

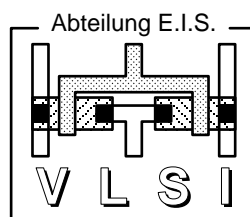
# Multimediales Lernen im Chip- und System-Entwurf

Übungen

Tamer Çatalkaya

14. Januar 2003

Abteilung Entwurf integrierter Schaltungen (E.I.S.)  
Technische Universität Braunschweig  
Prof. Dr. Ulrich Golze



## INHALTSANGABE

<b>1</b>	<b>EINLEITUNG.....</b>	<b>1-3</b>
<b>2</b>	<b>ARBEITSUMGEBUNG.....</b>	<b>2-4</b>
2.1	DIE BENUTZEROBERFLÄCHE .....	2-4
2.2	ICONS .....	2-5
2.2.1	Display-Icon.....	2-6
2.2.2	Motion-Icon.....	2-7
2.2.3	Erase-Icon.....	2-7
2.2.4	Wait-Icon.....	2-8
2.2.5	Navigate-Icon.....	2-8
2.2.6	Framework-Icon.....	2-8
2.2.7	Decision-Icon .....	2-9
2.2.8	Interaction-Icon .....	2-10
2.2.9	Calculation-Icon .....	2-11
2.2.10	Map-Icon .....	2-11
2.2.11	Digital Movie-Icon.....	2-12
2.2.12	Sound-Icon.....	2-12
2.2.13	Video-Icon .....	2-13
2.3	FUNKTIONEN UND VARIABLEN .....	2-13
2.4	START- UND STOPP-FLAGGE .....	2-14
2.5	CONTROL PANEL – DEBUGGING.....	2-14
<b>3</b>	<b>VORBEREITUNGEN.....</b>	<b>3-16</b>
3.1	PROJEKTVERZEICHNIS.....	3-16
3.2	VOREINSTELLUNGEN .....	3-16
<b>4</b>	<b>LEARNING BY DOING – ÜBUNGSAUFGABEN.....</b>	<b>4-18</b>
4.1	ÜBUNG – UEB00 .....	4-18
4.1.1	Lernziele.....	4-18
4.1.2	Aufgabe .....	4-18
4.2	ÜBUNG – UEB01 .....	4-18
4.2.1	Lernziele.....	4-18
4.2.2	Programmieranleitung.....	4-18
4.3	ÜBUNG – UEB02 .....	4-22
4.3.1	Lernziele.....	4-22
4.3.2	Programmieranleitung.....	4-22
4.4	ÜBUNG – UEB03 .....	4-27
4.4.1	Lernziele.....	4-27
4.4.2	Aufgabe .....	4-27
4.4.3	Hinweise.....	4-27
4.5	ÜBUNG – UEB04 .....	4-29
4.5.1	Lernziele.....	4-29
4.5.2	Aufgabe .....	4-29
4.5.3	Hinweise.....	4-29
4.6	ÜBUNG – UEB05 .....	4-31
4.6.1	Lernziele.....	4-31
4.6.2	Aufgabe .....	4-31
4.6.3	Thema: MOS-Transistoren.....	4-31

## 1 Einleitung

Authorware (AW) ist ein Autorensystem zum Erstellen multimedialer Applikationen, die unter anderem Video, Audio, Texte und Grafiken enthalten können.

AW liegt das Icon-Control-Paradigma zugrunde, d.h. durch Zusammenstellen von Icons entlang einer Flusslinie wird die Applikation entwickelt. Mit Hilfe der Icons wird, ähnlich einem Flussliniendiagramm, eine Struktur aufgebaut, die Icons repräsentieren dabei Funktionen und Objekte. Aus dem „AW-Flussliniendiagramm“ kann die Applikation automatisch generiert werden (Packen).

Der Programmierer entnimmt Icons aus der Icon-Palette und platziert diese per drag'n'drop im Flusslinien-Fenster. Die Icon-Palette stellt jeden Icon-Typ beliebig oft zur Verfügung. Auf diese Weise können bis zu 32.000 Icons zu einer Applikation zusammengestellt werden. Das Arrangement der Icons gibt die logische Struktur der Applikation wieder und legt dabei das Verhalten der Applikation fest, d.h. seinen Ablauf. Die Flusslinie stellt die Verbindung zwischen den Icons untereinander dar. Wenn keine Sprünge programmiert sind, werden die Icons entlang der Flusslinie von oben nach unten (bzw. links nach rechts) ausgeführt.

Dieser Leitfaden liefert für den zukünftigen AW-Programmierer das Basiswissen. Zunächst wird die Benutzeroberfläche von AW vorgestellt, anschliessend werden die unterschiedlichen Icon-Typen erklärt. Um den Aufbau von AW-Programmen zu vermitteln, werden Sie nach dem Prinzip *Learning by Doing* anhand von Übungsaufgaben Schritt für Schritt in die Programmierung eingeführt. Jedem Abschnitt sind dabei die Lernziele vorangestellt, in den folgenden Abschnitten wird darauf aufgebaut.

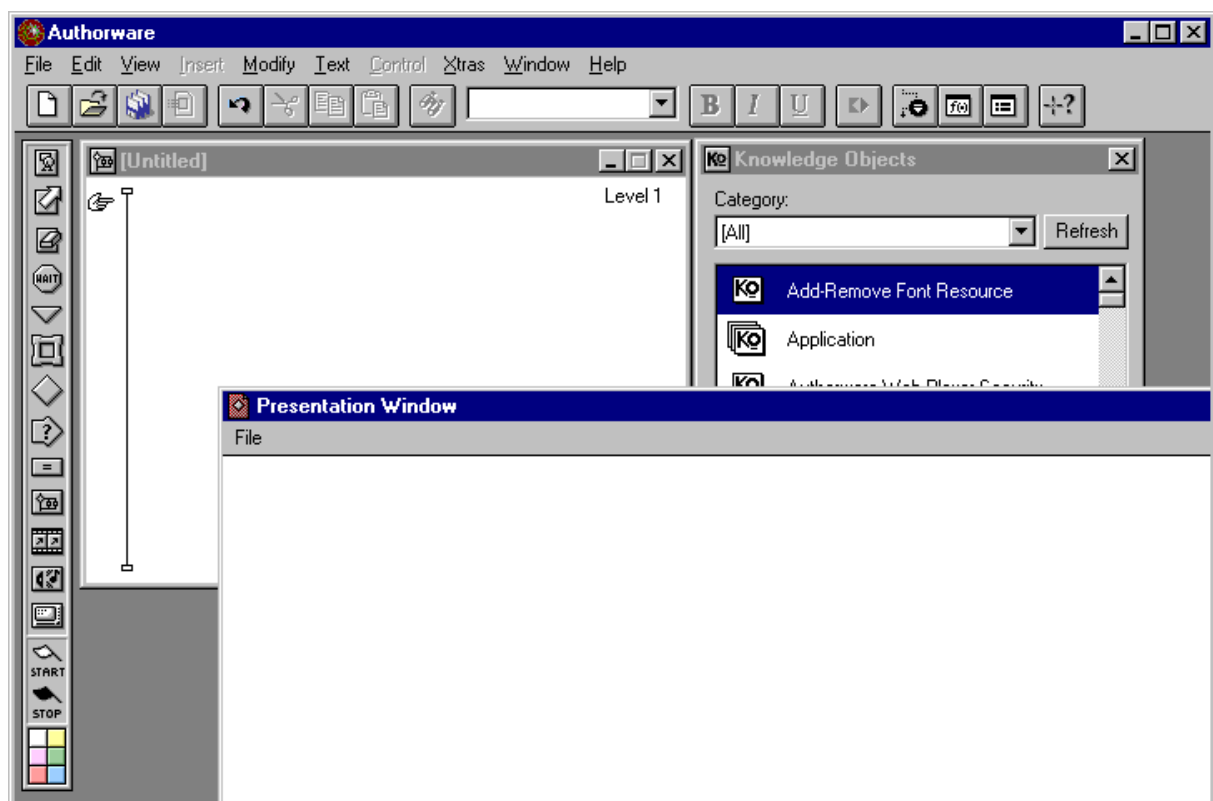
Nach dem Durcharbeiten dieser Einführung sollte man eine Vorstellung über die Arbeit mit AW haben. Wir empfehlen anschliessend im Internet nach AW-Beispielen zu suchen, von denen es sehr viele gibt. Durch Studieren solcher Beispiele lernt man schneller den Umgang mit AW als durch experimentelles Ausprobieren.

## 2 Arbeitsumgebung

Im Folgenden werden die Benutzeroberfläche, Icons sowie Funktionen und Variablen von AW vorgestellt. Dabei wird nur das Notwendigste betrachtet. Für weiterführende Erklärungen wird auf das Handbuch verwiesen. Es kann über das Menü *Help > Authorware Help Pages...* aufgerufen werden (wahlweise auch per F1) und erscheint als HTML-Hilfe mit Suchmöglichkeit in einem Browser-Fenster.

### 2.1 Die Benutzeroberfläche

In Bild 2.1 ist die Benutzeroberfläche von AW 5.2 nach dem Anstarten zu sehen. Unter der Menüleiste ist die Symbolleiste zu erkennen, am linken Rand die Icon-Palette und in der Mitte links das Flusslinien-Fenster mit der noch leeren Flusslinie. Rechts in der Mitte das *Knowledge Objects* Fenster, und unten ein Teil des Präsentationsfensters.



**Bild 2.1** Benutzeroberfläche von Authorware 5.2

Die Icon-Palette enthält 13 Icon-Typen, zwei Start-/ Stop-Flaggen und sechs Farben zum Einfärben von Icons. Die Flusslinie stellt die Beziehung zwischen den Icons dar. Das Präsentationsfenster zeigt die Applikation wie sie später der Benutzer sieht. In dem Präsentationsfenster kann man sich jederzeit den Ablauf – auch nur teilweise oder schrittweise – ansehen. Das *Knowledge Objects* Fenster wird hier nicht weiter betrachtet, nur soviel, es enthält *Knowledge Objects* (KOs), das sind Prozedur

ähnliche Programmteile. Die Anwendung der Knowledge Objekts bleibt den Fortgeschrittenen AW-Nutzern vorbehalten.

Startet man später die unter AW entwickelte Applikation, läuft sie in einem Fenster wie das Präsentationsfenster ab. Mit Hilfe des Präsentationsfenster zur Laufzeit stellt AW eine „What You See Is What You Get“ (WYSIWYG) Anwendung dar. Eigenschaften des Präsentationsfensters (z.B. Hintergrundfarbe, Fenstergrösse, etc.) können im Eigenschaftenfenster (STRG+SHIFT+D) eingestellt werden.

## 2.2 Icons

Icons enthalten Funktionen (z.B. Sprünge, Reaktionen auf Eingaben) und Objekte (z.B. Grafiken, Töne, Filme). Ein Programm wird durch Anordnen von Icons entlang der Flusslinie aufgebaut. Die Icons werden von oben nach unten (bzw. links nach rechts) abgearbeitet, sofern keine Sprünge implementiert sind. Die Flusslinie lässt sich durch sogenannte Map-Icons hierarchisch gliedern und durch Einfärben von beliebigen Icons mit Hilfe der Farben kennzeichnen.

Durch ein Doppelklick öffnet man ein Icon auf der Flusslinie und je nach Typ erscheint sein Inhalt (Objekte) oder das Eigenschaften-Fenster auf dem Bildschirm. Jedes Icon kann einen individuellen Namen erhalten, den man direkt am Icon durch Auswählen mit der Maus eingibt. Zusätzlich numeriert AW intern jedes Icon mit einer Icon-ID.

Ist ein Icon über eine Funktion referenziert, kann dieser Name nicht noch einmal vergeben werden. Wird ein referenziertes Icon per copy'n'past dupliziert, so hängt AW automatisch eine Zahl an diesen Namen, die er selber verwaltet. Bei wiederholtem copy'n'past wird die Zahl automatisch hoch gezählt.

Werden referenzierte Icons gelöscht, entsteht ein Link-Bruch. Dabei wird die Referenzierung umbenannt in „Unlinked“ oder „DeletedIcon“. Mit der Suchfunktion kann nach diesen Link-Brüchen gesucht werden.

Icons können per drag'n'drop verschoben werden. AW verwaltet die Verbindungen intern automatisch. Durch zu häufiges Verschieben, Löschen und Hinzufügen von Icons kann die Verbindungsliste anwachsen. Durch ein *Save and Compact...* im Menü *File*, führt AW eine *garbage collection* durch, dabei wird das Programm unter einem neuen Namen auf die Platte geschrieben (ähnlich einem *Save as...*). Ein *Save and Compact...* sollte man regelmässig durchführen, ansonsten können Probleme nicht ausgeschlossen werden.

Tool-Tips zeigen jeden Icon-Namen in der Palette, wenn mit der Maus auf dem Icon verweilt wird.

### 2.2.1 Display-Icon

Um Texte und Grafiken darzustellen, verwendet man das Display-Icon. Entweder werden Texte und Grafiken aus einer externen Quelle importiert oder mit Hilfe der Toolbox vom Designer eingegeben. Die Toolbox öffnet sich, wenn man ein Display-Icon doppelklickt.

