

Universität Kaiserslautern  
Fachbereich Elektrotechnik  
Lehrstuhl für Digitale Systeme  
Prof. Dr.-Ing. S. Wendt

# Diplomarbeit

Spezifikation und Entwurf der  
Navigationskomponente für ein “Corporate  
Memory” und Implementierung eines  
Demonstrationsprototyps

Christian Brand  
August 1997

Betreuer: Prof. Dr.-Ing. S. Wendt

Bearbeiter: Christian Brand  
Carl-Zimmermann-Straße 32  
67373 Dudenhofen



**Erklärung:**

Hiermit erkläre ich, die vorliegende Diplomarbeit unter Verwendung der angegebenen Quellen und ohne fremde Hilfe angefertigt zu haben.

Walldorf, den 29. August 1997

Christian Brand



# Danksagung

An dieser Stelle möchte ich allen danken, die zum Gelingen dieser Arbeit beigetragen haben.

Mein Dank gilt in erster Linie Herrn Prof. Dr.-Ing. Siegfried Wendt, der die Diplomarbeit im Rahmen seiner Forschungsaktivitäten bei der Firma SAP AG ermöglichte und betreute. Sein mir entgegengebrachtes Vertrauen und der mir zugestandene Entscheidungsfreiraum waren außerordentlich groß und Basis einer sehr angenehmen Arbeitsatmosphäre.

Die vorliegende Arbeit ist Teil eines Projekts, das im Januar 1997 ins Leben gerufen wurde. Mein Dank gilt meinen Kollegen Markus Cherdron, Jürgen Langhauer und Stefan Steißlinger, sowie den Mitarbeitern von Herrn Prof. Dr.-Ing. Siegfried Wendt Bernhard Gröne, Eberhard Iglhaut, Andreas Knöpfel und Johannes Otto. Ihnen verdanke ich viele wertvolle Anregungen und Ideen. Ihre stete Bereitschaft zu Diskussionen, die zur Klärung vieler Sachverhalte führten, war mir sehr hilfreich.

Christian Brand



# Inhaltsverzeichnis

<b>Kapitel 1: Einleitung und Aufgabenstellung</b>	<b>1</b>
<b>Kapitel 2: Die Ablagestruktur</b>	<b>5</b>
2.1 Bereits vorhandene Ablagestrukturen .....	5
2.1.1 Das Filesystem .....	5
2.1.2 Die reale Bibliothek .....	6
2.1.3 Ein Vergleich beider Ablagesysteme .....	7
2.2 Die öffentliche Ablagestruktur (Bibliothekswelt) .....	8
2.3 Die private Ablagestruktur (Bürowelt) .....	11
<b>Kapitel 3: Die Navigationskomponente</b>	<b>13</b>
3.1 Forderungen .....	13
3.2 Die Bildschirmdarstellung .....	13
3.2.1 Das View-Fenster .....	14
3.2.2 Das Pfad-Fenster .....	21
3.2.3 Das Informations-Fenster .....	28
3.2.4 Das Fenster für Bedienelemente .....	31
3.3 Erste Erkenntnisse aus dem Prototyp .....	33
<b>Kapitel 4: Die Katalogkomponente</b>	<b>35</b>
4.1 Das Seitenlayout .....	36
4.2 Die Erstellung eines Katalogs .....	37
4.2.1 Ein Beispiel .....	37
4.2.2 Die Generierungsprozedur .....	38

<b>Kapitel 5: Vorüberlegungen zum Finden einer geeigneten Systemarchitektur</b>	<b>43</b>
5.1 Der Aufbau des Gesamtsystems . . . . .	43
5.2 Die Anforderungen der Tool-Bus-Schnittstelle . . . . .	44
5.3 Der erste Entwurf einer Architektur des Gesamtsystems . . . . .	45
5.3.1 Der Aufbau des Gesamtsystems . . . . .	45
5.3.2 Das Zusammenwirken der Komponenten des Gesamtsystems . . . . .	47
5.4 Anmerkung . . . . .	50
<b>Anhang A: Benutzerrollen und Werkzeuge eines Corporate Memory</b>	<b>53</b>
A.1 Die Benutzerrollen . . . . .	53
A.2 Das Datenbanksystem. . . . .	54
A.3 Die Schnittstelle zwischen Werkzeugen und Datenbanksystem . . . . .	54
A.4 Die Werkzeuge. . . . .	55
A.5 Beispiele für unterschiedliche Aufgaben und Tätigkeiten von Benutzern. . . . .	56
<b>Anhang B: Bemerkungen zur Erstellung des Prototyps</b>	<b>57</b>
B.1 Ein interaktiver Film als Prototyp . . . . .	57
B.2 Die eingesetzten Produkte . . . . .	57
B.3 Grundbegriffe und -konzepte von <i>Director</i> . . . . .	58
<b>Anhang C: Der Aufbau des Prototyps</b>	<b>61</b>
C.1 Die Besetzungen . . . . .	61
C.2 Die Kanalbelegung . . . . .	63
<b>Anhang D: Der Ablauf des interaktiven Films</b>	<b>65</b>
D.1 Die Views . . . . .	65
D.2 Die Navigation zwischen Views benachbarter Hierarchieebenen . . . . .	66
D.3 Das Überspringen einer Hierarchieebene nach unten . . . . .	66
D.4 Das Verschieben des angezeigten Pfad-Ausschnitts . . . . .	69
D.5 Das Überspringen von Hierarchieebenen nach oben . . . . .	69
D.6 Die Anzeige im Informations-Fenster . . . . .	71
<b>Literaturverzeichnis</b>	<b>75</b>



# Abbildungsverzeichnis

Bild 1	Beispielbaum, wie er mit Hilfe eines Filesystems möglich ist . . . . .	6
Bild 2	Allgemeiner Aufbau einer Baumstruktur mit Hilfe eines Filesystems . . . . .	6
Bild 3	Neue – für Menschen anschauliche und vertraute – Ablagestruktur . . . . .	10
Bild 4	Aufbau der privaten Ablagestruktur . . . . .	12
Bild 5	Bildschirm-Layout der Navigationskomponente. . . . .	14
Bild 6	Land-View . . . . .	15
Bild 7	Campus-View. . . . .	16
Bild 8	Gebäude-View . . . . .	17
Bild 9	Etagen-View. . . . .	17
Bild 10	Saal-View. . . . .	18
Bild 11	Regal-View. . . . .	18
Bild 12	Brett-View . . . . .	19
Bild 13	Navigationsmöglichkeiten mittels View-Fenster . . . . .	20
Bild 14	Gebäude-View (transparent) . . . . .	20
Bild 15	Saal-View (Regal oben) . . . . .	22
Bild 16	Erweiterte Navigationsmöglichkeiten mit Hilfe des View-Fensters . . . . .	22
Bild 17	Beispiel für ein gefülltes Pfad-Fenster. . . . .	23
Bild 18	Navigationsmöglichkeiten mit View- und Pfad-Fenster. . . . .	25
Bild 19	Animation nach Anwählen eines Containers im View-Fenster . . . . .	26
Bild 20	Animation nach Anwählen eines Eintrags im Pfad-Fenster . . . . .	27
Bild 21	Beispiel für den Inhalt eines Informations-Fensters . . . . .	29

---

Bild 22	Fenster für Bedienelemente . . . . .	31
Bild 23	Layout einer Katalogseite . . . . .	36
Bild 24	Beispiel eines Strukturbaums. . . . .	37
Bild 25	Seiten des generierten Beispielkatalogs. . . . .	37
Bild 26	Ablauf zur Generierung eines Katalogs (rekursive Lösung) . . . . .	39
Bild 27	Teilschritte beim Generieren des Beispielkatalogs . . . . .	42
Bild 28	Grobe Darstellung eines Gesamtsystems mit integrierten Tools . . . . .	44
Bild 29	Architektur des Gesamtsystems. . . . .	46
Bild 30	Ablauf bei einer Benutzer-Anmeldung. . . . .	48
Bild 31	Benutzerrollen und Werkzeuge eines Corporate Memory. . . . .	53
Bild 32	Ausschnitt aus einem Drehbuch, wie es in <i>Director</i> angezeigt wird. . . . .	59
Bild 33	Navigation zwischen Views benachbarter Hierarchieebenen. . . . .	67
Bild 34	Überspringen einer Hierarchieebene nach unten . . . . .	68
Bild 35	Verschieben des angezeigten Pfad-Ausschnitts . . . . .	70
Bild 36	Überspringen von Hierarchieebene nach oben . . . . .	72
Bild 37	Anzeige im Informations-Fenster . . . . .	73

# Kapitel 1: Einleitung und Aufgabenstellung

Ein Softwarehaus kann im Gegensatz zu Firmen anderer Branchen als ein ausschließlich Information produzierendes Unternehmen angesehen werden. Nur ein Teil dieser Information ist in dem an den Kunden ausgelieferten Produkt – der Software – enthalten. Der andere Teil der Information, der bei der Entwicklung und der Konstruktion von Software produziert wird, verbleibt als Wissen in der Firma.

Häufig ist ein Großteil dieses Wissens nur in den Köpfen der Entwickler abgelegt, so daß für viele Mitarbeiter das in der Firma reichlich vorhandene Wissen schwer oder gar nicht erreichbar ist. Das kann zum einen daran liegen, daß nicht bekannt ist, wer über das gesuchte Wissen verfügt, andererseits kommt es auch vor, daß ein Entwickler seine Erfahrungen und Kenntnisse nicht preisgeben will, um sein "Wissensmonopol" nicht zu verlieren. Es ist außerdem möglich, daß ein Software-Entwickler zwar das Know-how über einen komplexen Sachverhalt besitzt, er dieses auch mit anderen teilen möchte, aber nicht in der Lage ist, einem Kollegen, der nur über begrenzte Vorkenntnisse verfügt, dieses Know-how zu vermitteln. Dazu fehlt häufig die Fähigkeit, die Komplexität durch geeignete Abstraktion zu reduzieren.

Die aus diesen Gründen beschränkte Verteilung der Information zieht Probleme und Gefahren für ein Softwareentwicklungsunternehmen nach sich:

- Viele Probleme werden mehrfach gelöst, da vorhandene Lösungen nicht bekannt gemacht werden.
- Fehler werden mehrfach begangen, da sie weder bekannt gemacht noch als Warnungen formuliert werden.
- Das Ausscheiden eines Mitarbeiters bedeutet für die Firma gleichzeitig den Verlust an wichtiger Information.
- Aufwendig zu lösende Aufgaben, für die eine große Zahl von Mannjahren nötig ist, können nicht durch den Einsatz vieler Mitarbeiter beschleunigt werden, da die Kommunikation der Mitarbeiter untereinander ineffizient ist.

Dieser Mißstand kostet ein Softwarehaus sehr viel Zeit und somit auch sehr viel Geld. Außerdem birgt der Verlust mehrerer Mitarbeiter das Risiko des Niedergangs eines Unternehmens.

Zur Beseitigung dieser Probleme sind folgende Änderungen anzustreben:

1. Es sollte zur Alltagsarbeit eines Entwicklers gehören, Informationen – d.h. Kenntnisse und angeeignetes Wissen – in Form von Dokumenten festzuhalten und diese abzulegen.
2. Die so entstandenen Pläne sowie ergänzende Dokumente, wie z.B. Texte, Filme oder Tonaufnahmen, sollten zentral und für alle Mitarbeiter leicht zugänglich abgelegt und verwaltet werden.
3. Jeder Entwickler sollte sich wenn nicht das Erstellen, so doch zumindest das Lesen von Modellplänen aneignen. Diese Pläne sollten die zwischenmenschliche Kommunikation unterstützen, d.h. sie sollten auch für Mitarbeiter verständlich sein, die weniger Vorkenntnisse mitbringen als der Entwickler selbst.

Diese Änderungsempfehlungen bauen aufeinander auf, d.h. jede Empfehlung ist Voraussetzung für ihren Nachfolger und wenig sinnvoll ohne Erfüllung ihres Vorgängers.

Die vorliegende Arbeit entstand als Teil eines Projekts, das als Ziel das Schaffen von Voraussetzungen zur Umsetzung der zweiten oben genannten Änderungsempfehlung hat. Innerhalb des Projekts soll ein System entwickelt werden, das die Logistik von Information innerhalb eines Unternehmens unterstützt. Ein solches System wird als **Corporate Memory** bezeichnet.

Als vorläufiges Ziel wurde die Entwicklung eines Dokumentenverwaltungssystems – eines Systems zur Ablage von Dokumenten – formuliert. Ein Dokumentenverwaltungssystem kann als Corporate Memory eingesetzt werden, indem Dokumente als Träger von Information benutzt werden.

Im Rahmen dieser Diplomarbeit wurden folgende Aufgaben bearbeitet:

- Die Entwicklung einer geeigneten Ablagestruktur zur Archivierung von Dokumenten nach den oben genannten Forderungen.
- Die Beschäftigung mit der Frage, wie ein Benutzer die Ablagestruktur sehen möchte.
- Das Entwickeln eines Konzepts zur Navigation in der Ablagestruktur, ohne daß sich ein Benutzer spezielles Bedienwissen aneignen muß.
- Das Erstellen eines Prototyps der Navigationskomponente.
- Das Zusammentragen von Überlegungen zur Konstruktion eines Corporate Memory mit integrierter Navigationskomponente. Dazu gehören der Entwurf und die Spezifikation von Schnittstellen sowohl zwischen Benutzer und programmiertem System, als auch zwischen den Komponenten des programmierten Systems. Das bedeutet das Finden und Festlegen einer geeigneten Architektur.



# Kapitel 2: Die Ablagestruktur

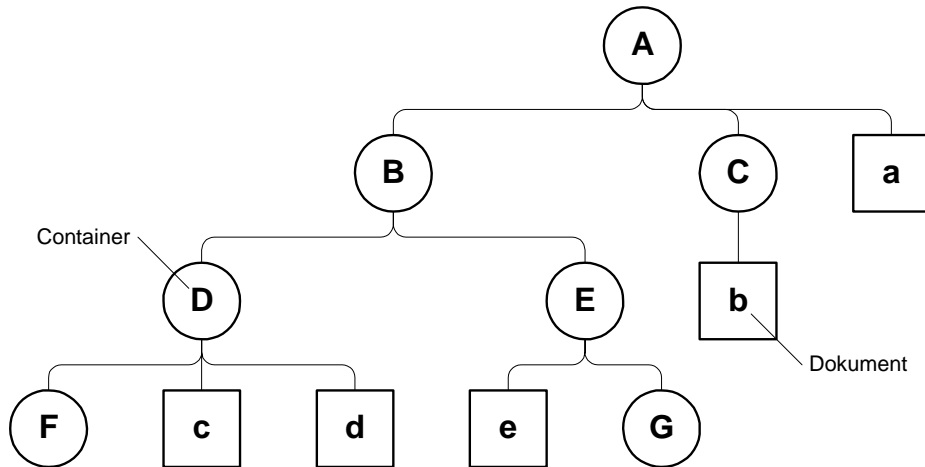
## 2.1 Bereits vorhandene Ablagestrukturen

Dieses Kapitel beschreibt das Ergebnis der Suche nach einer geeigneten Struktur zur Ablage von Dokumenten. Bevor man damit beginnt, neue Konzepte zu entwickeln, ist es sinnvoll, bereits vorhandene und im Einsatz befindliche Ablagestrukturen genauer zu untersuchen. Hier wurden zwei zu unterschiedlichen Zwecken eingesetzte Ablagesysteme betrachtet.

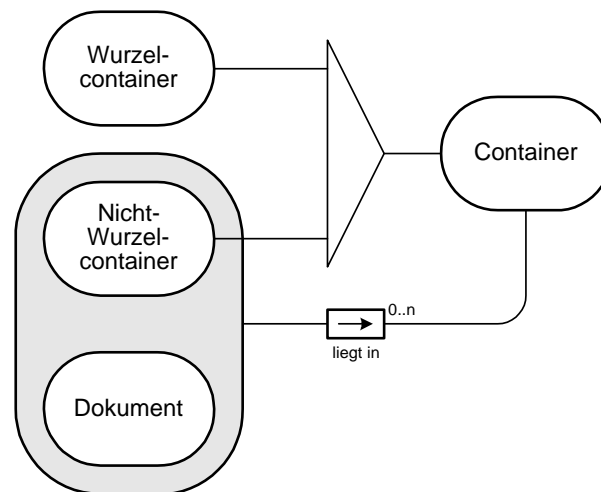
### 2.1.1 Das Filesystem

Bereits heute werden Dokumente der elektronischen Datenverarbeitung in Strukturen abgelegt. Gängige Filesysteme bieten die Strukturierungsmöglichkeit in Form eines Baumes an. Zu diesem Zweck gibt es Container – auch als Ordner oder Behälter bezeichnet – die sowohl Dokumente als auch weitere Container aufnehmen können. Bild 1 zeigt eine Beispielstruktur eines solchen Baumes. Dabei sind Container als Kreise – beschriftet mit Großbuchstaben – und Dokumente als Quadrate – beschriftet mit Kleinbuchstaben – dargestellt. Die Wurzel eines solchen Baumes ist immer ein Container (A), während die Blätter sowohl Dokumente als auch Container sein können. Im Gegensatz zum Dokument, das immer Blatt ist (a bis e), kommt ein Container nur dann als Blatt vor, wenn er leer ist (F und G), d.h. wenn er weder ein Dokument noch einen Container enthält.

Wie eine solche Baumstruktur allgemein aufgebaut ist, zeigt das E/R-Diagramm in Bild 2. Die beiden darin vorkommenden Entitätstypen sind Dokument und Container. Dabei gibt es in jedem auf diese Weise aufgebauten Strukturbaum einen ausgezeichneten Container, der Wurzel dieses Baumes ist. Das ist der einzige, der in keinem anderen Container enthalten ist und wird im Gegensatz zu allen anderen als Wurzelcontainer bezeichnet. Alle Nicht-Wurzelcontainer und Dokumente sind in einem Container, der Wurzel- oder Nicht-Wurzelcontainer sein kann, enthalten.



**Bild 1:** Beispielbaum, wie er mit Hilfe eines Filesystems möglich ist



**Bild 2:** Allgemeiner Aufbau einer Baumstruktur mit Hilfe eines Filesystems

## 2.1.2 Die reale Bibliothek

Ein zweites sehr bekanntes Ablagesystem stellt eine reale Bibliothek dar. Mit dem Begriff Bibliothek verbindet man im allgemeinen ein Gebäude mit einem oder mehreren Stockwerken. In einem Stockwerk befinden sich Säle, die mehrere Regale enthalten können. Ein Regal ist meist aus mehreren Regalbrettern aufgebaut, auf denen Dokumente und Ordner durch Buchstützen zu Mengen zusammengefaßt sind. Ordner dienen dabei der Ablage von Kleindokumenten, die aus nur wenigen Seiten bestehen. Innerhalb eines Ordners dienen Heftklammern und Trennblätter der Gruppierung von Dokumenten.



Eine Bibliothek besteht nicht notwendigerweise aus einem einzigen Gebäude. Man denke dabei zum Beispiel an den Verbund von Universitätsbibliotheken innerhalb eines Landes. Solch ein Verbund besteht aus mehreren Universitätscampi, d.h. aus Flächen, die auf verschiedene Städte verteilt sind. Jeder Campus kann mehrere Gebäude enthalten.

### 2.1.3 Ein Vergleich beider Ablagesysteme

Der folgende Vergleich der beiden betrachteten Ablagesysteme soll als Untersuchung auf Verwendbarkeit der darin enthaltenen Konzepte dienen:

- Sowohl ein Filesystem – im folgenden auch als elektronisches Lager bezeichnet – als auch eine reale Bibliothek besitzen die Struktur eines Baumes.
- Während in einem elektronischen Lager Strukturänderungen durch Erzeugen, Kopieren und Löschen von Dokumenten und Containern sehr einfach möglich sind, gestalten sich diese in einer realen Bibliothek, die ausschließlich aus materiell-energetischen Elementen aufgebaut ist, als schwierig. Dies ist jedoch nicht nur als Nachteil der realen Welt anzusehen, denn eine verminderte Dynamik einer Ablagestruktur vereinfacht das Wiederauffinden von Dokumenten ganz erheblich. Diese Dynamik kann jedoch auch bei einem Filesystem durch explizites Einführen entsprechender Regeln reduziert werden.
- Benutzer<sup>1</sup> einer realen Bibliothek können drei unterschiedliche Rollen spielen. In der Rolle des **Autors** ist ein Benutzer dafür zuständig, daß Information in Form von Dokumenten in die Bibliothek gelangt. Ein **Bibliothekar** hat die Aufgabe, Dokumente in der Bibliothek strukturiert abzulegen, um eine leichte Auffindbarkeit zu ermöglichen. Ein in der Rolle des **Lesers** befindlicher Benutzer verwendet eine Bibliothek um an gesuchte Information zu gelangen. Im Gegensatz dazu werden beim Entwurf eines Filesystems keine Benutzerrollen festgelegt. Dennoch können meist durch Nutzung eines integrierten Berechtigungswesens unterschiedliche Rollen generiert werden.
- In einer Bibliothek gibt es viele Regeln zu beachten. Dazu gehören Kapazitätsbeschränkungen, die beispielsweise bei einem Regalbrett durch dessen Breite festgelegt ist. Somit kann ein Regalbrett auch nur eine begrenzte Menge an Information beinhalten. Eine weitere Einschränkung besteht darin, daß Dokumente nicht direkt in einem Gebäude, in einer Etage oder einem Saal abgelegt werden, sondern entweder auf einem Regalbrett zwischen zwei Buchstützen oder in einem Ordner. Diese beiden Einschränkungen gelten nicht zwingend für ein elektronisches Lager, können jedoch explizit festgelegt werden.
- Innerhalb einer Bibliothek denkt man in Orten, zwischen denen man sich bewegen kann. Durch die individuelle Gestaltung der materiell-energetischen

---

1. An dieser Stelle soll unter einem Benutzer einer Bibliothek ein Mensch verstanden werden, der am Betrieb einer Bibliothek Anteil hat.

Elemente in der Bibliothek wird das Wiederauffinden bekannter Orte sehr gut unterstützt. Außerdem nimmt ein Benutzer beim Bewegen durch die Bibliothek nicht nur die Orte wahr, die er gezielt gesucht hat, sondern auch diejenigen, die er bei seiner Suche passiert. Dieser Nebeneffekt des unbewußten Kennenlernens der Bibliothek kann als weiterer Vorteil des realen Bibliothekssystems angeführt werden. Dies sind entscheidende Vorteile der realen Bibliothek gegenüber der künstlichen und unnatürlichen Welt, wie sie das Filesystem darstellt.

- Die praktische Nutzung einer realen Bibliothek beschränkt sich auf gebundene Bücher und Zeitschriften, d.h. auf statische Dokumente. Für alle Dokumente, die nicht statisch sind, werden zur Darstellung spezielle Werkzeuge, wie Projektoren oder Abspielgeräte, benötigt, die das Anzeigen von gespeicherter Information erschweren. Diesen Nachteil hat ein elektronisches Lager nicht. Dem Wunsch eines Benutzers, ein Dokument sehen zu wollen, wird bei komfortablen Systemen dadurch nachgegangen, daß das benötigte Werkzeug automatisch zur Verfügung gestellt wird.
- Möchten mehrere Benutzer zur selben Zeit die gleichen Dokumente betrachten, so ist dies in der realen Bibliothek nur möglich, wenn für jeden dieser Benutzer ein Exemplar zur Verfügung steht. Das ist in einem elektronischen Lager nicht notwendig, da hier ein paralleler Zugriff auf ein und dasselbe Dokument möglich ist.

Der Vergleich der beiden Ablagestrukturen macht deutlich, daß beide Vor- und Nachteile aufweisen. Wünschenswert wäre eine Ablagestruktur, die einem Benutzer eine Sicht auf eine Bibliothek bietet, und die die Vorzüge der elektronischen Lagerhaltung nutzt.

