

Multimedia-Autorensysteme

Graphisch-interaktive Werkzeuge zur Erstellung multimedialer Anwendungen

Dietrich Boles, Marco Schlattmann
Informatik-Institut OFFIS
Escherweg 2
D-26121 Oldenburg
{boles,schlattmann}@offis.uni-oldenburg.de

Zusammenfassung

Elektronische multimediale Dokumente und Anwendungen ermöglichen gegenüber den traditionellen Printmedien eine qualitative Verbesserung der Informationsvermittlung. Die Nutzung zusätzlicher Medien wie Audio, Video und Animation neben Text und Graphik sowie die Berücksichtigung von Interaktionsformen erfordert jedoch von Autoren sowohl ein konzeptionelles Umdenken bzgl. der Aufbereitung und Präsentation von Informationen als auch ein technisches Umdenken, was die Erstellung derartiger Anwendungen betrifft. Multimedia-Autorensysteme, die in diesem Artikel vorgestellt werden, unterstützen Autoren bei der Erstellung interaktiver multimedialer Anwendungen, und zwar insbesondere durch die Bereitstellung einfach zu bedienender graphisch-interaktiver Werkzeuge, die auf intuitiven Authoring-Metaphern basieren.

1 Einleitung

Multimedia ist zum Schlagwort der 90er Jahre geworden. Durch die Integration von Audio und Video können Computeranwendungen in vielen Bereichen interessanter gestaltet und effektvoller eingesetzt werden. Multimediale Lehr- und Lernprogramme erhöhen den Lernerfolg, multimediale Präsentationen im Geschäftsbereich steigern den Verkaufsumsatz, multimedial gestaltete Spiele bringen mehr Spaß und Spannung, multimediale Informationssysteme ermöglichen einen effektiveren Informationsaustausch. Multimedia-Systeme sehen dabei in ihrem Benutzer keinen passiven Konsumenten sondern einen aktiven Teilnehmer, der die Anwendungen durch Eingaben beeinflussen kann. Videos lassen sich vom Benutzer starten und stoppen. Je nach Korrektheit der Antworten können Lektionen eines elektronischen Sprachkurses wiederholt oder weiterführende Lektionen gestartet werden. Aus Produkt- bzw. Firmenpräsentationen kann ein Benutzer die Teile auswählen, die ihn gerade interessieren. Farbgebung und Layout der Benutzungsoberfläche, Musiklautstärke

oder Geschwindigkeit von elektronischen Spielen lassen sich vom Benutzer dynamisch je nach seinen momentanen Wünschen einstellen.

Auch im wissenschaftlichen Bereich ist ein Umbruch von der papierbasierten zu einer elektronischen Publikationsweise zu erkennen. Ein häufig genannter Kritikpunkt am traditionellen Publikationswesen ist seine Schwerfälligkeit, die sich unter anderem durch die lange Zeitspanne zwischen dem Einreichen eines Artikels bei einer Fachzeitschrift bis zu seiner endgültigen Veröffentlichung äußert. Des weiteren bereitet das exponentielle Wachstum wissenschaftlicher Literatur vielen wissenschaftlichen Institutionen und Bibliotheken heute nicht nur Kosten- sondern auch Platzprobleme. Auf Leserseite wird es immer schwieriger, aus der großen Menge von Publikationen für den eigenen Bedarf relevante Informationen zu finden. Insbesondere durch den Einsatz weltumspannender Rechnernetze und den Aufbau verteilter digitaler Bibliotheken bietet eine elektronische Publikationsweise hier unbestritten ein enormes zeitliches und ökonomisches Rationalisierungspotential. Durch die Möglichkeit der Integration von Audios, Videos und Animationen in elektronische Dokumente kann darüber hinaus eine qualitative Verbesserung der Informationsvermittlung erreicht werden. Über sogenannte Hyperlinks können Dokumente bzw. Dokumentteile miteinander verbunden und Informationsnetzwerke aufgebaut werden, die menschliche Formen der Wissensrepräsentation viel besser widerspiegeln als herkömmliche sequentielle Buchstrukturen.

So vielversprechend solche Szenarien auch klingen, ein Problem dabei ist, daß multimediale Dokumente und Anwendungen erst einmal entwickelt werden müssen. Ihre Erstellung bedeutet insbesondere für Autoren eine veränderte Arbeitsweise. So ist beispielsweise sowohl ein konzeptionelles als auch ein technisches Umdenken erforderlich. Konzeptionell neue Fragestellungen, die sich für einen Autor ergeben, sind: wie bereite ich die zu vermittelnden Informationen auf, welche Medientypen verwende ich, wie verbinde ich die einzelnen Informationseinheiten miteinander, welche Formen der Interaktion des Benutzers unterstützte ich und wie präsentiere ich die Informationen? Im technischen Sinne muß sich ein Autor mit Problemen auseinandersetzen, die inhärent mit den neuen Medien verbunden sind, wie die Synchronisation und Datenkompression.

Die Verwendung konventioneller Programmiersprachen stellt natürlich eine Realisierungsmöglichkeit dar. Sie ist jedoch mit einem hohen Zeit- und Kostenaufwand verbunden und nur erfahrenen Programmierern vorbehalten. Nicht-Programmierern müssen Software-Werkzeuge zur Verfügung gestellt werden, die eine einfache und schnelle Entwicklung multimedialer Anwendungen ohne Programmierkenntnisse ermöglichen. Derartige Werkzeuge stellen die Multimedia-Autorensysteme dar, die in diesem Artikel vorgestellt werden. Durch die Benutzung eines Autorensystems kann sich der Entwickler voll und ganz der Aufbereitung der zu präsentierenden Informationen und der Gestaltung der Benutzerschnittstelle der Anwendung widmen und muß sich nicht um programmiertechnische Details kümmern.

