinnen, Sachbearbeiter/innen und Direktionsmitglieder - einer Abteilung der Schweizerischen Bankgesellschaft in ihrer Arbeit unterstützen soll. Aufgabe der zukünftigen Systembenutzer ist es, alle Geschäftsbeziehungen, die gewisse Konzerne und ihre Tochterfirmen mit in- und ausländischen Geschäftsstellen unserer Bank unterhalten, zu koordinieren und zu kontrollieren. Sowohl die Menge der pro Kunde zu bearbeitenden Informationen wie auch deren komplexe Vernetzung stellen besondere Ansprüche an das zu entwickelnde System. Ziel des Projektes ist es, bis Ende 1992 eine erste produktiv eingesetzte Version des **Konzernbetreuer Arbeits-Platzes** (KAP-92) zu entwickeln, welcher die Benutzer in den Bereichen Bürotätigkeit, Management der Gesamtgeschäftsbeziehung und operative Abwicklung von einzelnen Geschäftsfällen unterstützt.

Der KAP-92 orientiert sich (als Metapher) an einem erweiterten Dokumentbegriff. Dabei wird ein Dokument nicht nur als eine statische Darstellung von Information aufgefasst. Vielmehr werden den Dokumenten zwei weitergehende Funktionalitäten zugeordnet: zum einen eine dynamische Verarbeitungsleistung, die den Zugriff auf Daten (aus Datenbanken und anderen Dokumenten) und deren Prozessierung im Dokument selber erlaubt; zum anderen ein Referenzierungsmechanismus (Links), der die sinnvolle Vernetzung der Dokumente untereinander und so das schnelle Aufsuchen von zusammenhängenden Informationen ermöglicht. Das Dokument wird zum Träger von sowohl strukturierter als auch unstrukturierter Information.

2. Überblick über den Entwicklungsprozess

Ziel dieses Artikels ist es, die von uns gesammelten Erfahrungen im Einsatz von "Hyper-Tools" für das Prototyping im Projekt KAP-92 darzustellen. Damit die Stellung der verschiedenen von uns entwickelten Prototypen verständlich wird, werden an dieser Stelle einige wichtige Punkte unserer Vorgehensweise kurz beleuchtet:

- Die Beteiligung der Benutzer am Entwicklungsprozess ist für uns äusserst wichtig und selbstverständlich. Dies widerspiegelt sich auch in der Zusammensetzung des Projektteams: 4 Informatiker und 2.5 Benutzer. Die anderen zukünftigen Benutzer werden regelmässig informiert und spezielle Fragestellungen werden intensiv diskutiert.
- Das Projekt wird nicht stur gemäss dem klassischen Wasserfallmodell (Vorstudie - Analyse - Detailspezifikation - Implementation - Test - Einführung) entwickelt. Nach einer ersten relativ grobkörnigen Analyse und ersten Prototypen wurden verschiedene Systemkomponenten identifiziert, die nacheinander - und teilweise auch gleichzeitig - in iterativen Entwicklungszyklen auf der Basis von Prototyping realisiert werden.
- Die einzelnen Komponenten wie auch das Gesamtsystem werden in Tests durch die Benutzer bezüglich Bedienbarkeit und Funktionalität evaluiert. Resultate dieser Tests fliessen in die weitere Entwicklung ein.
- Strukturierte Analyse- und Spezifikationsmethoden (Datenflussanalyse, semantische Datenmodellierung) werden für die Detailspezifikationen ebenfalls angewandt.

3. Prototyping im KAP-92 Projekt

Die Prototypingphilosophie und die von uns entwickelten Prototypen lassen sich am besten mit den Begriffen "horizontaler Prototyp", "vertikaler Prototyp" und "Szenario" beschreiben (Nielsen 89). Bild 1 illustriert diese Begriffe.