Ein Doppelklick auf einem Display-Icon zeigt die Toolbox (im Bild 2.2 oben rechts im Fenster zusehen). Die Toolbox stellt Funktionen zum Zeichnen und Schreiben zur Verfügung. Schliesst man die Toolbox, wird das Display-Icon geschlossen. Die Überschrift im Rahmen der Toolbox zeigt den Namen des geöffneten Display-Icons an.

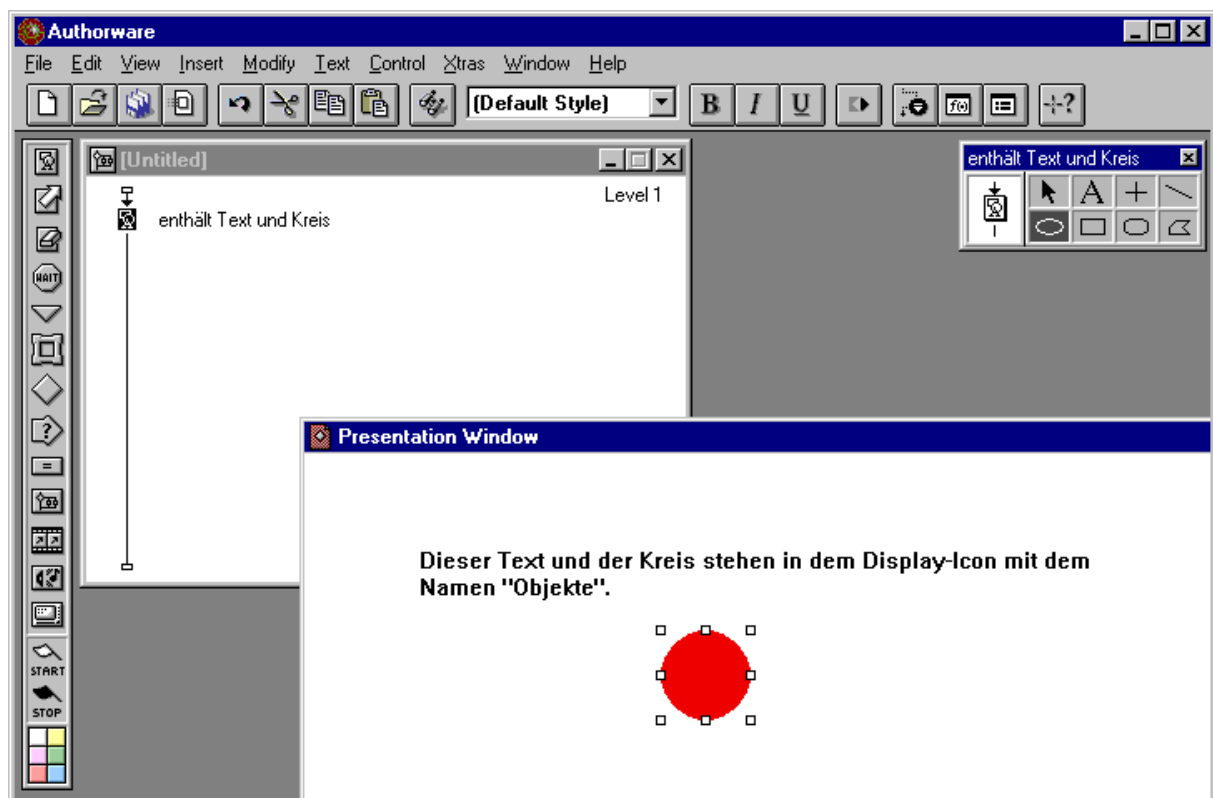
Bei geöffnetem Display-Icon kann man mittels Inspektoren Eigenschaften von Objekten wie Farbe, Linienstärke, Füllung etc. verändern. Die Tastaturkürzel für die Inspektoren lauten:

Linien-Inspektor (STRG-L)

Füllung-Inspektor (STRG-D)

Zustands-Inspektor (STRG-M)

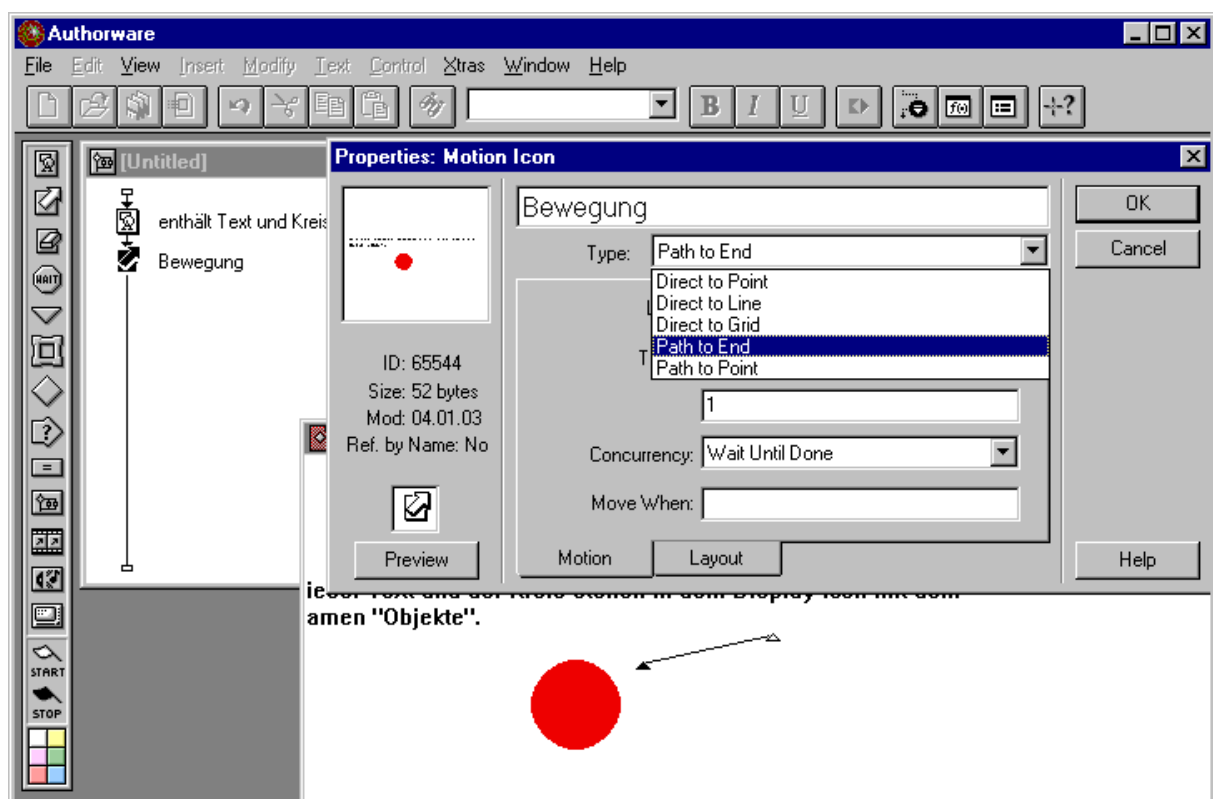
Farb-Inspektor (STRG-K)



**Bild 2.2** Ein Display-Icon mit zwei Objekten

### 2.2.2 Motion-Icon

Um Objekte auf dem Bildschirm zu bewegen, wird das Motion-Icon verwendet. Objekte können von A nach B direkt, über mehrere Zwischenpunkte oder entlang einem Pfad bewegt werden. Die Bewegung kann über die Zeit oder mit einer konstanten Geschwindigkeit erfolgen. Die Art der Bewegung und das zu bewegende Objekt wählt man in einer entsprechenden Dialogbox aus. Wählt man ein Objekt, das sich mit weiteren Objekten in einem Display-Icon befindet, zum Bewegen aus, so bewegen sich alle Objekte des Display-Icons. Um ein einzelnes Objekt zu bewegen, muss es separiert allein in einem Display-Icon enthalten sein. Das zu bewegende Objekt muss im Ablauf zunächst im Präsentationsfenster angezeigt werden bevor es bewegt werden kann, d.h. das Display-Icon ist vor dem Motion-Icon zu platzieren.



**Bild 2.3** Das Eigenschaftsfenster des Motion-Icon

Über die Option Type wird die Art der Bewegung definiert. In Bild 2.3 ist ein *Path to End* gewählt. Das zu bewegende Objekt wird durch Anklicken im Präsentationsfenster selektiert, dann per drag'n'drop des Objekts der Bewegungspfad definiert.

### 2.2.3 Erase-Icon

Zum Löschen von Objekten auf dem Bildschirm wird das Erase-Icon verwendet. Das Löschen bezieht sich auf alle Objekte eines Display-Icons, um ein einzelnes Objekt zu

löschen, muss es separiert in einem Display-Icon stehen. Nach dem Platzieren des Erase-Icons wird sein Eigenschaftenfenster durch ein Doppelklick geöffnet. Die zu löschenden Objekte werden im Präsentationsfenster durch Anklicken selektiert.

Wird im Ablauf ein Erase-Icon ausgeführt, werden die gewählten Objekte, sofern sie auf dem Bildschirm sichtbar sind, entfernt, andernfalls passiert nichts.

### 2.2.4 Wait-Icon

Um den Ablauf entlang der Flusslinie zu pausieren, wird das Wait-Icon verwendet. Der Ablauf kann eine bestimmte Zeit pausiert werden oder solange bis eine Taste gedrückt wird.

### 2.2.5 Navigate-Icon

Um zu einem Icon innerhalb eines Framework-Icons zu springen, verwendet man das Navigate-Icon. Man unterscheidet zwischen mehreren Arten von Sprüngen:

- Jump to Page: Ein einfacher Sprung.
- Call and Return: Sprung mit Rückkehr. AW speichert den Startpunkt der Rundreise und kehrt hierher zurück, sobald der Befehl *Return* aufgerufen wird.
- Weitere seltener benutzte Sprünge offeriert die Dialog-Box.

### 2.2.6 Framework-Icon

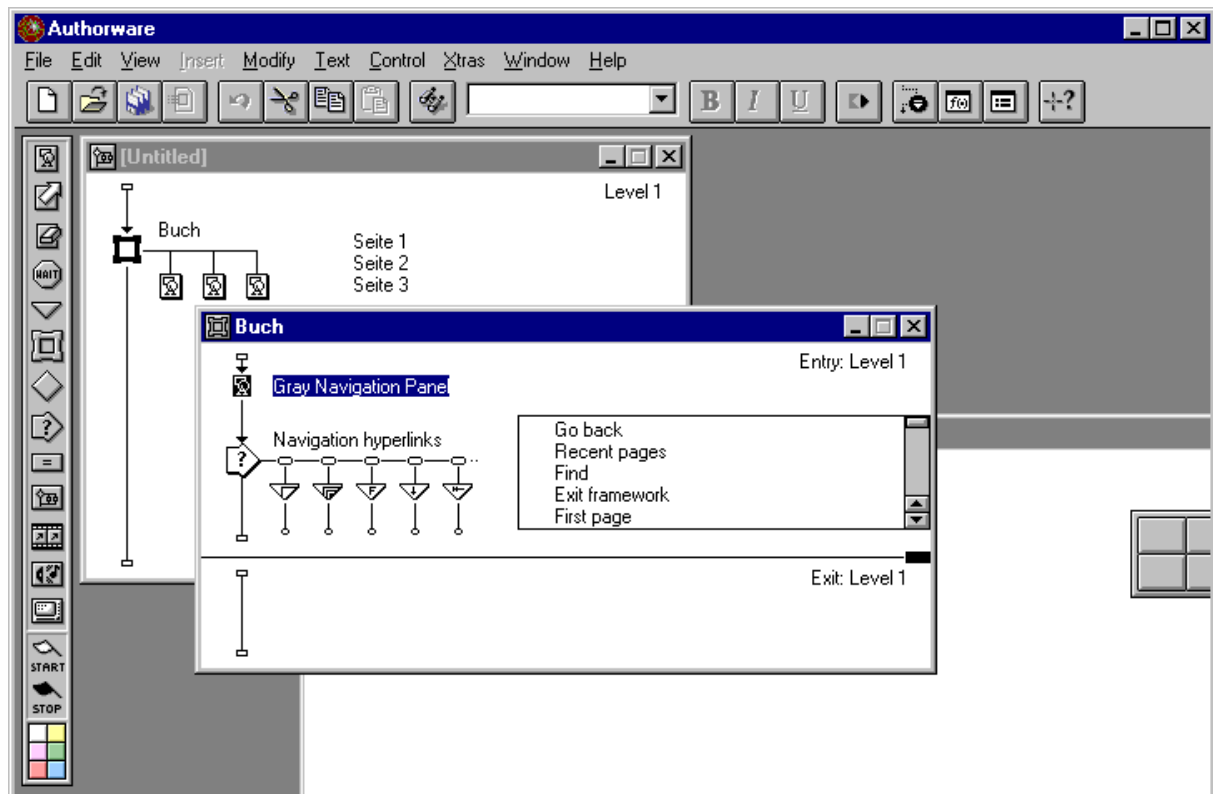
Die Icons, die an einem Framework-Icon hängen, heissen Seiten oder Pages. Das Framework-Icon verwaltet diese Seiten. Sprünge zwischen den Seiten erfolgen durch eine vorgegebene Navigation, die bereits in dem Framework-Icon implementiert ist. Diese kann übernommen, angepasst oder gelöscht werden.