Dieser Artikel ist so gegliedert, daß nach dieser motivierenden Einleitung im zweiten Abschnitt Autorensysteme unter verschiedenen Aspekten charakterisiert werden. Im dritten Abschnitt werden anschließend drei konkrete Autorensysteme etwas genauer vorgestellt. Der vierte Abschnitt enthält eine abschließende Bewertung und einen Blick in die Zukunft. Am Ende des Artikels finden sich eine Reihe von Verweisen auf weiterführende Informationsquellen.

2 Charakterisierung von Autorensystemen

Verallgemeinert lassen sich Multimedia-Autorensysteme definieren als graphisch-interaktive Softwareentwicklungsumgebungen zur Erstellung interaktiver multimedialer Anwendungen. Spezifischere Details über konkrete Anforderungen an Autorensysteme, ihren Aufbau und die generelle Arbeitsweise mit Autorensystemen werden in diesem Abschnitt vorgestellt. Außerdem werden existierende Autorensysteme klassifiziert, ihre Eignung für verschiedene Einsatzbereiche untersucht und Empfehlungen für die Auswahl von Autorensystemen gegeben. Zunächst wird jedoch eine kurze Charakterisierung interaktiver multimedialer Anwendungen, also der Produkte, die mit Autorensystemen erstellt werden, vorgenommen, auf der die Charakterisierung von Autorensystemen basiert.

2.1 Interaktive multimediale Anwendungen

Ziel einer multimedialen Anwendung ist es, einen Benutzer über einen bestimmten Sachverhalt zu informieren. Dieses Ziel wird dadurch erreicht, daß einzelne voneinander unabhängige Informationseinheiten in einen Zusammenhang zueinander gebracht werden. Multimediale Anwendungen lassen sich damit charakterisieren als ein Netzwerk aus Knoten und Kanten. Die Knoten repräsentieren die elementaren Informationseinheiten. Sie werden als Informationsobjekte bezeichnet. Die Kanten legen den Zusammenhang, d.h. Beziehungen zwischen den Informationsobjekten fest.

2.1.1 Informationsobjekte

Die einzelnen Informationseinheiten einer multimedialen Anwendung, wie Graphiken, Musikstücke und Videos, kapseln jeweils bestimmte Informationen und können daher, angelehnt an die Terminologie der objektorientierten Programmierung, als (Informations-) Objekte betrachtet werden. Neben Medienobjekten, wie Texte, Graphiken, Audios, Videos und Animationen, bilden Interaktionsobjekte, wie Buttons, Menüs, Schieberegler und Texteingabefelder, eine zweite Klasse von Informationsobjekten. In eine dritte Klasse lassen sich anwendungsspezifische Objekte einordnen, die beispielsweise eine Datenbank, eine Wissensbasis oder eine Statistikkomponente repräsentieren.

Mitunter ist es sinnvoll, einen Teil des Beziehungsnetzwerks als eine Einheit zu betrachten, beispielsweise einen Film zusammen mit seiner Vertonung. Derartige zusammengesetzte Einheiten können als komplexe Informationsobjekte bzw. Multimedia-Objekte angesehen werden, die durch Komposition von anderen – elementaren bzw. selbst wieder komplexen – Informationsobjekten – ihren Komponentenobjekten – und Beziehungen zwischen den Komponentenobjekten gebildet werden.

2.1.2 Beziehungen

Der Zusammenhang, in dem die einzelnen Informationseinheiten einer multimedialen Anwendung präsentiert werden, läßt sich über Beziehungen zwischen den Informationsobjekten modellieren. Beziehungen können dadurch charakterisiert werden, daß bestimmte auslösende Ereignisse (Auslöser) bestimmte Aktionen (Auswirkung) implizieren. Im

Multimedia-Netzwerk repräsentieren die Kanten diese Beziehungen. Auslöser und Auswirkungen können dabei verschiedenen Typs sein. Zeitliche Auslöser bzw. Auswirkungen sind beispielsweise der Start eines Videos oder der Abbruch eines Audios. Die Umpositionierung eines Textes auf dem Bildschirm oder die Vergrößerung einer Zeichnung sind Beispiele räumlicher Auslöser bzw. Auswirkungen. Darüber hinaus gibt es Auslöser und Auswirkungen gestalterischer Art (Laustärkenänderung eines Musikstückes, Farbänderung eines Graphik-Objektes, ...) oder auch anwendungsspezifischer Art (Eintrag eines bestimmten Tupels in eine Datenbank, Erreichen eines bestimmten Mittelwertes in einer Statistikkomponente, ...). Als weitere Klasse von Auslösern können Benutzerinteraktionen angesehen werden, wie die Auswahl aus einem Menü, die Eingabe eines Textes in ein Texteingabefeld, das Betätigen eines Schiebereglers oder das direkt-manipulative Verschieben eines Objektes mit Hilfe der Maus. Es ist durchaus möglich, daß sich der Typ des Auslösers und der Auswirkung einer Beziehung unterscheiden. Die Reduzierung der Lautstärke eines Audios (Auswirkung) aufgrund des Starts eines Videos (Auslöser) ist ein Beispiel für eine zeitlich-gestalterische Beziehung.

Von besonderem Interesse in multimedialen Anwendungen sind dabei zeitliche, räumliche und Navigationsbeziehungen. Über zeitliche Beziehungen (Beziehungen mit einem zeitlichen Auslöser und einer zeitlichen Auswirkung) wird der generelle Ablauf der Anwendung definiert, räumliche Beziehungen (räumlicher Auslöser und räumliche Auswirkung) dienen der Anordnung visueller Objekte auf dem Bildschirm und über Navigationsbeziehungen (NutzerAuswahl als Auslöser und zeitliche Auswirkung) wird festgelegt, an welcher Stelle ein Benutzer den Ablauf der Anwendung beeinflussen kann.