## 2.2 Die öffentliche Ablagestruktur (Bibliothekswelt)

Der Weg zur Erfüllung dieses Wunsches ist die Abbildung der dem Menschen vertrauten Bibliothekswelt auf ein Filesystem. Dabei entsteht eine neue Ablagestruktur, die in der Begriffswelt des Bibliothekswesens verankert ist und hier als **öffentliche Ablagestruktur** oder als **Bibliothekswelt** bezeichnet werden soll. In der öffentlichen Ablagestruktur finden sich mehrere unterschiedliche Containertypen, die insbesondere durch folgende Attribute spezifiziert sind:

- die Vorschrift, Elemente welchen Typs in einem Container enthalten sein dürfen
- die Kapazität eines Containers
- die Art der Präsentation eines Containers

Die unterschiedlichen Containertypen sind im E/R-Diagramm in Bild 3 neben dem Dokument als Entitätstypen dargestellt. Bei allen hier gezeigten Relationen handelt es sich um Enthaltenseinsrelationen. Der Wurzelcontainer aus Bild 2 ist jetzt ein Container vom Typ Land. Alle anderen Container sind Nicht-Wurzelcontainer. Dazu gehören Campi, die im Land verteilt sind und mehrere Gebäude enthalten. Ein Gebäude ist aus mehreren Etagen aufgebaut. Eine Etage enthält Säle, die mit mehreren Regalen gefüllt sind. Ein Regal ist in Bretter, und diese wiederum in Sektionen unterteilt, wobei eine Sektion eine durch Buchstützen begrenzte Gruppe von Ordnern und Dokumenten ist.

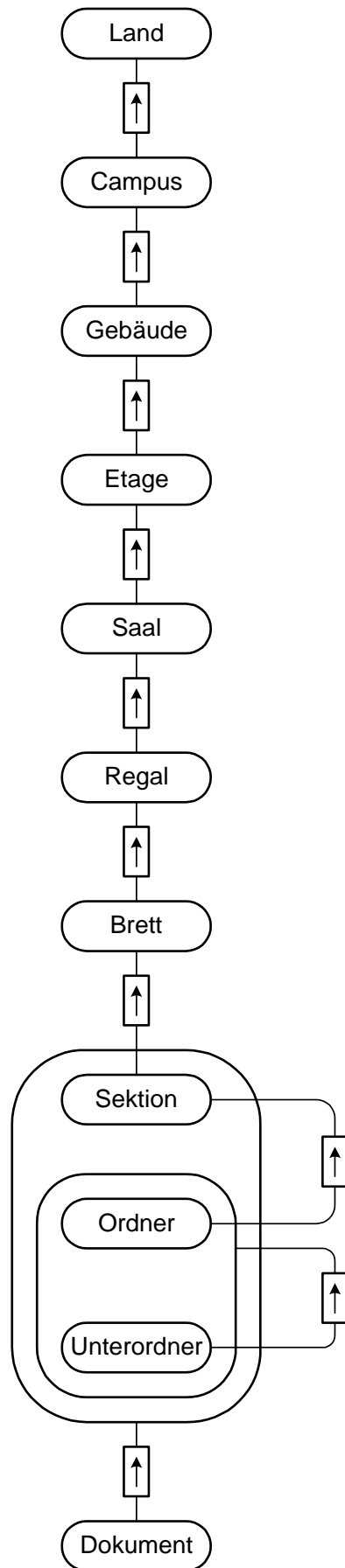
Bei dieser Struktur fällt auf, daß es Container gibt, deren Typ es von der Anschauung her verbietet, daß sie Dokumente unmittelbar<sup>1</sup> enthalten können. Dazu gehören alle Container der Typen Land, Campus, Gebäude, Etage, Saal, Regal und Brett, die ausschließlich zur ortsbezogenen Strukturierung der Ablage, jedoch nicht zur Aufnahme von Dokumenten dienen. Diese Restriktionen wurden nach Betrachtung der realen Bibliothek festgelegt, in der es beispielsweise nicht vorkommt, daß ein Buch unmittelbar in einem Gebäude liegt.

Im gewählten Modell enthält ein Brett nur deshalb keine Dokumente, weil diese durch Zusammenfassung zu Sektionen gruppiert werden sollen. In Bild 3 erkennt man, daß Dokumente sowohl in Sektionen als auch in Ordnern und Unterordnern enthalten sein können. Während man sich einen Ordner als Mappe mit Ringmechanik oder als Stehsammler für Zeitschriften und Magazine vorstellen kann, dienen Unterordner lediglich zur Strukturverfeinerung bei der Dokumentenablage. Unterordner sollen dabei die in der Realität vorhandenen Gruppierungsmöglichkeiten von Dokumenten innerhalb eines Ordners ersetzen. Solche Gruppierungen werden in der Realität beispielsweise durch Trennblätter, Gummiringe, Heft- oder Büroklammern ermöglicht. Mit Hilfe von Unterordnern ist eine Strukturierung der Ablage möglich, deren Tiefe nur durch die Kapazität eines Ordners begrenzt ist.

Alle Container haben eine begrenzte Kapazität. Durch die typspezifische Präsentation der Container kann ein Benutzer, der zu einem gesuchten Themenbereich zum Beispiel ein halb gefülltes Regalbrett vorfindet, abschätzen, welchen Aufwand er betreiben, d.h. wieviel Zeit er investieren muß, um die gesamte vorhandene Information zu erfassen. Solch ein Abschätzen ist deshalb möglich, weil bei der Festlegung der Kapazitätsgrenzen Werte aus der realen Welt benutzt wurden. So ist beispielsweise durch die Breite eines Regalbretts von einem Meter dessen maximal möglicher Informationsinhalt festgelegt. Mit Hilfe von Erfahrungen aus

---

1. Im Gegensatz zur Enthaltenseinsrelation, bei der es sich um eine transitive Relation handelt, ist für die Relation "enthält unmittelbar" die Transitivität nicht gegeben. Dennoch soll im folgenden Text – auch ohne die Ergänzung durch den Begriff "unmittelbar" – die Enthaltenseinsrelation als nicht-transitiv angesehen werden.



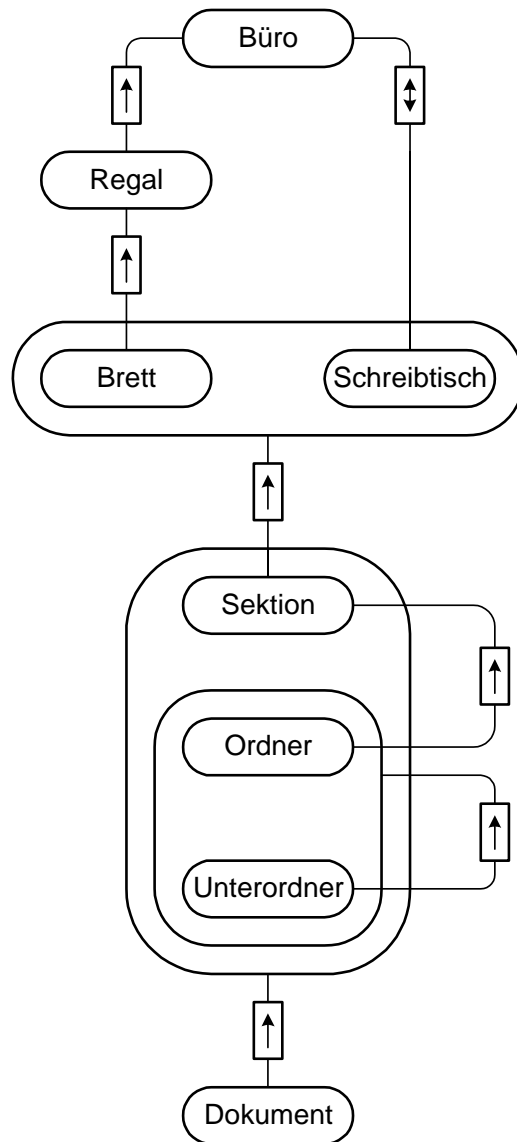
**Bild 3:** Neue – für Menschen anschauliche und vertraute – Ablagestruktur

der realen Welt, die jeder Benutzer besitzt, ist ohne eine explizite Unterrichtung bekannt, daß zum Beispiel in einem Gebäude mehr Information untergebracht werden kann als auf einem Regalbrett.

## 2.3 Die private Ablagestruktur (Bürowelt)

Für jeden Benutzer wird es eine Auswahl von Dokumenten geben, die er häufig benötigt. Da es sehr unwahrscheinlich ist, daß alle diese Dokumente in unmittelbarer Nachbarschaft abgelegt sind, wäre ein Benutzer gezwungen, viele Ortswechsel in der Bibliothek durchzuführen, um an bereits bekannte Dokumente zu gelangen. Um dies vermeiden zu können, ist für jeden Benutzer ein privater Ablagebereich vorhanden, zu dem ausschließlich er Zugang besitzt. Dieser Bereich wird durch die **private Ablagestruktur** dargestellt. In dieser seiner **Bürowelt** hat ein Benutzer die Möglichkeit, sich einen eigenen Handapparat aus häufig benötigten Dokumenten zusammenzustellen, und diesen nach seinen Wünschen und Vorstellungen zu verwalten. Dafür stehen ihm mehrere Regale zur Verfügung.

Die Ablagestruktur innerhalb eines Büros ist durch das E/R-Diagramm in Bild 4 dargestellt. Hier erkennt man, daß ein Büro neben Regalen auch genau einen Schreibtisch enthält, wodurch es sich von einem Saal der öffentlichen Ablagestruktur unterscheidet. Ein Schreibtisch kann, ebenso wie ein Regalbrett, mehrere Sektionen enthalten. Diese werden durch eine Arbeitsunterlage oder durch Buchstützenpaare auf der Oberfläche sowie durch Schubladen des Schreibtischs dargestellt. Dabei können mehrere auf der Arbeitsunterlage abgelegte Dokumente geöffnet, d.h. parallel gelesen werden. Die Strukturierung innerhalb einer Sektion entspricht der aus der Bibliothekswelt bekannten, so daß dem Benutzer ein Umdenken erspart bleibt.



**Bild 4:** Aufbau der privaten Ablagestruktur

# Kapitel 3: Die Navigationskomponente

## 3.1 Forderungen

An die zu entwickelnde **Navigationskomponente** sind mehrere Forderungen gestellt. Zum einen soll durch eine räumliche Darstellung der unterschiedlichen Container der Ablagestruktur beim Benutzer das Denken in Orten unterstützt werden – aus diesem Grund spricht man auch von **Raumnavigation** – zum anderen muß eine komfortable Navigation innerhalb der Ablagestruktur möglich sein. Darüber hinaus wird großen Wert auf die Bedienbarkeit gelegt, die so einfach sein sollte, daß ein Benutzer nach einer etwa fünfminütigen Einführung in der Lage ist, sich selbständig sowohl in der Bibliotheks- als auch in der Bürowelt zurechtzufinden.

## 3.2 Die Bildschirmdarstellung

Dieser Abschnitt beschreibt, wie eine Navigationskomponente aussehen könnte. Die dargestellten Bildschirmfotos sind Beispiele aus dem im Rahmen dieser Arbeit entwickelten Prototyps<sup>1</sup>. Auch der beschreibende Text bezieht sich auf diese Entwicklung.

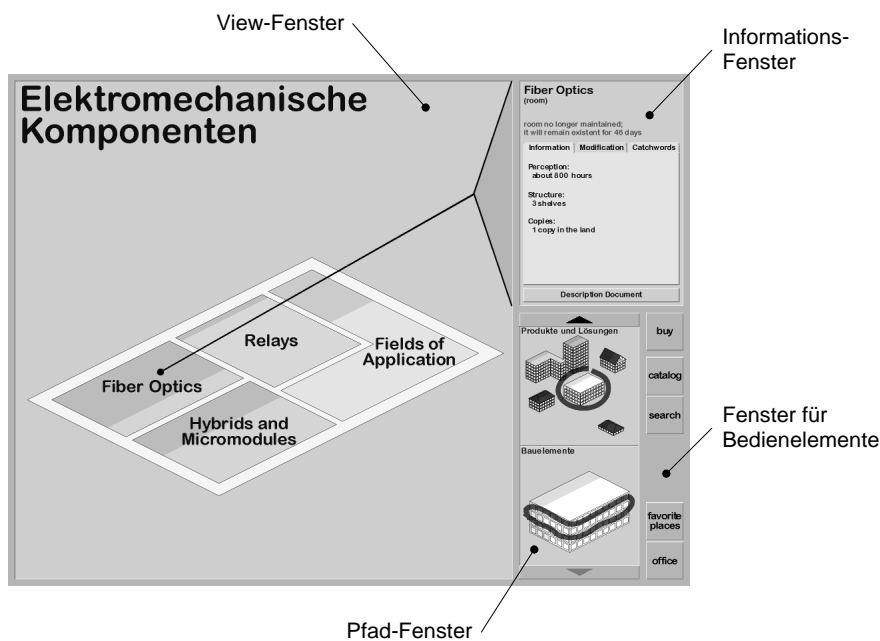
Der Prototyp wurde nicht konstruiert, um ausschließlich als Spezifikation für ein Produkt zu dienen. Er hat vielmehr die Aufgabe, die entwickelten Konzepte und getroffenen Entscheidungen einem größeren Personenkreis zu präsentieren und deren Reaktionen und Meinungen für die Weiterentwicklung zum fertigen Produkt zu benutzen. Dabei wurde die Prototypentwicklung auf die Betrachtung der Bibliothekswelt beschränkt.

---

1. vgl. Anhang C und Anhang D

Bild 5 zeigt die Bildschirmerteilung der Navigationskomponente. Der Bildschirm gliedert sich in vier unterschiedlich große Bereiche, die als Fenster bezeichnet und in den folgenden Abschnitten genauer beschrieben werden:

- View-Fenster
- Pfad-Fenster
- Informations-Fenster
- Fenster für Bedienelemente



**Bild 5:** Bildschirm-Layout der Navigationskomponente

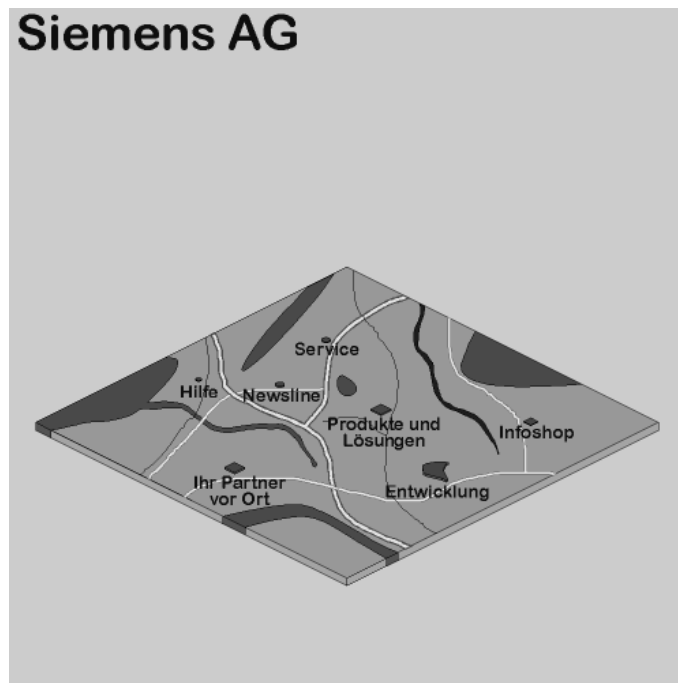
### 3.2.1 Das View-Fenster

Das View-Fenster nimmt die größte Fläche ein. Es stellt den Inhalt des Containers, in dem sich der Benutzer im Augenblick befindet, graphisch dar. Die graphische Inhaltsdarstellung eines Containers wird als **View** bezeichnet. Da dem Benutzer der Eindruck vermittelt wird, er würde sich in einem Container befinden, erlebt er diesen als Ort im Raum. Die jeweilige Bezeichnung des dargestellten Containers wird am linken oberen Fensterrand angezeigt. Im folgenden werden die Views der unterschiedlichen Containertypen benannt und kurz erläutert:

Die Land-View zeigt den Blick auf eine Landfläche, die mit Seen, Flüssen und Straßen durchzogen ist. Diese dienen ausschließlich der Orientierung und sollen das Erinnerungsvermögen des Benutzers unterstützen. Auf der Fläche verteilt gibt es mehrere Städte, die in Anlehnung an Universitätsgelände als Campi



bezeichnet werden. Alle Campi werden je nach Größe in unterschiedlichen Formen dargestellt. Jede Campus-Ikone ist mit einer Schattierung versehen, die den Füllstand anzeigen soll. Die Wahl der Farben wurde aus Landkarten übernommen (Flächen in grün, Flüsse und Seen in blau und Städte in rot). Bild 6 zeigt ein Beispiel für ein Land, auf dem 7 Campi – jeweils mit ihrer Bezeichnung beschriftet – zu sehen sind.



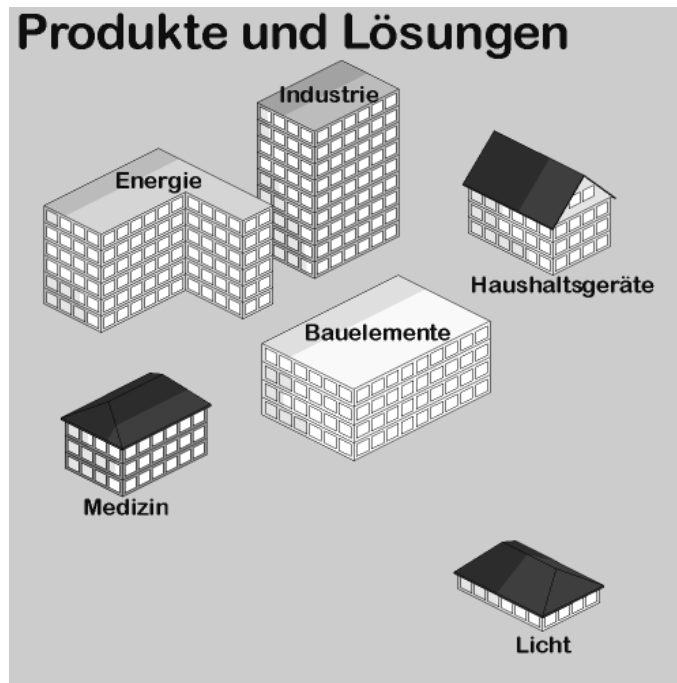
**Bild 6:** Land-View

Sowohl die Symbole der angezeigten Campi als auch die füllstandsanzeigenden Schattierungen werden generiert. Ein Bibliothekar ist ausschließlich für die Platzierung der Campi zuständig, nicht aber für deren Bildschirmdarstellung in der Land-View.

Das Anwählen bzw. Anklicken einer Campus-Ikone in der Land-View führt direkt zu einer detaillierten Darstellung dieses Campus'. Bild 7 zeigt dazu ein Beispiel.

Der Inhalt eines Campus besteht aus Gebäuden. Für ein Gebäude kann aus verschiedenen Bauformen und Farben gewählt werden. Je nach Gebäudetyp stehen unterschiedliche Wahlmöglichkeiten für die Anzahl der Etagen zur Verfügung. Auch in der Campus-View dienen Schattierungen zur Darstellung des Füllstands. Eine Schattierung auf dem Dach eines Gebäudes zeigt den Gebäude-, eine Schattierung auf einer Etage den Etagenfüllstand an.

Das Anwählen eines Gebäudes mit dem Mauszeiger führt zur Gebäude-View (Bild 8). Die Gebäude-View besteht aus einer Vergrößerung des angewählten Gebäudes aus der Campus-View. Zusätzlich sind die einzelnen Etagen beschriftet. Auf eine schrägstehende Beschriftung der Etagen wurde zugunsten einer besseren



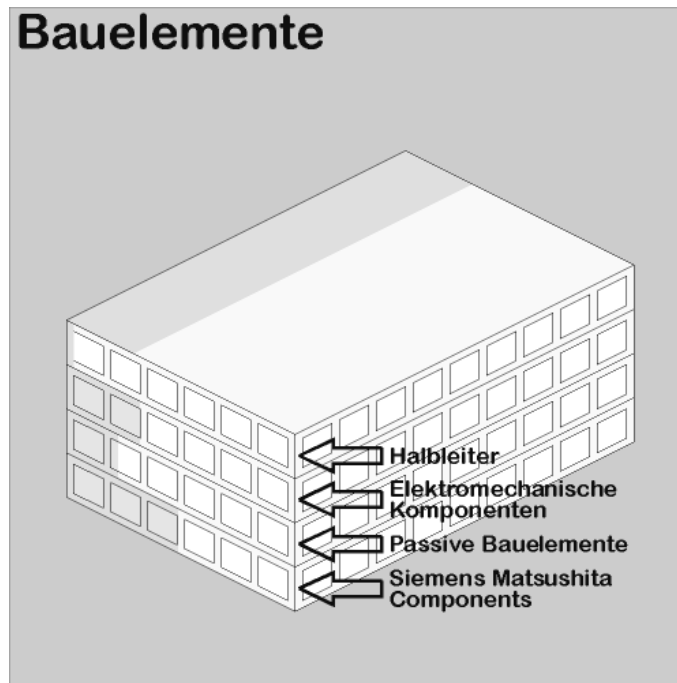
**Bild 7:** Campus-View

Lesbarkeit verzichtet. Um eine vereinfachte Zuordnung zwischen der Beschriftung und der zugehörigen Etage zu ermöglichen, wurden Pfeilsymbole eingesetzt. Die Schattierungen wurden aus der Campus-View übernommen und haben hier die gleiche Bedeutung.

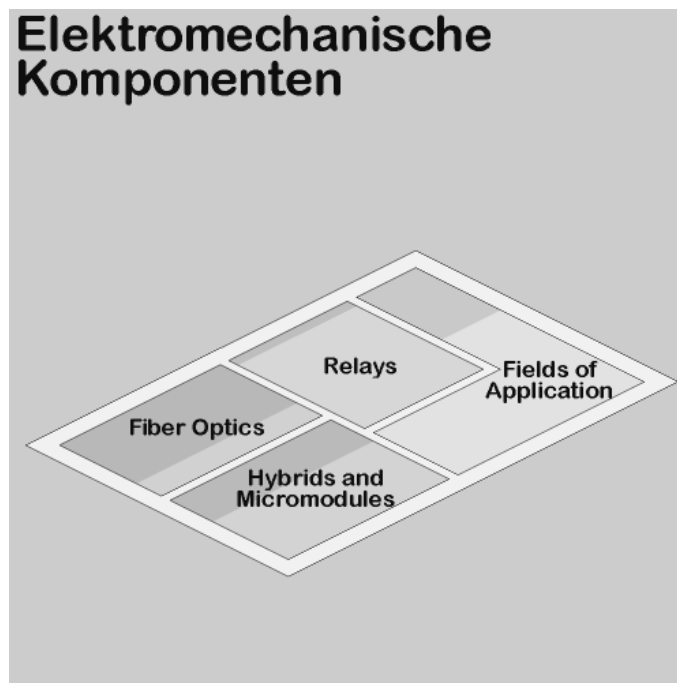
Wird in der Gebäude-View eine Etage ausgewählt, so erhält man eine entsprechende Etagen-View, wie sie in Bild 9 dargestellt ist. Eine Etage kann ein Bibliothekar durch beliebiges Einziehen von Zwischenwänden in Säle untergliedern. Im Beispiel ist eine Etage mit vier Sälen in Form eines Grundrißplans dargestellt, den man aus der Vogelperspektive betrachtet. Auch hier sind die Säle beschriftet und mit einer Schattierung zur Anzeige des Füllstands versehen.

Das Anklicken eines Saals führt in denselben. Auch ein Saal wird als Grundrißplan abgebildet. Dabei werden die den Saal begrenzenden Wände dargestellt, sowie deren weiterer Verlauf angedeutet (Bild 10). In einem Saal können Regale stehen, die lediglich durch ihren Grundriß symbolisiert werden. Diese Regale sind ebenfalls beschriftet und ihr Füllstand wird durch eine Schattierung angezeigt. Die Regale können von einem Bibliothekar beliebig positioniert werden.