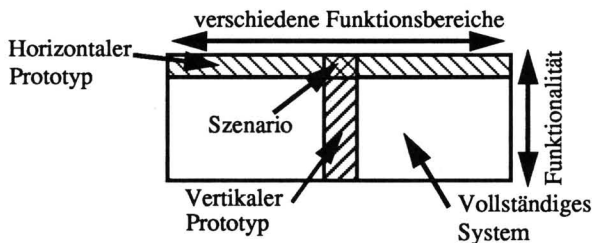


Bild 1: Die zwei Dimensionen des Prototyping. Nach Nielsen (89).

Ein **horizontaler Prototyp** deckt zwar alle Funktionsbereiche des Gesamtsystems ab, es fehlt ihm aber weitgehend eigentliche Funktionalität, d.h. die Fähigkeit, Daten zu prozessieren. Ein **vertikaler Prototyp** weist in einem Teil des Gesamtsystems sehr viel eigentliche Funktionalität auf, er kann unter Umständen auch als einsetzbares Subsystem betrachtet werden. Ein **Szenario** stellt weder die ganze Breite des Funktionenspektrums noch die Tiefe der Funktionalität dar, eignet sich aber durchaus, um bestimmte Einzelaspekte eines Systems zu modellieren und mit den Benutzern zu diskutieren.

Im Rahmen des KAP-92 Projektes wurden und werden in verschiedenen Prototypingphasen mehrere unterschiedliche Prototypen entwickelt. Sie können wie folgt beschrieben werden:

- Beschreibungen und Skizzen auf Papier: Einzelne Szenarien wurden auf Papier skizziert, um eine erste Diskussionsbasis zu haben. Es wurden weder alle Funktionsbereiche abgedeckt noch wurde die gesamte Funktionalität angedeutet. Erstelldauer: Wenige Stunden.
- Mit Intermedia erstellter Prototyp: Ein horizontaler Prototyp, der alle Funktionsbereiche abdeckt. Es konnte - und sollte - keinerlei Funktionalität implementiert werden. Neue Formen der Informationsdarstellung wurden getestet und Elemente des Dialogs angedeutet. Erstelldauer: 2 Wochen.
- Mit Supercard erstellter Prototyp: Ein vorwiegend horizontaler Prototyp, der aber gewisse Szenarien mittels einer Pseudofunktionalität präziser modelliert. Handlungsabfolgen können realistisch simuliert werden, Dialogelemente können optimiert werden. Erstelldauer: 8 Wochen.
- Mit ET++ und Ingres Windows 4GL erstellte Prototypen: Mit diesen Werkzeugen wurden und werden vertikale Prototypen entwickelt. Diese Prototypen verwenden Daten aus einer Datenbank und implementieren echte Funktionalität. Erstelldauer: Je nach Grösse des Prototyps mehrere Wochen bis Monate.

In jeder dieser vier Phasen werden spezifische Eigenschaften des zukünftigen Systems ausgestaltet. Leider ist uns bisher kein Werkzeug bekannt, das eine sowohl gleich gute als auch gleich schnelle Modellierung aller Aspekte erlaubt.

Bei den Prototypen, welche mit Intermedia und Supercard realisiert wurden, handelte es sich um reine Wegwerfprototypen. Die in C++/ET++, bzw. Ingres Windows 4GL implementierten Prototypen werden teilweise vom Prototyp zum Produkt weiterentwickelt und ins definitive System integriert. Das ET++ System wird hier nicht näher beschrieben, für eine Referenz siehe (Weinand, Gamma und Marty, 1988 und 1989). Für eine Beschreibung von Ingres Windows 4GL sei auf die entsprechende Dokumentation der Firma Ingres verwiesen.

In den folgenden Kapiteln sollen nun die Erfahrungen dargestellt werden, die wir im Einsatz der Hyper-Werkzeuge Intermedia und Supercard gesammelt haben. Wegen der starken Verbreitung von Hypercard präsentieren wir auch einen Vergleich zwischen Supercard und Hypercard. Alle 3 Hyper-Tools werden nur unter den für das Prototyping relevanten Aspekten beschrieben.