2.2 Anforderungen an Autorensysteme

Die Hauptanforderungen an Autorensysteme, die sich aus der obigen Charakterisierung interaktiver multimedialer Anwendungen ergeben, sind die Einbindung von Informationsobjekten unterschiedlichen Typs bzw. Formats in die Anwendung, die Verwaltung von Objekten und Beziehungen sowie die Bereitstellung von Mechanismen zur Definition von Beziehungen. Weiterhin müssen Autorensysteme technische Aspekte, die mit den neuen Medien verbunden sind, wie Synchronisation und Kompression, vor den Entwicklern verbergen. Die anfallenden Aufgaben sollten so weit wie möglich mit graphisch-interaktiven Hilfsmitteln durchgeführt werden können, um auch Nicht-Programmierern die Erstellung interaktiver multimedialer Anwendungen zu ermöglichen. Graphisch-interaktiv bedeutet in diesem Zusammenhang die Nutzung der Maus und direkt-manipulativer Interaktionsformen wie Cut-and-Paste- und Drag-and-Drop-Mechanismen unterstützt durch geeignete Metaphern anstelle der Notwendigkeit einer textuellen Programmierung.

Zusätzlich stellen die meisten Autorensysteme eigene Werkzeuge zur Erstellung bzw. Bearbeitung medientypspezifischer Objekte wie Graphik-, Animations-, Audio- und Video-Editoren zur Verfügung. Diese besitzen häufig jedoch nur rudimentäre Funktionalität. Viel wichtiger ist es deshalb, daß Autorensysteme den Import gängiger Formate unterstützen, so daß Medienobjekte mit professionellen Werkzeugen erstellt und importiert werden können.

Zur Verwaltung von Informationsobjekten und Beziehungen kann das Autorensystem entweder auf die Dateiverwaltung des Betriebssystems zurückgreifen, oder es kann ein eige-

nes oder fremdes Datenbanksystem verwenden. Existierende Autorensysteme basieren im allgemeinen auf der ersten Alternative, obwohl durch den Einsatz von Datenbanksystemen deren Vorteile wie flexible Suchmöglichkeiten, Integritätssicherung, Mehrbenutzerbetrieb, Datensicherung und Datenschutz genutzt werden könnten.

Der größte Unterschied existierender Autorensysteme besteht in der Art und Weise, wie die Definition von zeitlichen Beziehungen und von Navigationsbeziehungen unterstützt wird:

- In timeline-basierten Autorensystemen werden die Informationsobjekte symbolisch auf einer Zeitachse angeordnet, die den zeitlichen Verlauf der Anwendung festlegt. Navigationsinteraktionen können der Zeitachse zugeordnet werden und bewirken im allgemeinen einen Zeitsprung.
- Flowchart-basierte Autorensysteme sind dadurch gekennzeichnet, daß die Informationsobjekte — durch Ikonen bzw. Miniaturen repräsentiert — in Diagrammen durch Kanten miteinander verbunden werden, die den möglichen Verlauf der Anwendung widerspiegeln. Gehen dabei von einem Objekt mehrere Kanten zu unterschiedlichen Objekten aus, so wird die während des Ablaufs der Anwendung tatsächlich durchflossene Kante im allgemeinen durch eine Navigationsinteraktion bestimmt.
- Area-basierte Autorensysteme lassen sich dadurch charakterisieren, daß die zu präsentierenden Objekte auf Flächen angeordnet werden, die als Frames, Karten, Seiten, Fenster oder auch Dias bezeichnet werden und die im Prinzip einen Bildschirm repräsentieren, wie ihn ein Benutzer während der Präsentation für einen bestimmten Zeitraum zu sehen bekommt. Eine komplette interaktive multimediale Anwendung setzt sich aus einer Menge solcher Flächen zusammen, die einem Benutzer in einer bestimmten von ihm durch Navigationsinteraktionen beeinflussbaren Reihenfolge gezeigt werden. Im Prinzip sind auch die timeline- und flowchart-basierten Autorensysteme area-basierte Autorensysteme, die aber zusätzliche Metaphern für die Verknüpfung von Bildschirmseiten verwenden.

Für die Festlegung räumlicher Beziehungen verwenden quasi alle Autorensysteme — wie gewöhnliche Softwareentwicklungsumgebungen auch — sogenannte Layout-Editoren, in denen visuelle Objekte mit Hilfe von Drag-and-Drop-Mechanismen räumlich zueinander angeordnet werden können. Zur Definition einfacher Navigationsbeziehungen werden im allgemeinen spezielle Menüs oder Dialogboxen angeboten. Andere Typen von Beziehungen lassen sich nur schwierig oder gar nicht mit Hilfe graphisch-interaktiver Techniken spezifizieren. Hier greifen viele Autorensysteme auf spezielle Skript- oder sogar Programmiersprachen zurück.

Neben den oben genannten Hauptanforderungen sollten Autorensysteme als weitere Anforderungen insbesondere einen jederzeitigen Test der sich in Entwicklung befindlichen Anwendung, die Generierung von Stand-Alone-Anwendungen sowie die Anbindung externer Programme unterstützen. Ein jederzeitiger Test ermöglicht die Erstellung der Anwendung im Sinne des Rapid-Prototyping, dem insbesondere bei der Entwicklung von Benutzerschnittstellen eine hohe Bedeutung zukommt. Die Generierung von Stand-Alone-Anwendungen ist sinnvoll, damit die Anwendungen von zukünftigen Nutzern ohne zusätzliche Software ausgeführt werden können. Durch externe Schnittstellen sind bspw. Zugriffe auf Datenbanken oder Netzwerke möglich.

2.3 Aufbau von und Arbeitsweise mit Autorensystemen

Die typische Arbeitsweise eines Multimedia-Entwicklers mit einem Autorensystem und der Aufbau eines Autorensystems ergeben sich aus den obigen Anforderungen und werden in Abbildung 1 skizziert.