Die durch ein Framework-Icon repräsentierte Struktur ähnelt dem eines Buches. Das Framework-Icon stellt den Umschlag dar und bietet die Navigationsfunktionen an. An das Framework-Icon können beliebig viele Icons angehängt werden, die jeweils einer Seite eines Buches entsprechen (vgl. Bild 2.4). Anders als beim klassischen Buch, wo lediglich Text und Bilder auf einer Seite enthalten sind, können hier multimediale Objekte eingesetzt werden. Im Framework-Icon selbst können Abläufe definiert werden, die zu Beginn (beim Öffnen des Buches) und am Ende (beim Schliessen des Buches) ausgeführt werden (vgl. Bild 2.4). Mit Navigate-Icons kann von irgendwo zu einer Seite gesprungen werden.

In Bild 2.4 ist das Framework-Icon mit dem Namen *Buch* durch ein Doppelklick geöffnet. Das zugehörige Fenster, in dem sich die Navigationsfunktionen befinden, ist zweigeteilt in die Bereiche *Entry* und *Exit*.



Wird das Framework-Icon ausgeführt, werden zu Beginn zunächst die Icons in der *Entry* ausgeführt und anschließend die erste Seite (sofern kein Sprung zu einer anderen Seite in der *Entry* definiert wurde). Beim Verlassen des Framework-Icon werden dann die Icons in *Exit* ausgeführt. In Bild 2.4 ist *Exit* leer. Das Framework-Icon wird über ein Navigate-Icon verlassen. In Bild 2.4 ist in *Entry* unter den Navigationsfunktionen z.B. das Navigate-Icon *Exit framework* definiert. Im Präsentationsfenster erscheint diese Funktion wie die übrigen als Button. Alle Objekte in *Entry* erscheinen auf allen Seiten, sofern sie nicht auf einer Seite gelöscht werden.



**Bild 2.4** Im Inneren des Framework-Icon befindet sich die Navigationsfunktionen

Das Platzieren der Navigationsfunktionen in *Entry* ist sinnvoll, somit stehen sie auf allen Seiten zum Navigieren zur Verfügung. Redundant könnten sie auch auf allen Seiten platziert werden, was unsinnig ist.

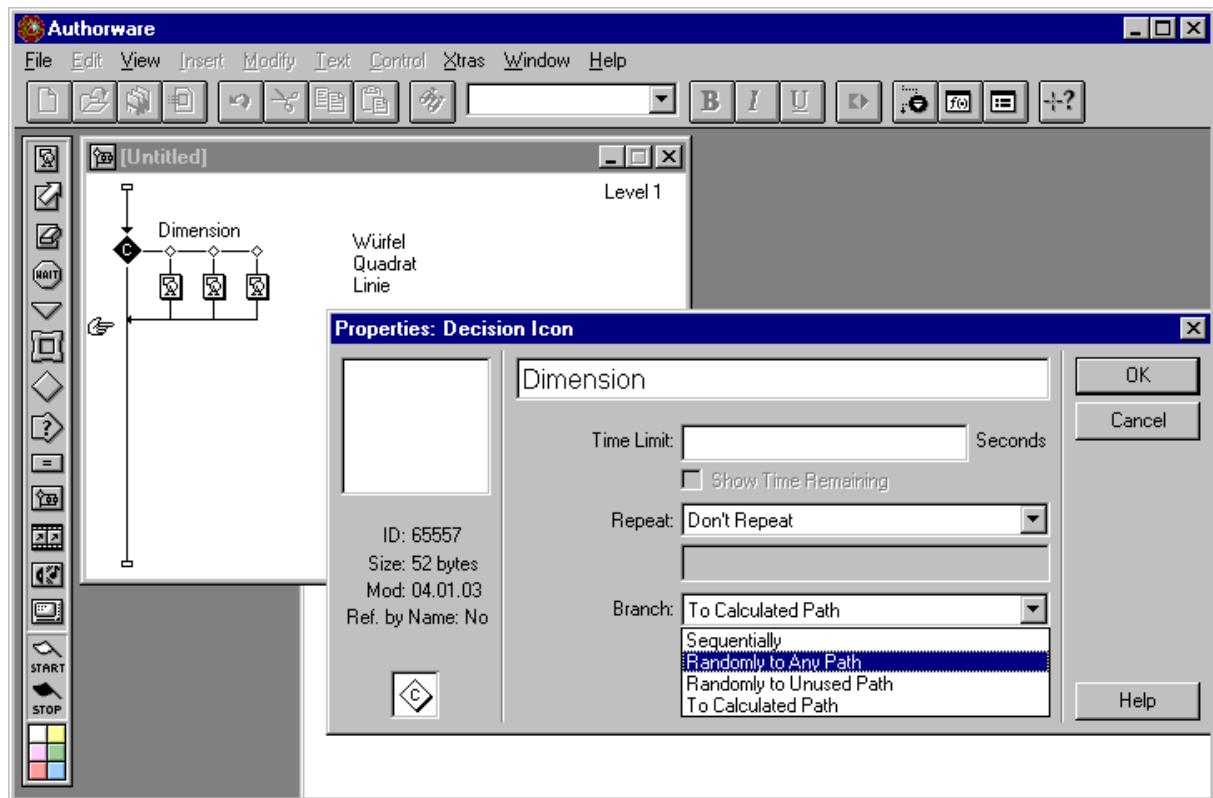
### 2.2.7 Decision-Icon

An das Decision-Icon kann man mehrere Icons anhängen. In dem Decision-Icon definiert man eine Bedingung, nach der eines der angehängten Icons ausgeführt wird. AW merkt sich die ausgeführten Icons sogar über mehrere Durchläufe. So können die angehängten Icons nacheinander, bei jedem Durchlauf, ausgewählt werden (*Sequentially*) oder zufällig eines (*Randomly to Any Path*) oder zufällig eines der noch

nicht benutzen (*Randomly to Unused Path*) oder ein individuell berechnetes (*To Calculated Path*) (vgl. Bild 2.5).

Wahlweise kann ein Decision-Icon in einer Schleife wiederholt ausgeführt werden, z.B. so oft bis alle angehängten Icons einmal ausgeführt wurden.

Mit Hilfe des Decision-Icons kann man z.B. eine Umordnung von Fragen implementieren, die dann bei jedem Durchlauf in unterschiedlicher Reihenfolge dem Benutzer präsentiert werden.



**Bild 2.5** Das Decision-Icon führt bedingt angehängte Icons aus

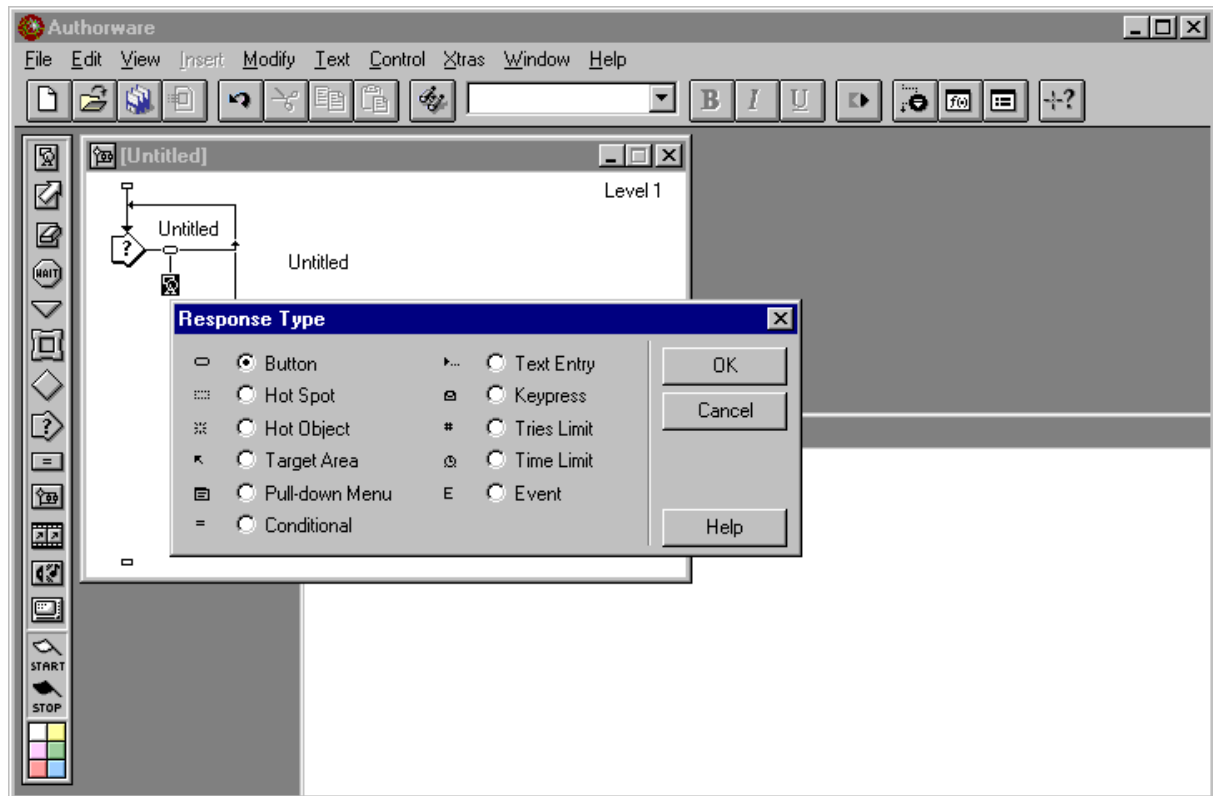
## 2.2.8 Interaction-Icon

Wann immer eine Interaktion mit dem Benutzer programmiert werden soll, kommt das Interaction-Icon zum Einsatz. An das Interaction-Icon kann man, ähnlich wie an das Decision-Icon, auch weitere Icons anhängen, dabei öffnet sich das *Response Type* Fenster (Bild 2.6).

Im *Response Type* Fenster wählt man den Interaktions-Typ aus, z.B. *Button*. Im Präsentationsfenster erscheint dann der Interaktions-Typ, in diesem Fall ein Button. Klickt der Benutzer auf den Button, wird das zugehörige am Interaktions-Icon hängende Icon ausgeführt.

Durch ein Doppelklick auf den Interaktions-Typen (als Knoten über dem angehängten Icon) kann dieser nachträglich, wenn bereits das *Response Type* Fenster geschlossen wurde, verändert werden.

Ein Doppelklick auf das Interaction-Icon öffnet, ähnlich wie beim Display-Icon, die Toolbox. Alle hier platzierten Objekte erscheinen dann zu allen Interaktionen.



**Bild 2.6** Beim Anhängen von Icons an das Interaction-Icon öffnet sich das *Response Type* Fenster

### 2.2.9 Calculation-Icon

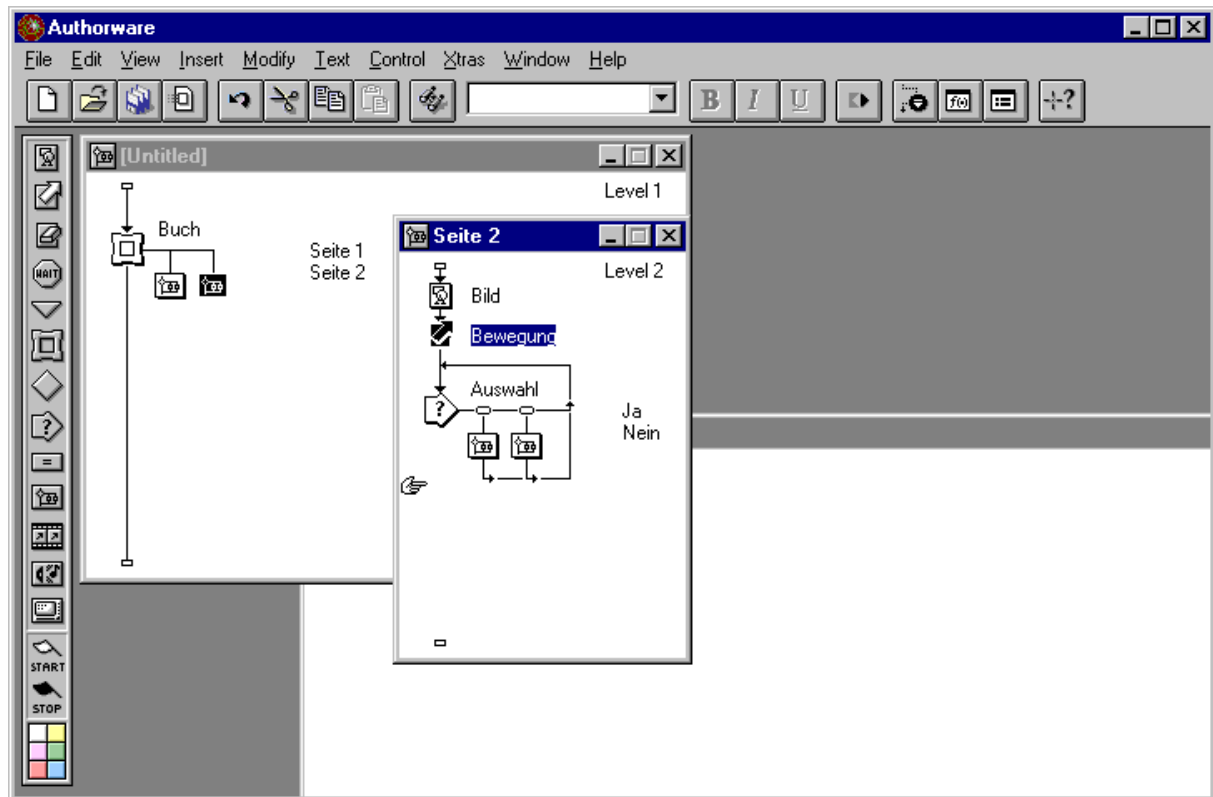
Innerhalb des Calculation-Icons können Funktionen und Variablen von AW verwendet werden (siehe 2.3). Wird z.B. ein Calculation-Icon mit der Funktion *Quit()* ausgeführt, so wird das Präsentationsfenster geschlossen und die Applikation beendet.