4. Prototyping mit Intermedia

Intermedia wurde vom Institute for Research in Information and Scholarship (IRIS) der Brown Universität, Providence Rhode Island, entwickelt. Intermedia ist ein Hypertext System, welches neben Textdokumenten auch Graphik- und andere spezialisierte Dokumenttypen unterstützt. Zwischen den Informationsblöcken in den Dokumenten können beliebige **bidirektionale Verweise (Links)** definiert werden. Durch die Aktivierung eines Linkmarkers mit der Maus wird das entsprechende Dokument geöffnet und auf dem Bildschirm angezeigt. Intermedia läuft auf Apple Macintosh unter A/UX. Eine detaillierte Beschreibung von Intermedia findet sich in (Brown University 1988; Yankelowich et al 1988). Bild 2 zeigt einen Ausschnitt aus dem KAP-92 Intermedia Prototyp.

The screenshot displays the KAP-92 Intermedia Prototyp interface, a hypertext system running on Apple Macintosh under A/UX. The interface is composed of several overlapping windows and a central timeline of events.

Top Menu Bar: File Edit Intermedia Font Arrange Print

Background Windows:

- Geschäftsbeziehungen mit SBG:** Contains a logo for 'S B G' (Union de Banque Suisse, Schweizerische Bankgesellschaft, Unione di Banche Svizzere, Union Bank of Switzerland) and the text 'Wir kauften fuer Sie am 3.5.90 in Zuerich'.
- Beurteilungen:** Contains the text 'KaufNestle 3.5.90'.
- Bezugspersonen:** Contains a small table with columns 'R' and 'GL'.

Central Form (KaufNestle 3.5.90):

110-0350	
10	999999
Saurer Nutzfahrzeuge A z.H. Hr. Suter Abt. Finanzen Schaffhausenstrasse 134 8152 Glattbrugg	
Aktien IRGENDWAS Fr. 100.- nom.	Valoren-Nr. 135799
40 a 540.-	in Ihr S
Courtage	2.1.800
Stempel/Gebuehr	138
Inkl. Boersengebuehr	14
	2.1.749
	Valuta
Titel in Depot 99999999	

Timeline (Zeitreihe):

- 1977 01.03.77 : Eröffnen Kontokorrent.....
- 03.04.77 : Kauf 10 Aktien Nestle.....
- 06.07.77 : Aufnahme Darlehen von.....
- 06.06.77 : Uebernahme UNKONWN AG....
- 1978 12.01.78 : Aufnahme Hypothek.....
- 12.04.78 : Emission Obligation zu.....
- 14.06.78 : Kauf Obligation Akusuisse.....
- 1979 31.07.79 : Wechsel Management:
Neu Finanzchef
- 1990 03.06.90 : Kauf 40 Aktien IRGENDWAS
- 23.06.90 : Vergabe neue Rahmenlimite blanko in H
Streichen Betriebslimite blanko in Hoeh
Erhoehung Investitionslimite blanko um
97'684

Bottom Navigation Bar:

- Kreditprotokoll
- Kreditvertrag
- ER
- Spartenerfolgsrechnung
- Kapitalf...
- Kennzahlen
- abgewickelte Aktivitäten

Bild 2: Ausschnitt aus dem KAP-92 Intermedia Prototyp. Es sind drei Dokumente sichtbar, rechts eine Darstellung von Ereignissen als Zeitreihe, links ein Formular und im Hintergrund ein Übersichtsdocument. Die kleinen Rechtecke mit den Pfeilen sind die standardisierten Intermedia Linkmarker

Intermedia zeichnet sich aus der Sicht des Prototyping durch folgende Eigenschaften aus:

- Jedes Dokument wird in einem eigenen Fenster angezeigt, mehrere Fenster können gleichzeitig sichtbar sein, die Grösse der Fenster ist beliebig wählbar.
- Durch die Aktivierung von Links werden Bildwechsel (=Dokumentwechsel) erzeugt. So können typische Handlungsabfolgen simuliert werden.
- Die Definition von Links ist so einfach wie "Cut und Paste". Von und zu einem Ankerpunkt (Informationsblock) können mehrere Links definiert werden. Alle Links sind bidirektional.
- Die Text- und Graphikdokumente können in der typischen Macintosh Philosophie bearbeitet und erstellt werden.