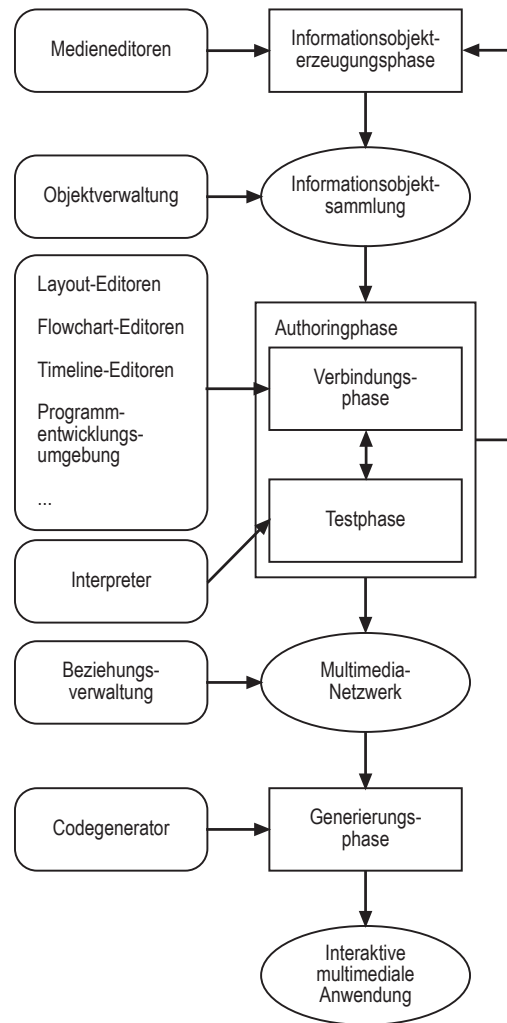


Abbildung 1: Aufbau von und Arbeitsweise mit Autorensystemen

In der Informationsobjekterzeugungsphase werden die elementaren Informationsobjekte erzeugt bzw. bearbeitet. Hierzu werden entweder fremde oder ins Autorensystem integrierte Werkzeuge verwendet. Die noch voneinander unabhängigen Objekte werden in einer Informationsobjektsammlung abgelegt, die von der Objektverwaltung des Autorensystems verwaltet wird. Die eigentliche Authoringphase besteht aus der Verbindungs- und Testphase. In der Verbindungsphase werden Beziehungen zwischen den Objekten definiert. Dazu werden unterschiedliche Werkzeuge zur Beziehungsmodellierung eingesetzt wie Layout-Editoren, Timeline-Editoren, Flowchart-Editoren oder auch Programm-entwicklungsumgebungen für die Anfertigung von Skripten bzw. Programmen. In der Testphase kann sich ein Entwickler durch Aktivierung eines Interpreters jederzeit Ausschnitte der Anwendung anschauen und damit überprüfen, ob die Anwendung seinen Vorstellungen

gen entspricht. Ergebnis der Authoringphase Phase ist ein Multimedia-Netzwerk, das von der Beziehungsverwaltung des Autorensystems verwaltet wird. Letztendlich wird als Endprodukt mit Hilfe eines Generators eine interaktive multimediale Anwendung als Stand-Alone-Anwendung erzeugt.

2.4 Klassifikation von Autorensystemen

Eine Klassifikationsmöglichkeit für Autorensysteme auf der Basis der Art und Weise, wie die Definition zeitlicher Beziehungen unterstützt wird, wurde oben bereits vorgestellt. Weitere Klassifikationen von Autorensystemen sind bspw. aufgrund der Authoring-Plattformen bzw. der Abspielplattformen möglich. In der Fachliteratur hat sich eine nicht-orthogonale Klassifikation auf der Grundlage bestimmter Authoring-Paradigmen etabliert, die eine Erweiterung der obigen Klassifikation darstellt. Diese Klassifikation enthält die im folgenden erläuterten Kategorien, für die jeweils Vor- und Nachteile, die geeignetsten Einsatzbereiche und exemplarisch einige Autorensysteme aufgeführt werden.

2.4.1 Stapel-Karten-Metapher

Die Autorensysteme, die gemäß der Stapel-Karten-Metapher arbeiten, bilden eine Teilmenge der area-basierten Autorensysteme. Multimedia-Anwendungen werden durch Kartenstapel repräsentiert, bei denen die Karten einzelne Bildschirmseiten darstellen. Die Autorensysteme dieser Kategorie sind insbesondere geeignet für die Erstellung statischer Hypermedia-Dokumente. Vorteile für den Autor bestehen in einer übersichtlichen Strukturierung und einer zumeist kurzen Einarbeitungszeit. Typische Vertreter diese Kategorie sind HyperCard, MetaCard und SuperCard.

2.4.2 Buch-Seiten-Metapher

Die Buch-Seiten-Metapher ist der Stapel-Karten-Metapher sehr ähnlich. Insbesondere sind Einsatzbereich sowie Vor- und Nachteile der Autorensysteme dieser beiden Kategorien identisch. Bei den Systemen, die die Buch-Seiten-Metapher nutzen, wird die Produktion einer Multimedia-Anwendung verglichen mit der Erstellung eines Buches, das sich aus einzelnen Seiten zusammensetzt, wobei die Seiten jedoch nicht linear sondern netzwerkartig miteinander verbunden sind. Typische Vertreter dieser Kategorie sind ToolBook, Create Interactive, ClickWorks und Click&Create.

2.4.3 Zeitachsen-Metapher

Die Zeitachsen-Metapher wurde bereits oben beschrieben. Autorensysteme dieser Kategorie eignen sich insbesondere für die Entwicklung dynamischer, interaktiver Multimedia-Präsentationen. Die Entwicklung wird dabei häufig mit der Produktion eines Filmes verglichen. Der wesentliche Vorteil liegt in der jederzeitigen Kontrolle des Ablaufs des Geschehens anhand der Zeitachse. Bei größeren Projekten wird die Anordnung der Objekte jedoch schnell unübersichtlich. Typische Vertreter dieser Kategorie sind Director, Action! und Media Mogul.