### 2.2.10 Map-Icon

Mit einem Map-Icon lassen sich beliebig viele Icons in der Flusslinie zusammenfassen. Eine solche Gruppierung erlaubt eine übersichtliche Strukturierung über mehrere Ebenen (Level).

Bild 2.7 zeigt eine einfache Programmstruktur. Ein Framework-Icon mit zwei Seiten ist auf *Level 1* aufgebaut. Eine Seite wird in der Regel aus mehreren Icons aufgebaut, die hier durch ein Map-Icon zusammengefasst sind. Das Map-Icon mit dem Namen *Seite 2* ist durch ein Doppelklick geöffnet und sein Inhalt wird in einem separaten Fenster auf *Level 2* dargestellt.

Auch die Interaktionen *Ja* und *Nein* am Interaction-Icon *Auswahl* sind durch Map-Icons gekapselt. Ein Doppelklick auf eines dieser Map-Icon würde uns auf *Level 3* führen. Auf diese Weise lassen sich beliebige Schachtelungstiefen erreichen.



**Bild 2.7** Mit dem Map-Icon können Hierarchien aufgebaut werden

### 2.2.11 Digital Movie-Icon

Will man als multimediale Komponenten digitale Filme wie AVIs oder QuickTime einbinden, so erfolgt dies über das Digital Movie-Icon. Über das Eigenschaftensfenster kann der digitale Film in die Applikation eingelesen und in Grenzen manipuliert werden.

### 2.2.12 Sound-Icon

Sprachausgaben und Töne werden über das Sound-Icon in die Applikation eingebunden. Über das Eigenschaftensfenster kann die Sound-Datei in die Applikation eingelesen und in Grenzen manipuliert werden. Sounds sind mit digitalen Filmen die platzfressendsten Komponenten. Anstelle von üblichen WAV-Dateien sollte man das SWA-Format (ShockWave Audio-Format) wählen. Das SWA-Format ist ein Abkömmling des MP3-Formats und entsprechend platzsparend.

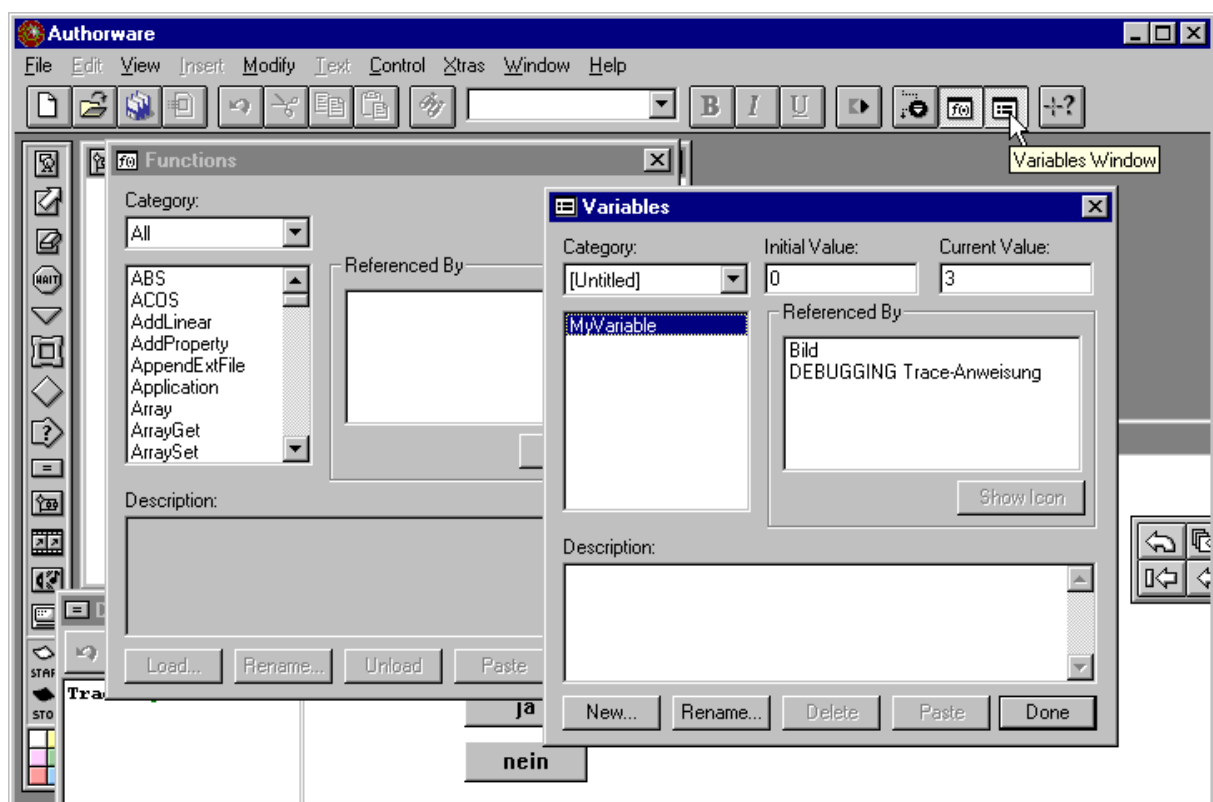
WAV-Dateien können mit einer Applikation, die aus AW über das Menü *Xtras > Other > Convert WAV to SWA...* aufgerufen wird, konvertiert werden.

### 2.2.13 Video-Icon

Zum Abspielen von analogen Videos. Dieses Icon wird höchst selten verwendet. Weitere Informationen entnehmen Sie bitte dem Handbuch.

## 2.3 Funktionen und Variablen

Neben den Icons können auch Funktionen und Variablen zum Programmieren verwendet werden. Alle Funktionen und Variablen sind in Kategorien eingeteilt und können über jeweils ein Fenster verwaltet werden. Über STRG+Shift+F und STRG+Shift+V wird das jeweilige Fenster aufgerufen (Bild 2.8).



**Bild 2.8** Funktionen- und Variablen-Fenster

AW wertet eine Vielzahl von Ereignissen im Programmlauf aus, diese werden teilweise in Variablen abgelegt. Per Zugriff auf diese Variablen kann z.B. der weitere Programmverlauf gesteuert werden.

Das Erklären aller Funktionen und Variablen würde den Rahmen dieser Einführung sprengen. Wir empfehlen dem fortgeschrittenen Anfänger, bei Gelegenheit alle Funktionen und Variablen mal zu überfliegen. Nähere Informationen finden sich in dem jeweiligen Fenster unter *Description*. Wahlweise kann auch in der Hilfe nachgeschlagen werden. Wählen Sie nach Aufrufen der Hilfe oben rechts im Browser-Fenster die Option *MAIN MENU*, rechts erscheint eine Themenauswahl. Wählen Sie hier *System Functions* oder *System Variables* nach Bedarf.

Die meisten Funktionen und Variablen werden in der Regel in einem Calculation-Icon eingesetzt. Unter den Funktionen sind auch Konstrukte wie eine repeat while-Schleife oder eine if then-Anweisung enthalten. Damit lassen sich Abläufe auch per Programmcode aufbauen, im Gegensatz zu Abläufen mit Icons. Für Anfänger empfehlen wir den Einsatz von Icons, der Fortgeschrittene wird sicher Funktionen und Variablen verwenden wollen.

Ein Begriff, der in einem Calculation-Icon auftaucht und keine Funktion oder Variable ist, wird als individuell deklarierte Variable erkannt. Diese Variablen verwaltet AW zentral ebenfalls über das Variablen-Fenster unter der Kategorie des Applikationsnamen (Bild 2.8).

## 2.4 Start- und Stopp-Flagge

Mit den auf der Icon-Palette vorhandenen Flaggen können Anfangs- und Endpunkte im Programmlauf auf der Flusslinie abgesteckt werden. Der Ablauf beginnt bzw. endet dann an den entsprechenden Stellen.

Wird die Position einer Flagge durch Schliessen von Fenstern mal vergessen, kann durch Anklicken des leeren Platzes in der Icon-Palette die Flagge zurückgeholt werden.

## 2.5 Control Panel – Debugging

Über das Control Panel kann der Ablauf überwacht, kontrolliert und auch debugged werden. Die Ausführung von jedem Icon wird in dem Control Panel Fenster protokolliert (Bild 2.9). Mit den Funktionsknöpfen kann die Applikation jederzeit angehalten und fortgesetzt werden.

In der Regel betrachtet man einen Abschnitt, den man programmiert hat, im Präsentationsfenster. An einer Stelle, wo man eingreifen will, hält man den Ablauf z.B. über das Control Panel an, macht die gewünschten Änderungen an den Objekten im Präsentationsfenster und setzt den Ablauf fort. Wahlweise kann man auch über die Tastenkombination STRG+P den Ablauf stoppen und durch erneutes Betätigen fortsetzen.

Ruft man bei angehaltenem Programm das Variablen-Fenster auf (STRG+SHIFT+V) und wählt eine Variable so wird ihr aktueller Wert angezeigt. Wahlweise kann man auch in den Programmfluss Calculation-Icon mit der Trace-Funktion einsetzen. In Bild 2.9 ist z.B. das Calculation-Icon *DEBUGGING Trace-Anweisung* eingefügt. Der Inhalt des Calculation-Icons und die Syntax der Trace-Anweisung werden unten links im Bild gezeigt. Der Wert der individuell deklarierten Variable *MyVariable* wird in der Trace-Anweisung in dem protokollierten

Ablauf des Control Panels ausgegeben, in diesem Fall nach dem Display-Icon *Bild*, somit mit der Ausgabe --3.

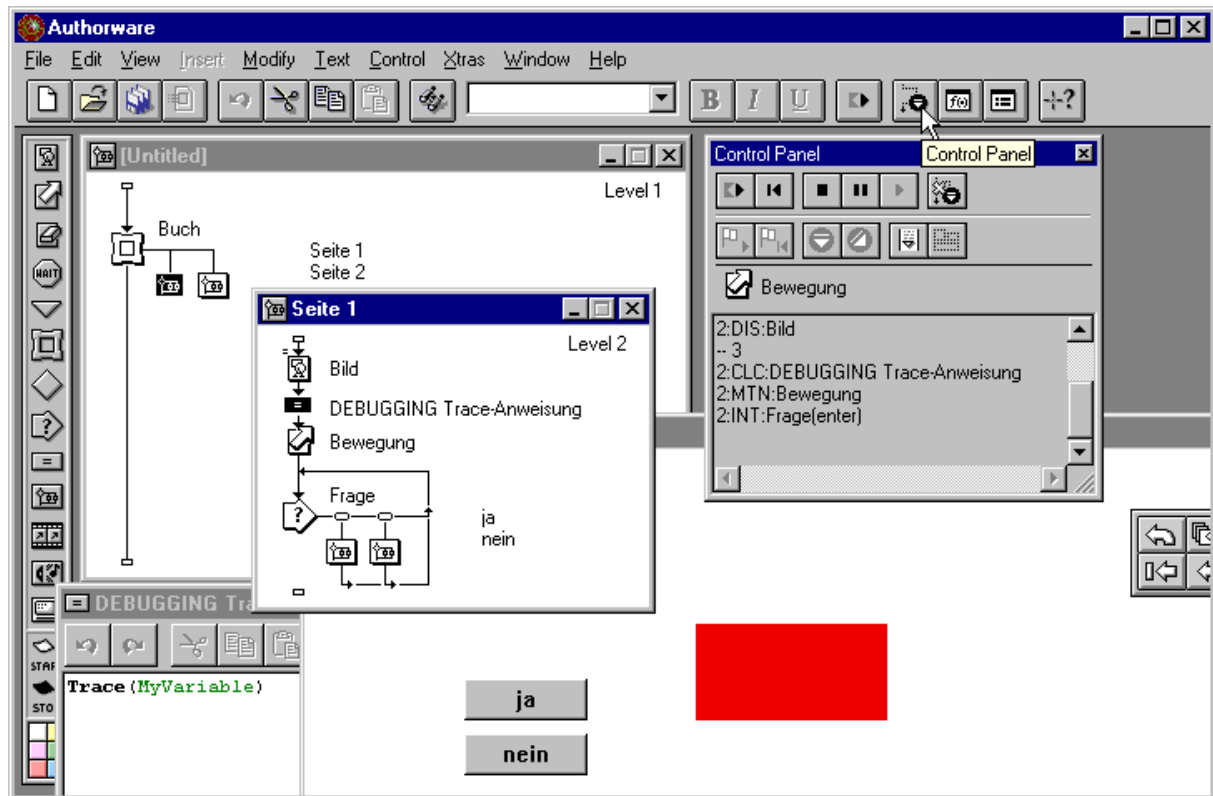


Bild 2.9 Debugging mit dem Control Panel

## 3 Vorbereitungen

Vor Beginn der Implementierung sollten gewisse Vorkehrungen und Einstellungen vorgenommen werden.