Da mit diesem Werkzeug keine Programmierung möglich ist, ist man gezwungen, sich auf die Gesamtzusammenhänge und reinen Darstellungsaspekte zu konzentrieren. Uns ging es in erster Linie darum, einen Rahmen abzustecken und so eine geeignete Ausgangslage für Diskussionen mit den späteren Benutzern zu schaffen, ohne sich dabei in diesem ersten Schritt bereits in Details zu verlieren. Zumal so auch keine Programmierfehler auftreten konnten, entfiel die Fehlersuche, die ja meist sehr zeitaufwendig ist. Dies und die einfache Definition von Links trugen dazu bei, dass mit diesem Instrument ausserordentlich schnell, mit einem Einarbeitungsaufwand von einem Tag, erste Resultate erzielt werden konnten.

In unserem Fall verwendeten wir Intermedia hauptsächlich für die Modellierung des Schreibti-schoberflächen-Grundbildes, für die Ausarbeitung von Beispiel-Dokumenten und Masken und insbesondere für das Aufzeigen der Navigationsmöglichkeiten im System (Wechsel zwischen verschiedenen Dokumenten und Masken).

5. Prototyping mit Supercard

Supercard, ein Produkt der Firma Silicon Beach Software, ist eine Weiterentwicklung von Apple's Hypercard. Beide Hypermedia-Pakete bieten dem Benutzer eine relativ einfache Entwicklungsumgebung für Macintosh-Programme unter Verwendung verschiedenster Medien wie Text, Graphik, Animation und Ton an. Applikationen werden in der Scriptsprache Hypertalk, bzw. Supertalk programmiert. Eine nähere Beschreibung findet sich in folgenden Publikationen: Supercard (1989), Gookin (1989), Goodman (1988), Waite et al (1989), Mourant (1989). Hypertalk und Supertalk werden verbreitet für Prototyping eingesetzt. Als ein Beispiel siehe Hofer (1990).

Die für das Prototyping relevanten Unterschiede zwischen Hypercard und Supercard sind in Tabelle 1 zusammengefasst:

Im Projekt KAP-92 wurde Supercard zur Ausarbeitung wichtiger Details der Benutzungsoberfläche und Systemfunktionalität eingesetzt. Durch den geschickten Einsatz der Möglichkeiten von Supercard konnten mit relativ geringem Aufwand komplexe Szenarien realistisch durch den Pro-

WAS	HYPERCARD	SUPERCARD
Karten	Fixe Kartengröße (9 " Diagonale), ausgerichtet auf den klassischen Mac Plus/SE Bildschirm. (wird in Version 2.0 geändert)	Beliebige, variable Kartengröße. Dies erleichtert das Arbeiten auf 19" Zoll Bildschirmen.
Stapel	Ein Stapel besteht aus mehreren Karten. Nur eine Karte kann gleichzeitig auf dem Bildschirm angezeigt werden. (wird in Version 2.0 geändert)	Eine Applikation besteht aus verschiedenen Fenstern. Zu jedem Fenster gehört ein Stapel. In einem Fenster kann gleichzeitig nur eine Karte des Stapels angezeigt werden.
Fenster	Fenstergröße entspricht immer der Kartengröße (9").	Unterschiedliche Fenstertypen werden unterstützt. Es können mehrere Fenster gleichzeitig geöffnet sein. Kartengröße kann verschieden von der Fenstergröße sein.
Farbe	Nicht vorhanden.	Wird gut unterstützt.
Menüs	Keine eigenen Menüs definierbar	Eigene Menüs können erstellt werden (Pull-down, Pop-Up und Floating Palette)
Geschwindigkeit	Auf allen Mac's genügend schnell	Relativ langsam, benötigt für effizientes Arbeiten teurere Maschine (Mac IIfx)