Durch den Klick auf ein Regalsymbol gelangt man in die Regal-View (Bild 11). Hier wurde auf die in den bisher beschriebenen Views konsequent eingesetzte Vogelperspektive verzichtet, da ein Regal in der realen Welt in etwa die Größe eines Menschen hat, und dieser somit nicht mehr von einem erhöhten Standort auf ein Regal blicken muß, um einen Überblick zu erhalten. Stattdessen steht er unmittelbar vor dem Regal. Die Bezeichnungen der Regalbretter sind neben dem Regal positioniert. Da auf jedem Brett bereits die Anzahl der enthaltenen Sektio-

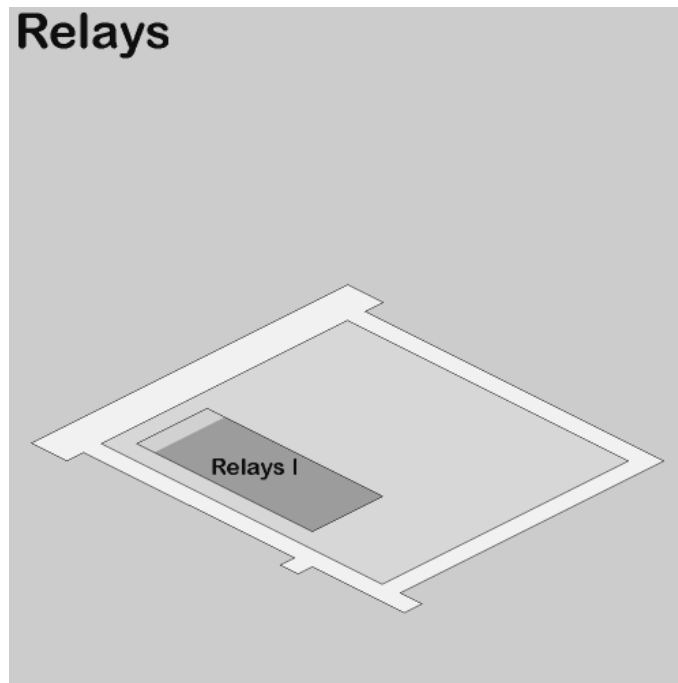


**Bild 8:** Gebäude-View

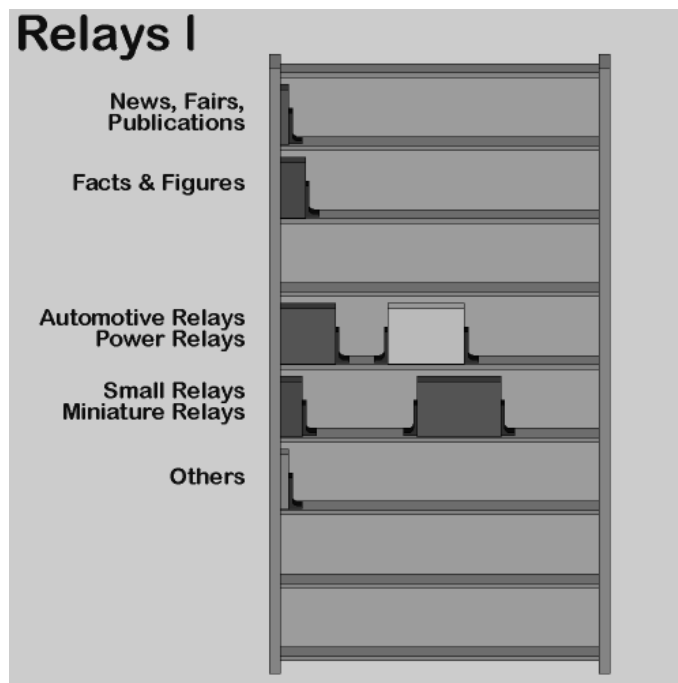


**Bild 9:** Etagen-View

nen und somit auch deren Größe sichtbar ist, ist eine Füllstandsanzeige der Regalbretter nicht mehr notwendig. Die Sektionen sind in der Regal-View nicht beschriftet und können im Gegensatz zu den Brettern nicht angewählt werden.



**Bild 10:** Saal-View



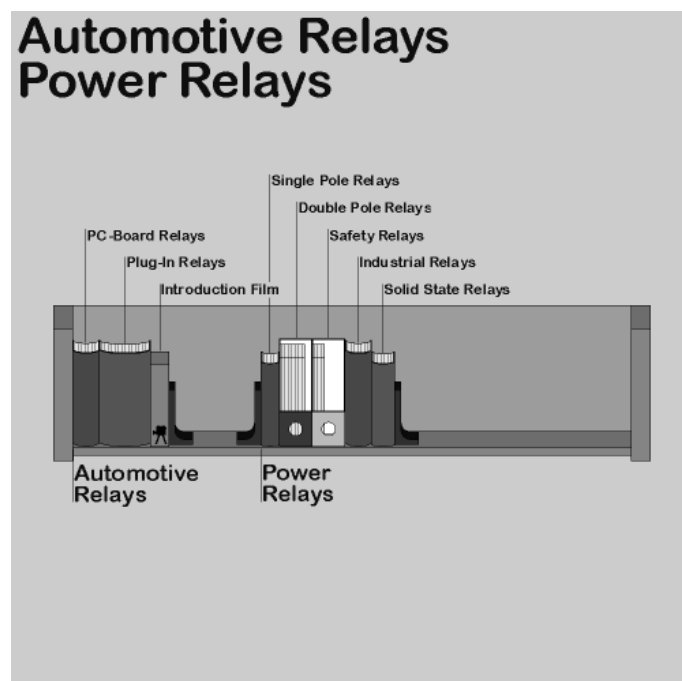
**Bild 11:** Regal-View

Ein Klick auf ein Brett führt zur gleichnamigen Brett-View (Bild 12). Dargestellt wird ein Brett als Ausschnitt aus einem Regal, d.h. aus Teilen der Seitenwände und der Rückwand sowie dem eigentlichen Brett. Darauf sind Dokumente und Ordner gruppiert und durch Buchstützen zu Sektionen zusammengefaßt. Die

Beschriftung der Sektionen wird unten, die der Dokumente und Ordner oben angezeigt. Dünne Bezugslinien vereinfachen den Zusammenhang zwischen den Beschriftungstexten und den zugehörigen Ikonen.

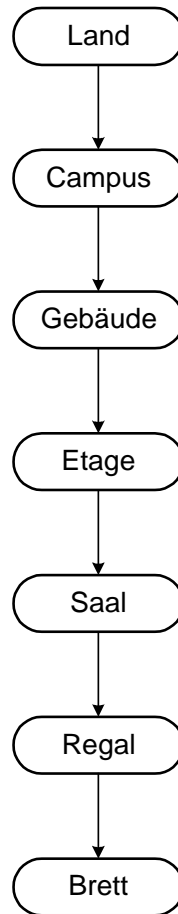
Eine vergrößerte Darstellung einer Sektion wäre nur dann sinnvoll, wenn mehrere Sektionen auf einem Brett stehen würden. Ein Regalbrett kann jedoch auch mit nur einer einzigen Sektion gefüllt sein. Eine im Vergleich zum Brett vergrößerte Darstellung der Sektion wäre dann nicht möglich. Da bereits in der Brett-View alle Dokumente und Ordner deutlich sichtbar sind, wurde auf die zusätzliche Möglichkeit der Darstellung einer Sektion verzichtet.

Der Füllstand eines sichtbaren Ordners wird durch die Dicke eines darin befindlichen Papierstapels symbolisiert. In der Sektion vorhandene Dokumente werden durch unterschiedliche Ikonen dargestellt – statisch-flächige Dokumente als Bücher, Filme als Video-, und Tondaten als Musikkassetten. Den Umfang eines statisch-flächigen Dokumentes kann man anhand der Größe des zugehörigen Buchsymbols abschätzen.

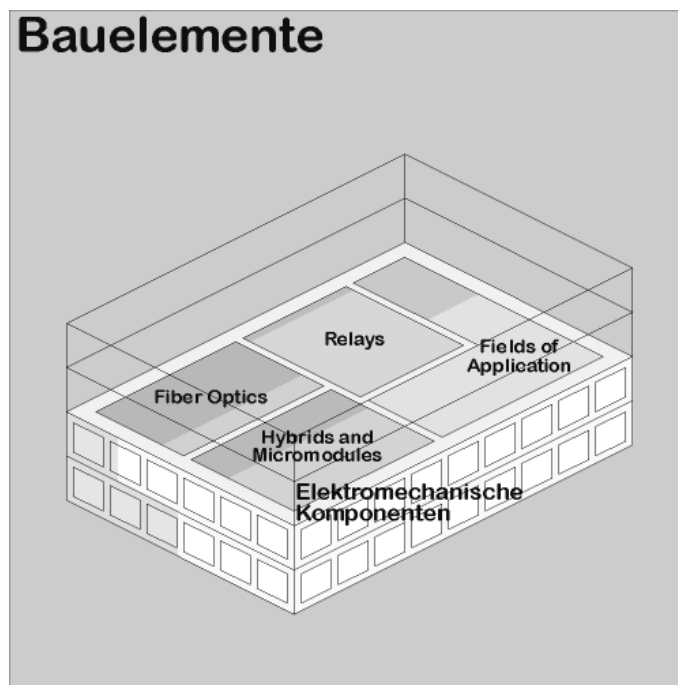


**Bild 12:** Brett-View

Die bisher beschriebenen Navigationsmöglichkeiten sind in Bild 13 dargestellt. Dieser Zustandsgraph zeigt die Navigationsmöglichkeiten von der Wurzel der Ablagestruktur der Bibliothekswelt – der Land-View – bis zur Inhaltsdarstellung eines Regalbretts – der Brett-View.



**Bild 13:** Navigationsmöglichkeiten mittels View-Fenster



**Bild 14:** Gebäude-View (transparent)

Die Inhalte von Ordnern und Unterordnern werden nicht von der Navigationskomponente dargestellt. Diese Aufgabe übernimmt ein anderes Tool – die sogenannte Katalogkomponente (siehe Kapitel 4). In der Bibliothekswelt ist man beschränkt auf die Navigationsmöglichkeiten ausgehend vom Land bis hin zu den Regalbrettern.

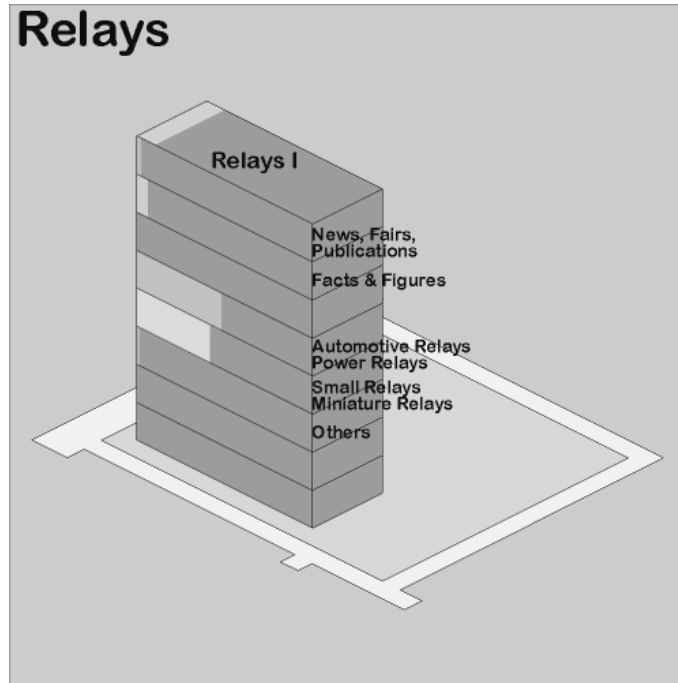
Bisher wurden nicht alle Navigationsmöglichkeiten, die innerhalb des View-Fensters möglich sind, beschrieben. Hier können noch zwei weitere Möglichkeiten, die lediglich dem Bedienungskomfort dienen, ergänzt werden. Zum einen ist dies der direkte Übergang von der Gebäude- zur Saal-View. Dazu kann – alternativ zum normalen Mausklick auf eine Etage – durch zusätzliches Festhalten einer Umschalttaste die Gebäudeaußenwand sowie das Dach transparent gemacht werden. Dadurch wird der Blick auf die ausgewählte Etage frei. Das ermöglicht nun das direkte Auswählen eines Saals dieser Etage. Beim Blick auf ein transparentes Gebäude wird die Gebäudeform durch dünne Umrißlinien symbolisiert (siehe Beispiel in Bild 14). Die ausgewählte Etage wird dabei genauso dargestellt, wie dies für die Etagen-View bereits beschrieben wurde.

Eine weitere Möglichkeit, eine View bei der Navigation zu überspringen, besteht im Übergang von der Saal- zur Brett-View. Wie bereits beschrieben, werden beim Blick auf einen Saal die Regale durch deren Grundfläche symbolisiert. Ein Klick auf ein solches Regalsymbol bei gedrückter Umschalttaste führt dazu, daß dieses Regal in Form eines Quaders aus dem Boden nach oben geschoben und die Bezeichnung der Regalbretter sichtbar wird. Im Gegensatz zur Regal-View, bei der bereits Anzahl und Größe der Sektionen sichtbar werden, sind hier lediglich die Füllstände der Regalbretter durch verschiedenfarbige Schattierungen angezeigt, die auf der Regaloberseite angebracht sind (Bild 15). Diese erweiterten Navigationsmöglichkeiten sind im Zustandsgraphen von Bild 16 dargestellt.

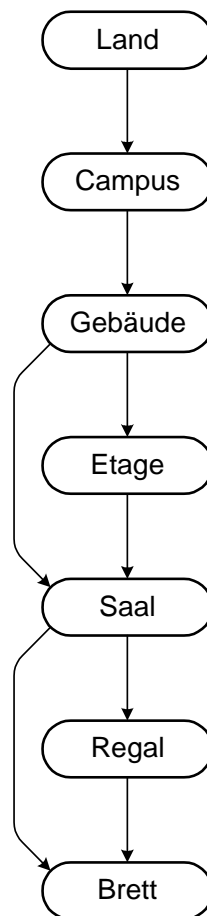
### 3.2.2 Das Pfad-Fenster

Während das View-Fenster den Ort darstellt, an dem sich der Benutzer im Augenblick befindet, zeigt das Pfad-Fenster den Weg von der Wurzel der Ablagestruktur – dem Land – zu diesem aktuellen Ort. Dieser Weg wird im folgenden als **Pfad** bezeichnet. Zur Darstellung ist im Pfad-Fenster für jeden Baumknoten des aktuellen Pfads ein Eintrag enthalten. So ist für jede Hierarchieebene der Ablagestruktur, durch die der Benutzer navigiert hat, ein Eintrag im Pfad-Fenster vorhanden. Als Eintrag dient die verkleinerte Darstellung der zugehörigen View, die zudem in ein Graustufenbild umgewandelt wird. In diesem Bild ist zusätzlich jeweils die Ikone des Containers markiert, den der Benutzer in der View angewählt hatte.

Die Markierung äußert sich durch eine breite, transparente, rot eingefärbte Linie um die entsprechende Ikone. Eine auf diese Weise dargestellte Markierung ist deutlich auf einem Graustufenbild zu erkennen.

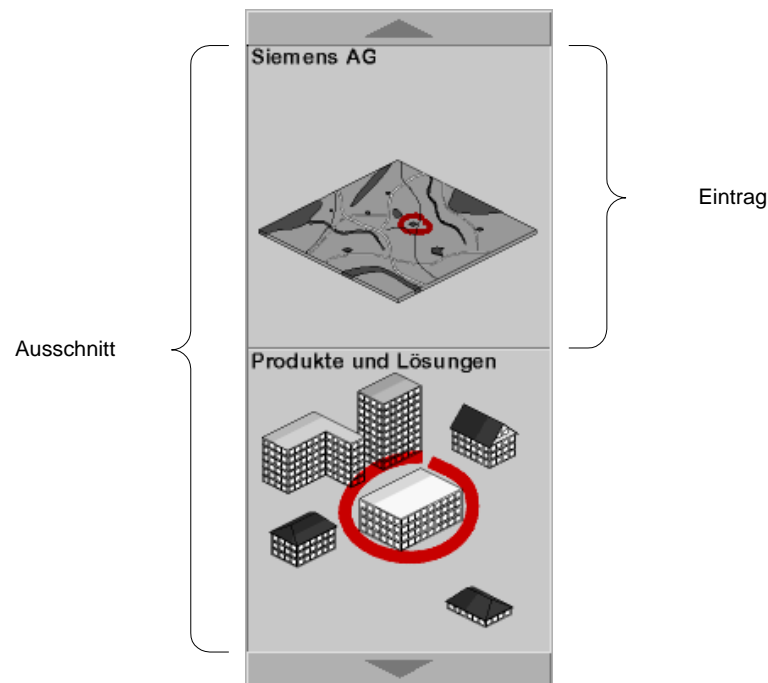


**Bild 15:** Saal-View (Regal oben)



**Bild 16:** Erweiterte Navigationsmöglichkeiten mit Hilfe des View-Fensters





**Bild 17:** Beispiel für ein gefülltes Pfad-Fenster

Zum einfacheren Verständnis der Anzeigemöglichkeit innerhalb des Pfad-Fensters soll hier ein Beispiel dienen (vgl. Bild 17).

Befindet sich ein Benutzer in einem Gebäude, so verläuft der aktuelle Pfad vom Land über den Campus zu diesem Gebäude. Der Inhalt des Pfad-Fensters besteht in diesem Fall aus einem Abbild der Land-View, in der der angewählte Campus markiert ist, und einem Abbild des Campus, in dem das angewählte Gebäude markiert ist. Das angewählte Gebäude ist in diesem Fall genau das Gebäude, in dem sich der Benutzer im Augenblick befindet.

Der Inhalt des Pfad-Fensters wird nach dem Stapelprinzip verwaltet. Ein Eintrag kommt genau dann hinzu, wenn sich der Benutzer in der Ablagestruktur um eine Hierarchieebene nach unten bewegt. Dieser Eintrag wird im Pfad-Fenster unten angefügt.

Obwohl der Inhalt des Pfad-Fensters aus mehr als zwei Einträgen bestehen kann, sind stets nur zwei Einträge gleichzeitig sichtbar. Mit Hilfe der beiden Pfeiltasten am oberen und unteren Rand des Pfad-Fensters ist jedoch ein Verschieben des Inhalts möglich, so daß der aktuelle Ausschnitt frei gewählt werden kann. Die Tasten sind mit Dreiecksymbolen versehen, die die Richtung der dadurch auslösbaren Ausschnittsverschiebung symbolisieren.

Um anzuzeigen, daß der aktuelle Ausschnitt des Pfad-Fensters in eine Richtung nicht weiter verschiebbar ist, wird das Dreieck der entsprechenden Taste in grauer Farbe dargestellt. Das ist genau dann der Fall, wenn der erste bzw. letzte Eintrag im Pfad-Fenster sichtbar ist. Ein Dreieck erscheint dagegen schwarz, wenn durch das Drücken der Taste eine Verschiebung möglich ist.

Neben der Darstellung des Pfads dient das Pfad-Fenster auch der Navigation. Mit seiner Hilfe werden die bisher beschriebenen Navigationsmöglichkeiten erweitert.

Während man beim Anwählen einer Container-Ikone im View-Fenster eine – in den beiden beschriebenen Sonderfällen auch zwei (vgl. Bild 16) – Hierarchieebenen der Ablagestruktur nach unten gelangt, ermöglicht die Benutzung des Pfad-Fensters das freie Bewegen in der Ablagestruktur in Richtung der Wurzel der Hierarchie. Durch Anwählen eines Eintrags im Pfad-Fenster gelangt man in den Container, der im Eintrag dargestellt ist. Das bedeutet, daß der Inhalt des View-Fensters durch das Original der im Pfad-Fenster angewählten Verkleinerung ersetzt wird.

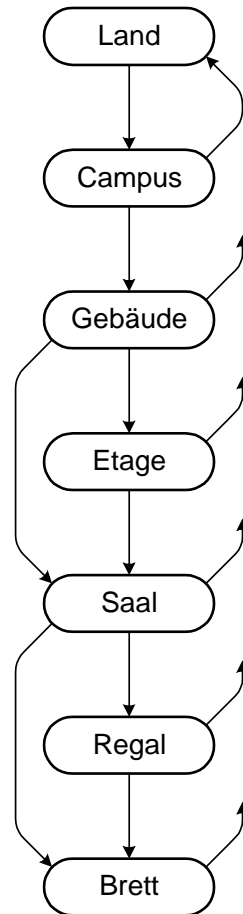
Da im Pfad-Fenster – möglicherweise erst nach dem Verschieben des Ausschnitts – jeder vorhandene Eintrag angezeigt und somit auch angewählt werden kann, ist eine direkte Navigation zu allen im Pfad enthaltenen Baumknoten möglich. Dadurch kann sich ein Benutzer beliebig viele Hierarchieebenen nach oben bewegen kann.

Die durch View- und Pfad-Fenster gegebenen erweiterten Navigationsmöglichkeiten werden durch den Zustandsgraphen in Bild 18 dargestellt. Dabei bedeuten die Pfeile mit nach oben weisender Pfeilspitze, daß die Navigation im Ablagestrukturbaum beliebig weit nach oben erfolgen kann.

Alle in Bild 18 enthaltenen Pfeile zeigen mögliche **Navigationsschritte** an. Unter einem Navigationsschritt soll im folgenden eine Bewegung zwischen zwei Containern verstanden werden, die durch Anwählen einer Ikone im View-Fenster oder eines Eintrags im Pfad-Fenster hervorgerufen wird.

Ein solcher Navigationsschritt soll nicht als blitzartiger Wechsel des Inhalts von View- und Pfad-Fenster erlebt, sondern vielmehr durch geeignete Animationen als angenehm empfunden werden. Durch diese Animationen wird die intuitive Bedienbarkeit der Navigationskomponente gefördert. Die Animationen sind je nach Navigationsrichtung – nach oben oder unten – im Ablagestrukturbaum grundlegend unterschiedlich.

Nach Anwählen einer Container-Ikone im View-Fenster erklingt das Geräusch eines Fotoapparats bei der Aufnahme. Zugleich wandert eine Gruppe von Rechtecken, die zu Beginn die Größe des View-Fensters und am Ende die eines Pfad-Fenster-Eintrags besitzen, zum Zentrum des View-Fensters. Nach Abspielen des Geräusches und des durch den Verkleinerungsprozeß der Rechtecke realisierten Zoomeffekts erscheint ein verkleinertes Graustufenbild des aktuellen View-Fenster-Inhalts, das als Eintrag im Pfad-Fenster benutzt wird, in der Mitte des View-Fensters. Bevor jedoch die Verkleinerung der View ins Pfad-Fenster bewegt werden kann, wird der Pfad-Fenster-Inhalt um die Höhe eines Eintrags nach oben geschoben. Auf diese Weise wird der Inhalt des Pfad-Fensters nach unten ergänzt und somit aktualisiert.

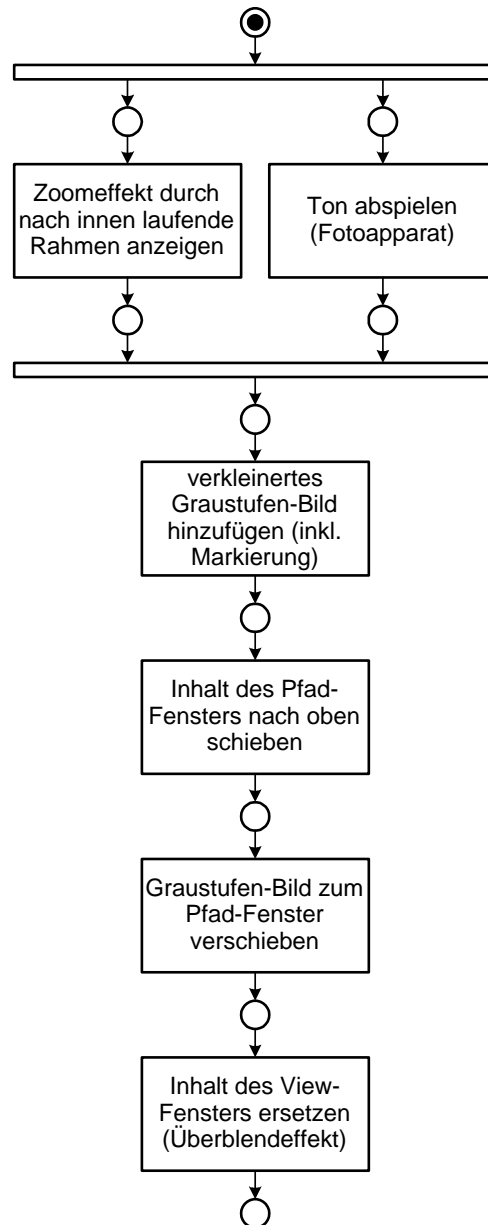


**Bild 18:** Navigationsmöglichkeiten mit View- und Pfad-Fenster

Als Abschluß dieser Animation vollzieht sich der Wechsel des View-Fenster-Inhalts. Um auch diesen Wechsel nicht blitzartig erfolgen zu lassen, wird das Bild, das den alten Inhalt des View-Fensters darstellt, in das neue überblendet. Dabei entsteht eine kurzzeitige Überlagerung beider Bilder. Nach Ende dieses Überblendeffekts ist die Animation eines Navigationsschritts abgeschlossen. Der gesamte Ablauf der Animation, der etwa zwei Sekunden andauert, ist im Petrinetz in Bild 19 dargestellt.

Die Animation verläuft bei den direkten Übergängen von der Gebäude- zur Saal- und von der Saal- zur Brett-View in etwas abgewandelter Form.