### 3.1 Projektverzeichnis

Erstellen Sie entsprechend Ihrer Gruppennummer ein Projektverzeichnis, in dem Sie alle Dateien zu der jeweiligen Applikation ablegen:

zum Beispiel: C:\USER\GRUPPE*ij*\UEBUNG*xy*

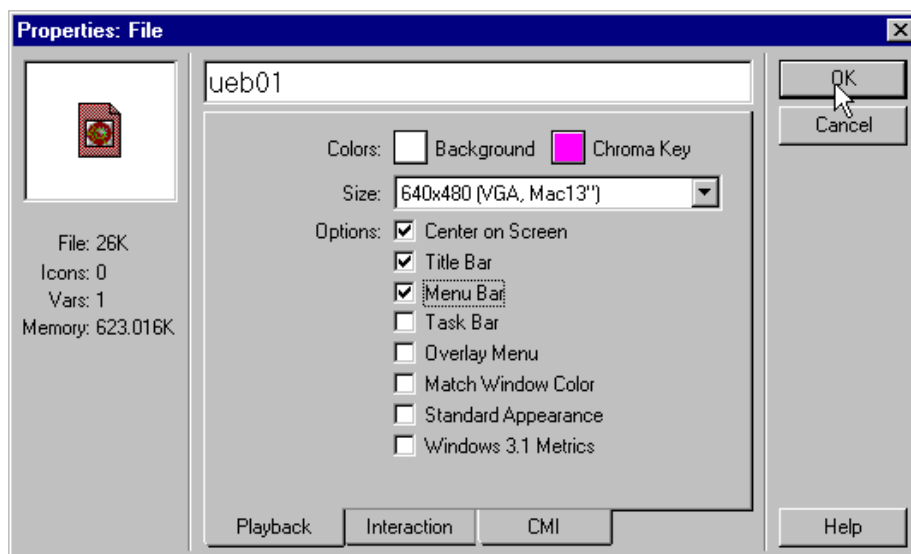
Dabei steht *ij* für die jeweilige Gruppennummer und *xy* für die jeweilige Übungsaufgabe.

Für jede Übungsaufgabe erstellen Sie ein neues Unterverzeichnis. Im übrigen werden Ihnen zu den Übungsaufgaben einzelne Dateien zur Verfügung gestellt, die Sie bitte dann in Ihr jeweiliges Projektverzeichnis übernehmen.

### 3.2 Voreinstellungen

Jede Applikation muss vor der Implementierung analysiert, geplant und entsprechend der Zielgruppe gestaltet werden. Vor der Implementierung einer Applikation müssen Voreinstellungen, bestimmt durch die Planung, vorgenommen werden. Diese Voreinstellungen werden in einem Eigenschaftensfenster (Bild 3.1) vorgenommen und behalten ihre Gültigkeit dann für die Dauer der Applikation.

- Öffnen Sie dazu aus dem Menü *Modify > File > Properties...*



**Bild 3.1** Das Properties Fenster



Vor Beginn jeder Implementierung sollten diese Einstellungen überprüft und ggf. angepasst werden. Die Entwicklung baut auf diesen Einstellungen auf, nachträgliche Änderungen können zusätzlichen Mehraufwand bedeuten.

- In das erste Feld kommt der Applikationsname (z.B. *ueb01*).
- In dem Popup-Menü **Size** wird die Grösse des Präsentationsfensters festgelegt (für uns immer *640x480 (VGA, MAC13“)*).
- Unter der Überschrift Options wählen Sie: *Center on Screen*
- Der Rest bleibt wie er ist. Schliessen Sie das Fenster mit OK.
- Sichern Sie über *File > Save* die gemachten Einstellungen.

## 4 *Learning by doing* – Übungsaufgaben

In diesem Abschnitt werden mehrere Übungsaufgaben vorgegeben, die dann von Ihnen nachprogrammiert werden sollen.

### 4.1 Übung – ueb00

#### 4.1.1 Lernziele

- Kennenlernen der Oberfläche von Authorware.
- Was sind Xtras?

#### 4.1.2 Aufgabe

Spielen sie mit der Authorware-Oberfläche und schauen Sie sich alle Menüpunkte an. Ziehen Sie jedes Icon einmal auf die Flusslinie und erstellen Sie ein beliebiges Flussliniendiagramm und beobachten Sie, wie Authoware die Flusslinie selbständig anpasst.

Rufen Sie die AW-Hilfe mit *F1* auf. Suchen Sie nach *Xtras*. Lesen Sie das Dokument *Determining which files you need to distribute* und schmökern Sie auch in den anderen Dokumenten.

### 4.2 Übung – ueb01

#### 4.2.1 Lernziele

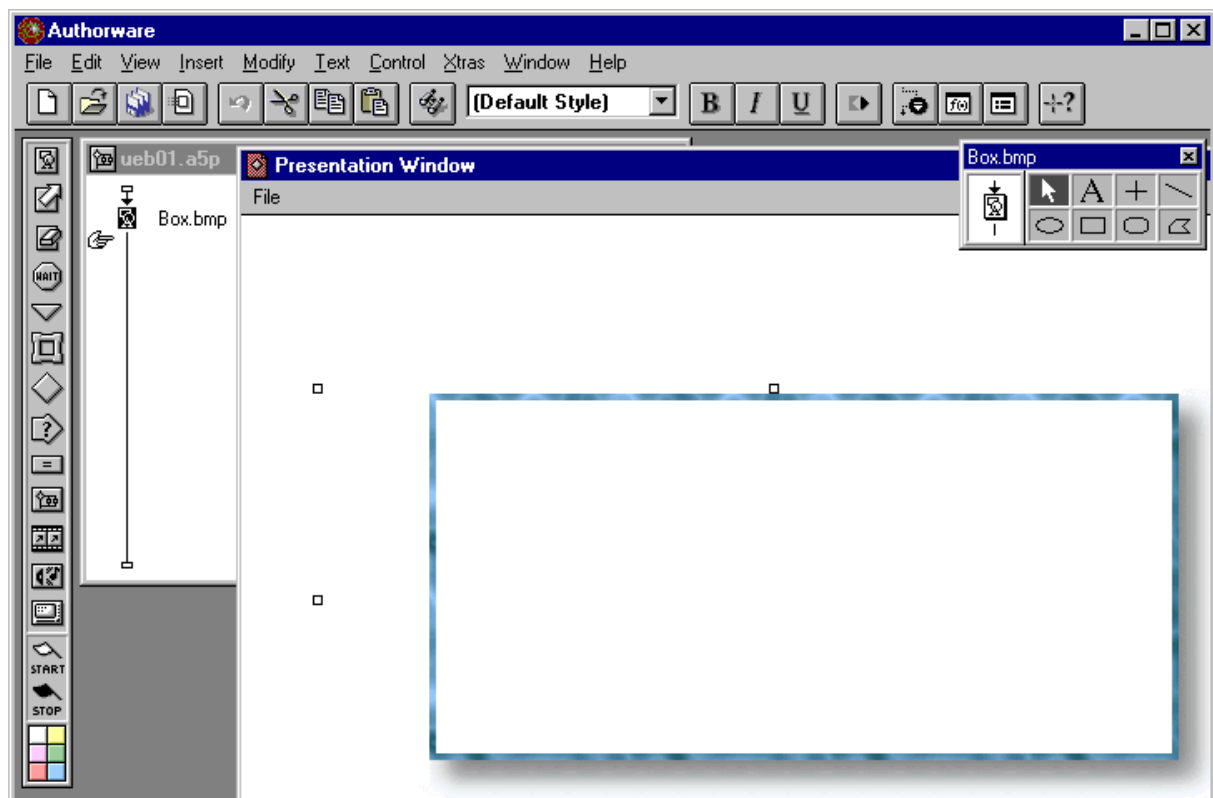
- Platzieren von Icons auf der Flusslinie.
- Importieren von Grafiken.
- Erstellen von Texten, Ändern von Font, Farbe und Grösse.
- Zuordnen einer Transition zu einem Display-Icon.

#### 4.2.2 Programmieranleitung

- Richten Sie ein neues Projektverzeichnis ein und speichern Sie alles unter *ueb01* ab.

AW stellt eine kleine Werkzeugpalette zur Verfügung, mit der primitive Zeichnungen erstellt werden können. In der Regel werden Zeichnungen, Grafiken und Bilder in Grafikprogrammen erstellt und in AW importiert.

- Platzieren Sie ein Display-Icon auf der Flusslinie und nennen Sie es *Box*.
- Per Doppelklick öffnen Sie das Display-Icon. Das Präsentationsfenster und die Toolbox erscheinen.
- Wählen Sie *File > Import* und aus dem Projektverzeichnis die Datei *box.bmp* (Bild 4.1).
- Sichern Sie die Applikation über *File > Save*. Es kann wichtig sein, die erstellten Teile immer wieder in Zwischenschritten zu sichern. Bei einem unerwarteten Absturz erspart man sich viel Ärger und Arbeit.
- Platzieren Sie ein weiteres Display-Icon nach dem vorhandenen und benennen es *Text 1*.
- Wählen Sie nun aus dem Menü *Windows > Panel > Control Panel*.
- Starten Sie die Applikation über den *Restart-Knopf*.

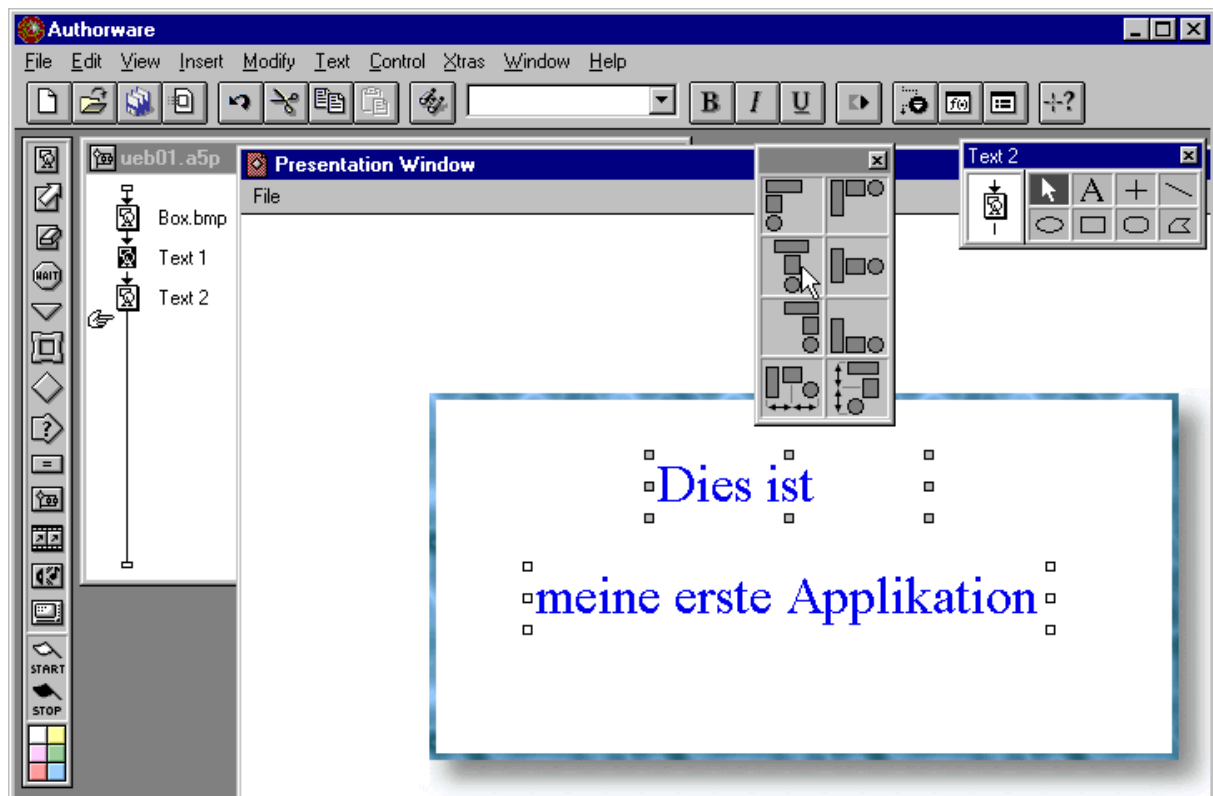


**Bild 4.1** Die importierte Grafik *box.bmp*.

Zunächst wird in dem Präsentationsfenster der Inhalt vom Display-Icon *Box* dargestellt, anschliessend hält der Ablauf an. AW öffnet das zweite Display-Icon *Text 1* und wartet auf die Eingabe von Inhalten. Immer wenn AW ein Icon ausführt,

das noch keinen Inhalt (bzw. Einstellungen) hat, wird die Ausführung gestoppt und das Icon geöffnet.