Tabelle 1: Vergleich zwischen Hypercard und Supercard.

totyp simuliert werden. Als besonders geeignet erwies sich Supercard für die Modellierung von Dialogboxen, von Selektionsdialogen und Aktionen, die durch das Drücken von Tasten, Graphiken oder Ikonen ausgelöst werden. Es wurden auch einige echt direktmanipulative Dialogelemente mit "Drag and Drop" Mechanismus implementiert, wobei allerdings die Grenzen von Supercard erreicht wurden. Die Benutzer konnten sich aufgrund von Vorführungen ein Bild machen und den Entwicklern entsprechenden "Feedback" geben, inwieweit Darstellung, Funktionalität und Dialog ihren Bedürfnissen entsprechen. Bild 3 zeigt einen Ausschnitt des mit Supercard entwickelten KAP-92 Prototyps.

Trotz komfortablem Editor, Tracer und Debugger schnellte die Entwicklungszeit im Vergleich zum Intermedia-Prototyp in die Höhe. Die Einarbeitungszeit betrug etwa 2 Wochen statt nur einen Tag. Der Mangel einer klaren und konsequenten Syntax von Hypertalk/Supertalk, die ungenügende Dokumentation diverser Befehle sowie die Schwierigkeit, sich einen Überblick über alle unterstützten, bzw. vorhandenen Möglichkeiten zu verschaffen, erhöhten den Einarbeitungsaufwand und führten zu Fehlern und anschließender Fehlersuche.