Der erste Teil der Animation – der Verlauf des Zoomeffekts, das Erklingen des Sounds sowie das Erscheinen des verkleinerten Graustufenbilds – läuft zweimal nacheinander ab. Es sind also zwei Bilder zu sehen, die als Pfad-Fenster-Einträge verwendet werden und auch bereits entsprechend angeordnet sind. Bei diesen verkleinerten Bildern handelt es sich um die gleichen, die entstanden wären, wenn ein Benutzer navigiert hätte ohne die Möglichkeit zum Überspringen einer Hierarchieebene zu benutzen. Das bedeutet, daß kein Abbild der Gebäude-View mit sichtbarem Etagegrundriß (vgl. Bild 14) bzw. keine Saal-View mit nach oben geschobenem Regal (vgl. Bild 15) in der Größe eines Pfad-Fenster-Eintrags ent-



**Bild 19:** Animation nach Anwählen eines Containers im View-Fenster

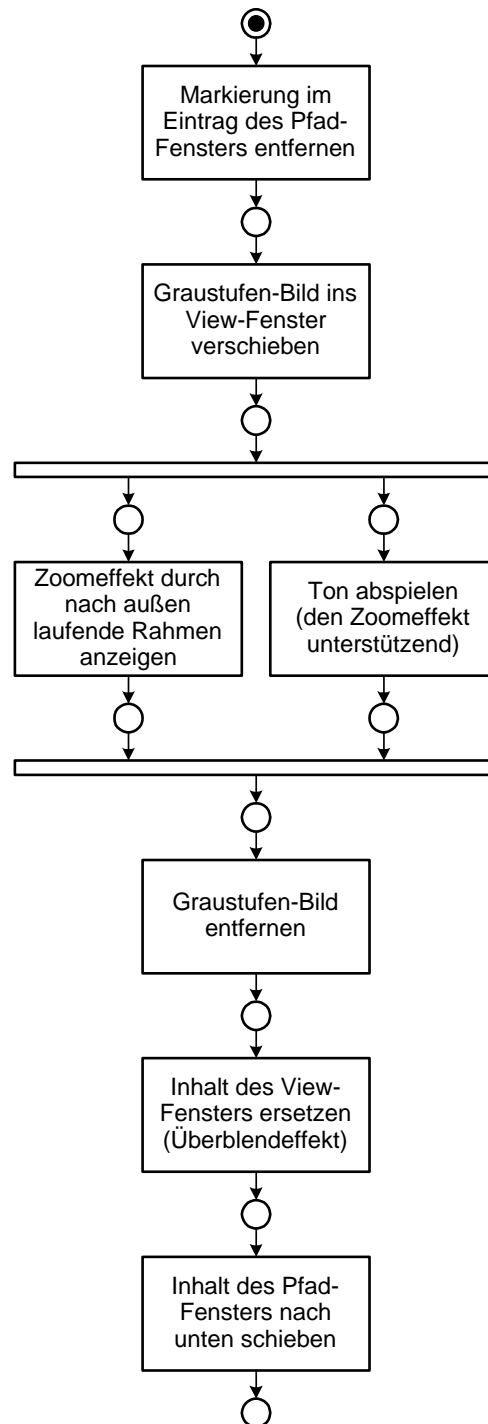
steht. Stattdessen werden genauso wie bei der Navigation ohne Benutzung der Umschalttaste Abbilder der Gebäude- und Etagen- bzw. der Saal- und Regal-View erzeugt.

Der Inhalt des Pfad-Fensters ist also ausschließlich vom Ort abhängig, an dem sich der Benutzer befindet, nicht aber davon, wie er an diesen Ort gelangte.

Nachdem der Inhalt des Pfad-Fensters um die Höhe zweier Einträge nach oben geschoben wurde, werden beide Graustufenbilder an die nun freien Plätze im Pfad-Fenster bewegt. Danach wird der Inhalt des Pfad-Fensters durch einen Überblendeffekt aktualisiert.

Bei umgekehrter, d.h. nach oben führender Navigationsrichtung verläuft die Animation wie folgt (siehe Bild 20):

Zunächst verschwindet die Markierung im angewählten Eintrag des Pfad-Fensters. Anschließend wird dieser Eintrag in die Mitte des View-Fensters verschoben, d.h. im Inhalt des Pfad-Fensters verbleibt eine freie Stelle.



**Bild 20:** Animation nach Anwählen eines Eintrags im Pfad-Fenster

Das Ersetzen des View-Fenster-Inhalts wird durch einen akustisch unterstützten Zoomeffekt vorbereitet. Mehrere sich vergrößernde Rechtecke verändern dabei ihre Größe vom Rahmen des Graustufenbilds zum Rahmen des View-Fensters. Das Graustufenbild wird entfernt und der Inhalt des View-Fensters durch einen Überblendeffekt aktualisiert.

Schließlich wird noch der Inhalt des Pfad-Fensters so weit nach unten geschoben, bis die freie Stelle verschwunden ist.<sup>1</sup> Der Inhalt des Pfad-Fensters ist jetzt wieder auf dem aktuellen Stand.

Auch die Zeitdauer dieses Animationsprozesses beträgt ungefähr zwei Sekunden.

### 3.2.3 Das Informations-Fenster

Das Informations-Fenster dient der Anzeige von Informationen zu einem im View-Fenster ausgewählten<sup>2</sup> Element. Dabei kann entweder die Ikone eines Containers oder die eines Dokuments ausgewählt sein, was durch eine Verbindungslinie zwischen der Ikone und dem Informations-Fenster angezeigt wird. Diese Verbindungslinie endet links in einer schwarzen Kreisfläche auf oder direkt neben der ausgewählten Ikone. Am anderen Ende befindet sich ein Dreieck, das links an das Informations-Fenster anschließt. Dazu zeigt die Gesamtübersicht in Bild 5 auf Seite 14 ein Beispiel, in dem in einer Etagen-View der Saal mit der Beschriftung "Fiber Optics" ausgewählt ist.

Der Inhalt des Informations-Fensters (Beispiel in Bild 21) gliedert sich wie folgt: Die im View-Fenster angezeigte Bezeichnung des aktuell ausgewählten Elements ist auch am oberen Rand des Informations-Fensters zu sehen. Unmittelbar darunter ist der Typ des Elements sowie dessen Darstellungsform angegeben. Ein Dokument ist beispielsweise vom Typ Buch, Video oder Tonaufnahme, ein Container vom Typ Land, Campus,..., Regal, Brett, Sektion oder Ordner. Die Darstellungsform eines Elements kann optional angegeben sein. Ein Beispiel hierfür ist die Form eines Gebäudes.

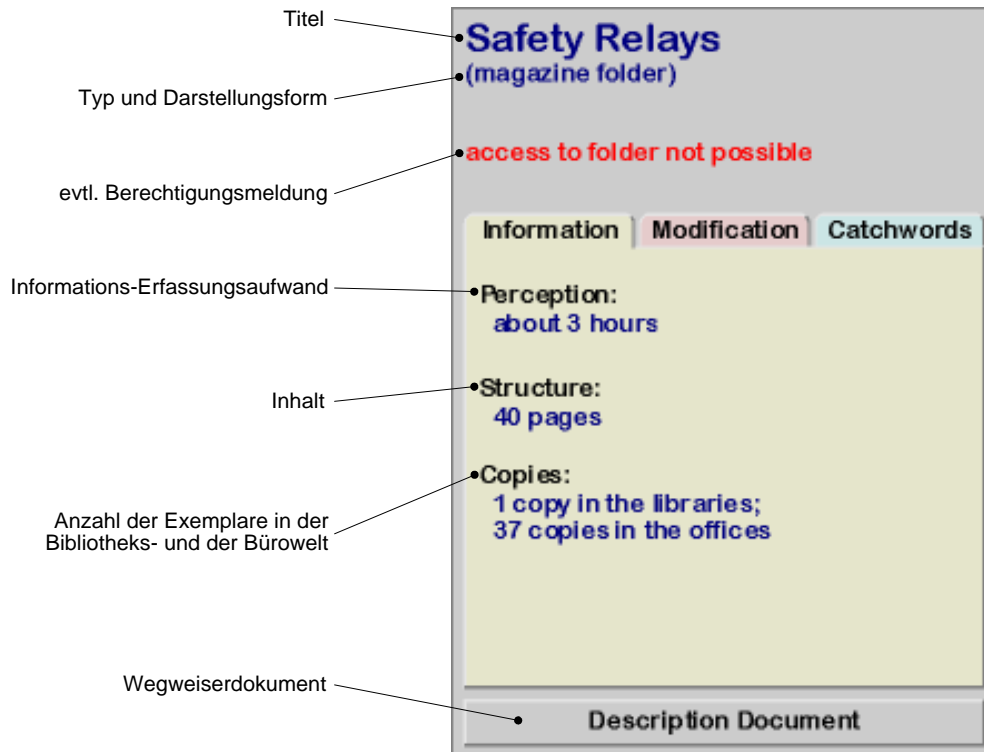
Im Informations-Fenster kann auch eine Berechtigungsmeldung angezeigt werden. Sie erscheint, wenn der aktuelle Benutzer eine eingeschränkte Zugriffsberechtigung bezüglich des ausgewählten Elements besitzt. So kann gegebenenfalls kenntlich gemacht werden, daß einem Benutzer der Zugang zu einem Container oder das Sehen von Informationen im Informations-Fenster nicht gestattet (keine

---

1. Eine Ausnahme stellt die Navigation zum Land dar. Wird die Land-View angezeigt, so ist das Pfad-Fenster leer.

2. Im Gegensatz zum Anwählen eines Elements, was beispielsweise im Falle eines Containers die Navigation zu diesem Container hervorruft, handelt es sich beim Auswählen eines Containers lediglich um das Identifizieren eines Elements.

In dem im Rahmen dieser Arbeit erstellten Prototyp muß zum Auswählen eines Elements die linke, zum Anwählen die rechte Taste der Maus gedrückt werden.



**Bild 21:** Beispiel für den Inhalt eines Informations-Fensters

Erlaubnis vorhanden) ist, da die entsprechenden Berechtigungen fehlen. Ebenso ist es möglich, daß der Zugang zu einem Container zeitweise für alle Benutzer untersagt (Zugang nicht möglich) ist. Dies geschieht dann, wenn dieser Container gerade von einem Bibliothekar, der ungestört arbeiten möchte, gewartet wird. Die hier beschriebenen Berechtigungsmeldungen werden durch auffälligen roten Text dargestellt.

In der öffentlichen Ablagestruktur ist es möglich, Container zu fixieren. Das bedeutet, daß die Wartung eines Containers durch einen Bibliothekaren beendet und der Container nach endlicher Zeit aus der Ablagestruktur entfernt werden wird<sup>1</sup>. Vor der Fixierung eines Containers muß ein neuer aufgestellt werden, der je ein Exemplar aller Dokumente des fixierten Containers enthalten muß.

Die Anzeige, daß ein Container fixiert ist, wird durch einen Text in grüner Schrift angegeben. Dieser Text enthält auch Information darüber, wie lange der Container noch in der Ablagestruktur verbleibt.

Die Informationen über das ausgewählte Element selbst werden in der Mitte des Informations-Fensters angezeigt. Aus Platzgründen und zur Steigerung der Übersichtlichkeit werden diese Informationen in drei Gruppen unterteilt und auf getrennten Flächen dargestellt. Jede dieser unterschiedlich gefärbten Flächen hat

1. Diese Möglichkeit ist insbesondere für den Fall vorgesehen, daß es zu einem Themenbereich mehr Dokumente gibt, als der dafür vorhandene Container aufnehmen kann.

die Form eines Hängeordners mit aufgestecktem Beschriftungsschild. Bei hintereinander liegenden Flächen sind die Beschriftungsschilder versetzt angebracht, so daß sie nebeneinander sichtbar sind. Das Klicken auf ein Beschriftungsschild führt dazu, daß die betreffende Fläche nach vorne rückt und somit der Informationstext lesbar wird.

Die erste Gruppe der Informationen enthält Angaben zum Informationserfassungsaufwand, den strukturellen Inhalt sowie die Anzahl der in der Bibliotheks- bzw. Bürowelt enthaltenen Exemplare des im View-Fenster ausgewählten Elements. Unter dem Informationserfassungsaufwand eines Dokuments versteht man die Zeit, die ein Benutzer benötigt, um einen Text zu lesen, ein Bild oder einen Film zu betrachten oder sich eine Tonaufnahme anzuhören. Bei einem Container ist der Informationserfassungsaufwand die Summe der Zeiten, die für die Informationsaufnahme der im Container enthaltenen Dokumente benötigt wird.

Der strukturelle Inhalt eines Containers besteht aus Typ und Anzahl der in ihm unmittelbar enthaltenen Elemente. So enthält beispielsweise ein Gebäude eine Anzahl von Etagen.

Die Einheit für die Angabe des strukturellen Inhalts eines Dokuments ist die Seite, sofern es sich um ein statisch-flächiges Dokument handelt. Für andere Dokumente ist die Angabe des strukturellen Inhalts nicht sinnvoll.

Die zweite Gruppe fällt für Dokumente und Container unterschiedlich aus. Während von Dokumenten der Autor, das Erstellungsdatum und die Versionsnummer angegeben ist, interessiert bei einem Container der Name des Bibliothekars, der die letzte Änderung im Container vorgenommen hat, sowie das Datum dieser Änderung.

Die dritte Gruppe von Informationen enthält die Liste der für dieses Element eingetragenen Suchbegriffe, deren Verwendung den Benutzer beim Auffinden von Dokumenten oder Containern mit Hilfe des Suchwerkzeugs unterstützt.

Ist im View-Fenster ein Container ausgewählt, so enthält das Informations-Fenster am unteren Rand eine Taste, die das Anzeigen des zum Container gehörenden Wegweiserdokuments<sup>1</sup> ermöglicht. Wegweiserdokumente werden ebenso wie Dokumente, die in die Ablagestruktur eingeordnet sind, mit einem speziell dafür vorgesehenen Werkzeug – dem Dokumenten-Viewer – betrachtet.

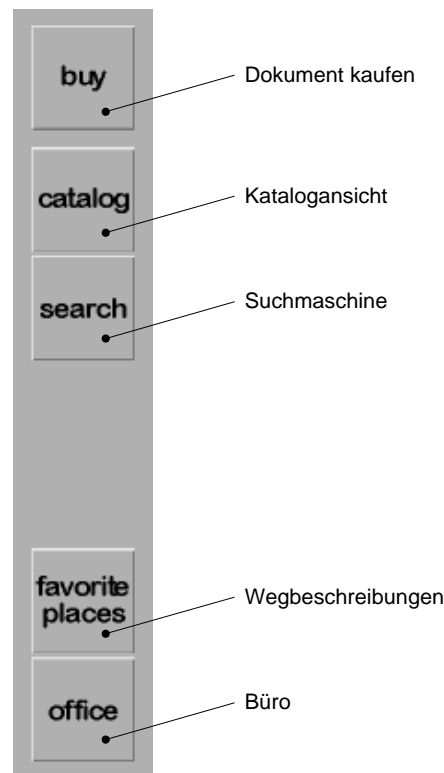
---

1. Ein Wegweiserdokument ist einem Container zugeordnet, und enthält Informationen zu dessen Inhalt. Weitere Einzelheiten können [Langhauser 97] entnommen werden.



### 3.2.4 Das Fenster für Bedienelemente

Das Fenster für Bedienelemente (siehe Bild 22) enthält Tasten, die zur Erweiterung der Funktionalität der Navigationskomponente dienen und im folgenden näher beschrieben werden:



**Bild 22:** Fenster für Bedienelemente

Nach Drücken der Taste mit der Aufschrift “buy” erhält ein Benutzer ein Exemplar des ausgewählten Dokuments oder des ausgewählten Ordners. Diese “gekauften” Elemente erscheinen im Büro des jeweiligen Benutzers und können nach dessen Rückkehr ins Büro nach Wunsch abgelegt und verwaltet werden.

Die “catalog”-Taste erlaubt dem Benutzer die Verwendung der Katalogkomponente (sie wird in Kapitel 4 beschrieben). Während die Katalogkomponente beim Anwählen eines Ordners im View-Fenster implizit aufgerufen wird, kann sie durch Drücken dieser Taste explizit gestartet werden. Somit hat ein Benutzer die Möglichkeit, sich die Ablagestruktur außerhalb der Ordner in Katalogform zu betrachten. Dabei kann auf den im View-Fenster angewählten Container Bezug genommen werden. Der Benutzer kann sich demnach die gesamte Ablagestruktur oder aber den Inhalt eines beliebigen Containers in Form eines Katalogs ansehen.

Durch Drücken der Taste “search” erhält der Benutzer ein Werkzeug, das ihn bei der Suche nach Dokumenten oder Containern unterstützt. Dabei können sowohl Abfrageattribute manuell eingegeben als auch auf eine im View-Fenster ausgewählte Ikone eines Containers oder eines Dokuments Bezug genommen werden. Wäre ein Dokument ausgewählt, so könnte nach Dokumenten des gleichen Autors, nach Dokumenten mit teilweise gleichen Suchbegriffen, nach anderen Versionen des Dokuments oder nach weiteren Exemplaren desselben Dokuments gesucht werden. Bei Containern ist die Suche nach weiteren Exemplaren oder nach Containern mit teilweise gleichen Suchbegriffen möglich.

Das Werkzeug zum Suchen in der Ablagestruktur liefert als Anfrageergebnis eine Liste, deren Einträge den gefundenen Elementen entsprechen. Das Anwählen eines Listeneintrags löst eine “geführte Navigation” zu dem gefundenen Element aus. Unter geführter Navigation soll die vollautomatisch erfolgende Bewegung in der Ablagestruktur verstanden werden. Das bedeutet, daß die Navigationskomponente ohne Eingreifen eines Benutzers Bewegungen – durch das Wechseln der Inhalte von View- und Pfad-Fenster – anzeigt. Der Wechsel des aktuellen Ortes erfolgt nicht blitzartig sondern animiert.

Nach Drücken der Taste “favorite places” erscheint ein Dialogfenster, das dem Benutzer ermöglicht, den Ort, an dem er sich augenblicklich befindet, zu speichern oder zu einem bereits gespeicherten Ort zu navigieren. Auch hierbei wird eine geführte Navigation ausgelöst.

Die Taste mit der Beschriftung “office” dient dem Benutzer zur Rückkehr ins Büro. Zur Navigation in der Ablagestruktur der Bürowelt wird eine spezielle Büro-Navigationskomponente<sup>1</sup> eingesetzt, die sich sicherlich in einigen Punkten von der im Rahmen dieser Arbeit spezifizierten Navigationskomponente unterscheiden wird.

Im Fenster für Bedienelemente wurde Platz für weitere Tasten freigehalten, deren Bedarf beim aktuellen Entwicklungsstand der Navigationskomponente noch nicht erkennbar ist.

---

1. Eine Spezifikation dieses Werkzeugs liegt noch nicht vor.

## 3.3 Erste Erkenntnisse aus dem Prototyp

Sowohl durch das Bedienen als auch durch das Vorführen des Prototyps konnten bereits wertvolle Hinweise zum Bau der Navigationskomponente gewonnen werden:

- Das Bedienkonzept wurde als intuitiv anerkannt. Ein Benutzer fühlt sich bereits nach kurzer Gewöhnungsphase sicher im Umgang mit den durch View- und Pfad-Fenster gegebenen Navigationsmöglichkeiten.
- Die Verwendung der zu Beginn der Entwicklung noch kritisierten isometrischen statt der aufwendigeren perspektivischen Darstellung wurde akzeptiert, nachdem die Ikonen im View-Fenster mit einer Beschriftung versehen waren.
- Die Dauer der realisierten Animationen von zwei Sekunden erwies sich als maximal hinnehmbare Länge. Die Zeitdauer einer Animation sollte möglichst verkürzt werden, ohne dabei die Vorzüge dieser Effekte zu mindern. So könnten beispielsweise einige Teile der Animation parallel ablaufen. Die Animation sollte außerdem vom Benutzer abschaltbar sein.
- Bei weiter steigender Rechenleistung der zur Verfügung stehenden Plattformen könnte statt des durch Rechtecke imitierten Zoomeffekts ein "echter" Zoom-Übergang eingesetzt werden. So würde die Ausführung der Navigationsschritte für einen Benutzer noch angenehmer.
- Trotz durchaus positiver Resonanz bei der Vorführung des Prototyps ist sich der Autor bewußt, daß eine Überarbeitung durch Fachleute – wie Designer oder Ergonomie-Experten – veranlaßt werden sollte.



# Kapitel 4: Die Katalogkomponente

Wie in Kapitel 3 bereits beschrieben, ermöglicht die Navigationskomponente die Raumnavigation in der Ablagestruktur. Da jedoch innerhalb eines Ordners das Denken in Orten nicht mehr angebracht ist, wurde die Raumnavigation auf den Bereich zwischen dem Land und Ordnern bzw. Dokumenten der Ablagestruktur beschränkt. Um auch Ordnerinhalte einsehen zu können, muß dem Benutzer eine weitere Art der Navigation zur Verfügung gestellt werden.

Dabei sollte neben dem Inhalt auch die Baumstruktur der Ablage erkennbar sein. Weiterhin muß die Ausgabe der Darstellung sowohl auf dem Bildschirm als auch auf einem Drucker möglich sein. Auf Papier gedruckte Dokumente haben gegenüber elektronisch gespeicherten den Vorteil, unabhängig von der Verfügbarkeit eines Rechensystems verwendet werden zu können. Zusätzlich wird das Lesen gedruckter Dokumente von vielen Menschen als angenehmer empfunden, als das auf einem Bildschirm möglich ist.

Diese Anforderungen können durch die Abbildung des Ablagestrukturbaums oder eines darin enthaltenen Teilbaums auf ein statisch-flächiges Dokument erfüllt werden. Dabei entsteht ein Dokument in Form eines Katalogs, den der Benutzer seitenweise durchblättern kann.

Das Durchblättern eines Katalogs bietet die zweite Möglichkeit zur Navigation, die hier als **Katalognavigation** bezeichnet wird. Diese Katalognavigation wird durch die **Katalogkomponente** ermöglicht. Sie kann nicht nur zur Navigation innerhalb eines Ordners, sondern für die gesamte Ablagestruktur eingesetzt werden. So ist es einem Benutzer möglich, neben den für alle Ordner zur Verfügung stehenden Katalogen – den Ordnerkatalogen – auch auf einen Katalog zur gesamten Bibliothekswelt sowie beispielsweise auf Kataloge zu Gebäuden oder Regalen zuzugreifen.

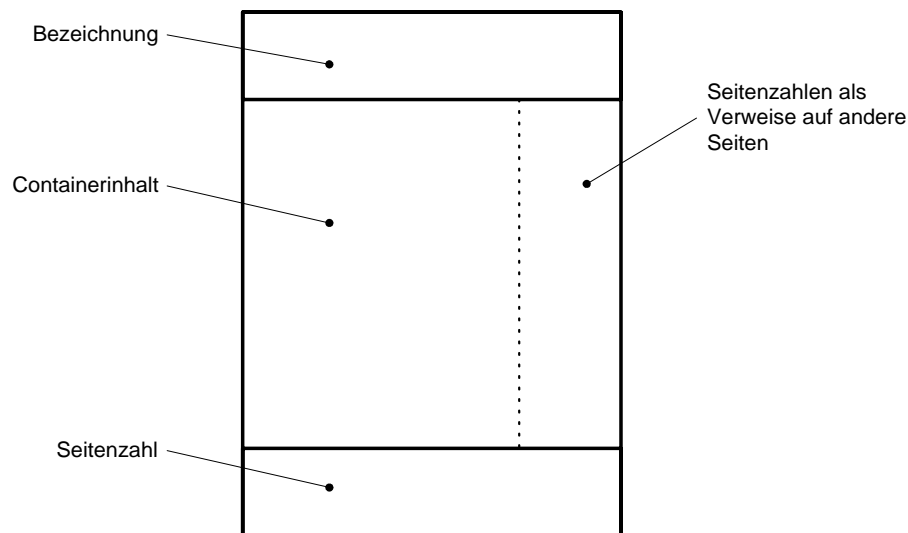
Während zur Entwicklung der Navigationskomponente aufwendige Ergonomieüberlegungen angestellt und viele Konzepte erarbeitet wurden, sind Überlegungen dieser Art für die Entwicklung der Katalogkomponente weniger entscheidend und wurden bisher bewußt vernachlässigt.

Im Rahmen dieser Arbeit wurden ausschließlich grundlegende Konzepte zur Katalogkomponente entwickelt, die im folgenden vorgestellt werden.

## 4.1 Das Seitenlayout

Alle Seiten eines Katalogs sind gleich groß und gleich gestaltet. Jede Seite ist in vier Bereiche unterteilt, die in Bild 23 dargestellt sind. Eine Seite zeigt immer den Inhalt eines Containers. Dieser Inhalt ist im größten Bereich in der Mitte der Seite enthalten. Die Bezeichnung des Containers, dessen Inhalt dargestellt ist, befindet sich im Kopfteil, während der Fußteil der Seite die jeweilige Seitenzahl enthält.