- Aus der Toolbox wählen Sie die Texteingabe. Platzieren Sie mit der Maus den Textanfang im Präsentationsfenster in dem Rahmen (box.bmp).
- Wählen Sie über *Text > Font > Other...* den Font *Times New Roman* aus.
- Wählen Sie über *Text > Font > 32* die Schriftgröße.
- Tippen Sie: *Dies ist*
- Wählen Sie aus der Toolbox den Zeiger, der Text wird mit Ziehpunkten dargestellt.
- Wählen Sie *Windows > Inspectors > Colors* und aus dem Farb-Inspektor eine blaue Farbe als Textfarbe.
- Wählen Sie *Windows > Inspectors > Modes* und Transparent als Eigenschaft.
- Sichern Sie die Applikation über *File > Save*.
- Wiederholen Sie die Texteingabe mit einem dritten Display-Icon *Text 2*. Geben Sie als Text *meine erste Applikation* ein.
- Selektieren Sie nun den ersten Text durch Doppelklicken und zusätzlich den zweiten Text durch Festhalten von STRG+SHIFT und Anklicken.
- Wählen Sie über *Modify > Align...* und ordnen Sie die selektierten Objekte mittig an (Bild 4.2).



**Bild 4.2** Mittig positionieren von Objekten

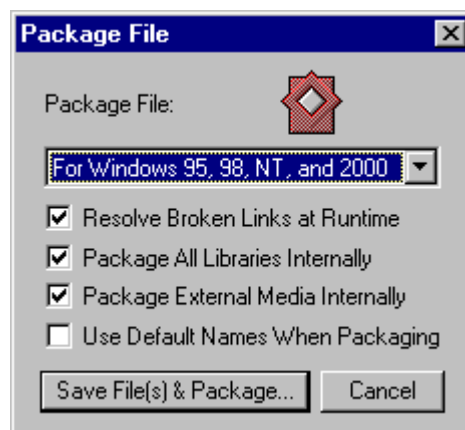
- Halten Sie die STRG-Taste gedrückt und führen ein Doppelklick auf die erste Textzeile. Es öffnet sich das Eigenschaftsfenster.
- Drücken Sie den Knopf *Transition*, wählen Sie *Iris Out* als Transition.

Das Transitionsfenster zeigt zwei Listen (Categories und Transitions). Hier sollten Sie für alle Ihre Projekte nur Transitionen aus der Kategorie *[internal]* wählen. Andere Transitionen können Probleme verursachen und sollten gemieden werden.

- Weisen Sie dieselbe Transition auch der zweiten Zeile zu.
- Sichern Sie die Applikation über *File > Save*.
- Starten Sie die Applikation.

Ihre erste Applikation ist fertig.

- Wählen Sie *File > Package*.
- In dem *Package File*-Fenster wählen Sie die Optionen aus Bild 4.3.
- Achten Sie auf die Einstellung im Drop-Down-Fenster:  
*For Windows 95,98,NT, and 2000* .



**Bild 4.3** Packen (EXE-Erstellen) einer Applikation.

- Starten Sie das Packen mit *Save File(s) & Package*.

AW erstellt nun die entsprechende EXE-Datei. Die EXE-Datei und das Verzeichnis \Xtras müssen zusammen im selben Verzeichnis liegen. Das Xtras-Verzeichnis enthält die benötigten Xtras-Bibliotheken. Welche Xtras Ihre Applikation benötigt, hängt von den verwendeten Medien, Transitionen und Efekten ab. Info hierzu gibt es über die AW-Hilfe im Dokument *Determining which files you need to distribute*.

## 4.3 Übung – ueb02

### 4.3.1 Lernziele

- Umgang mit der Toolbox
- Buttons
- Hierarchien
- Löschen von Icons
- Interaction-Icon

### 4.3.2 Programmieranleitung

- Richten Sie ein neues Projektverzeichnis ein und speichern alles unter *ueb02* ab.
- Platzieren Sie ein Display-Icon und nennen es *Schaltung*.
- Zeichnen Sie in dem Display-Icon die Schaltung wie in Bild 4.4. Verwenden Sie im Menü die Funktionen *View > Grid* und *Snap To Grid* für orthogonale Linien.

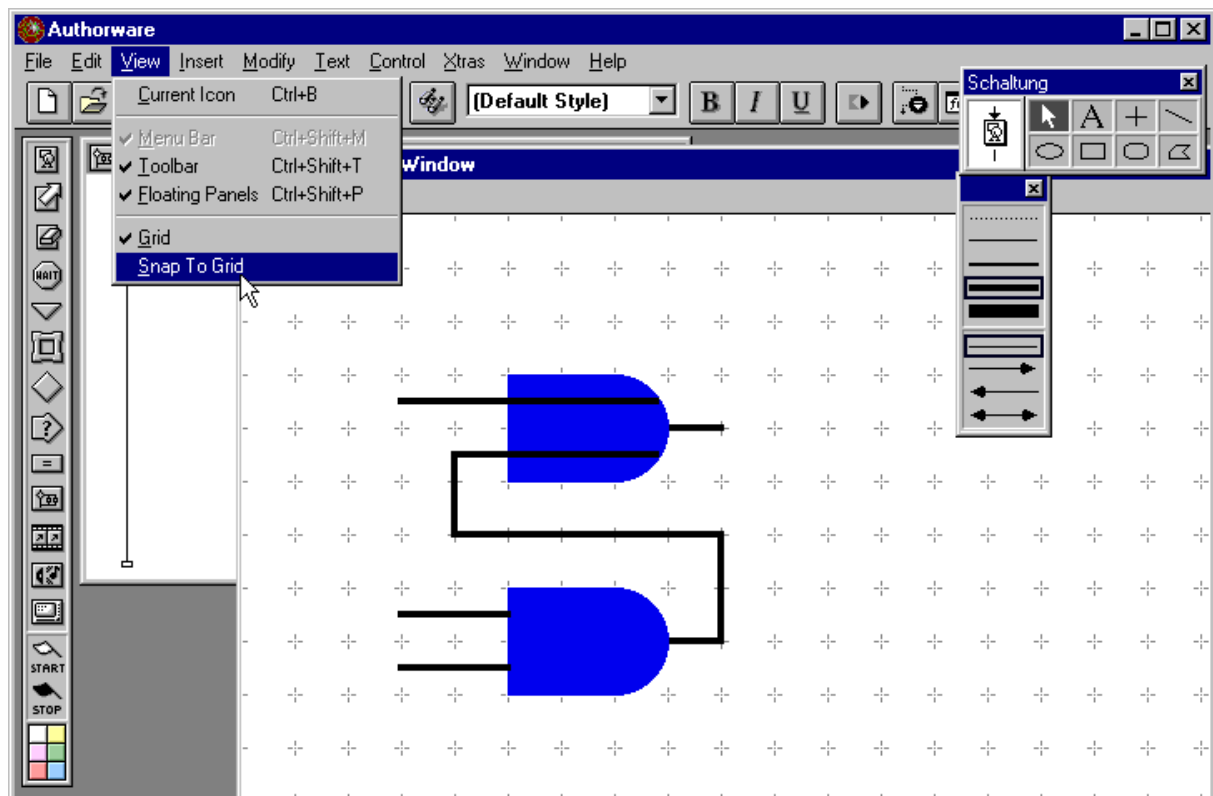


Bild 4.4 Inhalt des Display-Icons Schaltung

- Platzieren Sie nun ein Interaction-Icon mit drei Map-Icons. Wählen Sie als Response-Type Button.
- Benennen Sie die Map-Icon mit *Input A*, *Input B* und *Input C*.
- Halten Sie die SHIFT-Taste gedrückt und doppelklicken Sie das Interaction-Icon. Die SHIFT-Taste verhindert, dass zuvor der Bildschirm gelöscht wird.
- Wenn die Grid-Funktionen an sind, können Sie Objekte leichter platzieren (Bild 4.5).

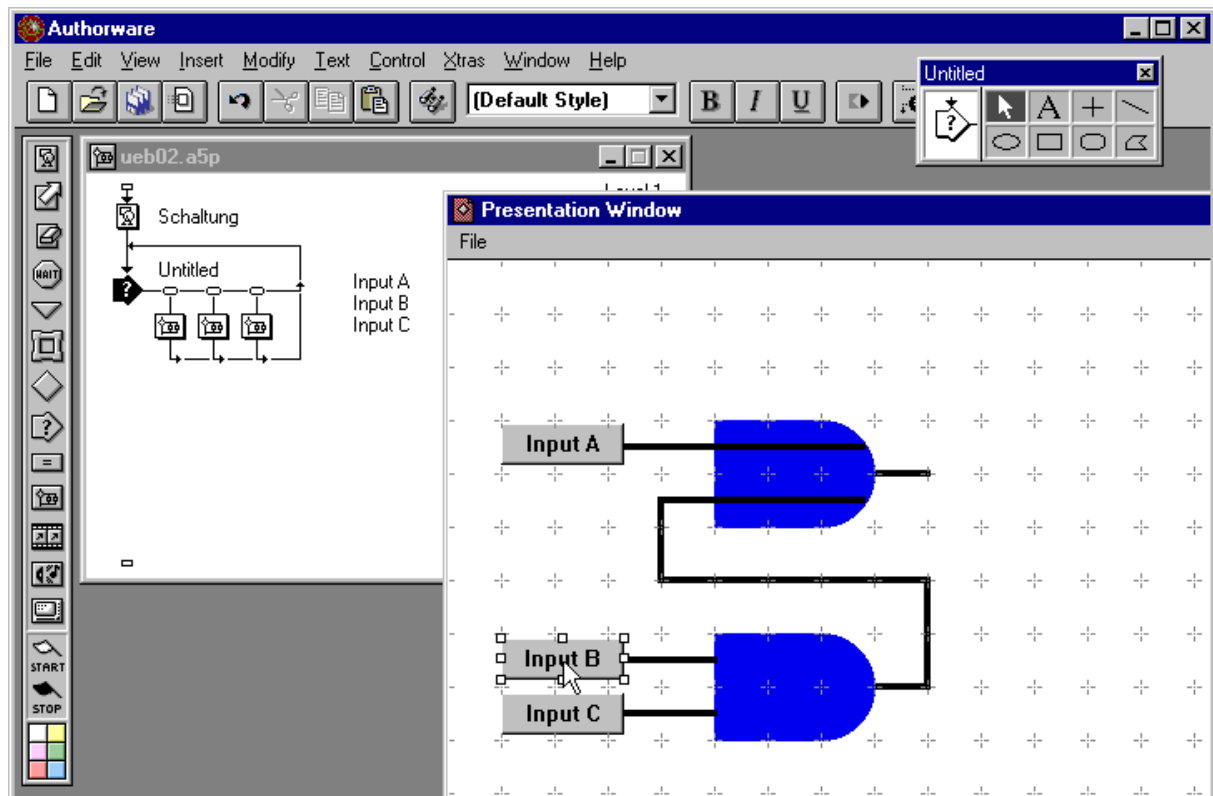


Bild 4.5 Platzieren der Buttons

- Kopieren Sie per bekannter Tastenkombination STRG-C und STRG-V das Display-Icon *Schaltung* in die Map-Icons.
- Entfernen Sie alle nicht benötigten Objekte aus dem kopierten Display-Icons. In dem ersten Map-Icon sind das alle, bis auf die Leitung vom Knopf *Input A* zum Gatter und der Gatterausgang des OR-Gatters.
- Diese beiden Linien färben Sie rot (Bild 4.6).
- Entsprechend gestalten Sie die anderen Leitungen.
- Sichern Sie die Applikation über *File > Save*.
- Starten Sie die Applikation und testen Sie aus.