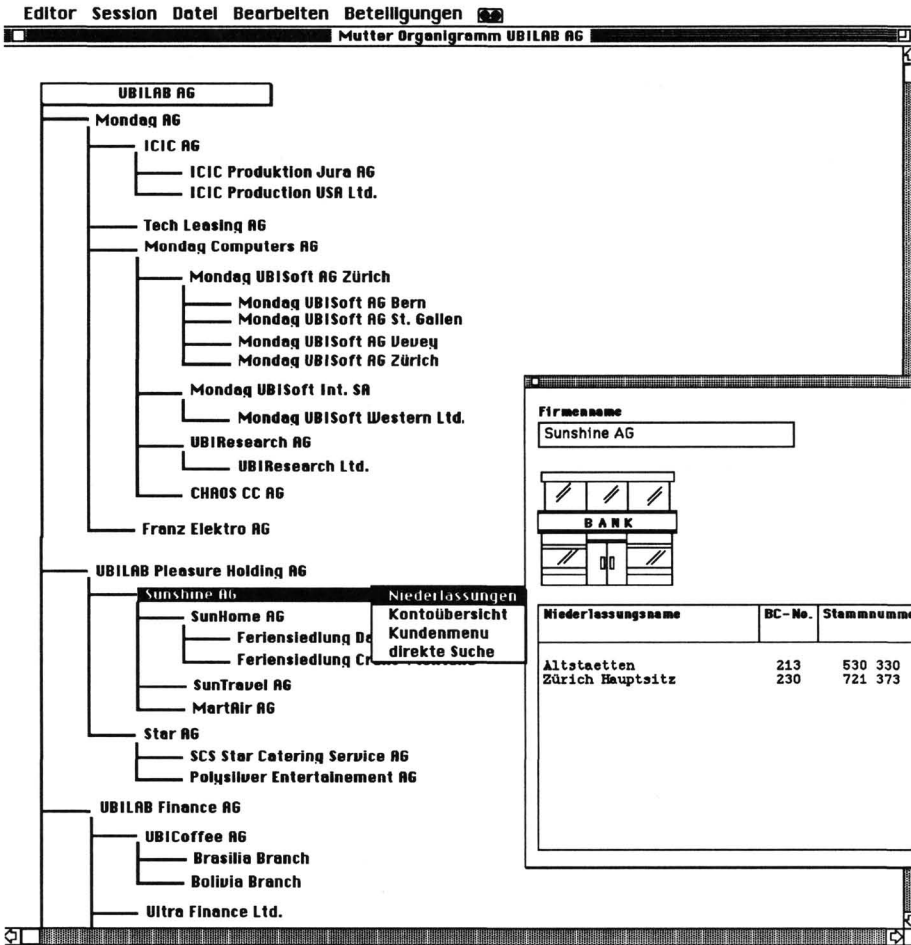


Bild 3: Ausschnitt aus dem KAP-92 Supercard Prototyp. Es sind zwei Dokumente sichtbar. Links ist das Organigramm einer Firmen-
gruppe dargestellt. Durch die Selektion der "Sunshine AG" er-
scheint ein Pop-Up-Menü, aus dem der Menüpunkt
"Niederlassungen" ausgewählt wurde. Dies öffnete das kleinere
Fenster rechts, in dem nun, aufgrund einer simulierten Daten-
bankabfrage, alle Geschäftsstellen der SBG angezeigt werden,
mit denen die "Sunshine AG" Geschäftsbeziehungen pflegt.

Weil die Hypertalk/Supertalk-Programme lediglich interpretiert werden, sind zwar Programm-
änderungen jederzeit möglich (selbst der in Supertalk geschriebene Supertalk-Editor kann abge-
ändert werden) und Compile-Link-Zeitverluste treten nicht auf, aber das Geschwindigkeitsver-

halten des Systems lässt bei komplexeren Programmen zu wünschen übrig. Es muss auf einen effizienten Programmierstil geachtet werden, da sonst selbst auf einer schnellen Maschine (Mac IIx) bereits längere Wartezeiten feststellbar sind. Trotz dieser kleineren Probleme ist Supercard unserer Ansicht nach ein Produkt, welches sich für viele Prototypingarbeiten bestens eignet.

6. Wann soll wofür welches Werkzeug eingesetzt werden?

Nach unseren bisherigen Erfahrungen mit Prototyping teilen wir die Werkzeuge gemäss ihren maximalen Möglichkeiten in 4 Klassen ein:

1. Erstellung von statischen Bildern, Bildschirmmasken und Skizzen: Papierskizzen, allgemeine Zeichnungsprogramme (z.B. MacPaint, MacDraw etc).
2. Erstellung von durch den Benutzer aktivierbaren Bildfolgen: z.B. Intermedia
3. Erstellung von Prototypen mit Pseudofunktionalität, Simulation von Dialogeigenschaften: z.B. Hypercard/Supercard.
4. Erstellung von Prototypen mit echter Funktionalität: 4. Generations Werkzeuge (z.B. Ingres Windows 4 GL), Application Frameworks (z.B. ET++, MacApp)

Grundsätzlich weisen die Werkzeuge der höheren Klassen auch alle Möglichkeiten der niedrigeren Klassen auf. So können beispielsweise Layouts von Bildschirmmasken auch sehr gut mit Hypercard oder Supercard modelliert werden. Im Grunde genommen könnte man auch alle Prototypingarbeiten mit den Werkzeugen der 4. Klasse durchführen, wenn nicht folgende drei Umstände zu beachten wären:

1. Mit zunehmendem Funktionsumfang der Prototypingwerkzeuge steigt meist auch die Komplexität der Bedienung und der Lernaufwand.
2. Spezialisierte Werkzeuge eignen sich für bestimmte Aufgaben besser als breiter einsetzbare Werkzeuge.
3. Die Verfügbarkeit von mehr Funktionalität in den Werkzeugen der höheren Klassen verführt den Entwickler dazu, diese auch zu gebrauchen.

Besonders der dritte Punkt verdient grösste Beachtung: Je nach Prototypingphase oder Zweck des Prototyps kann ein Mangel an Selbstdisziplin und -beschränkung des Entwicklers dazu führen, dass die Erstellung des Prototyps viel zu lange dauert, beziehungsweise im Prototyp Merkmale implementiert sind, die im entsprechenden Entwicklungsstand gar nicht von Interesse sind. Aus eigener Erfahrung wissen wir, dass Werkzeuge wie Hypercard, Supercard und auch Ingres Windows 4GL sehr verführerisch sind und den Spieltrieb fördern. Der Entwickler findet in seinem Prototyp immer etwas, was besser oder noch raffinierter gelöst werden könnte. Auch besteht die Gefahr, dass schlecht implementierte Prototypen plötzlich unkontrolliert zum eigentlichen System weiterentwickelt werden. Den Preis zahlt man dann meist in Form von hohen Wartungskosten und grosser Unzuverlässigkeit des Systems.