Die im Container enthaltenen Elemente – Dokumente oder Container – werden zeilenweise durch Eintragen ihrer Bezeichnung dargestellt. Der Inhalt des in einem Zeileneintrag enthaltenen Containers muß in der Regel auf einer weiteren Seite dargestellt werden. Um einem Benutzer das Auffinden einer solchen Seite zu erleichtern, sind am rechten Seitenrand die Nummern dieser Seiten eingetragen. Die Spalte rechts neben dem Bereich für den Containerinhalt dient also zur Aufnahme von Verweisen zu anderen Seiten.



**Bild 23:** Layout einer Katalogseite

Bei dieser Form eines Katalogs bildet jede Seite, die Verweise enthält, ein Inhaltsverzeichnis für weitere Seiten. Durch diese Art der Gestaltung ist ein gedruckter Katalog komfortabel nutzbar, obwohl seitenweise geblättert werden muß. Bei der interaktiven Benutzung der Katalogkomponente, die das Betrachten eines elektronisch gespeicherten Katalogs ermöglicht, erfolgt nach dem Anwählen eines Zeileneintrags für einen Container ein automatischer Wechsel zu der Seite, die den Inhalt dieses Containers enthält.

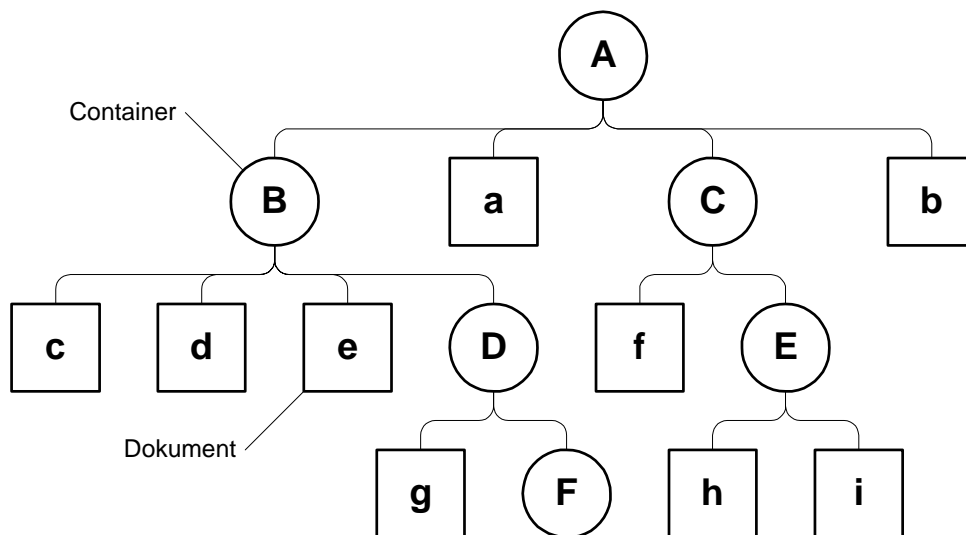
## 4.2 Die Erstellung eines Katalogs

Bevor der Ablauf zur Generierung eines Katalogs vorgestellt wird, soll ein einfaches Beispiel die bisher genannten Konzepte verdeutlichen.

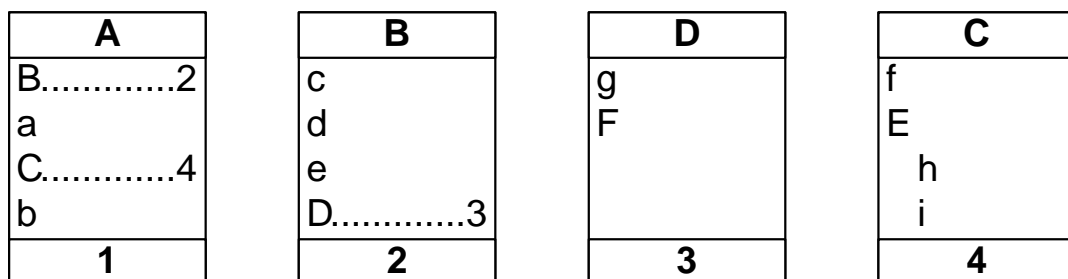
### 4.2.1 Ein Beispiel

Bild 24 zeigt ein Beispiel für den Teilbaum einer Ablagestruktur. Dabei stellen Kreise mit Großbuchstaben Container, Quadrate mit Kleinbuchstaben Dokumente dar. Die Wurzel dieses Teilbaums – der Container mit der Bezeichnung A – soll hierbei einen Ordner, alle anderen Container (B bis F) sollen Unterordner darstellen.

Die Seiten des aus dieser Baumstruktur generierten Katalogs zeigt Bild 25. Es handelt sich hierbei um einen Ordnerkatalog, der sowohl den Inhalt als auch die Struktur des Ordners A beschreibt.



**Bild 24:** Beispiel eines Strukturbaums



**Bild 25:** Seiten des generierten Beispielkatalogs

Das Ergebnis bei der Generierung eines Katalogs ist abhängig von der Anzahl der Zeilen, die für das Eintragen des Containerinhalts auf einer Seite zur Verfügung stehen. Bei der Realisierung der Katalogkomponente sollten mindestens so viele Zeilen auf einer Seite Platz finden, wie zur Darstellung eines Containerinhalts maximal benötigt werden.<sup>1</sup> Bei der Generierung des Beispielkatalogs wurde jedoch, um das Beispiel einfach halten zu können, für die maximale Anzahl von Zeileneinträgen die Zahl vier festgelegt.

Im Beispiel zeigt die erste Seite des Katalogs den Inhalt der Wurzel des Teilbaums (Ordner A). Sie enthält die beiden Dokumente a und b sowie zwei Container – die beiden Unterordner B und C. Der Inhalt von Unterordner B ist auf Seite zwei dargestellt. B enthält neben den drei Dokumenten c, d und e noch den Unterordner D. Dessen Inhalt wird vor dem des Unterordners C, d.h. auf einer Seite mit kleinerer Seitenzahl, angezeigt. Der Grund dafür ist die Anwendung des Prinzips der Linksabwicklung eines Baumes bei der Generierung eines Katalogs.

D enthält ein Dokument (g) sowie den Unterordner F. Da es sich bei F jedoch um einen leeren Container handelt, muß für seinen Inhalt keine Seite angelegt werden. Dadurch entfällt auch die verweisende Seitenzahl hinter dem Zeileneintrag. Auf diese Weise wird dem Betrachter des Katalogs angezeigt, daß es sich um einen leeren Container handelt.

Seite vier zeigt den Inhalt des Unterordners C. Er enthält das Dokument f sowie den Unterordner E. Diese beiden Elemente belegen zwei Zeilen auf der Seite, so daß zwei weitere Zeilen für Einträge zur Verfügung stehen. Da E nur zwei weitere Dokumente (h und i) enthält, und somit sein gesamter Inhalt noch auf Seite vier Platz findet, muß für die Inhaltsdarstellung des Containers E keine neue Seite angelegt werden.

Daß die beiden Dokumente h und i in Unterordner E und nicht unmittelbar in C enthalten sind, wird durch das Einrücken der beiden Zeileneinträge nach rechts angezeigt. Wie bei einem leeren Container entfällt auch hier die verweisende Seitenzahl.

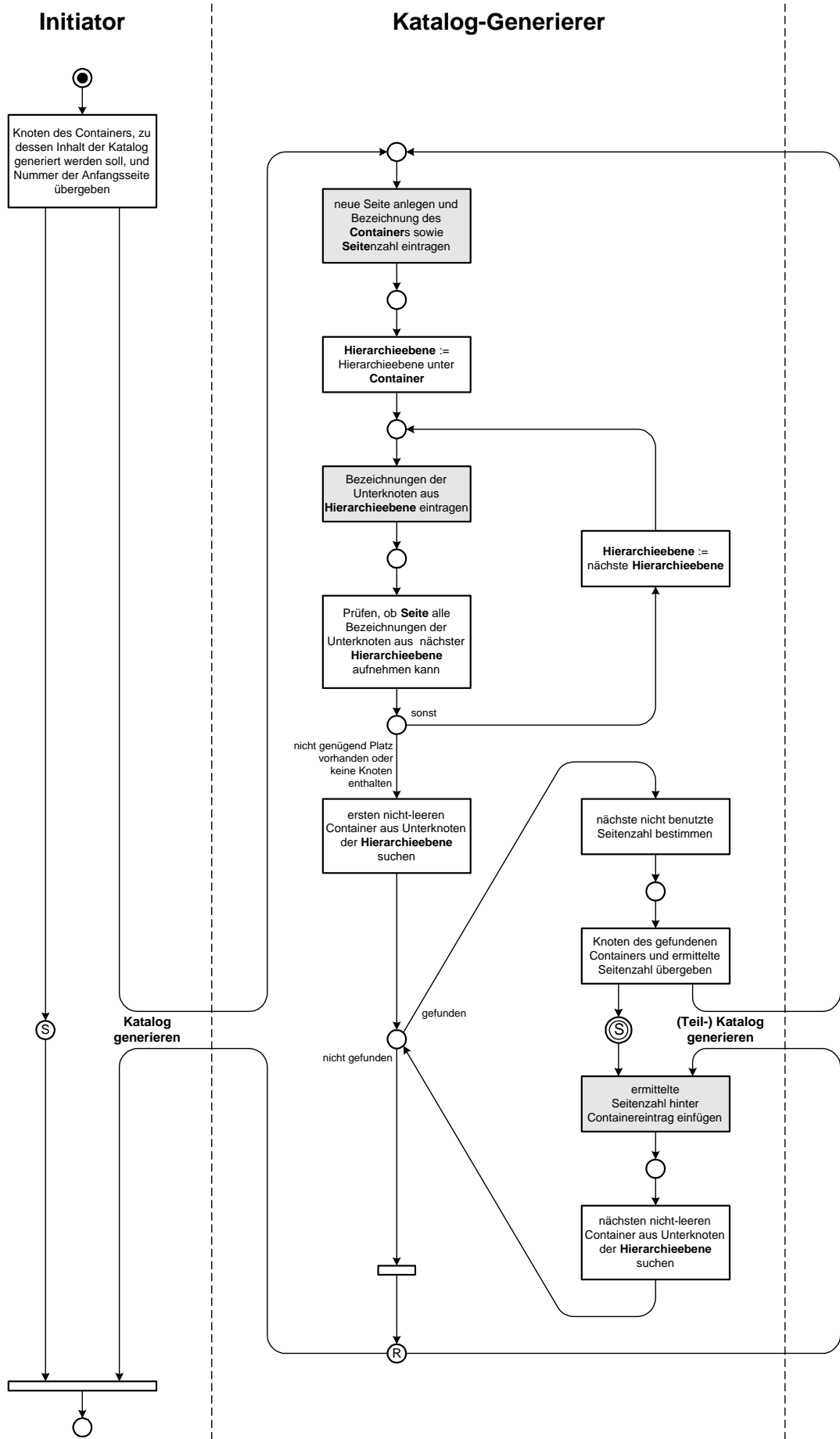
## 4.2.2 Die Generierungsprozedur

Bild 26 zeigt den Ablauf zur Generierung eines Katalogs. Das dargestellte Ablaufbild enthält zwei senkrechte gestrichelte Linien. Die linke Linie stellt eine Zuständigkeitsgrenze zwischen dem Initiator und dem Katalog-Generierer dar. Der Initiator hat die Aufgabe, den Generierungsprozeß, den der Katalog-Generierer ausführt, zu starten. Die rechte gestrichelte Linie soll andeuten, daß es sich um einen wiederholten (rekursiven) Aufruf des Generierungsprozesses handelt.

---

1. Für die praktische Nutzung eines Katalogs sollte eine Zahl zwischen 20 und 50 ausreichen.





**Bild 26:** Ablauf zur Generierung eines Katalogs (rekursive Lösung)

Die drei grau markierten Transitionen des Petrinetzes stehen für das Anlegen bzw. teilweise Ausfüllen einer Katalogseite.

Ein Katalog kann entweder aus dem gesamten oder aus einem Teil des Ablagestrukturbaums, auf den der Katalog-Generierer Zugriff hat, generiert werden. Um dem Generierer mitzuteilen, was der zu erstellende Katalog enthalten soll, wird vom Initiator der Containerknoten übergeben, der Wurzel des gewünschten (Teil-)Baums ist. Zusätzlich wird ihm die Nummer der ersten Katalogseite mitgeteilt.

Der Katalog-Generierer legt zunächst eine neue Seite an, auf der oben die Bezeichnung des übergebenen Containerknotens und unten die übergebene Seitenzahl eingetragen werden. Diese beiden Werte sowie die Hierarchieebene des Strukturbaums, die unter dem mitgeteilten Containerknoten liegt, werden gespeichert, was durch die Hervorhebung der Begriffe Container, Seite und Hierarchieebene in der Beschriftung der Transitionen verdeutlicht werden soll. Jedes Schalten einer Transition mit solchen hervorgehobenen Begriffen bedeutet einen Zugriff auf einen der gespeicherten Werte.

Nach dem Anlegen einer Seite werden die Bezeichnungen aller Unterknoten des gespeicherten Containers, die in der ebenfalls gespeicherten Hierarchieebene enthalten sind, in den Bereich in der Mitte der Seite eingetragen. Danach erfolgt die Prüfung, ob alle Unterknoten der nächsten Hierarchieebene ebenfalls auf dieser Seite Platz finden<sup>1</sup> würden. Ist dies der Fall, so wird statt der aktuellen Hierarchieebene die nächste gespeichert und die Bezeichnungen aller darin enthaltenen Unterknoten des vermerkten Containers auf der Seite eingetragen. Diese Einträge werden nach rechts eingerückt und stehen unmittelbar unter der Bezeichnung ihres übergeordneten Containers (vgl. vierte Seite des Beispielkatalogs in Bild 25). Danach erfolgt erneut die oben genannte Prüfung.

Dieser Vorgang wiederholt sich, bis eine der beiden Abbruchbedingungen erfüllt ist. Die erste Abbruchbedingung ist erfüllt, wenn die Seite nicht alle Unterknoten der nächsten Hierarchieebene aufnehmen kann, die zweite dann, wenn die nächste Hierarchieebene keine Knoten enthält.

Nach Verlassen der Schleife erfolgt die Suche nach dem ersten nicht-leeren Container, der sowohl Unterknoten des aktuellen Containers als auch in der aktuell gespeicherten Hierarchieebene enthalten ist. Kann hier ein Knoten gefunden werden, so wird eine Wiederholschleife so oft durchlaufen, bis kein Container mehr entdeckt werden kann. Mit Verlassen der Schleife ist zugleich das Ende der Prozedur erreicht.

Innerhalb der Wiederholschleife wird zu Beginn die nächste nicht benutzte Seitenzahl bestimmt. Danach erfolgt ein rekursiver Aufruf der Prozedur zum Generieren eines Katalogs (durch einen S-Platz im Petrinetz angedeutet). Im Falle des rekur-

---

1. Das ist abhängig von der Anzahl der Zeilen, die auf dieser Seite zur Verfügung stehen. Diese Zahl wird vor der Laufzeit festgelegt.

siven Aufrufs handelt es sich um die Generierung eines Teilkatalogs. Dabei wird sowohl der gefundene Containerknoten als auch die ermittelte Seitenzahl übergeben, an der der Teilkatalog beginnen soll. Bei jedem Aufruf der Prozedur werden für die drei zu speichernden Werte neue Speicherplätze<sup>1</sup> angelegt, die erst nach Beendigung der Prozedur wieder freigegeben werden.

Nach Rückkehr aus der rekursiv aufgerufenen Prozedur wird die Kataloggenerierung damit fortgesetzt, daß die vor dem Aufruf ermittelte Seitenzahl<sup>2</sup> als Verweis hinter dem Eintrag des gefundenen Containers vermerkt wird.

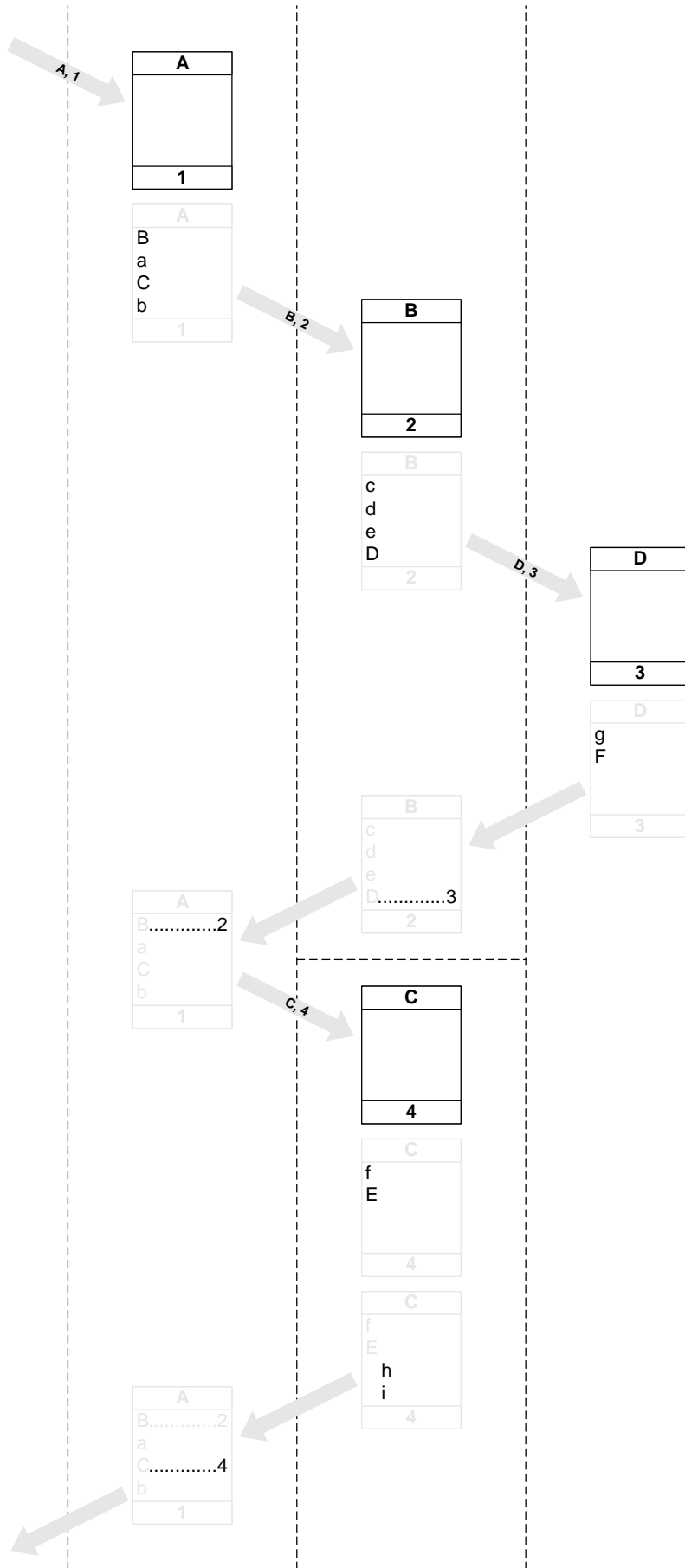
Um den gesamten Ablauf besser zu verstehen, sollte er anhand eines Beispiels nachvollzogen werden. Als Beispiel kann dabei die Generierung des Katalogs in Bild 25 aus der Baumstruktur in Bild 24 dienen. Um den Nachvollzug zu vereinfachen, wurden in Bild 27 die Teilschritte beim Entstehen des Beispielkatalogs aufgezeichnet. Dabei verläuft die zeitliche Abfolge der Seitenerstellung von oben nach unten. Alle Abbildungen einer Seite stehen für einen Teilschritt bei der Katalogentstehung und können dem Schalten einer grau dargestellten Transition (im Petrinetz von Bild 26) zugeordnet werden. Die beim Schalten der entsprechenden Transition hinzukommenden Teile einer Seite sind schwarz dargestellt. Im Gegensatz dazu sind grau gezeichnete Teile bereits vorhanden.

Die grauen Pfeile bedeuten ein Fließen der Marke im Petrinetz über eine gestrichelte Linie. Nach rechts unten zeigende Pfeile entsprechen einem Aufruf der Generierungsprozedur. Dabei ist das jeweils übergebene Wertepaar – bestehend aus Containerknoten und Seitenzahl – in den Pfeil eingetragen. Nach links unten gerichtete Pfeile bedeuten ein Ende der Generierungsprozedur und somit ein Fließen der Marke vom R-Platz über eine gestrichelte Linie des Petrinetzes.

In Bild 27 ist deutlich zu erkennen, daß die Generierungsprozedur genau einmal vom Initiator und in diesem Beispiel dreimal rekursiv aufgerufen wird.

---

1. Man spricht in diesem Falle von prozedurlokal benutzten Speicherplätzen.
2. Während der Ausführung einer aufgerufenen Prozedur werden die Werte der aufrufenden Prozedur auf einem Stapel zwischengespeichert. (siehe [Wendt 91])



**Bild 27:** Teilschritte beim Generieren des Beispielkatalogs

# Kapitel 5: Vorüberlegungen zum Finden einer geeigneten Systemarchitektur

Die im Rahmen dieser Arbeit entworfene Navigationskomponente soll als Komponente eines zu entwickelnden **Gesamtsystems**<sup>1</sup> eingesetzt werden. Die Architektur für ein System dieser Größe sollte so allgemein wie möglich entworfen werden, um eine Übertragbarkeit der Architektur auf weitere noch zu entwickelnde Softwaresysteme zu gewährleisten. In der Phase der Vorentwicklung sollte also eine Art **Universalarchitektur** für komplexe Softwaresysteme entstehen.

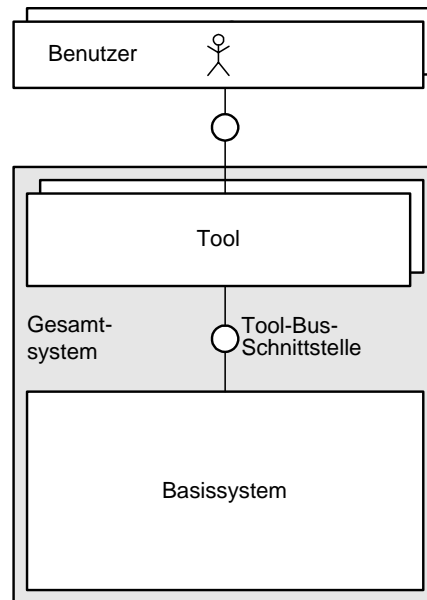
## 5.1 Der Aufbau des Gesamtsystems

Bild 28 zeigt den allgemeinen Aufbau eines Gesamtsystems, das eine zweigeteilte Struktur – bestehend aus einem **Basissystem** und einer Menge von **Tools** – aufweist.

Ein Tool ermöglicht die Interaktion eines Benutzers mit dem Basissystem. Der Kanal zwischen einem Benutzer und einem Tool ist durch einen Bildschirm, Tastatur und Maus realisiert.

Im Basissystem ist die maximal mögliche Funktionalität des Gesamtsystems festgelegt, die an einer definierten Schnittstelle in Form von Diensten angeboten wird. Diese Schnittstelle soll im folgenden als **Tool-Bus-Schnittstelle**<sup>2</sup> bezeichnet werden und stellt einen Universalanschluß zur Anbindung von Tools an das Basissystem dar.

- 
1. Bei diesem Gesamtsystem handelt es sich um ein Dokumentenverwaltungssystem, das als Corporate Memory (vgl. Kapitel 1) eingesetzt werden soll.
  2. Der Begriff "Bus" soll darauf hinweisen, daß diese Schnittstelle voraussichtlich als Bussystem realisiert wird.



**Bild 28:** Grobe Darstellung eines Gesamtsystems mit integrierten Tools

Ein Kanal zwischen einem Menschen und einem programmierten System sollte strengen Ergonomieprüfungen unterliegen. So müßte bei der Entwicklung eines Tools großer Wert auf intuitive, einfache und angenehme Bedienbarkeit gelegt werden, wie dies beim Entwurf der Navigationskomponente geschehen ist.