2.4.4 Flußdiagramm-Metapher

Auch die Flußdiagramm-Metapher wurde bereits oben beschrieben. Autorensysteme, die auf dieser Metapher basieren, werden sehr häufig für die Entwicklung statischer multimedialer CBT-Anwendungen (Computer Based Training) eingesetzt. Sie bieten insbesondere einen guten Überblick über den Aufbau der Anwendung. Typische Vertreter dieser Kategorie sind Authorware, IconAuthor und CourseBuilder.

2.4.5 Objekt-Metapher

Den Autorensystemen dieser Kategorie liegt die aus der Welt der Objektorientierung bekannte Objekt-Metapher zugrunde. Objekte bzw. ihre Eigenschaften werden dabei visuell repräsentiert und graphisch-interaktiv manipuliert. Objektorientierte Autorensysteme eignen sich insbesondere für die Erstellung dynamischer Multimedia-Anwendungen in der Unterhaltungsbranche und im Edutainmentbereich. Ihre Vorteile leiten sich aus den Vorteilen der Objektorientierung ab, das sind insbesondere eine klare Strukturierung, eine einfache Erweiterbarkeit und die Wiederverwendung von Teilen der Anwendung in anderen Projekten. Autoren sollten jedoch ein Grundverständnis der objektorientierten Programmierung besitzen. Typische Vertreter dieser Kategorie sind mTropolis und SK8. Auch die Skriptsprachen von einigen Autorensystemen wie HyperCard oder Director besitzen objektorientierte Eigenschaften.

2.4.6 Skript-Metapher

Wie bereits erwähnt stellen viele Autorensysteme eine Skriptsprache zur Verfügung, um auch Aspekte, die sich nicht graphisch-interaktiv modellieren lassen, umsetzen zu können. Diese Skriptsprachen enthalten bspw. Befehle zum Abfragen von Maus-Events, zum Seitenwechsel oder zum Einbinden externer Programme. Des weiteren sind häufig auch aus Programmiersprachen bekannte Kontrollstrukturen integriert. Der Vorteil von Skriptsprachen ist natürlich eine wesentlich größere Flexibilität und Leistungsfähigkeit. Der Nachteil besteht in der Voraussetzung von Programmierkenntnissen beim Autor. Bekannte Skriptsprachen sind HyperTalk in HyperCard und Lingo in Director.

2.5 Auswahl von Autorensystemen

Eine sehr häufig gestellte Frage ist: Was ist eigentlich das beste Autorensystem? Diese Frage ist nicht pauschal zu beantworten. Vielmehr sollte jeder, der eine eigene Multimedia-Anwendung erstellen will, zunächst konkrete Anforderungen ermitteln:

- Was ist der genaue Einsatzbereich?
- Auf welcher Plattform wird entwickelt?
- Auf welchen Plattformen soll die Anwendung lauffähig sein?
- Müssen externe Programme eingebunden werden?

- Sind beim Entwickler Programmierkenntnisse vorhanden?
- Ist eine Anbindung ans Internet geplant?
- Welche Werkzeuge zur Medienbearbeitung werden benötigt?
- Welche Formate liegen den Medienobjekten zugrunde?
- Generelle Anforderungen wie Preis, Service, ...

Gemäß dieser Anforderungen sollte dann das geeignetste Autorensystem ausgewählt werden. Das bedeutet auch, daß es für einen Autor unter Umständen notwendig ist, den Umgang mit unterschiedlichen Autorensystemen zu erlernen. Wenn nämlich die Anforderungsanalyse ergibt, daß sich die zu entwickelnde Anwendung nicht oder nur sehr schwierig mit dem bisher von ihm eingesetzten System realisieren läßt, ist ein Umstieg dringend zu empfehlen.

Vor dem Kauf eines Autorensystems ist anzuraten, sich zunächst eine Demo-Version zu beschaffen, um den Umgang mit dem System und seine Funktionalität zu testen. Existierende Autorensysteme weisen große Unterschiede in der Bandbreite ihrer Funktionalität und auch im Preis auf (kostenlos bis hin zu einigen zehntausend DM).

3 Autorensysteme im Detail

Um die im vorangehenden Abschnitt vorgenommen Charakterisierung von Autorensystemen zu verdeutlichen, werden in diesem Abschnitt exemplarisch drei weit verbreitete Autorensysteme etwas ausführlicher vorgestellt, und zwar HyperCard (Stapel-Karten- und Skript-Metapher), Director (Zeitachsen-, Skript- und Objekt-Metapher) und Authorware (Flußdiagramm-Metapher).

3.1 HyperCard

HyperCard, das ausschließlich auf Apple Macintosh Rechnern verfügbar ist, war eines der ersten Multimedia-Autorensysteme. Zur Zeit ist es in der Version 2.4 auf dem Markt. Es basiert auf der Stapel-Karten-Metapher, enthält aber zusätzlich eine Skriptsprache namens HyperTalk. In HyperCard existieren fünf verschiedene Typen von Elementen: Stapel, Hintergründe, Karten, (Texteingabe-)Felder und Buttons. Ein Stapel ist eine Macintosh-Datei und entspricht im Prinzip einer HyperCard-Anwendung. Ein Stapel besteht aus einer Menge von Karten. Karten stellen die Basiseinheiten zur Präsentation von Informationen dar. Mit jeder Karte ist ein sogenannter Hintergrund assoziiert, der dieselbe Größe wie die Karte hat und räumlich gesehen auf der Karte liegt. Ein Hintergrund gehört meistens zu mehreren Karten. Auf ihm liegen solche Informationen, die allen assoziierten Karten gemeinsam sind. Die eigentlichen Informationsobjekte, die auf die Hintergründen bzw. Karten gelegt werden können, sind Bilder, Buttons und Felder. Während Bilder als Bitmap-Graphiken realisiert sind und nicht als HyperCard-Elemente im engeren Sinn zählen, bilden Buttons und Felder aktive Elemente, die miteinander „kommunizieren“