Es hat sich gezeigt, dass es nützlich und notwendig ist, folgende Fragen **vor** Beginn der Entwicklung eines Prototyps zu beantworten:

1. Welchen Zweck hat der Prototyp? Welche Fragen und Probleme soll er lösen? In welcher Form wird der Prototyp den Benutzern zugänglich gemacht?
2. Wie lange darf die Entwicklung maximal dauern?
3. Soll die Möglichkeit bestehen, den Prototyp zum eigentlichen System weiterzuentwickeln?

In der Tabelle 2 ist dargestellt, welche Prototypingwerkzeuge sich unserer Meinung nach für die verschiedenen Arten von Prototyping eignen.

Werkzeug Medium Zweck des Prototyps Prototypingphase	Papierskizzen, Zeichnungswerkzeuge	Intermedia, Hypercard/Supercard ohne Programmierung	Hypercard/Supercard mit Programmierung	4. Generationssprachen (Ingres Windows 4GL) Application Frameworks (ET++, MacApp)
Brainstorming, erste Entwürfe	***	***	*	
Abstecken des Gesamtrahmens, Navigation im System, (Horizontaler Prototyp)	*	***	**	
Darstellung von Information, Layouts, Bildschirmmasken, (Horizontaler Prototyp, Szenarien)	**	***	**	*
Dynamische Dialogaspekte, Pseudofunktionalität, (Szenarien)			***	**
Echte Funktionalität mit echten Daten, Weiterentwicklung zum Produkt möglich (Vertikaler Prototyp)			*	***

Tabelle 2: Zuordnung von geeigneten Werkzeugen zu Prototypingphasen.

Legende: *** = sehr gut geeignet, kein * = ungeeignet.

Werden im Laufe eines Projektes verschiedene Prototypen gemäss dieser Philosophie entwickelt, kommen zwangsläufig mehrere Werkzeuge zum Einsatz. Der Werkzeugwechsel mag zwar auf den ersten Blick lästig sein, weist aber auch einige bedeutende Vorteile auf:

- In jeder Phase kann ein wirklich geeignetes Werkzeug zum Einsatz kommen, ohne die Notwendigkeit, stets unnötigen Ballast mitzuschleppen. Bei Wegwerfprototypen muss weniger auf eine saubere Programmierung und gute Dokumentation geachtet werden.
- Erste Prototypen lassen sich schneller und billiger entwickeln und erlauben es, die Benutzer schon früh in Diskussionen einzubeziehen. Die Kommunikation zwischen Anwendern und Entwicklern wird dadurch positiv beeinflusst.
- Man ist gezwungen, gewisse Aspekte des Systems zweimal zu gestalten. Dies gibt die Möglichkeit, beim zweiten Versuch Benutzerreaktionen einzubeziehen, Verbesserungen anzubringen und neue Erkenntnisse umzusetzen.

7. Fazit

Prototyping ist nur attraktiv, wenn man relativ schnell und mit geringem Aufwand zu ersten Resultaten gelangt. Die Einfachheit der Erlernung eines Werkzeugs ist ebenso wichtig wie die problemlose, effiziente Bedienung nach der Lernphase. Kostet die Erstellung fast soviel (Zeit, Mitarbeiter, Maschinen, Geld) wie die Entwicklung des eigentlichen Systems, ist Prototyping uninteressant.