## 5.2 Die Anforderungen der Tool-Bus-Schnittstelle

Zu Beginn der Entwurfphase wurden folgende Forderungen an die Tool-Bus-Schnittstelle – und damit auch an die Entwicklung des Basissystems – formuliert:

- Die Implementierung von Tools und deren Anbindung an das Basissystem sollte so einfach möglich sein, daß bei ihrer Entwicklung ein Schwerpunkt auf Ergonomie und Bedienbarkeit gelegt werden kann.
- Ein Tool-Entwickler sollte sich nicht um die Vergabe und Einhaltung von Berechtigungen kümmern, obwohl das Gesamtsystem mit einem Berechtigungswesen ausgestattet sein kann. Desweiteren sollte es nicht möglich sein, Dienste des Basissystems unter Umgehung dieses Berechtigungswesens zu nutzen.
- Innerhalb eines Basissystems wird es mehrere unterschiedliche Typen von Elementen – sogenannte **Basiselemente** – geben, für die gewisse Regeln formuliert sind, d.h. eine bestimmte Logik festgelegt ist. Man denke dabei an Container unterschiedlicher Typen, die im Rahmen der Entwicklung einer geeigneten Ablagestruktur (vgl. Kapitel 2) entstanden sind. So sind beispielsweise für alle Containertypen unterschiedliche Kapazitätsbeschränkungen

festgelegt.

Die Einhaltung dieser Regeln muß vom Basissystem garantiert werden, so daß das Gesamtsystem weder durch fehlerhaft arbeitende noch durch bewußt manipulierte Tools in einen logisch falschen Zustand versetzt werden kann.

Eine solche Überprüfung, die vor der Erfüllung eines Auftrags durchgeführt werden muß, wird als **logische Prüfung** bezeichnet.

- Der Tool-Bus soll nicht nur zum Anschluß von mit dem Benutzer interagierenden Tools dienen. Er soll außerdem eine Verbindung des Gesamtsystems mit anderen programmierten Systemen ermöglichen, die ohne Interaktion mit einem Menschen Aufgaben erledigen. Zu solchen Aufgaben gehören beispielsweise die Sicherung und Aktualisierung von Systemdaten. Beim Anschluß solcher maschineller Systemen müssen Berechtigungen und Logik ebenso überprüft werden, wie dies für Tools notwendig ist.
- Tools und Basissystem befinden sich im allgemeinen auf verschiedenen Rechnern. Trotz einer möglicherweise großen räumlichen Distanz zwischen den Orten, an denen diese Rechner aufgestellt sind, sollten ausreichend kurze Reaktionszeiten garantiert werden können, um die angenehme Bedienbarkeit eines Tools zu gewährleisten.

## 5.3 Der erste Entwurf einer Architektur des Gesamtsystems

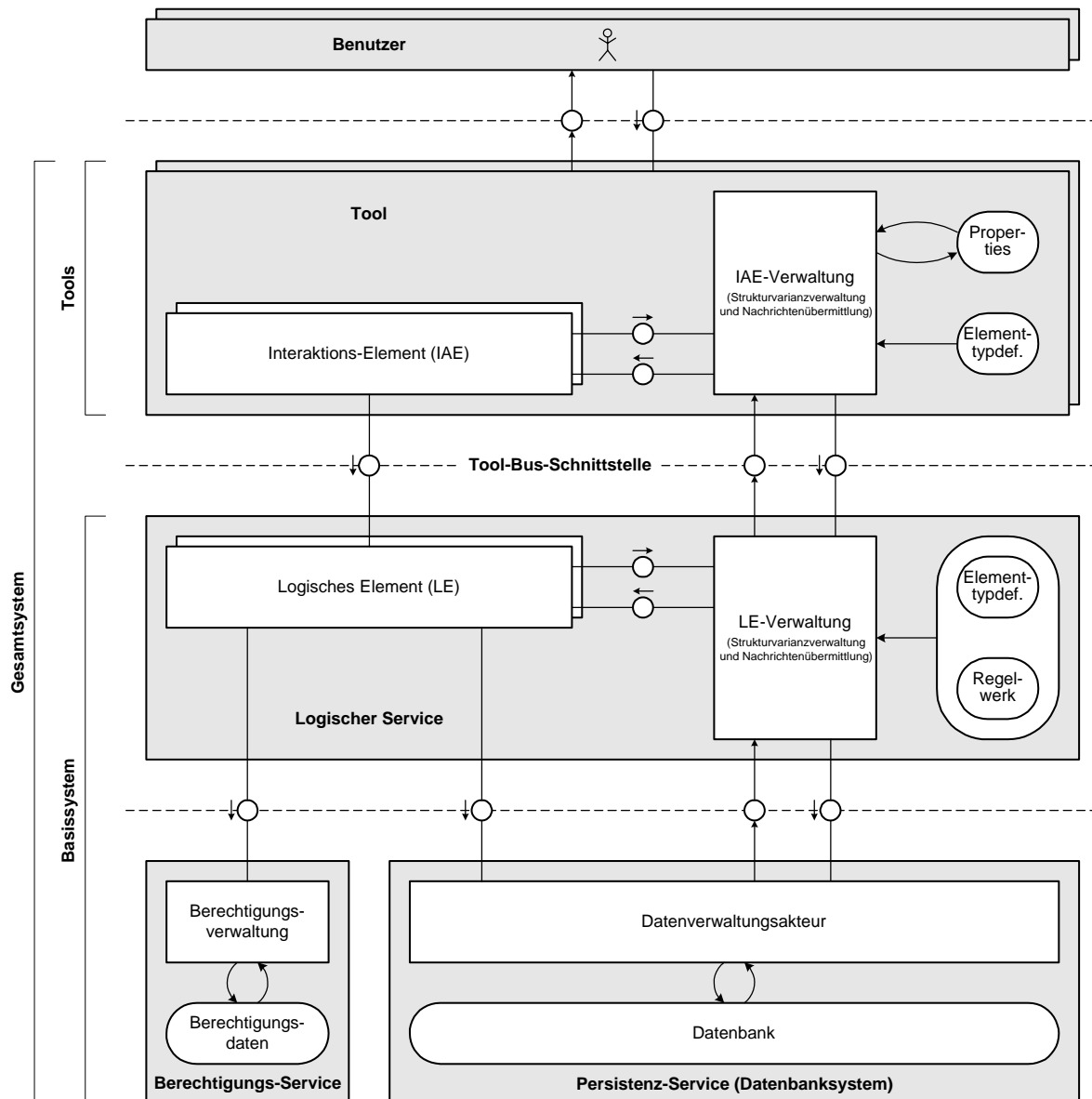
Anhand der oben genannten Forderungen entstand die in Bild 29 gezeigte Architektur, die eine Verfeinerung des Aufbaus aus Bild 28 darstellt.

### 5.3.1 Der Aufbau des Gesamtsystems

Wie bereits erwähnt, soll ein Tool, an das der Benutzer Aufträge absetzen und Anfragen stellen kann, die komfortable Kommunikation eines Benutzers mit dem Gesamtsystem ermöglichen. Im allgemeinen besteht die Bildschirmdarstellung (nach oben gerichteter Kanal zwischen Tool und Benutzer) eines solchen Tools aus mehreren grafischen Elementen<sup>1</sup>. Ein Akteur, der eine beliebig zusammengesetzte Gruppe von grafischen Elementen darstellen, und darüber hinaus Eingabe-Informationen eines Benutzers empfangen kann, wird als **Interaktionselement (IAE)** bezeichnet.

---

1. Beispiele für grafische Elemente sind Knöpfe (Buttons), Eingabefelder, Auswahllisten, Piktogramme (Grafiken), Texte, Skalen sowie Roll- und Trennbalken. Einzelheiten können in [Balzert 96] nachgelesen werden (Achtung: für den hier benutzten Ausdruck *grafisches Element* wird dort das Wort *Interaktionselement* verwendet!).



**Bild 29:** Architektur des Gesamtsystems

Während ein Benutzer mit einem Tool arbeitet, können Interaktionselemente entstehen und verschwinden. Die dafür notwendige Strukturvarianzverwaltung übernimmt ein **IAE-Verwalter**. Er ermöglicht außerdem eine Kommunikation der Interaktionselemente untereinander, indem er Nachrichten übermittelt. Das Verhalten sowie die Darstellung der IAEs auf dem Bildschirm wird bereits zur Bauzeit eines Tools festgelegt. Die dafür notwendigen Daten werden als Elementtypdefinitionen bezeichnet, auf die die IAE-Verwaltung Lesezugriff hat. Ein Tool kann innerhalb bestimmter Grenzen an die Wünsche des Benutzers angepaßt werden. Diese Anpassungsdaten bezeichnet man als Properties, die zur Laufzeit vom Benutzer über den IAE-Verwalter modifiziert werden können.



Das Basissystem (vgl. Bild 28) ist in drei Komponenten aufgeteilt:

Ein **Datenbanksystem** dient der Verwaltung und Bereitstellung persistenter Daten – es wird deshalb auch als **Persistenz-Service** bezeichnet. Zur Verwaltung und Bereitstellung von Daten hat der Datenverwaltungsakteur Zugriff auf eine Datenbank.

Der **Berechtigungs-Service** erlaubt die Festlegung und Verwaltung von individuellen Benutzerrechten, die als Berechtigungsdaten persistent gehalten werden. Die Berechtigungsverwaltung stellt Dienste zur Anforderung bzw. Abfrage von Benutzerrechten zur Verfügung.

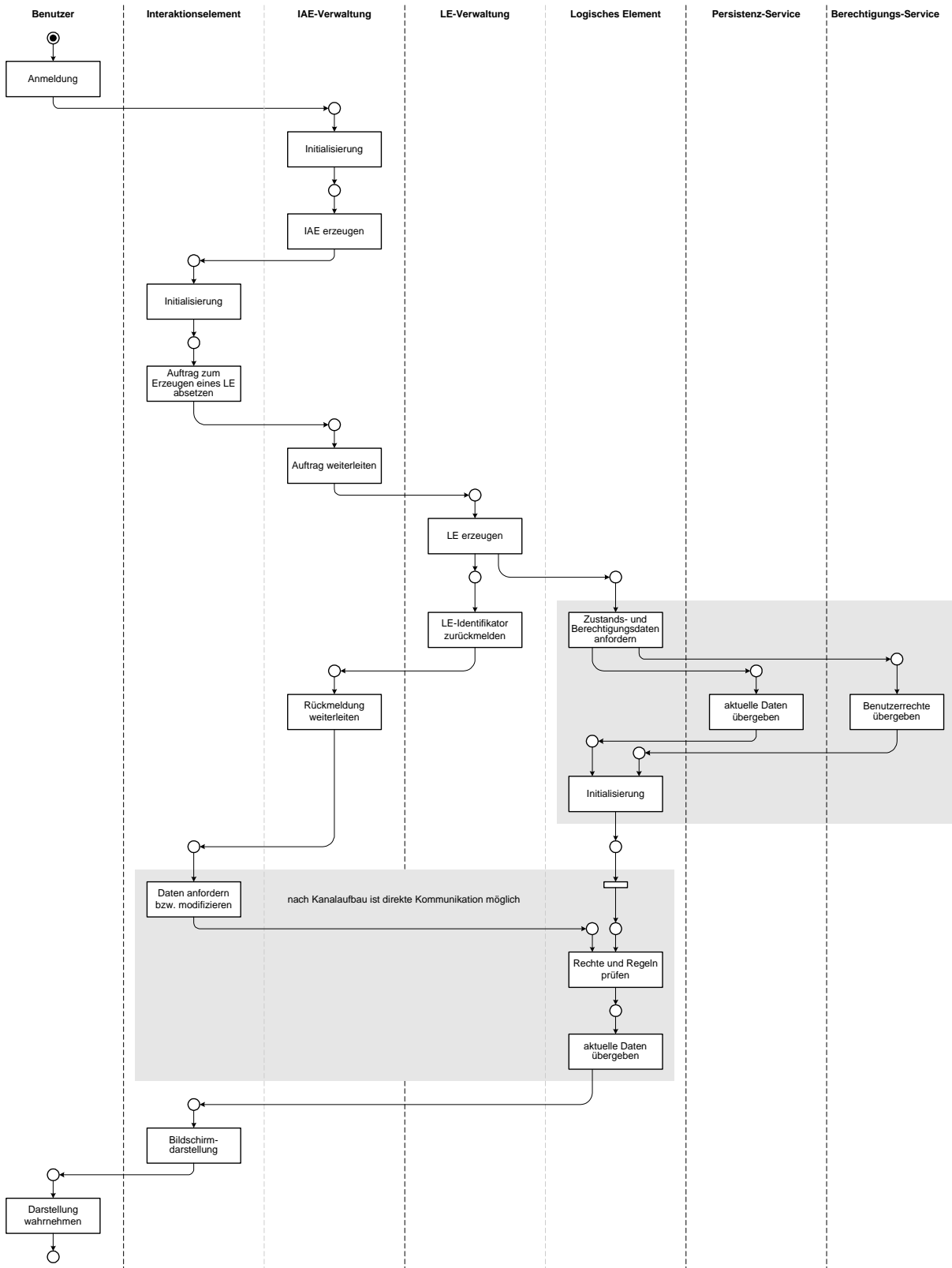
Der **Logische Service** stellt die Funktionalität des Basissystems nach außen dar und legt somit die Spezifikation der Tool-Bus-Schnittstelle fest. Er stellt – ebenso wie ein Tool – ein strukturvariantes System dar, in dem **Logische Elemente (LE)** entstehen und verschwinden können. Ein Logisches Element ist ein Stellvertreter für ein Basiselement. Die Strukturvarianzverwaltung sowie den Nachrichtenaustausch der LEs untereinander übernimmt die **LE-Verwaltung**. Das Verhalten der LEs ist in den Elementtypdefinitionen und dem Regelwerk des Logischen Service festgelegt, auf die die LE-Verwaltung Lesezugriff besitzt. Diese Daten werden zur Erzeugung von Logischen Elementen benötigt. Während die Elementtypdefinitionen die Grundkonzepte des Basissystems festlegen, dient das Regelwerk der flexiblen Verhaltensanpassung<sup>1</sup> vor der Laufzeit des Systems. So würden beispielsweise Containertypen einer Ablagestruktur in den Elementtypdefinitionen festgelegt, wogegen das Regelwerk die Festsetzung der zugehörigen Kapazitätsbeschränkungen enthalten würde.

### 5.3.2 Das Zusammenwirken der Komponenten des Gesamtsystems

Bild 30 zeigt den Ablauf innerhalb des Gesamtsystems bei einer Benutzer-Anmeldung. Zunächst identifiziert sich der Benutzer durch Angabe eines Benutzer-Identifikators bei der IAE-Verwaltung. Zur Benutzung eines Tools muß dieses durch Erzeugen von IAEs eine Bildschirmdarstellung aufbauen. Werden zur Darstellung eines IAE Daten eines Basiselements benötigt, so wird der Auftrag zur Erzeugung eines entsprechenden LE vom IAE über die IAE-Verwaltung an die LE-Verwaltung abgesetzt. Die LE-Verwaltung erzeugt ein LE unter Verwendung einer Elementtypdefinition und Teilen des Regelwerks. Dabei wird der Benutzer-Identifikator übergeben.

---

1. Statt Verhaltensanpassung wird häufig der Begriff "Customizing" verwendet.



**Bild 30:** Ablauf bei einer Benutzer-Anmeldung

Das LE fordert unmittelbar nach seiner Erzeugung selbständig Zustands- sowie Berechtigungsdaten bezüglich des Basiselements an und steht nach Erhalt dieser Daten als Auftragsempfänger für das auftraggebende IAE bereit. Das diesem Vorgang entsprechende Teilnetz ist in Bild 30 (rechts) durch Unterlegung mit einem grauen Rechteck hervorgehoben.

Das auftraggebende IAE erhält als Rückmeldung der LE-Verwaltung (über die IAE-Verwaltung) den Identifikator des Logischen Elements. Damit ist ein Kanal zwischen beiden aufgebaut und das IAE kann angebotene Dienste des LE nutzen um Daten anzufordern oder zu modifizieren. Innerhalb eines LE wird ein vom Interaktionselement abgesetzter Auftrag vor seiner Bearbeitung auf Berechtigungen des Benutzers sowie auf Einhaltung von Regeln hin überprüft.

Nach dem Erhalt von Daten des logischen Elements kann eine Darstellung grafischer Elemente auf dem Bildschirm erfolgen.

An dieser Stelle ist bereits das Schlüsselkonzept dieses ersten Entwurfs erkennbar. Es besteht darin, **benutzerspezifische** LEs zu erzeugen, die die Dienste des Basissystems abhängig vom angemeldeten Benutzer zur Verfügung stellen.

Auf diese Weise ist es einem Entwickler möglich, Interaktionselemente zu definieren – und somit komplette Tools zu erstellen –, ohne das Berechtigungswesen zu kennen oder sich darum kümmern zu müssen. Desweiteren ist durch die garantierte Einhaltung des Regelwerks innerhalb der LEs stets ein logisch korrekter Systemzustand sichergestellt. Auch diese Aufgabe wird einem Tool-Entwickler abgenommen.

Ein weiterer Vorteil dieses Konzepts besteht in der Entlastung des Persistenz-Service, wenn benötigte Daten bereits in LEs vorhanden sind. Durch Verteilung des Logischen Service auf mehrere parallel arbeitende und verteilt aufgestellte Rechner kann eine zusätzliche Performanzsteigerung erreicht werden. Man spricht dabei von Skalierbarkeit des Logischen Service.

Während eine vom Benutzer initiierte Datenmodifikation über A/R-Schnittstellen<sup>1</sup> vom IAE zum LE bzw. vom LE zum Persistenz-Service – im Aufbaubild nach unten – weitergeleitet wird, ist die Meldung einer Datenänderung in die entgegengesetzte Richtung nicht so einfach möglich. Das liegt daran, daß weder ein LE etwas über die Existenz eines IAE noch der Persistenz-Service etwas über die Existenz von LEs weiß. Dennoch müssen Änderungen persistenter Daten allen betroffenen LEs bzw. IAEs mitgeteilt werden, um eine Zustandsaktualisierung bzw. eine Aktualisierung der Bildschirmdarstellung zu gewährleisten.

Dieser Forderung wird nachgekommen, indem Änderungsmeldungen vom Datenverwaltungsakteur zur LE-Verwaltung und von dieser wiederum zur IAE-Verwaltung gesendet werden. Sowohl LE-Verwaltung als auch IAE-Verwaltung werden

---

1. abgekürzte Schreibweise für Auftrags- und Rückmelde-Schnittstelle

nach dem Empfang einer solchen Meldung Nachrichten an die betroffenen Elemente weiterleiten<sup>1</sup>. Anschließend aktualisieren LEs und IAEs ihren Zustand, indem sie aktuelle Daten anfordern.

Das zu entwickelnde Gesamtsystem sollte auch mit einem Sperr- sowie einem Transaktionswesen ausgestattet sein. Diese Dienste werden vom Persistenz-Service angeboten und von der LE-Verwaltung über die in Bild 29 dargestellte A/R-Schnittstelle genutzt.<sup>2</sup>

## 5.4 Anmerkung

Bei dem in diesem Kapitel vorgestellten System handelt es sich um einen ersten Entwurf, der bei der weiteren Entwicklung des Systems – in diesem Falle bei der Entwicklung eines Corporate Memory – helfen soll. Der vorgestellte Systemaufbau dient nicht als “Bauplan”, d.h. als Vorschrift zur System-Implementierung. Die Betrachtung der Architekturbeschreibung ist jedoch zur zwischenmenschlichen Kommunikation, zum Festhalten von Zwischenergebnissen sowie zur Weiterentwicklung des Systems bzw. der Systemarchitektur innerhalb einer Gruppe von Entwicklern äußerst nützlich, und kann damit zur Erstellung eines “Bauplans” beitragen.

- 
1. Zur gezielten Weiterleitung solcher Nachrichten muß über ein geeignetes Verfahren nachgedacht werden. Eine Möglichkeit wäre ein Publish-Subscribe-Verfahren, wie es in [DesignPatt 95] vorgestellt wird.
  2. Ein erster Entwurf zur Integration von Sperr- und Transaktionswesen kann in [Langhauser 97] nachgelesen werden.

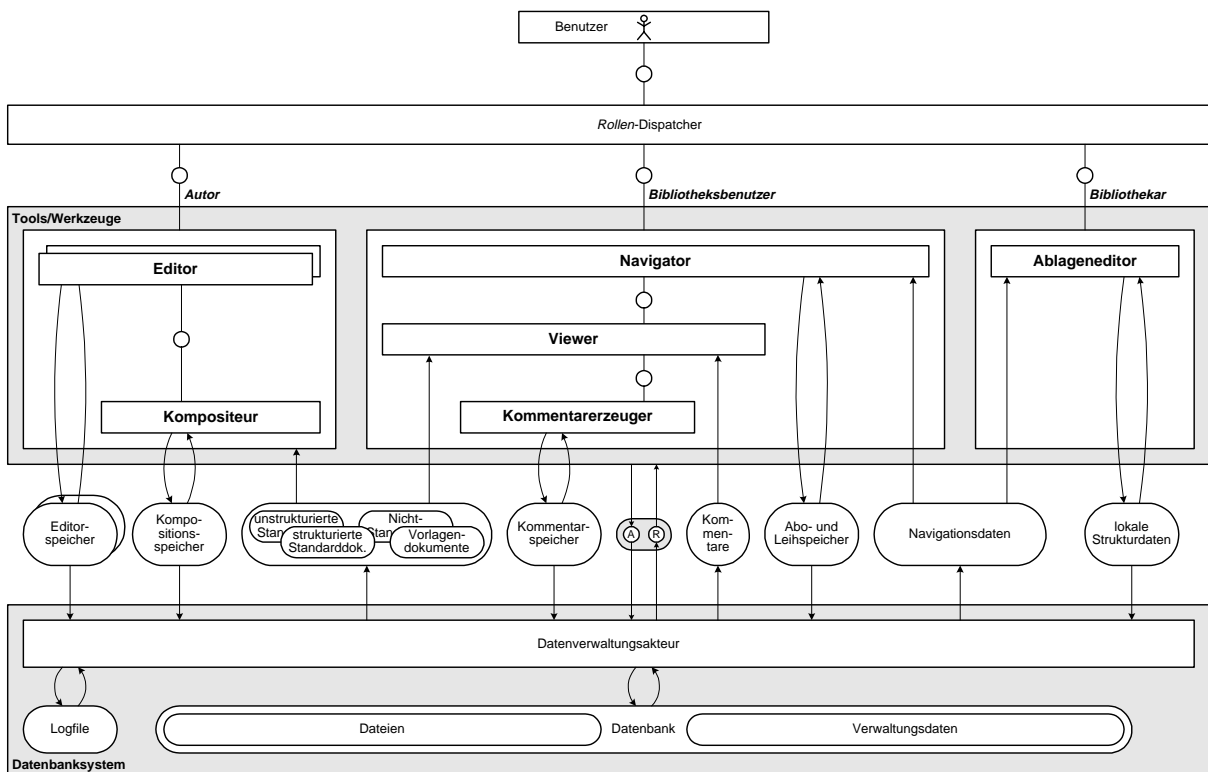
# Anhang



# Anhang A: Benutzerrollen und Werkzeuge eines Corporate Memory

## A.1 Die Benutzerrollen

Beim Arbeiten mit einem Corporate Memory kann ein Benutzer (siehe Bild 31 oben) mehrere Rollen spielen. Das Wechseln zwischen den verschiedenen Rollen **Autor**, **Bibliotheksbenutzer** und **Bibliothekar**, ermöglicht ein **Rollen-Dispatcher**. Beim Spielen dieser Rollen stehen einem Benutzer unterschiedliche Werkzeuge – auch als Tool bezeichnet – zur Verfügung.



**Bild 31:** Benutzerrollen und Werkzeuge eines Corporate Memory

## A.2 Das Datenbanksystem

Bevor auf diese Werkzeuge im einzelnen eingegangen wird, soll das **Datenbanksystem** betrachtet werden:

Die Datenbank ist in zwei Bereiche unterteilt. Sie enthält zum einen einen Speicher für Dateien, die durch ein Filesystem verwaltet werden, zum anderen sind in der Datenbank Verwaltungsdaten abgelegt, die man sich als eine Sammlung von Karteikarten – sortiert in Karteikästen – vorstellen kann.<sup>1</sup>

Ein **Datenverwaltungsakteur** ermöglicht den Zugriff auf die Datenbank. Er ist außerdem für das Mitschreiben eines Protokolls zuständig, das als Logfile abgelegt wird. Für die unterschiedlichen Werkzeuge kann der Datenverwaltungsakteur Navigationsdaten, Kommentare sowie Dokumente unterschiedlichen Typs zur Verfügung stellen. Alle diese Daten, auf die die Werkzeuge ausschließlich Lesezugriff besitzen, werden vom Datenverwaltungsakteur aus dem Inhalt der Datenbank generiert.