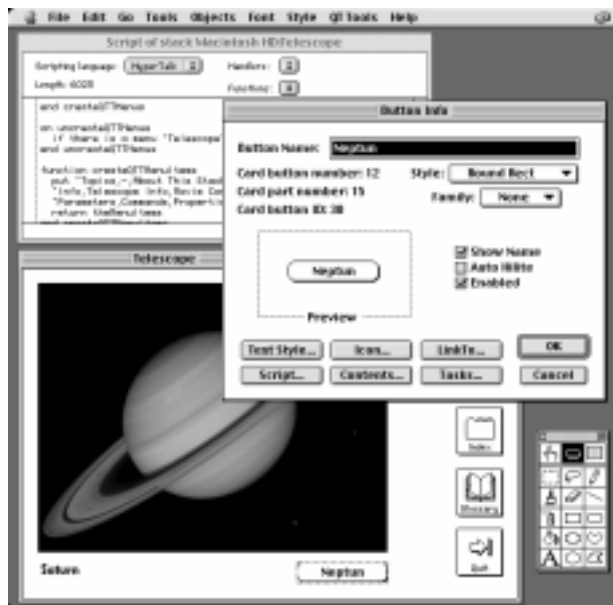


Abbildung 2: Benutzungsoberfläche von HyperCard

können. Als weitere Elemente können der Start-Stapel, über den HyperCard gestartet wird, und das HyperCard-System selbst angesehen werden.

HyperCard-Elemente besitzen Attribute, die festlegen, wie ein Element aussieht und wie es agiert. Diese Attribute werden Properties genannt. Properties eines Feld-Elementes sind beispielsweise sein Name, seine Position auf dem Bildschirm, seine Größe oder auch der verwendete Textfont. Die Properties eines Elementes werden normalerweise beim Entwickeln einer Applikation direkt-manipulativ oder mit Hilfe von Dialogboxen festgelegt. Sie können aber auch über HyperTalk-Anweisungen abgefragt bzw. gesetzt werden.

HyperCard-Elemente kommunizieren miteinander, mit dem Benutzer und mit dem Macintosh über Nachrichten. Zur Spezifikation der Kommunikation ist jedem HyperCard-Objekt ein Skript zugeordnet. Ein Skript ist eine Kollektion von sogenannten Handlern, die jeweils aus einem oder mehreren HyperTalk-Statements bestehen. Es gibt zwei Arten von Handlern. Nachrichtenhandler definieren das Verhalten des Elementes beim Erhalten einer bestimmten Nachricht. Funktionshandler spezifizieren Funktionen, die von anderen Elementen aufgerufen werden können.

Das Versenden von Nachrichten kann verschiedene Ursachen haben. Eine Quelle stellen Benutzerinteraktionen dar, wie das Bewegen der Maus oder Tastatureingaben. Bei bestimmten Systemereignissen, wie dem Öffnen oder Schließen von Karten werden ebenfalls Nachrichten erzeugt. Eine dritte Form der Nachrichtengenerierung bildet die Ausführung der Handler-Aktionen. Ihre Statements werden in Form von Nachrichten versendet.

3.2 Director

Macromedia Director, das zur Zeit in der Version 6.0 vorliegt, ist das am weitesten verbreitete Multimedia-Autorensystem. Es ist sowohl für Macintosh als auch für PC (Windows95, WindowsNT) verfügbar. In Director, das von der Zeitachsen-, Skript- und Objekt-

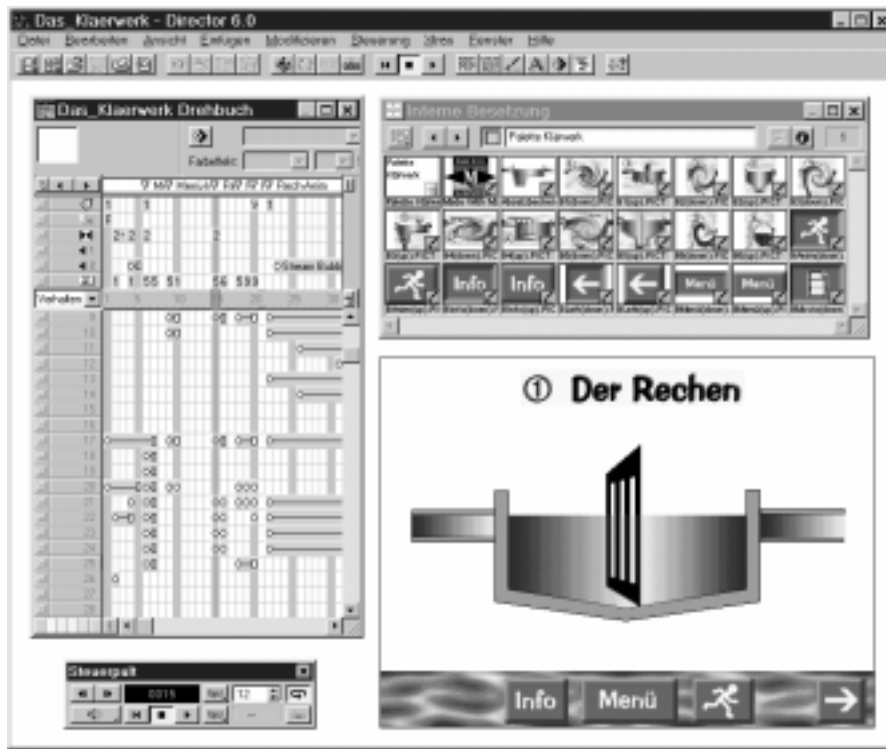


Abbildung 3: Benutzungsoberfläche von Director

Metapher Gebrauch macht, wird die Erstellung einer multimedialen Anwendungen mit der Produktion eines Filmes verglichen. Multimediale Anwendungen werden als komplexe Filme angesehen, die sich aus einer Reihe einfacher Szenen zusammensetzen. Wie im richtigen Film auch, werden diese Szenen als eine Serie von Einzelbildern realisiert. Der Filmeffekt kommt dadurch zustande, daß die Bilder in schneller Folge gezeigt werden. Ein Darsteller ist eine individuelle Graphik, zum Beispiel ein Ball. Wird dieser auf hintereinanderfolgenden Bildern an unterschiedliche Bildschirmpositionen gesetzt, läßt sich ein hüpfender Ball simulieren.