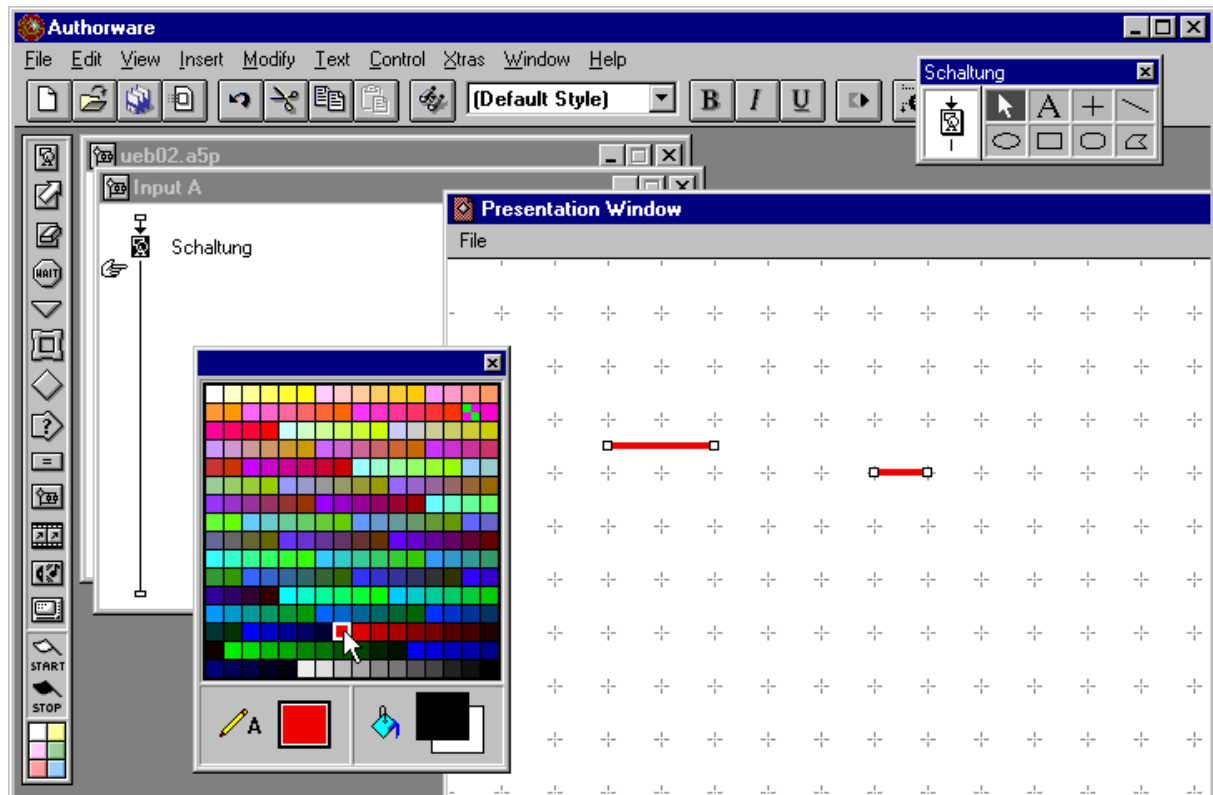
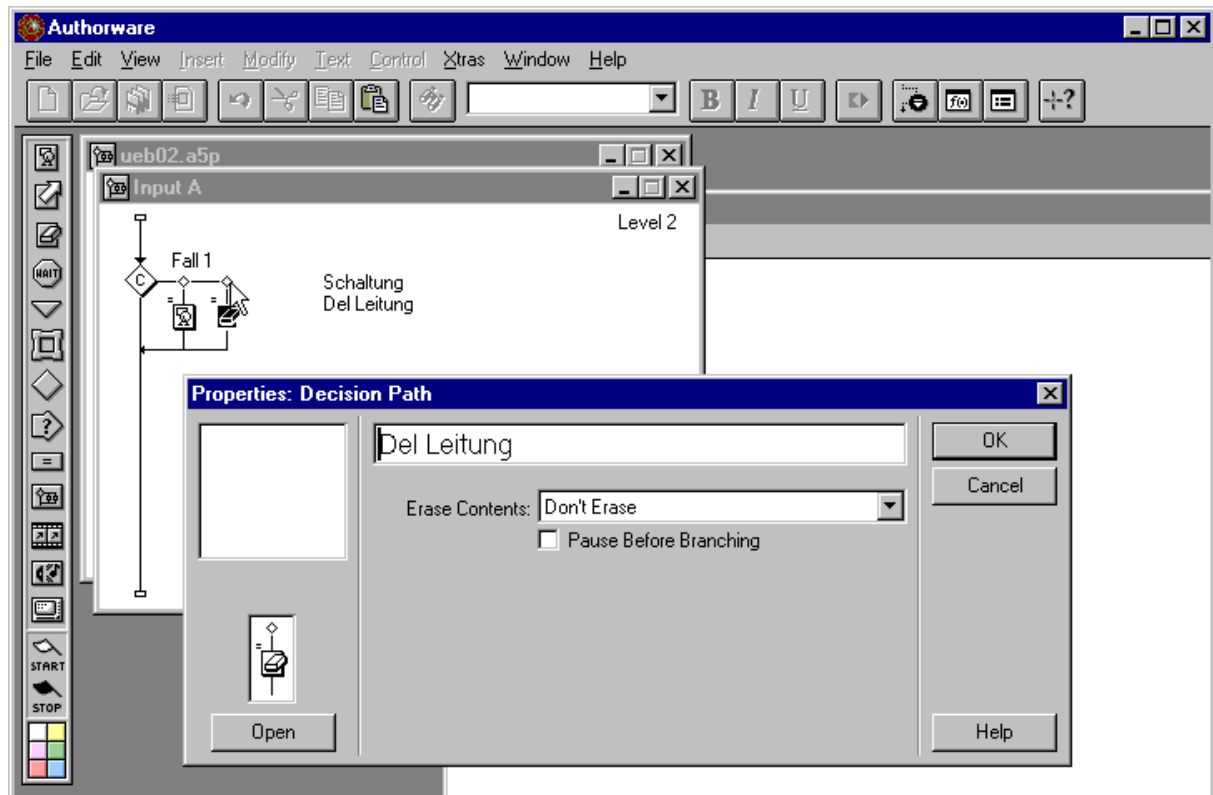


Bild 4.6 Färben der Leitungen

Die Knöpfe können betätigt werden, jedoch erscheinen die roten Leitungen selektiv, nicht additiv. Dies liegt daran, dass am Response-Typ der Map-Icons die Option *Erase* als Vorgabe *After Next Entry* hat. Hier muss *Don't Erase* eingestellt werden.

- Öffnen Sie das erste Map-Icon und platzieren ein Decision-Icon vor das enthaltene Display-Icon. Benennen Sie es in *Fall 1* um.
- Öffnen Sie das Decision-Icon durch ein Doppelklick.
- Unter der Option *Branch* wählen Sie *To Calculated Path*. Darunter in das Feld tragen Sie *A+1* ein.
- Per drag'n'drop ziehen Sie das Display-Icon nun an das Decision-Icon.
- Drücken Sie nun (während das Display-Icon selektiert ist) die Tastenkombination STRG+0. Es öffnet sich ein Calculation-Fenster.
- Tragen Sie hier *A:=TRUE* ein und beenden Sie mit ENTER (**nicht RETURN – ENTER ist das RETURN auf dem Nummernblock der Tastatur**).
- Als zweites platzieren Sie ein Erase-Icon (*Del Leitung*), hinter dem Display-Icon am Decision-Icon. Öffnen mit STRG+0 das Calculation-Fenster und geben *A:=FALSE* ein.
- Wählen Sie die Knoten über dem Display-Icon und stellen die Option *Erase Content* auf *Don't Erase*. Dasselbe machen Sie für das Erase-Icon (Bild 4.7).





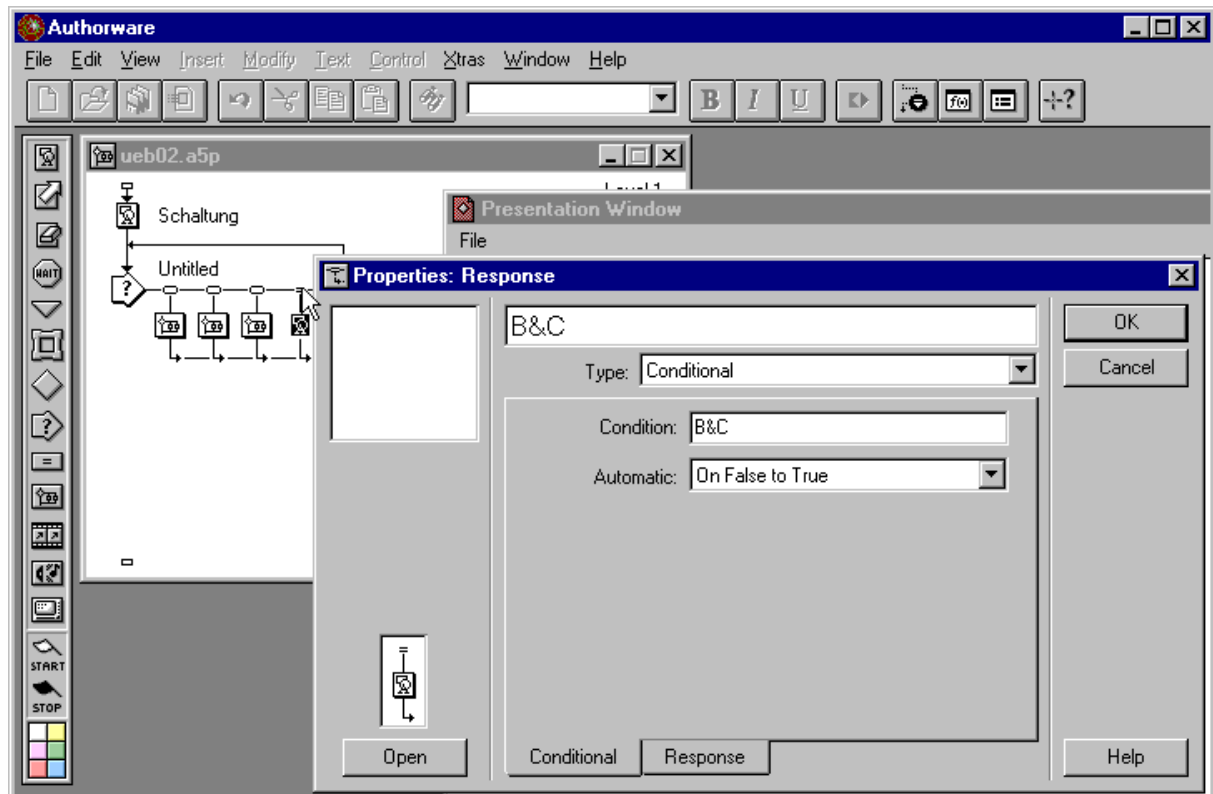
**Bild 4.7** Über die Knoten können weitere Optionen eingestellt werden.

- Sichern Sie die Applikation über *File > Save*.
- Starten Sie die Applikation und testen *Input A* aus.
- Beim ersten Klicken kommt die rote Leitung, beim zweiten Klicken erscheint das Eigenschaftensfenster des Erase-Icons. Schieben sie es ggf. zur Seite und klicken Sie die rote Leitung. Damit haben Sie die rote Leitung für die Zukunft als zu löschen markiert.

Der Knopf *Input A* sollte nun funktionieren. Ähnlich bauen Sie die Funktionalität von *Input B* und *Input C* auf, jedoch mit den Variablen B und C. Die Ausgabe von dem AND-Gatter betrachten wir anschliessend.

- Machen Sie nun eine Kopie des Display-Icons *Schaltung* per STRG-C in die Zwischenablage.
- Hängen Sie die Kopie an das Interaktion-Icon, als letztes, hinter den drei Map-Icons.
- Als Response-Typ wird automatisch *Button* eingestellt, ein Knopf erscheint auf dem Bildschirm. Ändern Sie den Response-Typ in *Conditional* um. Ändern Sie den Icon-Namen in *B & C* um.

- Öffnen Sie das Display-Icon und löschen Sie alles bis auf die Leitung vom AND-Gatter zum OR-Gatter und die Ausgangsleitung vom OR-Gatter. Färben Sie diese rot.
- Öffnen Sie den Knoten (das kleine Gleich-Zeichen) über dem Display-Icon und ändern die Option *Automatic* auf dem Reiter *Conditional* in *On False to True* um (Bild 4.8).



**Bild 4.8** Das Display-Icon wird nur ausgeführt, wenn die Bedingung von falsch nach wahr kippt.

- Hängen Sie nun ein Erase-Icon an das Interaction-Icon an. Es wird automatisch der Response-Typ *Conditional* gewählt und die Einstellungen des vorherigen Icon übernommen. Vergeben Sie hier den Namen  $\sim(B \& C)$ .
- Sichern Sie die Applikation über *File > Save*.
- Starten Sie die Applikation und testen Sie sie aus. Wenn das Eigenschaften-Fenster des Erase-Icon erscheint, klicken Sie die Leitung vom AND- zum OR-Gatter an.
- Sichern Sie die Applikation über *File > Save*.
- Packen Sie die Applikation.

## 4.4 Übung – ueb03

### 4.4.1 Lernziele

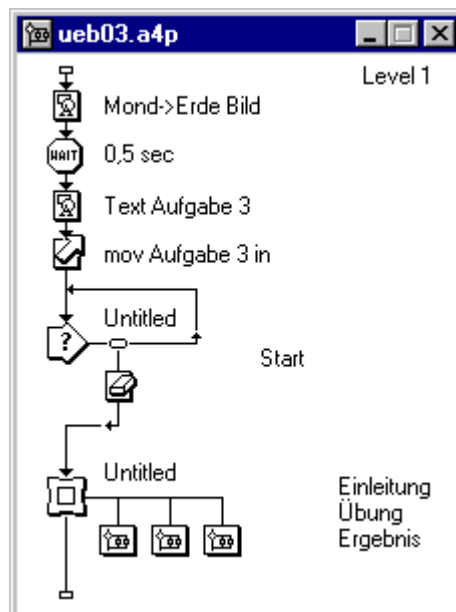
- Ausführen einer Bewegung
- Bewegen von Objekten und Platzieren an vordefinierten Stellen
- Framework-Icon
- Funktionsaufrufe
- Calculation-Icon

### 4.4.2 Aufgabe

Gegeben ist ein lauffähiges Authorware-Programm mit dem Namen *ueb03*. Starten Sie es und analysieren Sie den Ablauf. Anschliessend versuchen Sie selber, dieses Programm zu schreiben. Als Hilfsmittel stehen Ihnen einige Dateien zur Verfügung.