## A.3 Die Schnittstelle zwischen Werkzeugen und Datenbanksystem

Jedes der Werkzeuge ist über eine Auftrags-Rückmelde-Schnittstelle mit dem Datenverwaltungsakteur, d.h. mit dem Datenbanksystem, verbunden. Somit können von jedem Werkzeug Aufträge abgesetzt und Daten angefordert werden.

Zwischen den Werkzeugen und dem Datenbanksystem besteht eine weitere Verbindung. Mit Ausnahme des Viewers besitzen alle einen modifizierenden Zugriff auf einen eigenen lokalen Speicher. Die darin abgelegten Daten können dem Datenbanksystem übergeben werden.

Diese Art der Schnittstelle weist darauf hin, daß es sich mit Ausnahme des Viewers um Werkzeuge zur Erzeugung bzw. Modifikation von Daten handelt.

---

1. Auf die Strukturierung der Daten soll an dieser Stelle nicht eingegangen werden. Eine ausführliche Beschreibung dieser Struktur findet der Leser in [Langhauser 97].



## A.4 Die Werkzeuge

Der Benutzer hat in der Rolle des Autors mehrere **Editoren** (Editier-Werkzeuge) zur Verfügung. Mit diesen kann der Benutzer Dokumente dreier unterschiedlicher Typen erstellen:<sup>1</sup>

- Vorlagendokumente (Dokumente mit Platzhaltern; systemeigenes Format)
- unstrukturierte Standarddokumente (Dokumente ohne Platzhalter; systemeigenes Format)
- Nicht-Standarddokumente (Dokumente, die mit Editoren erstellt werden, die nicht das systemeigene Format unterstützen)

Das in Bearbeitung befindliche Dokument ist im Editorspeicher abgelegt. Ist das Editieren beendet, kann die Übergabe an das Datenbanksystem erfolgen.

Zur Erstellung von Dokumenten kann auch auf bereits vorhandene zurückgegriffen werden. Dazu wird die Kopie eines Dokuments in den Editorspeicher geholt, modifiziert und anschließend der gesamte Inhalt an das Datenbanksystem übergeben.

Zur Erstellung eines strukturierten Standarddokuments dient der **Kompositeur**. Zur Komposition wird ein Vorlagendokument ausgewählt und eine Kopie davon im Kompositionsspeicher abgelegt. Die in dieser Kopie enthaltenen Platzhalter können mit beliebigen (statisch-flächigen) Dokumenten belegt werden. Ist die Belegung abgeschlossen, wird der Inhalt des Kompositionsspeichers dem Datenverwaltungsakteur übergeben.

In der Rolle des Bibliotheksbenutzers stehen drei Werkzeuge zur Verfügung. Mit dem **Navigator**, der Lesezugriff auf die bereitgestellten Navigationsdaten besitzt, kann man sich in der Bibliothekswelt bewegen. Mit diesem Werkzeug ist es auch möglich, Dokumente, Ordner oder komplette Sektionen aus der Bibliothek ins Büro zu kopieren. Der Navigator ermöglicht das Anfordern von Mitteilungen über eine gewünschte Auswahl von neu erschienenen Dokumenten. Dazu dient der Abo- und Leihspeicher, den man sich als in der Bibliothek mitgeführten Einkaufskorb, bzw. als Abonnement-Antrag vorstellen kann. Der Navigator ermöglicht einem Benutzer auch das Bewegen in seinem eigenen Büro. Beim Ortswechsel von der Bibliothek zum Büro wird der Inhalt des Leihspeichers an den Datenverwaltungsakteur übergeben. Dem Benutzer werden kopierte Dokumente im Büro angezeigt.

Zum Betrachten von Dokumenten - inklusive der daran angehängten Leserkommentare - dient der **Viewer**. Mit diesem ist weder eine Erzeugung noch eine Bearbeitung von Daten möglich.

---

1. Eine eingehendere Beschreibung findet der Leser in [Langhauser 97].

Neben dem Lesen von Dokumenten ist es auch möglich, auf diesen Leserkommentare zu plazieren. Zum Erstellen solcher Kommentare dient der **Kommentarerzeuger**. Der Wunsch des Plazierens und den Ort, an dem der Leserkommentar angebracht werden soll teilt ein Benutzer dem Viewer mit. Dieser ruft den Kommentarerzeuger auf und übergibt ihm die Koordinaten. Ein in Bearbeitung befindlicher Kommentar ist im Kommentarspeicher abgelegt und wird nach der Fertigstellung an das Datenbanksystem übergeben.

In der Rolle als Bibliothekar steht einem Benutzer ein einziges Werkzeug, der **Ablageditor** zur Verfügung. Mit ihm können Strukturen in der Bibliothek erstellt oder verändert werden. Neue bzw. modifizierte Strukturen werden zunächst in den lokalen Strukturdaten abgelegt. Während mit dem Ablageditor eine Navigation in den, durch die lokalen Daten beschriebenen, Strukturen möglich ist, sind diese für Bibliotheksbenutzer, die mit dem Navigator arbeiten, nicht sichtbar. Im Bild ist dies dadurch zu erkennen, daß der Navigator keinen Zugriff auf die lokalen Strukturdaten besitzt. Nach der Übergabe dieser Daten an das Datenbanksystem können daraus neue Navigationsdaten generiert werden. Erst danach sind die neuen Strukturen für alle Benutzer sichtbar.

## A.5 Beispiele für unterschiedliche Aufgaben und Tätigkeiten von Benutzern

### **Bibliotheksbenutzer:**

Navigation in der Bibliothek; Betrachten von Dokumenten in der Bibliothek; Ausleihen von Dokumenten; Modifizieren der Ablagestruktur im Benutzerbüro; Erstellen von Wegweiserdokumenten; Einsortieren von Dokumenten in Ablagestruktur; Lesen von Dokumenten im Büro (bspw. auf dem Schreibtisch); Erstellen und Anbringen von Kommentaren an Dokumente.

### **Autor:**

Erstellen oder Suchen von Vorlagen; Erstellen von Dokumenten; Belegen von Platzhaltern einer Vorlage mit Dokumenten.

### **Bibliothekar:**

Erweitern der Ablagestruktur; Einsortieren von Dokumenten; Erstellen von Dokumenten als Inhaltsbeschreibung von Containern

# Anhang B: Bemerkungen zur Erstellung des Prototyps

## B.1 Ein interaktiver Film als Prototyp

Der **Prototyp** der Navigationskomponente dient in erster Linie zur Entwicklung von Bedienkonzepten und der damit verbundenen Gestaltung der Oberfläche. Aus diesem Grund sollte die Implementierung des Prototyps auf einfache Weise möglich, und bereits während der Erstellungsphase in Teilen vorführbar sein.

Zur reinen Vorführung hätte die Erstellung eines Films ausgereicht. Um jedoch auch die Bedienbarkeit beurteilen zu können, entschied sich der Autor für die Erstellung eines **interaktiven Films**. Ein Film wird als interaktiv bezeichnet, wenn dessen Ablauf durch das Einwirken des Betrachters beeinflusst werden kann. Die Möglichkeiten der Beeinflussung werden bei der Filmerstellung festgelegt.

Ein interaktiver Film kann so aufgebaut sein, daß dem Betrachter der Eindruck vermittelt wird, eine bereits fertiggestellte Applikation zu bedienen. Ein solcher Film dient als Prototyp, dessen Bedienung durch den Benutzer in Wirklichkeit eine Beeinflussung des Filmablaufs darstellt.

## B.2 Die eingesetzten Produkte

Zur Implementierung des Prototyps wurden mehrere Produkte eingesetzt:

**Corel Draw** (Version 7) wurde zur Erstellung von Zeichnungen und Grafiken benutzt. *Corel Draw* ist ein vektor-orientiertes Zeichen-Werkzeug, das eine sehr umfangreichen Funktionalität bietet.

Alle erstellten Bilder wurden in Pixel-Grafiken umgewandelt und mit dem als Shareware erhältlichen Bildbearbeitungs-Werkzeug **PaintShop Pro** nachbearbeitet. Diese Nachbearbeitung beschränkte sich im wesentlichen auf Farb- und Ausschnittskorrekturen von Grafiken.

Die auf diese Weise entstanden Grafiken wurden zusammen mit dem Werkzeug **Director** (Version 5) der Firma *Macromedia* zur Erstellung des interaktiven Films benutzt.

## B.3 Grundbegriffe und -konzepte von *Director*

Im folgenden werden grundlegende Begriffe<sup>1</sup> und Konzepte vorgestellt, die für das Arbeiten mit *Director* unerlässlich sind.<sup>2</sup>

*Director* gestattet die Erstellung mehrerer **Besetzungen**. Eine Besetzung beinhaltet eine Ansammlung von Komponenten, wie Grafiken, Texte oder Töne, die als **Darsteller** bezeichnet werden. Die semantische Gruppierung von Darstellern und deren Verwaltung in unterschiedlichen Besetzungen erhöht die Übersichtlichkeit beim Arbeiten mit *Director*.

Der Ablauf eines Films wird im sogenannten **Drehbuch** festgelegt, wovon Bild 32 einen Ausschnitt zeigt. Das Drehbuch hat die Form einer Matrix. Eine Spalte dieser Matrix wird als **Bild**, eine Zeile als **Kanal** bezeichnet. Beim Abspielen eines Films wird eine Spalte nach der anderen durchlaufen. Der Inhalt einer Spalte legt fest, was zu einem bestimmten Zeitpunkt auf dem Bildschirm zu sehen bzw. zu hören ist. Jede **Zelle** der Drehbuchmatrix nimmt genau ein Exemplar eines Darstellers – einen sog. **Sprite** – auf. Von einem Darsteller können beliebig viele Sprites erzeugt und mit unterschiedlichen Attributen versehen werden. Zu diesen Attributen gehören beispielsweise die Lautstärke eines Tons, die Größe oder Position einer Grafik oder die Schriftfarbe eines Textes. Das Vorhandensein von Kanälen ermöglicht das nebenläufige Agieren mehrerer Sprites auf dem Bildschirm.

Zur Positionierung und Festlegung der Größe eines Sprites dient die **Bühne**. Sie ist ein Bereich auf dem Bildschirm, der die Darstellungsgröße des zu erstellenden Films besitzt. Die Bühne dient neben der Komposition auch zum Betrachten einzelner Bilder.

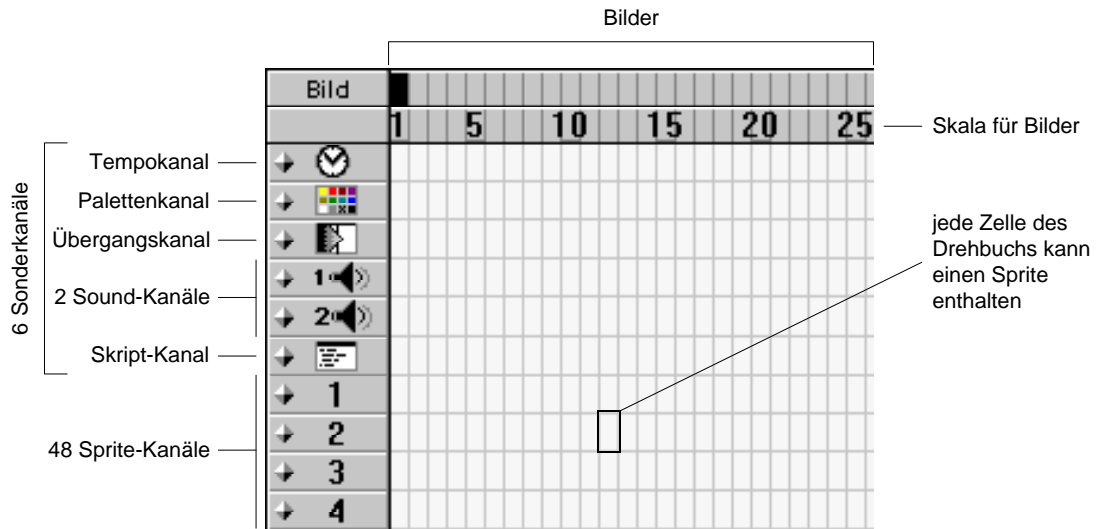
Um beispielsweise die Bewegung eines Objekts zu simulieren, werden Sprites eines Darstellers in aufeinanderfolgenden Bildern an benachbarte Positionen auf der Bühne angeordnet. Beim Abspielen des Films mit ausreichend schneller Bildfolge<sup>3</sup> entsteht beim Betrachter der Eindruck von Bewegung auf dem Bildschirm.

---

1. Die hier benutzten Begriffe stammen aus den mitgelieferten Handbüchern ([MDBenutz 97], [MDEinsteig 97]) sowie [Welsch 97]. Man begegnet diesen Begriffen auch bei der Bedienung der Anwendung (deutsche Version).

2. Eine übersichtlich dargestellte Einführung ist auch unter [Rosenzweig 97] zu finden.

3. Die Bildfolge sollte bei mindestens 10-15 Bildern pro Sekunde liegen.



**Bild 32:** Ausschnitt aus einem Drehbuch, wie es in *Director* angezeigt wird

Neben 48 **Sprite-Kanälen**<sup>1</sup> für grafische Elemente stehen sechs **Sonderkanäle** zur Verfügung. Diese können zur Veränderung der Ablaufgeschwindigkeit, zur Farbveränderung, zur Festlegung von Übergangseffekten beim Bildwechsel, zur Tonausgabe (zwei getrennte Kanäle) sowie zur Ausführung bestimmter Anweisungsfolgen eingesetzt werden (vgl. Bild 32).

Eine Anweisungsfolge wird als **Skript** bezeichnet und in der *Director*-eigenen Sprache **Lingo**<sup>2</sup> formuliert. Skripte werden – wie Elemente anderer Typen auch – als Darsteller in Besetzungen abgelegt.

Die Gruppe der am häufigsten benutzten Anweisungen sind die Sprunganweisungen. Bei der Ausführung einer Sprunganweisung wird der Ablauf des aktuellen Films mit dem angegebenen Bild fortgesetzt. Dabei kann ein Bild über seine Nummer – das Drehbuch enthält dafür eine Skala – oder über einen einem Bild zugewiesenen Bezeichner (Bildbezeichner) referenziert werden.

1. Sprites in Kanälen mit kleinerer Nummer werden vor Sprites mit größerer Nummer angezeigt. Auf diese Weise können Sprites durch andere verdeckt werden.
2. Einzelheiten zu Lingo können [MLEinsteig 97] und [MLLexikon 97] entnommen werden.



# Anhang C: Der Aufbau des Prototyps

## C.1 Die Besetzungen

In der nachfolgenden Tabelle sind alle benutzten Besetzungen aufgelistet und die darin enthaltenen Darsteller beschrieben.

<b>Besetzung</b>	<b>Beschreibung</b>
Hintergrund	enthält ein Hintergrundbild und eine Maske, die das Layout der Navigationskomponente festlegen; Die Maske ist eine graue Fläche, bei der die Bereiche der Fenster (View-, Pfad- und Informations-) ausgespart sind; Sie dient dazu, Sprites an bestimmten Stellen zu verdecken; hier ist außerdem eine Grafik enthalten, die die Bedienelemente darstellt
Rahmen	enthält rechteckige Rahmen, die Fasen an den Rändern der Fenster (View-, Pfad- und Informations-) darstellen
Skripts für Sprünge zwischen Bildern	enthält eine Rechteckfläche, die auf dem Bildschirm unsichtbar dargestellt wird, sowie Skripts mit unterschiedlichen Sprunganweisungen; diese Sprunganweisungen führen dazu, daß der Ablauf des Films an einer anderen Stelle fortgesetzt wird
verkleinerte Graustufen-Bilder und Markierungen	enthält verkleinerte Graustufen-Bilder und Markierungen; die Markierungen werden durch Kreise um eine Ikone dargestellt
View-Bilder	enthält Grafiken für mögliche Inhalte des View-Fensters

<b>Besetzung</b>	<b>Beschreibung</b>
Klänge, Geräusche und Effekte	enthält Geräusch eines Fotoapparats bei der Aufnahme und einen Ton, der den Zoom-Effekt unterstützt;
Zoom-Kästchen	enthält rechteckige Rahmen zur Implementierung des Zoom-Effekts
Pfeiltasten des Pfad-Fensters	enthält Bilder von Pfeiltasten, die am unteren und oberen Rand des Pfad-Fensters dargestellt werden
Skripts zur Bewegung des Pfad-Ausschnitts	enthält Skripts mit Sprunganweisungen zu Bildsequenzen, die die Verschiebung des Pfad-Ausschnitts animieren
Informations-Fenster	enthält Grafiken, die den Inhalt des Informations-Fensters darstellen



## C.2 Die Kanalbelegung

In der nachfolgenden Tabelle sind alle verwendeten Kanäle<sup>1</sup> aufgelistet. Der Inhalt der Kanäle wird beschrieben. Diese Tabelle ist beim Lesen des Drehbuchs nützlich.

<b>Kanal</b>	<b>Beschreibung</b>
Übergangskanal	wird zur Implementierung der Überblendeffekte im View-Fenster eingesetzt
Sound-Kanäle	Abspielen des Sounds eines Fotoapparats bei der Aufnahme und Abspielen des zur Unterstützung des Zoom-Effekts benutzten Tons
Skript-Kanal	Anweisung zur Ausführung von Sprüngen zu anderen Bildern; das Unterbrechen des Ablaufs bei einem Bild; das Ausführen des Zoom-Effekts. Zur Festlegung der Anfangs- und Endpunkte des Zoom-Effekts dienen die Rechtecke in den Kanälen 2 und 3.
SK 1	Grafik, die ihre Position während des gesamten Ablaufs des Films nicht verändert, dient als Hintergrund. Sie legt das Layout der Navigationskomponente fest.
SK 2-3	zwei transparente und somit vom Betrachter nicht sichtbare Rechtecke, die die Anfangs- und Endpositionen der Zoom-Effekte festlegen
SK 4	Grafik, die den jeweiligen Inhalt des View-Fensters festlegt
SK 5-12	Inhalt des Pfad-Fensters; Implementierung von Verschiebungen des Ausschnitts. Dabei müssen sowohl die Graustufen-Bilder als auch die zugehörigen Markierungen verschoben werden.
SK 13-15	Grafikrahmen, die die Einträge des Pfad-Fensters mit gefasteten Kanten erscheinen lassen. Diese Rahmen werden mit den Einträgen verschoben.
SK 16-21	Implementierung der Bewegung von Pfad-Fenster-Einträgen zwischen View- und Pfad-Fenster
SK 22	Grafik, die den Inhalt des jeweiligen Informations-Fensters anzeigt

1. Die Abkürzung "SK" steht für den Begriff "Sprite-Kanal".

<b>Kanal</b>	<b>Beschreibung</b>
SK 23	Grafik, in der rechteckige Flächen ausgespart sind, verdeckt Sprites, die außerhalb der festgelegten Fenster positioniert sind
SK 24-26	Grafikrahmen zur Darstellung der Fasen von View-, Informations- und Pfad-Fenster
SK 27	Darstellung der Bedienelemente
SK 28-29	Pfeiltasten des Pfad-Fensters
SK 30-33	Implementierung der Umschaltmöglichkeiten im Informations-Fenster durch Klicken auf Beschriftungsschilder
SK 34-37	Transparente Rechtecke, die Darstellungsbereiche anderer Sprites überlagern. Beim Klicken des Benutzers auf eines dieser Rechtecke werden definierte Skripts abgewickelt. Auf diese Weise sind Interaktionen des Benutzers mit dem Film möglich.

# Anhang D: Der Ablauf des interaktiven Films

Der Ablauf eines Films wird implementiert durch einen ausreichend schnellen Wechsel von einzeln dargestellten Bildern. Bei einem interaktiven Film sind nicht ständig Bildwechsel zu sehen. Es kommt des öfteren zu **Pausen**, wobei die Bildschirmanzeige für einen längeren Zeitraum konstant bleibt und auf eine Aktion des Benutzers gewartet wird.

In den Ablaufbildern, die in diesem Teil des Anhangs enthalten sind, werden solche Pausen durch Stellen eines Petrinetzes dargestellt. Diese sind mit der entsprechenden – auch im Drehbuch enthaltenen – Bildbezeichnung beschriftet. **Animierte Übergänge**, d.h. durch Darstellung einer Folge von Bildern angezeigte Übergänge werden durch beschriftete Transitionen symbolisiert. Diese Transitionen enthalten die Bezeichnung<sup>1</sup> des ersten Bildes der Sequenz. Zwischen zwei konstant angezeigten Bildern kann auch ein **abrupter Wechsel** der Bildschirmdarstellung erfolgen. Solch ein Übergang wird durch eine NOP-Transition zwischen zwei Stellen repräsentiert.

## D.1 Die Views

Mit dem entwickelten Prototyp muß es nicht möglich sein, in einer baumstrukturierten Ablage zu navigieren. Es ist ausreichend, in jeder Hierarchieebene der Ablagestruktur ein Beispiel für eine View, d.h. einen exemplarischen Pfad, sehen zu können. Der hier implementierte Beispiel-Pfad entstand aus dem Inhalt des WWW-Servers der *Siemens AG*. Die dort gefundenen Bezeichnungen, die als Verweise zu anderen Seiten dienen, wurden zur Beschriftung der Grafiken benutzt, die den Inhalt des View-Fensters darstellen.

---

1. Für die Bezeichnungen der Bilder sowie die der Transitionen wurde wegen Platzmangels im Drehbuch eine abgekürzte Schreibweise gewählt. Dabei wurden häufig nur die Anfangsbuchstaben der Bezeichnungen von Petrinetzstellen benutzt.

Die acht implementierten Views (*Land, Campus, Gebäude, Etage, Saal, Regal* und *Brett*) sind durch die Stellen des Petrinetzes in Bild 33 symbolisiert. Die in der linken Hälfte des Bildes dargestellten Grafiken zeigen die Bildschirmdarstellungen des Films für diese Views. An der Anfangsbelegung im Petrinetz erkennt man, daß zu Beginn des Films die Land-View dargestellt wird.

## D.2 Die Navigation zwischen Views benachbarter Hierarchieebenen

Die Übergänge bei der Navigation zwischen den Views benachbarter Hierarchieebenen werden animiert. Der Ablauf einer solchen Animation entspricht dem Schalten einer Transition des Petrinetzes aus Bild 33.

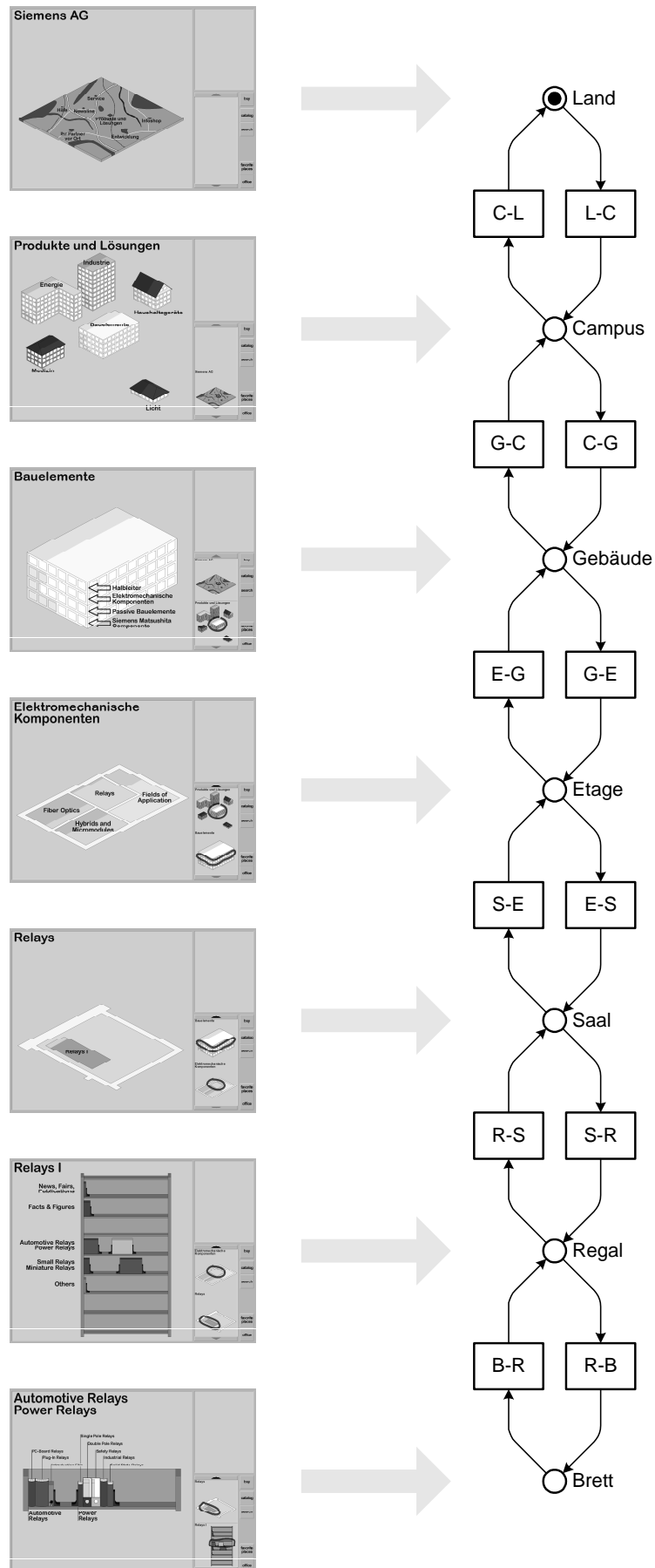
Transitionen, die eine Marke nach unten transportieren, entsprechen der Animation, wie sie in Bild 19 auf Seite 26 dargestellt ist. Die in Gegenrichtung transportierenden Transitionen entsprechen der Animation, die durch Bild 20 auf Seite 27 dargestellt ist.