Das sogenannte Studio dient dem Autor, auch Regisseur genannt, zur Regieführung. Das Studio besteht aus mehreren Fenstern, von denen die Bühne, das Steuerpult, die Besetzung und das Drehbuch die wichtigsten sind. Die Bühne ist das Fenster, in dem der Film abläuft. Mit Hilfe des Steuerpults kann die Filmausführung kontrolliert werden. In der Besetzung werden Darsteller gespeichert. Sie repräsentiert damit eine Informationsobjektsammlung. Ein Film besteht aus einer Reihe von Einzelbildern. Bilder setzen sich wiederum im allgemeinen aus einer Menge von Darstellern zusammen. Aus diesem Grund wird im Drehbuchfenster eine Gitterstruktur zur Definition der Ablaufverhalten eines Films verwendet. Jede Zelle repräsentiert einen Darsteller mit zusätzlichen Attributen, wie zum Beispiel die Position auf der Bühne. Eine Spalte konstruiert ein Gesamtbild, stellt also ein komplexes Informationsobjekt dar. Die Zeilen, auch Kanäle genannt, spiegeln den Filmablauf wider.

Lingo ist die Skriptsprache von Director. Sie enthält objektorientierte Konzepte. Wie in HyperTalk können auch in Lingo die Objekten mit Skripten versehen werden. Ein Film-Skript wird dem gesamten Film zugeordnet. Es definiert, was geschieht, wenn ein Film

startet, endet oder pausiert. Den einzelnen Zellen des Drehbuchfensters können Drehbuch-Skripte zugeordnet werden. Sie werden ausgeführt, wenn der Filmablauf die entsprechende Spalte erreicht. Ein Darsteller-Skript gehört zu einem Darsteller. Schließlich gibt es noch Ereignis-Skripte, in denen Reaktionen auf die Betätigung der Maustaste oder der Tastatur spezifiziert werden. Über sogenannte Xtras können mit Lingo externer Code und externe Anwendungen angebunden werden.

3.3 Authorware

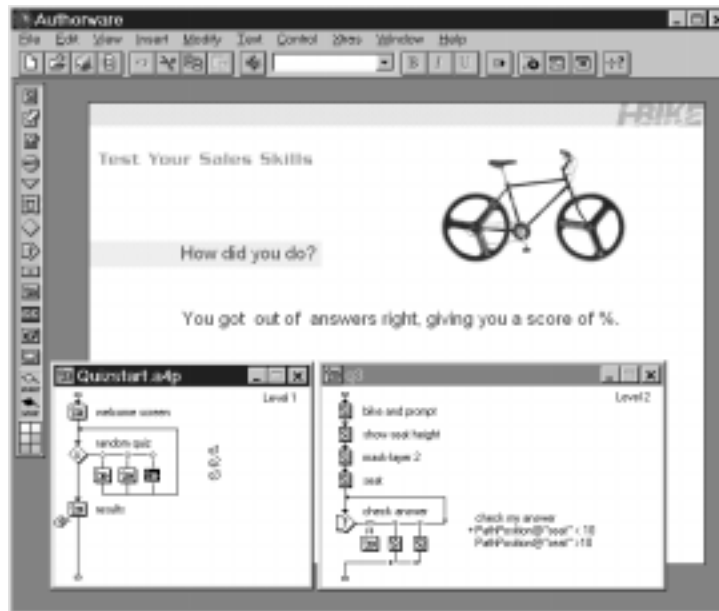


Abbildung 4: Benutzungsoberfläche von Authorware

Authorware Professional, zur Zeit in der Version 4.0 für Macintosh und PC (OS/2, Win3.x, Windows95, WindowsNT) erhältlich, basiert auf der Flußdiagramm-Metapher. Nach dem Start von Authorware erscheint ein Fenster mit einer Leiste von mehreren Ikonen am linken Rand. Jedes Ikon repräsentiert einen bestimmten Typ mit einer bestimmten Bedeutung. Durch Mausklick auf ein Typ-Ikon kann ein Autor Instanzen des jeweiligen Typs erzeugen, die durch ein identisches Ikon im Fenster dargestellt und über Kanten, sogenannte Flußlinien, miteinander verbunden werden können. Die Kanten zeigen den möglichen Präsentationsverlauf der Anwendung an. Eine Teilmenge der Ikonen repräsentiert die unterstützten Medientypen. Durch Anklicken der Instanz-Ikonen öffnet sich ein typspezifischer Editor, in dem das Medienobjekt erzeugt und seine Gestaltungsattribute festgelegt werden können. Ein spezielles Ikon ist für die Strukturierung der Flußdiagramme zuständig. Es ermöglicht die Bildung komplexer Informationsobjekte. Zum Austesten von Programmteilen ist jederzeit ein Interpreter aufrufbar.

An Interaktionsobjekten stehen in Authorware bspw. Buttons, Hotspots, Texteingabefelder, Schieberegler und Pulldown-Menüs zur Verfügung. Sie werden durch ein Interaktions-Ikon repräsentiert. Navigationsinteraktionen lassen sich mit Hilfe von Variablen realisieren. Dazu werden Benutzereingaben in bestimmten Variablen abgelegt. In

Abhängigkeit vom Inhalt einer Variablen lassen sich Bedingungen für Verzweigungen und die Fortsetzung der Anwendung formulieren. Außerdem kann ein Autor eine Menge von Systemfunktionen und mathematischen Funktionen zur Manipulation und Berechnung von Werten zur Steuerung des Präsentationsverlaufs der Anwendung einsetzen.