Die rasche Implementierung einiger weniger Basisfunktionen gibt Gelegenheit, diese durch den Benutzer erproben und verbessern zu lassen. Zumal es vielen Benutzern Schwierigkeiten bereitet, abstrakte Systemspezifikationen zu verstehen und umzusetzen, bietet erst diese Vorgehensweise eine echte Diskussionsgrundlage zwischen Anwender und Entwickler und beseitigt die bestehende Kommunikationskluft.

Es lohnt sich, verschiedene Prototypingphasen, die sich durchaus zeitlich und inhaltlich überlappen dürfen, zu unterscheiden und jeweils die für die entsprechende Aufgabe adäquaten Prototypingwerkzeuge einzusetzen.

Literaturverzeichnis

- Apple Computer Inc., 1987. Human Interface Guidelines: The Apple Desktop Interface. Addison-Wesley, Reading, MA
- Belew R.K., Rentzepis J., 1990. Hyper Mail: Treating Electronic Mail as Literature. Conference on Office Information Systems, ACM Press
- Brown University Institute for Research in Information and Scholarship, 1988. IRIS Intermedia User's Guide Release 3.0. Brown University Institute for Research in Information and Scholarship, Providence, Rhode Island
- Conklin, J., 1987. Hypertext: An Introduction and Survey. IEEE Computer 20.9 : 17-41
- Goodman D., 1988. The Complete HyperCard Handbook. Bantam Books, New York
- Gookin D., 1989. The Complete SuperCard Handbook. COMPUTE! Books, Radnor, Pennsylvania

- Heeg F., Neuser R., 1988. Nutzergerechte Ausgestaltung von Software durch Prototyping - Grundlagen, Vorgehensweise, Wirtschaftlichkeitsaspekte-VDI Verlag GmbH, Düsseldorf
- Hirsch M.C., 1990. Neue Horizonte (Hypercard 2.0). Macwelt 8'90
- Hofer E and Ruggiero F., 1990: Hypermedia as Communication and Prototyping Tools in the Concurrent Design of Commercial Airplane Products. INTERACT 90. D. Diaper et al (Editors). Elsevier Science Publ.
- Meyrowitz N., 1986: Intermedia: The Architecture and Construction of an Object Oriented Hypermedia System and Applications Framework. In OOPSLA 86 Proc. , pp 186-201, ACM Press.
- Mourant R., 1989. Designing Human Interfaces with Hypercard. In Salvendy G. and Smith M.: (eds): Designing and Using Human-Computer Interfaces and Knowledge Based Systems, Elsevier Science Publishers, Amsterdam ,
- Nielsen J., 1989: Usability Engineering at a discount. In Salvendy, G. and Smith, M.: (eds): Designing and Using Human-Computer Interfaces and Knowledge Based Systems, Elsevier Science Publishers, Amsterdam , pp 394-401
- Pomberger G., Bischofberger W., Keller R., Schmidt D., 1987. Prototypingorientierte Softwareentwicklung theoretische und organisatorische Aspekte Teil 1. Institut für Informatik Universität Zürich Irchel, Zürich
- Waite M., Prata S., and Jones T., 1989. The Waite Group's HyperTalk Bible. Hayden Books, Indianapolis, IN
- SuperCard, 1989. SuperCard: The personal software toolkit, User Manual. Silicon Beach Software Inc., San Diego
- Weinand A., Gamma E., Marty R., 1988: ET++ - An Object Oriented Application Framework in C++. OOPSLA 88, Special Issue of SIGPLAN Notices, Vol. 23, No. 11.
- Weinand A., Gamma E., Marty R., 1989: Design and Implementation of ET++, a Seamless Object -Oriented Application Framework; Structured Programming, Vol. 10, No. 2.
- Yankelovich N., Haan B.J., Meyrowitz K., Drucker M., 1988. Intermedia: The Concept and the Construction of a Seamless Information Environment. IEEE Computer, no 1, pp 81- 96.

Josef Bösze, Doris Aschacher
 Schweizerische Bankgesellschaft
 Abt. UBILAB (Union Bank Informatics Laboratory)
 Universitätsstr 84
 Postfach
 CH-8033 Zürich