### 4.4.3 Hinweise

- Richten Sie wie in Kapitel 3 beschrieben ein neues Projektverzeichnis ein und übernehmen Sie die dort beschriebenen Voreinstellungen.
- Sichern Sie die Applikation über *File > Save* mit dem Dateinamen *ueb03* ab.



**Bild 4.9** Aufbau des Flussdiagramms zum Programm *ueb03.exe* auf Level 1.

## Programmaufbau

Als eine Programmierhilfe ist in Bild 4.9 der Aufbau des Flussdiagramms auf *Level 1* zu sehen. Bild 4.10 zeigt den Inhalt des Map-Icons *Übung*.

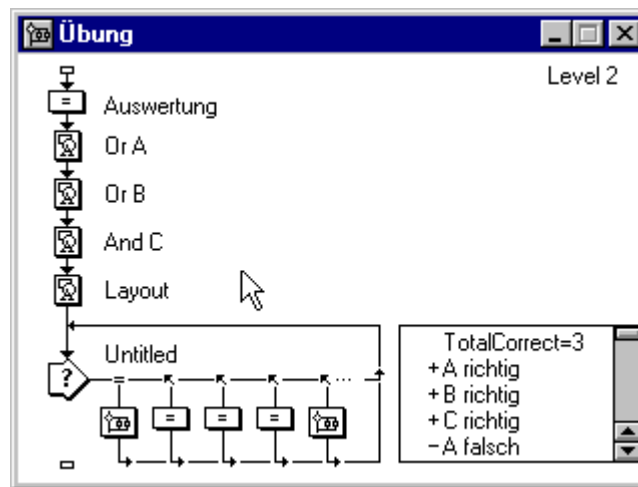


Bild 4.10 Interner Aufbau des Map-Icons *Übung* aus Bild 4.9

## Objektbewegung

In dem Vorspann zu Aufgabe 3 wird ein Textobjekt in das Bild bewegt. Die Bewegung von Objekten wird mit dem Motion-Icon realisiert.

## Target Area

Mit Hilfe der Eigenschaft *Target Area* wird in dem Präsentationsfenster ein Bereich definiert, der einem Objekt zugeordnet wird. Durch diese Zuordnung kann der Benutzer dieses Objekt<sup>1</sup> mit der Maus anfassen (gedrückte linke Maustaste), bewegen und ablegen (Maustaste loslassen). Wenn Sie ein Icon als erstes Icon an ein Interaction-Icon anhängen, fragt Sie AW nach dem Response-Typ. Hier wählen Sie ggf. *Target Area*.

## Variablen

Wird ein Objekt in dem als *Correct Response* definierten Bereich abgelegt, so wird die Variable *TotalCorrect* um eins erhöht. Ist der Bereich als *Wrong Response* definiert, dann wird *TotalWrong* inkrementiert. Initialisieren Sie vor jedem Test diese beiden Variablen.

Durch Zuweisen von FALSE zu *movable@“IconTitle“* werden die Objekte eines Display-Icons unbeweglich. Bei korrekter Antwort sollten die Objekte so fixiert werden.

<sup>1</sup> Voraussetzung ist, dass die Eigenschaft *Movable* des Display-Icons auf *Anywhere* gesetzt ist.

## 4.5 Übung – ueb04

### 4.5.1 Lernziele

- Bewegung entlang einem Pfad
- Sound-Icon
- Parallele Aktionen

### 4.5.2 Aufgabe

Gegeben ist ein Authorware-Programm mit dem Namen *ueb04*.. Starten Sie es und analysieren Sie den Ablauf. Anschliessend versuchen Sie selber dieses Programm zu schreiben. Als Hilfsmittel stehen Ihnen einige Dateien zur Verfügung.

#### Umzusetzendes

Achten Sie besonders auf die im Programm parallel ablaufenden Aktionen und versuchen Sie, diese in Ihrem Programm umzusetzen.

- Nach dem Start ist am Heizkessel eine Flamme zu sehen, die bis zum Ende des Programms brennt.
- Während ein Blubbern zu hören ist, steigt aus dem Herd Rauch auf, und der gelbe Ball wird zur Alarmanlage bewegt.
- Während bei der Waschmaschine das grüne Feld blinkt, wird der Ball zum Fernseher bewegt.

### 4.5.3 Hinweise

- Das offene Fenster wurde mit Hilfe der *Toolbox* erstellt.
- Den Rauch erstellen Sie, indem Sie mit Photoshop einen Teil des Herdes aus dem Hausbild (*haus.gif*) kopieren und in ein Display-Icon platzieren. Durch Layern wird der Rauch zwischen dem Hausbild und dem Teilbild des Herdes nach oben bewegt.
- Die Flamme wird als BMP-Sequenz eingebaut.
- Bild 4.11 zeigt Ausschnitte der Flussline.

#### BMP-Sequenz

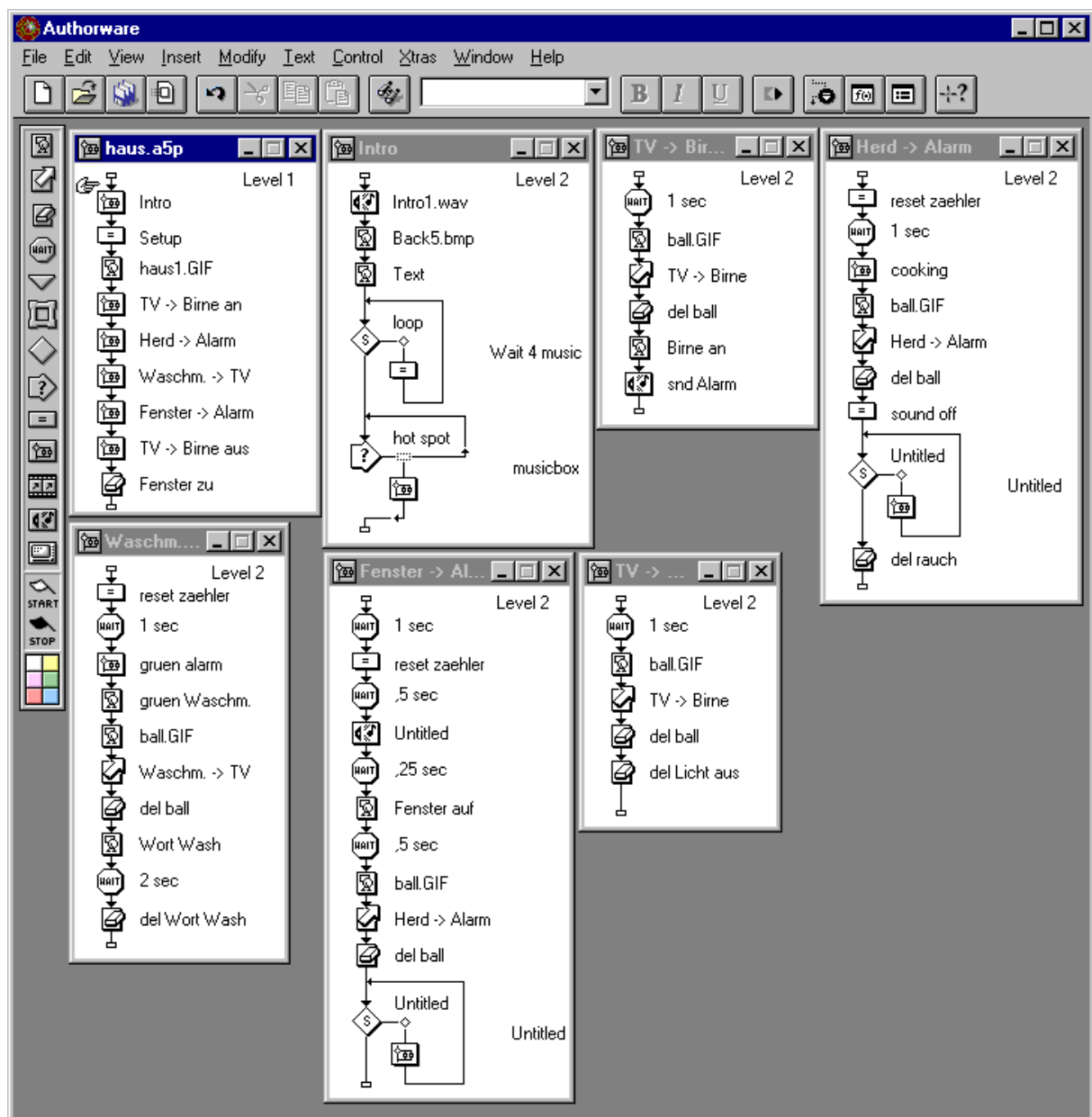
Digitale Filme können in unterschiedlichen Formaten vorliegen, eines davon ist die BMP-Sequenz. Bei der **Bit-Map-Picture-Sequenz** liegen Filme als Folgen von BMP-Bildern vor. Die Namen definiert die Reihenfolge. Für AW müssen die

Bildernamen folgendes Format haben: *AAAAxxxx.bmp*. *AAAA* steht für eine Zeichenfolge, *xxxx* für eine Zahl mit führenden Nullen.

BMP-Sequenzen haben den Vorteil, dass sie im Gegensatz zu AVIs oder MOVs gelayert werden können. AVIs oder MOVs laufen immer auf der obersten Ebene ab.

## Layern

Ist einem Objekt keine Ebene zugeordnet, erscheint es auf der Ebene 0. Objekte der selben Ebene werden nach ihrer Reihenfolge im Ablauf dargestellt, d.h. der zuletzt erscheinende verdeckt alle anderen auf seiner Ebene.



**Bild 4.11** Aufbau des Programms *ueb04.exe* in Authorware.

## 4.6 Übung – ueb05

### 4.6.1 Lernziele

- Selbständiges Entwerfen eines CBTs (Computer Based Training).

### 4.6.2 Aufgabe

Gegeben ist ein Thema und Informationen, die in einem CBT präsentiert werden sollen.

Das Thema ist aus dem Vorlesungsskript „Einführung in den VLSI-Entwurf“, Kapitel 2 entnommen. Weitere Informationen stehen dort.

#### Aufbau

Das Programm soll ein Hauptmenü haben, von dem man in zwei Abschnitte springen kann. Der erste Abschnitt enthält die Einführung, der zweite Abschnitt eine Aufgabe dazu.

Der erste Abschnitt soll aus mindestens fünf Seiten bestehen. Der Benutzer soll von Seite zu Seite (vor- und rückwärts) mit Buttons blättern können. Alle fünf Seiten zusammen präsentieren das Wissen.

Der zweite Abschnitt besteht aus min. einer Übungsaufgaben-Seite und einer letzten Seite mit Auswertung der Übungsaufgabe(n) (entsprechend aus ueb03.exe).

### 4.6.3 Thema: MOS-Transistoren

Integrierte MOS-Systeme enthalten mindestens drei leitende, voneinander isolierte Schichten. Geht man im Querschnitt eines Chips von oben nach unten, so sind dies die Schichten *Metall*, *Polysilizium* und *Diffusion*. Ein Full-Custom-Designer oder ein entsprechendes CAD-Werkzeug entwirft für diese drei Schichten bestimmte geometrische Muster, etwa in Form von Leitungspfaden oder von Kontaktlöchern, die zwei Leitungen durch die trennende Isolierschicht hindurch verbinden.

Verbindungsleitungen auf der Metallebene überkreuzen andere Leitungen auf der Poly- oder Diffusions-Ebene ohne wesentliche Beeinflussung, solange nicht Kontaktlöcher diese Ebenen miteinander verbinden. Falls jedoch eine Poly-Leitung eine Diffusions-Leitung kreuzt, entsteht ein MOS-Transistor. Dieser Transistor hat die Eigenschaften eines einfachen Schalters, wobei eine Spannung auf der Poly-Ebene den Stromfluss auf der Diffusions-Leitung kontrolliert.

Wir erläutern hauptsächlich den *n-Transistor*, der *p-Transistor* hat „duale“ Eigenschaften. Der *Source*- und *Drain-Anschluss* sind physikalisch nicht festgelegt;

vielmehr sind diese Anschlüsse beim n-Transistor dadurch definiert, dass die Drain-Source-Spannung  $V_{ds}$  positiv ist.

Unter dem Poly-Gate ist keine Diffusions-Leitung, und es gibt keine direkte Verbindung in der Diffusions-Schicht zwischen dem Source- und dem Drain-Anschluss des Transistors.

Ohne Ladung auf dem Gate ist der Transistor für die Drain-Source-Leitung ein offener Schalter. Das vom *Substrat* durch eine dünne Oxidschicht der *Dicke*  $D$  getrennte Gate bildet einen Kondensator. Legt man an das Gate eine positive Gate-Source-Spannung  $V_{gs}$  an, werden Elektronen in den Bereich unter dem Gate gezogen. Übersteigt  $V_{gs}$  eine *Schwellschpannung*  $V_{th}$ , bilden die Elektronen einen leitenden Kanal zwischen Source und Drain. n-Transistoren haben normalerweise eine positive Schwellspannung, typischerweise etwa  $0,2V_{dd}$ , wobei  $V_{dd}$  die positive Versorgungsspannung ist.