4 Abschließende Bemerkungen

In diesem Artikel wurden charakteristische Eigenschaften von Multimedia-Autorensystemen analysiert und existierende Systeme exemplarisch vorgestellt. Multimedia-Autorensysteme ermöglichen durch die Bereitstellung graphisch-interaktiver Werkzeuge die Entwicklung interaktiver multimedialer Anwendungen ohne umfangreiche Programmiererfahrung.

Dabei muß jedoch betont werden, daß Autorensysteme nur einen Teil des gesamten Entwicklungsprozesses multimedialer Anwendungen unterstützen. Wie bei der gewöhnlichen Softwareentwicklung werden auch bei der Multimedia-Softwareentwicklung mehrere Phasen durchlaufen. In der Analysephase werden insbesondere funktionale, operative, softwareergonomische, designerische, didaktische und psychologische Anforderungen an die zu entwickelnde Anwendung ermittelt. In der Entwurfsphase werden die Anforderungen dann in ein Gesamtkonzept umgesetzt, das in der Implementierungsphase in eine maschinenverständliche Form gebracht wird. Die Testphase dient der Überprüfung der Korrektheit der Implementierung und der Erfüllung der gestellten Anforderungen. Autorensysteme unterstützen hier lediglich die Implementierungsphase sowie in Ausschnitten die Entwurfs- und Testphase. Konfigurationsmanagement- bzw. Versionsmanagement-Systeme sind im allgemeinen nicht integriert, kooperative Arbeitsformen werden ebenfalls nicht unterstützt. Autorensysteme stellen also keine vollständigen CASE-Tools (Computer-Aided-Software-Engineering) dar. Insbesondere liegt den Autorensystemen keine vollständige Entwicklungsmethodik zugrunde. Im Gegensatz zu anderen Softwarebranchen scheint eine allgemeingültige Methodik bei der Entwicklung interaktiver multimedialer Anwendungen derzeit auch noch in weiter Ferne zu liegen. Zur Zeit müssen sich Autoren mit vagen Richtlinien und Vorschlägen zufrieden geben, existierende Anwendungen evaluieren oder sich einfach auf ihre Intuition verlassen.

Neben den klassischen Multimedia-Autorensystemen sind in den letzten Jahren eine Reihe von sogenannten Web-Autorensystemen auf den Markt gekommen. Web-Autorensysteme unterscheiden sich von Multimedia-Autorensystemen dadurch, daß sie primär für die Erstellung einzelner bzw. zusammenhängender WWW-Seiten geeignet sind. Sie ermöglichen nicht die Erstellung dynamischer, hochgradig interaktiver Multimedia-Anwendungen. Im Prinzip handelt es sich bei den Web-Autorensystemen meist um graphisch-interaktive HTML-Editoren.

Während die klassischen Multimedia-Autorensysteme bis vor wenigen Jahren ausschließlich die Entwicklung von Offline-Produkten ermöglichten, wird in neueren Versionen häufig eine Anbindung ans Internet unterstützt. Hier reichen die Alternativen von dem einfachen Start eines WWW-Browsers über die Generierung von HTML- und Java-Code, der Bereitstellung eigener Browser-Plug-ins bis hin zur Entwicklung hybrider Multimedia-Anwendungen, die zwar über CD-ROM distribuiert werden, aber zur Laufzeit ans Internet gekoppelt sind. Eine zur Zeit spannende Frage sowohl für Autorensystem-Entwickler

als auch -Nutzer ist, welche Rolle Java in der Zukunft im Authoring-Bereich einnehmen wird.

5 Weiterführende Informationen

Detaillierte Beschreibungen spezieller Autorensysteme sind den gängigen Informatik- bzw. Multimedia-Fachzeitschriften zu entnehmen. Dort finden sich häufig auch Vergleiche und Übersichten, welche Autorensysteme welche Anforderungen erfüllen. Nachfolgend einige empfehlenswerte Beispiele:

- Screen Multimedia, Heft 11/97: Authoring-Konzepte
- Screen Multimedia, Heft 11/97: Marktübersicht Autorenprogramme
- CT, Heft 3/96: Vergleich Authorware, Director, ToolBook, Delphi und Visual Basic
- CT, Heft 1/96: Vergleich Authorware und ToolBook im Hinblick auf CBT-Softwareentwicklung
- CBT ComputerBased Training (HighText-Verlag), Heft 1/97: Marktübersicht Autorensysteme für die Multimedia- und CBT-Entwicklung

Auch im WWW existieren diverse Seiten zum Thema Multimedia-Autorensysteme:

- <http://www.tiac.net/users/jasiglar/MMASFAQ.HTML>
Multimedia Authoring Systems FAQ: Einführung in Autorensysteme und Zusammenstellung dutzender von Autorensystemen mit Links zu den jeweiligen Herstellern bzw. Distributoren.
- <http://www.mcli.dist.maricopa.edu/authoring/>
Multimedia Authoring Web: Suchbarer Meta-Server
- <http://www.phil.uni-sb.de/FR/Infowiss/topics/autsyst.html>
Autorensystem Test: Beschreibung und Bewertung einzelner Autorensysteme
- http://www.man.ac.uk/MVC//SIMA/author_systems.html
Authoring Systems: Zusammenstellung von Informationen über Autorensysteme

Zusammenstellungen von Links zu allgemeinen Informationen im Bereich Multimedia finden sich bspw. unter:

- <http://www-is.informatik.uni-oldenburg.de/~dibo/teaching/mm/links/>
- <http://viswiz.gmd.de/MultimediaInfo/>
- <http://www.yahoo.com/Computers/Multimedia>