## D.3 Das Überspringen einer Hierarchieebene nach unten

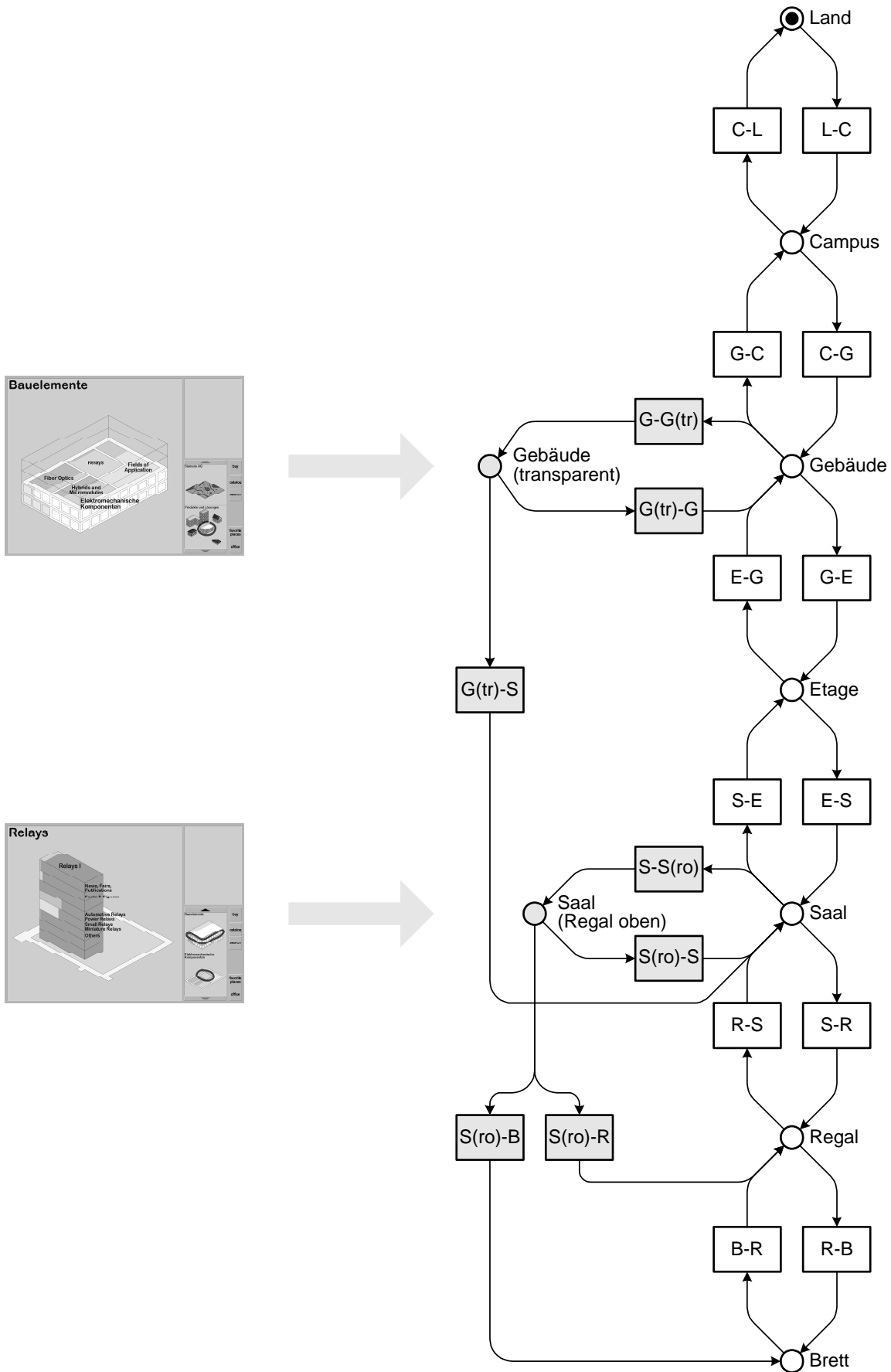
Bei der Navigation in der Ablagestruktur nach unten kann sowohl die Etagen- als auch die Regal-View übersprungen werden. Dennoch benötigen diese Übergänge einen **Zwischenschritt**, bei dem auf das Agieren des Benutzers gewartet wird. Das Petrinetz in Bild 34 enthält zwei grau markierte Stellen, die diesen Zwischenschritten entsprechen. Die links davon angeordneten Grafiken zeigen die jeweils zugehörige Bildschirmdarstellung.

Beim Übergang von der Gebäude- zur Saal-View, der durch einen Mausklick auf die Etage *Elektromechanische Komponenten* bei gedrückt gehaltener Umschalttaste initiiert wird, wird die Gebäudehülle teilweise transparent und der Inhalt der angewählten Etage sichtbar. Dieser Zwischenschritt wird durch die Stelle mit der Beschriftung *Gebäude (transparent)* symbolisiert. Der Benutzer hat an dieser Stelle die Möglichkeit, durch einen Mausklick neben den Gebäudeumriß zurück zur Gebäude-View zu gelangen. Erst nach Anwählen des Saals *Relays* wird der Übergang zu diesem Saal vollzogen.

Beim Zwischenschritt, der beim Übergang von der Saal- zur Brett-View – initiiert durch einen Mausklick auf das Regal *Relays I* bei gedrückt gehaltener Umschalttaste – durchlaufen wird, wurde das durch einen Quader dargestellte Regal nach oben geschoben. Der Benutzer hat nun drei Möglichkeiten, seine Navigation fort-



**Bild 33:** Navigation zwischen Views benachbarter Hierarchieebenen



**Bild 34:** Überspringen einer Hierarchieebene nach unten

zusetzen. Er kann neben das Regal klicken, um zur Saal-View zurückzukehren, oder er kann das Brett mit der Bezeichnung *Automotive Relays Power Relays* anwählen, um zu dessen View zu gelangen. Die dritte Alternative besteht im Klicken auf die Deckfläche des Quaders. In diesem Fall gelangt der Benutzer zur entsprechenden Regal-View.

Alle Bilder bzw. Bildsequenzen, die zur Simulation der oben genannten Navigationsmöglichkeiten zusätzlich zusammengestellt werden mußten, entsprechen den grau markierten Knoten im Petrinetz von Bild 34.

## **D.4 Das Verschieben des angezeigten Pfad-Ausschnitts**

Die Verschiebung des angezeigten Pfad-Ausschnitts ist gleichbedeutend mit der Verschiebung des aktuell sichtbaren Pfad-Fenster-Inhalts in die entgegengesetzte Richtung.

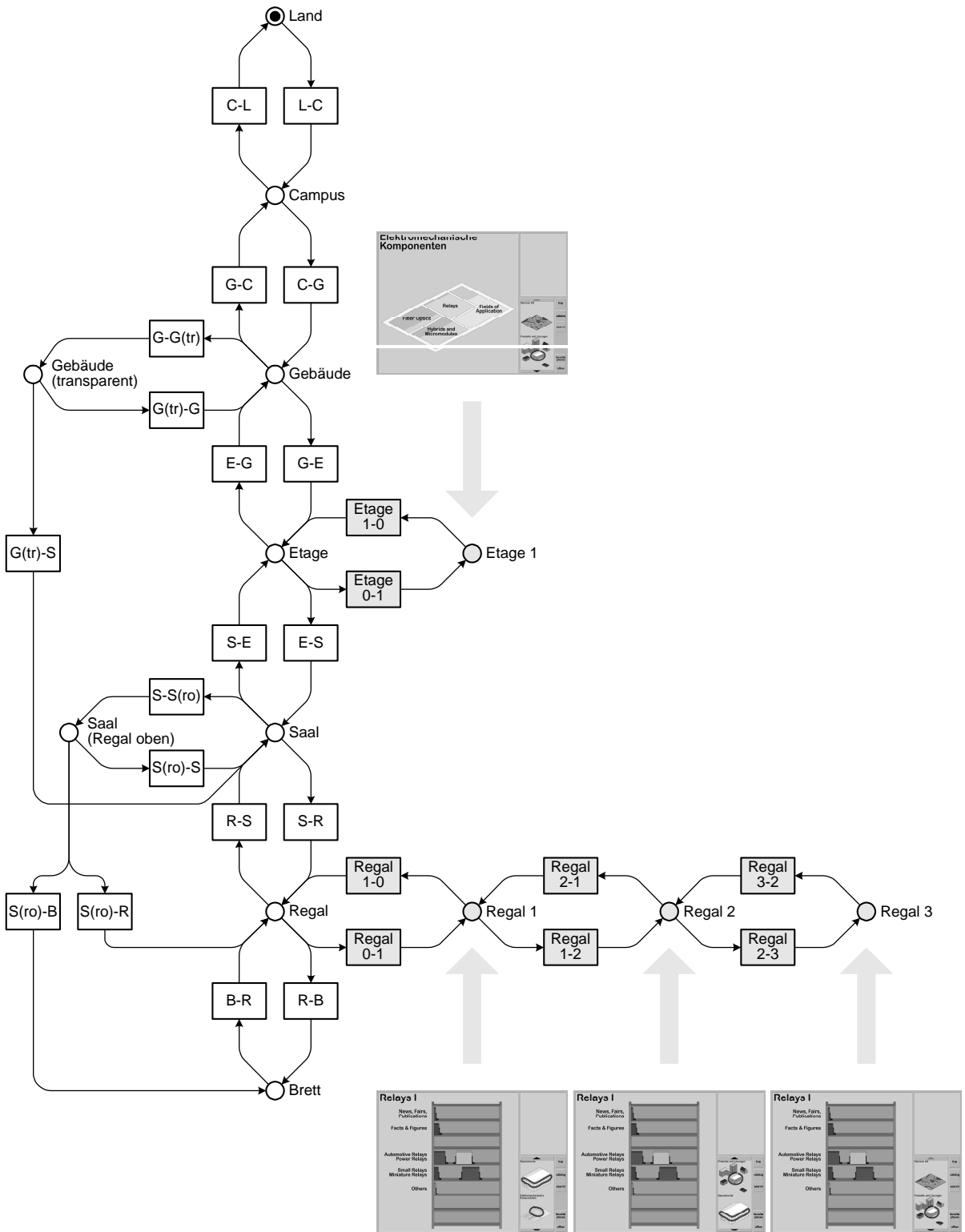
Die Möglichkeit zur Verschiebung des Pfad-Ausschnitts wurden im Prototyp nicht bei allen Views implementiert. Eine Implementierung wurde lediglich bei der Etagen- und Regal-View durchgeführt.

Die im Petrinetz in Bild 35 grau markierten Transitionen entsprechen jeweils der Verschiebung des Inhalts um die Höhe eines Eintrags. Ein solcher Verschiebungsschritt wird durch Drücken einer der beiden Pfeiltasten des Pfad-Fensters ausgelöst.

In der Etagen-View kann der Pfad-Fenster-Inhalt um einen, in der Regal-View um bis zu drei Einträge nach unten geschoben werden. Alle grau markierten Petrinetzstellen entsprechen einer Bildschirmdarstellung – durch die entsprechenden Bildschirmfotos in Bild 35 angezeigt – bei der der Inhalt des Pfad-Fensters um mindestens einen Verschiebungsschritt bewegt wurde. Die jeweilige Anzahl der Verschiebungsschritte ist in der Bezeichnung der Stellen enthalten.

## **D.5 Das Überspringen von Hierarchieebenen nach oben**

Bei der Navigation ist das Überspringen beliebig vieler Hierarchieebenen in Richtung der Wurzel des Ablagestrukturbaums möglich. Im Prototyp sind drei Navigationsschritte implementiert, wobei mindestens eine Hierarchieebenen übersprungen wird. Die zugehörigen Übergänge des interaktiven Films sind in Bild 36 durch grau markierte Transitionen dargestellt und entsprechen jeweils einer durch Bild 20 auf Seite 27 festgelegten Animation.



**Bild 35:** Verschieben des angezeigten Pfad-Ausschnitts



Folgende Übergänge wurden implementiert:

- Das Anwählen des oberen Pfad-Fenster-Eintrags in der Brett-View führt zur Saal-View. (Transition mit der Bezeichnung *B-S* in Bild 36)
- Nach zwei Verschiebungsschritten (Pfad-Fenster) führt das Anwählen des unteren Pfad-Fenster-Eintrags zur Gebäude-View. (*R2-G*)
- Nach drei Verschiebungsschritten (Pfad-Fenster) führt das Anwählen des oberen Pfad-Fenster-Eintrags zur Land-View. (*R3-L*)

## D.6 Die Anzeige im Informations-Fenster

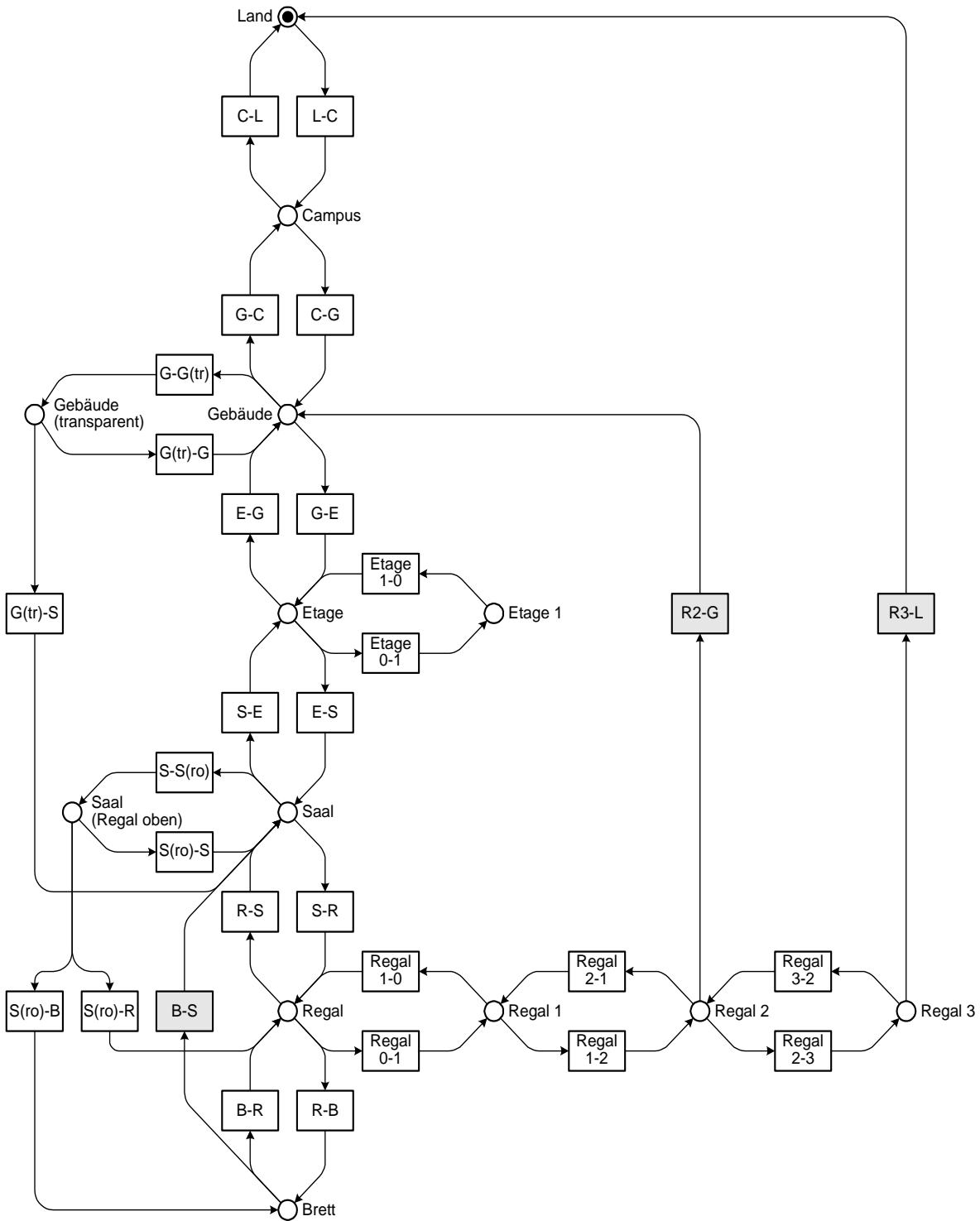
Die Anzeige von Informationen im Informations-Fenster zu einer im View-Fenster ausgewählten Ikone, ist im Prototyp lediglich in Land, Campus, Etagen oder Brett möglich. Die Darstellung dieser Information nach dem Auswählen einer Ikone im View-Fenster ist jeweils durch ein abruptes Wechseln des dargestellten Bildes implementiert.

Bild 37 enthält vier Teilnetze, die Verfeinerungen der Petrinetzstellen *Land*, *Campus*, *Etage* und *Brett* aus Bild 36 darstellen. Die Stellen werden durch grau hinterlegte Flächen angezeigt<sup>1</sup>. In jeder dieser vier Flächen ist eine weiß gefüllte Stelle enthalten, die einer Bildschirmanzeige bei leerem Informations-Fenster entspricht. Im Gegensatz dazu korrespondieren grau gefüllte Stellen einer Bildschirmdarstellung mit ausgefülltem Informations-Fenster.

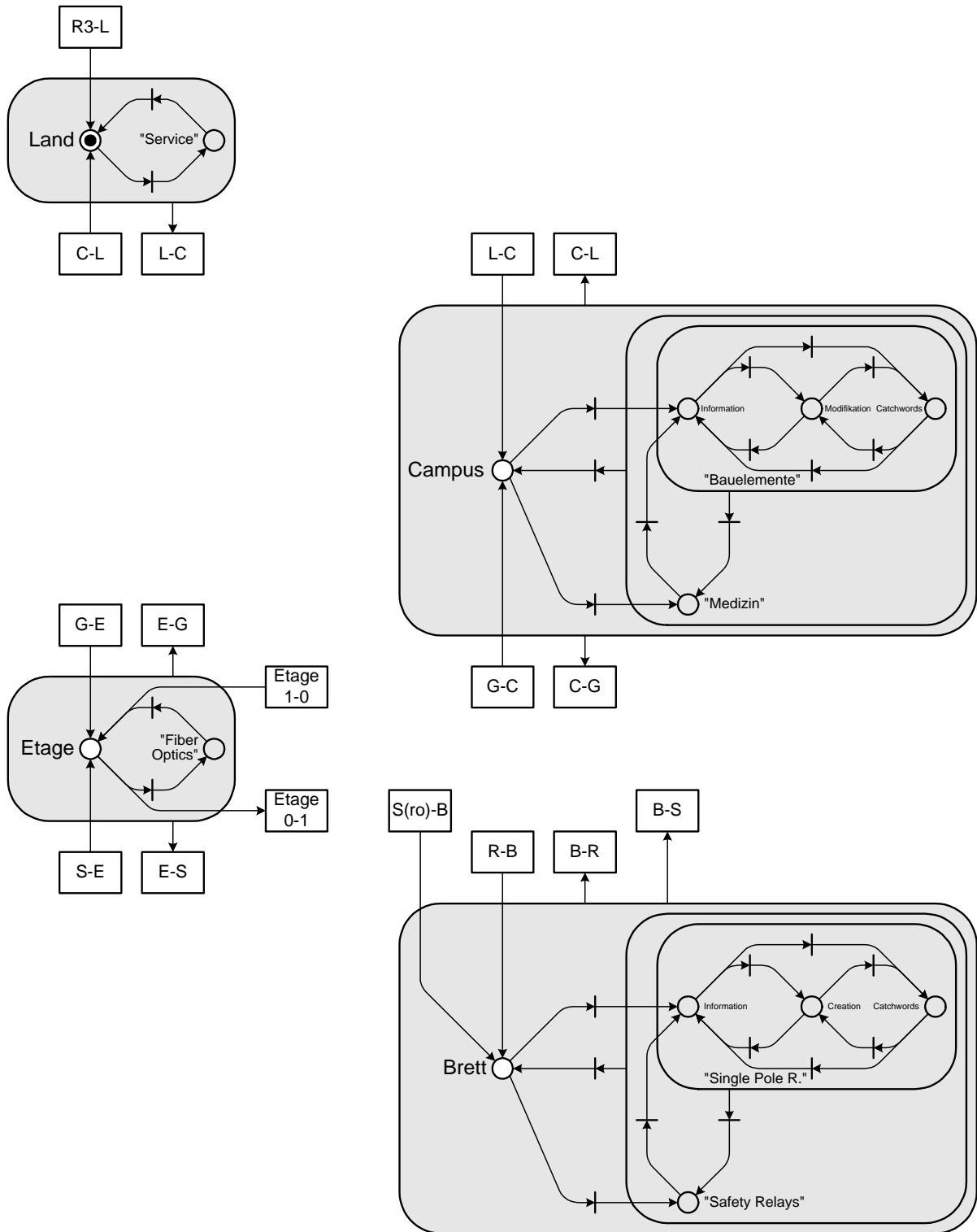
In den Teilnetzen erkennt man, daß sowohl in der Land- als auch in der Etagen-View nur jeweils eine Ikone angewählt werden kann. In der Campus- und der Brett-View dagegen können jeweils zwei Ikonen angewählt werden. Zusätzlich kann im Informations-Fenster, beim Auswählen einer der beiden Ikonen zwischen den Anzeigen unterschiedlicher Informationsgruppen gewechselt werden.

---

1. Die außerhalb der grauen Flächen plazierten Transitionen wurden aus Bild 36 übernommen, um den Bezug zwischen beiden Bildern zu verdeutlichen.



**Bild 36:** Überspringen von Hierarchieebene nach oben



**Bild 37:** Anzeige im Informations-Fenster



# Literaturverzeichnis

- [Balzert 96]      Helmut Balzert  
**Lehrbuch der Software-Technik - Software-Entwicklung**  
Spektrum Akademischer Verlag
- [DesignPatt 95]      Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides  
**Design Patterns - Elements of Reusable Object-Oriented Software**  
Addison Wesley
- [Langhauser 97]      Jürgen Langhauser  
**Spezifikation und Entwurf des Schemas der Verwaltungsdaten eines "Corporate Memory" und Implementierung der Zugriffskomponente**  
Diplomarbeit am Lehrstuhl für digitale Systeme, Universität Kaiserslautern, 1997
- [MDBenutz 97]      (Ben Melnick, Karen Olsen-Dunn)  
**Macromedia Director - Benutzerhandbuch**  
Macromedia, 1996
- [MDEinsteig 97]      (Ben Melnick, Jeff Schwamberger)  
**Macromedia Director - Einsteigerhandbuch**  
Macromedia, 1996
- [MLEinsteig 97]      (Joe Schmitz)  
**Macromedia Lingo - Einsteigerhandbuch**  
Macromedia, 1996

- [MLLexikon 97] (Ben Melnick, Jeff Schwamberger)  
**Macromedia Lingo-Lexikon**  
Macromedia, 1996
- [Otto 96] Johannes Otto  
**Bereitstellung einer Experimentierumgebung zur  
Entwicklung von Client-Server-Systemen in Smalltalk**  
Diplomarbeit am Lehrstuhl für digitale Systeme, Universität  
Kaiserslautern, 1996
- [Rosenzweig 97] Gary Rosenzweig  
**Combining Lingo & Director**  
<http://www.netscapepress.com/zine/9706/four>
- [Welsch 97] Norbert Welsch  
**Multimedia-Entwicklung mit Macromedia Director**  
Springer, 1997
- [Wendt 91] Siegfried Wendt  
**Nichtphysikalische Grundlagen der  
Informationstechnik**  
Springer, 1991
- [Wiegert 95] Oliver Wiegert  
**Änderbarkeit durch Objektorientierung**  
Vieweg, 1995